7universum.com
UNIVERSUM:
ТЕХНИЧЕСКИЕ НАУКИ

UNIVERSUM:
ТЕХНИЧЕСКИЕ НАУКИ

Научный журнал
Издается ежемесячно с декабря 2013 года
Является печатной версией сетевого журнала
Universum: технические науки

Выпуск: 10(127)

Октябрь 2024

Часть 1

Москва
2024

УДК 62/64+66/69
ББК 3
U55

Главный редактор:
Ахметов Сайранбек Махсутович, д-р техн. наук;

Члены редакционной коллегии:
Алланиязов Гуломжон Шерниязович, д-р философии техн. наук;
Горбачевский Евгений Викторович, канд. техн. наук;
Демин Анатолий Владимирович, д-р техн. наук;
Дехканов Зульфикахар Киргизбаевич, д-р техн. наук;
Звездина Марина Юрьевна, д-р. физ.-мат. наук;
Ким Алексей Юрьевич, д-р техн. наук;
Козьминых Владислав Олегович, д-р хим. наук;
Ларионов Максим Викторович, д-р биол. наук;
Манасян Сергей Керопович, д-р техн. наук;
Мажидов Кахрамон Халимович, д-р наук, проф;
Мартышкин Алексей Иванович, канд.техн. наук;
Мерганов Аваз Мирсултанович, канд.техн. наук;
Пайзуллаханов Мухаммад-Султанхан Саидвалиханович, д-р техн. наук;
Радкевич Мария Викторовна, д-р техн наук;
Романов Андрей Александрович, спец. по направлению гидрогеология и инженерная геология;
Серегин Андрей Алексеевич, канд. техн. наук;
Старченко Ирина Борисовна, д-р техн. наук;
Усманов Хайрулла Сайдуллаевич, д-р техн. наук;
Юденков Алексей Витальевич, д-р физ.-мат. наук;
Фозилов Садриддин Файзуллаевич, д-р техн. наук;
Tengiz Magradze, PhD in Power Engineering and Electrical Engineering.

U55 Universum: технические науки: научный журнал. – № 10(127). Часть 1., М., Изд. «МЦНО», 2024. – 72 с. – Электрон. версия печ. публ. – http://7universum.com/ru/tech/archive/category/10127

ISSN : 2311-5122

DOI: 10.32743/UniTech.2024.127.10

Учредитель и издатель: ООО «МЦНО»

ББК 3

© ООО «МЦНО», 2024 г.

Содержание

Статьи на русском языке **4**

Информатика, вычислительная техника и управление **4**

АНАЛИЗ СИСТЕМ ИДЕНТИФИКАЦИИ И ФОРМИРОВАНИЯ БАЗЫ ДАННЫХ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ 4
Бердымуратов Данияр Базарбаевич

СИМБИОТИЧЕСКОЕ ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА СОТРУДНИКОВ С ИСПОЛЬЗОВАНИЕМ ГЕНЕРАТИВНОГО ИИ В БИЗНЕС-СЕКТОРЕ 9
Воропаев Сергей Валерьевич

ОПТИМИЗАЦИЯ ПРОИЗВОДИТЕЛЬНОСТИ БАЗ ДАННЫХ MONGODB В МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ 19
Глумов Константин Сергеевич

УЛУЧШЕНИЕ КАЧЕСТВА ТИКЕТОВ В ТЕСТИРОВАНИИ ПО: СНИЖЕНИЕ ВОЗВРАТОВ ЗАДАЧ И УСКОРЕНИЕ ИСПРАВЛЕНИЯ БАГОВ 26
Домнин Евгений Константинович

ИСПОЛЬЗОВАНИЕ МОДУЛЬНЫХ МОНОЛИТОВ ДЛЯ РАЗРАБОТКИ МАСШТАБИРУЕМЫХ WEB-ПРИЛОЖЕНИЙ 32
Земсков Максим Александрович

ПЕРСПЕКТИВА ПРИМЕНЕНИЯ 3D-БИОПРИНТИНГА В МЕДИЦИНЕ И В СОЗДАНИИ КОНСТРУКЦИИ С ИСПОЛЬЗОВАНИЕМ ЖИВЫХ КЛЕТОК 40
Ибрагимов Шаботыр Бахрамович
Ходжамова Вера Сафаралиевна

РЕАЛИЗМ И АДАПТИВНОСТЬ ИИ-АВАТАРОВ В ОБУЧАЮЩИХ МОДУЛЯХ ГЕЙМИНГА 43
Любохинец Антон Андреевич

СИСТЕМА НАКОПЛЕНИЯ СПУТНИКОВЫХ ДАННЫХ LANDSAT И SENTINEL ДЛЯ СИСТЕМЫ ТОЧНОГО ЗЕМЛЕДЕЛИЯ 50
Павличенко Евгений Александрович
Якубайлик Олег Эдуардович

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ СПОРТИВНОЙ ФЕДЕРАЦИИ С ПРОГРАММНЫМ МОДУЛЕМ ПЛАНИРОВАНИЯ ТРЕНИРОВОЧНОГО ПРОЦЕССА 55
Сосегова Виктория Андреевна
Крутиков Александр Константинович

МЕТОДЫ РАЗРАБОТКИ АДАПТИВНЫХ ИНТЕРФЕЙСОВ ДЛЯ ANDROID ПРИЛОЖЕНИЙ С УЧЁТОМ ЭКРАННЫХ ХАРАКТЕРИСТИК 61
Терехов Владислав Игоревич

СТАТЬИ НА РУССКОМ ЯЗЫКЕ

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

АНАЛИЗ СИСТЕМ ИДЕНТИФИКАЦИИ И ФОРМИРОВАНИЯ БАЗЫ ДАННЫХ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

Бердымуратов Данияр Базарбаевич
докторант, кафедра технического обеспечения,
Национальный университет обороны Республики Казахстан,
Республика Казахстан, г. Астана
E-mail: for@topmail.kz

ANALYSIS OF UAV IDENTIFICATION SYSTEMS AND DATABASE DEVELOPMENT

Daniyar Berdymuratov
PhD student, Department of Technical Support,
The national Defense University of the Republic of Kazakhstan,
Kazakhstan, Astana

АННОТАЦИЯ

В статье подробно анализируются системы идентификации беспилотных летательных аппаратов (БПЛА), а также технологии создания базы данных для их классификации. Цель исследования – изучение существующих методов идентификации, таких как радиолокация, GPS, RFID и нейросетевые алгоритмы, для повышения эффективности противодействия БПЛА. Методология основана на сравнении технологий сбора и обработки данных, а также их структурировании с использованием ROLAP-хранилищ. В качестве примеров приводятся кейсы применения данных технологий для охраны воздушного пространства в мегаполисах и на стратегически важных объектах. Полученные результаты подчеркивают важность создания базы данных БПЛА для автоматизации процессов идентификации и усиления защиты инфраструктур. Выводы статьи указывают на необходимость формирования базы данных БПЛА как неотъемлемой части системы противодействия беспилотникам.

ABSTRACT

The article provides a detailed analysis of unmanned aerial vehicle (UAV) identification systems and database technologies for their classification. The goal of the research is to study existing identification methods, such as radar, GPS, RFID, and neural network algorithms, to improve UAV countermeasure systems. The methodology is based on comparing data collection and processing technologies and structuring them using ROLAP storage. Case studies of these technologies in urban airspace monitoring and security applications are presented. The results highlight the importance of creating a UAV database to automate identification processes and enhance infrastructure protection. The conclusions indicate the necessity of developing a UAV database as an essential part of UAV countermeasure systems.

Ключевые слова: БПЛА, беспилотные летательные аппараты, идентификация, база данных, радиолокация, нейросети.

Keywords: UAVs, unmanned aerial vehicles, identification, database, radar, neural networks

---

Введение. В последние годы вопрос идентификации БПЛА приобретает особую актуальность. Важно создать системы, способные эффективно распознавать и классифицировать различные типы беспилотников для повышения безопасности и оптимизации работы с этими устройствами. В данной статье рассматриваются современные подходы к идентификации БПЛА, а также методы формирования базы данных для эффективного управления информацией о БПЛА.

Цель исследования – проведение анализа существующих систем идентификации БПЛА и рассмотрение технологий, применяемых для создания и структурирования баз данных, для дальнейшей разработки рекомендаций по противодействию беспилотным угрозам.

Библиографическое описание: Бердымуратов Д.Б. АНАЛИЗ СИСТЕМ ИДЕНТИФИКАЦИИ И ФОРМИРОВАНИЯ БАЗЫ ДАННЫХ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18382*

Задачи исследования:

1. Изучить современные системы идентификации и технологии, применяемые для создания и структурирования баз данных беспилотных летательных аппаратов.

2. Проанализировать их эффективность в зависимости от технических характеристик и условий применения.

3. Сформулировать основные выводы и перспективы дальнейшего развития систем противодействия БПЛА.

Материалы и методы. В процессе исследования были использованы материалы из открытых источников, а также научная литература по рассматриваемой тематике. Применялись методы анализа, обобщения, синтеза и интерпретации.

Результаты и их обсуждение. Учитывая увеличение количества БПЛА и разнообразие их моделей, задача идентификации этих аппаратов приобретает критическое значение. Системы идентификации БПЛА должны быть способны быстро и точно определять тип и характеристики каждого летательного аппарата, а также различать их по параметрам, что требует интеграции передовых технологий, включая радиолокацию, GPS, RFID и методы машинного обучения.

Важным аспектом является не только процесс идентификации, но и эффективное управление информацией о БПЛА. Для этого необходимо создавать и структурировать базы данных, которые смогут хранить и обрабатывать большие объемы данных о различных моделях БПЛА и их характеристиках.

Системы идентификации беспилотных летательных аппаратов (БПЛА) играют ключевую роль в обеспечении безопасности воздушного пространства, особенно в связи с ростом использования дронов в гражданских, военных и коммерческих областях. В рамках классификации современных систем идентификации можно выделить несколько основных категорий, основанных на технологиях, которые используют различные принципы для обнаружения, отслеживания и распознавания БПЛА. Рассмотрим наиболее распространенные системы и их характеристики.

Радиолокационные системы. Радиолокационные системы для идентификации БПЛА широко используются благодаря своей способности обнаруживать объекты на больших расстояниях и в сложных погодных условиях. Такие системы фиксируют радиоволны, отраженные от летательных аппаратов, что позволяет точно определить местоположение и траекторию полета дрона.

Примером радиолокационной системы является ARSR-4, используемая в воздушном пространстве США для контроля воздушных судов, включая БПЛА. В Европе и России применяются системы типа ADS-B (Automatic Dependent Surveillance-Broadcast), которые позволяют отслеживать летательные аппараты с использованием данных о местоположении, передаваемых через спутниковую навигацию [1].

Принцип работы радиолокационных систем основан на отправке радиосигнала и его отражении от объекта. Возможности систем включают дальнее обнаружение, определение скорости и направления движения, что особенно важно в условиях быстрого роста числа БПЛА в гражданском и военном воздушном пространстве.

Системы на основе спутниковой навигации (GPS). Спутниковая навигация, в частности GPS, часто используется для идентификации БПЛА, особенно в случае автономных аппаратов. Такие системы позволяют точно определять координаты объекта и отслеживать его перемещения в режиме реального времени.

Одним из примеров использования технологии GPS является система DJI Aeroscope, разработанная китайской компанией DJI. Она позволяет отслеживать дроны в радиусе до 50 км, фиксируя их серийные номера, координаты и траектории полета. Принцип работы этой системы заключается в улавливании сигналов, передаваемых GPS-модулями дронов, что позволяет эффективно управлять воздушным пространством.

Модель Aeroscope обладает возможностями отслеживания даже небольших дронов на дальних расстояниях, что делает ее популярной среди правоохранительных органов и служб безопасности для защиты объектов критической инфраструктуры [2].

RFID-системы. RFID-системы (радиочастотная идентификация) используются для маркировки и идентификации БПЛА, особенно в закрытых и ограниченных зонах, таких как склады, производственные объекты и логистические центры. В отличие от радиолокационных и GPS-систем, RFID работает на меньших расстояниях, обеспечивая точную идентификацию меток, закрепленных на дроне.

Один из примеров — RFID-система от компании Zebra Technologies, которая применяется для отслеживания дронов и других устройств на складах и производственных объектах. Эти системы позволяют интегрировать данные с внутренними ERP-системами, что улучшает управление производственными и логистическими процессами [3].

Принцип работы основан на передаче сигнала с RFID-метки, который считывается при прохождении через специальные устройства. Возможности системы включают идентификацию большого количества объектов одновременно и обеспечение высокой точности идентификации на малых дистанциях.

Визуальные системы идентификации. Визуальные системы на основе технологий машинного обучения и компьютерного зрения становятся все более популярными благодаря их способности анализировать визуальные изображения БПЛА. Эти системы используют камеры высокого разрешения и алгоритмы распознавания образов для идентификации дронов по их внешним характеристикам.

Примером является система Dedrone, которая использует камеры и сенсоры для захвата изображений летательных аппаратов, а затем анализирует их

с помощью программного обеспечения на основе искусственного интеллекта (AI). Dedrone способна распознавать различные модели БПЛА, определять их размеры, форму и движение, а также классифицировать угрозы [4].

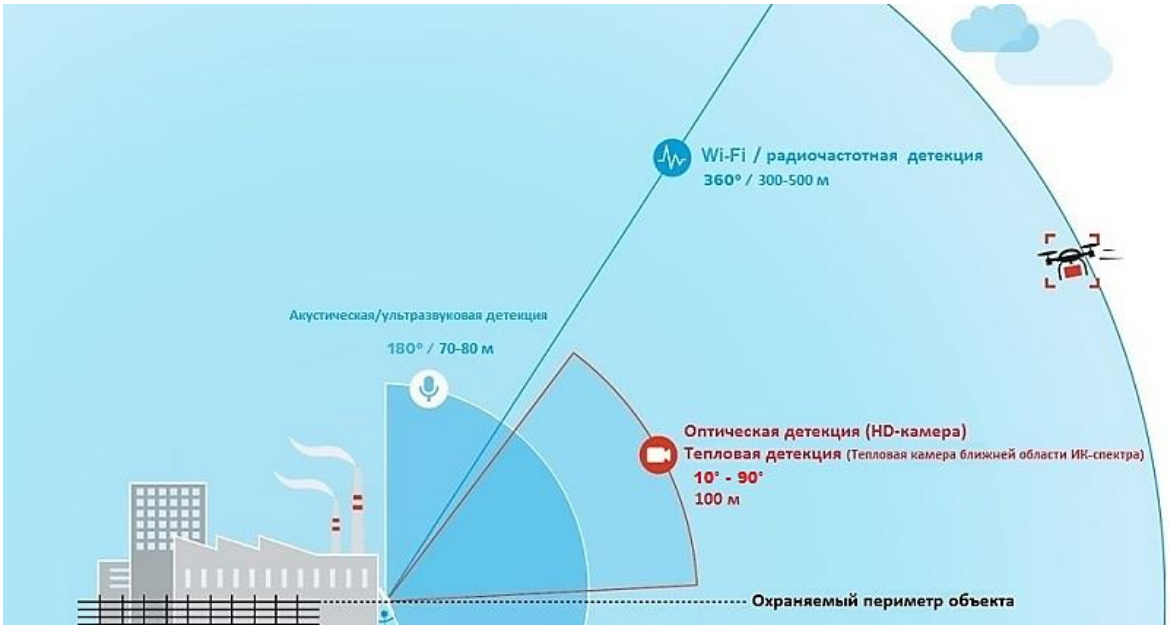

Рисунок 1. Система DroneTracker

Принцип работы таких систем основан на распознавании внешних характеристик дронов и их анализе с использованием нейросетевых алгоритмов. Возможности систем включают классификацию по типам аппаратов, обнаружение необычных траекторий и распознавание опасных объектов в реальном времени.

Сигнатурные базы данных. Сигнатурные системы идентификации БПЛА работают по принципу сопоставления уникальных сигнатур, таких как радиолокационные отражения, тепловые сигнатуры или электромагнитные излучения, с уже известными профилями, хранящимися в базе данных. Эти системы позволяют не только идентифицировать тип дрона, но и его модель и производителя.

Примером является система DroneShield, разработанная для военных и правоохранительных органов. DroneShield использует сигнатурные базы данных для идентификации и классификации дронов на основе их радиолокационных и радиочастотных характеристик. Такие системы особенно эффективны для обнаружения и нейтрализации дронов, применяемых в военных конфликтах [5].

Системы на основе сигнатур работают, анализируя данные о частоте, модуляции сигнала и других характеристиках, которые сопоставляются с эталонными данными в базе. Это позволяет идентифицировать не только тип аппарата, но и его конкретную модель.

Примеры использования систем идентификации:

Безопасность аэропортов. Системы, такие как Aeroscope и Dedrone, используются для защиты воздушного пространства вблизи аэропортов, предотвращая случайные или преднамеренные полеты дронов в запрещенных зонах.

Охрана стратегических объектов. Военные базы и правительственные учреждения применяют системы типа DroneShield для защиты от шпионских дронов и беспилотников, которые могут быть использованы для сбора данных или нанесения вреда.

Городское воздушное пространство. В крупных мегаполисах по всему миру активно внедряются системы мониторинга дронов на основе радиолокационных и визуальных технологий, обеспечивающих контроль над воздушным движением в густонаселенных районах [6].

Идентификационные признаки. Идентификация БПЛА основывается на множестве параметров, включая физические и технические характеристики. Основными признаками являются:

- Размеры и вес устройства;
- Форма корпуса;
- Частота и тип сигналов связи;
- Уникальные полетные траектории;
- Цифровые подписи, такие как радиолокационные и тепловые сигнатуры.

Эти параметры используются для создания уникальных профилей каждого типа БПЛА, которые затем могут быть использованы для идентификации при помощи автоматизированных систем [7].

Формирование базы данных для БПЛА. Создание и эффективное использование базы данных (БД) для беспилотных летательных аппаратов (БПЛА) является одним из ключевых элементов системы идентификации. Современные системы БД для БПЛА направлены на накопление, хранение, обработку и анализ данных, что позволяет не только идентифицировать летательные аппараты, но и прогнозировать

их поведение, отслеживать траектории полетов и предупреждать потенциальные угрозы. В данном разделе рассматриваются основные подходы к созданию баз данных, примеры современных систем и их использование.

Основной задачей формирования базы данных для БПЛА является сбор разнородной информации, которая может включать:

- визуальные данные (изображения и видео, полученные с камер наблюдения и сенсоров);
- характеристики радиосигналов, передаваемых БПЛА;
- параметры полета (траектория, высота, скорость и др.);
- физические характеристики БПЛА (размеры, вес, тип двигателей, форма корпуса);
- производственные данные (информация о производителе, модели, дате выпуска).

Сбор данных происходит как автоматически (с использованием сенсоров, камер, радаров), так и вручную операторами. Современные системы включают широкий спектр сенсоров для сбора данных: оптические камеры, радиолокационные установки, инфракрасные датчики, которые обеспечивают многомерный анализ полетов БПЛА.

Примером системы сбора данных является SkyPatriot, разработанная компанией Dedrone, которая использует множество сенсоров для обнаружения, отслеживания и классификации БПЛА. Система автоматически собирает данные о полетах БПЛА, их размерах, скорости и направлении движения, создавая детализированную базу данных для дальнейшей обработки [4].

Хранилища данных. Для хранения и обработки больших объемов данных, поступающих в режиме реального времени, используются современные системы управления базами данных (СУБД). В данном контексте важную роль играют две технологии: ROLAP (Relational Online Analytical Processing) и MOLAP (Multidimensional Online Analytical Processing).

ROLAP использует реляционные базы данных и позволяет гибко управлять данными, что делает его удобным для аналитики больших объемов информации о БПЛА. Примером такой системы может служить PostgreSQL, которое широко применяется для хранения больших объемов структурированных данных и выполнения сложных аналитических запросов.

MOLAP представляет собой более специализированное решение, где данные агрегируются и индексируются, что ускоряет анализ. Cognos Analytics и Microsoft Analysis Services — примеры систем, которые могут быть использованы для MOLAP-анализа данных о БПЛА [8].

Эти системы позволяют не только хранить, но и быстро обрабатывать большие объемы данных, предоставляя пользователю возможность делать прогнозы и моделировать поведение беспилотников на основе полученных данных.

Системы идентификации на основе базы данных БПЛА работают по следующему принципу:

1) Обнаружение: Система фиксирует присутствие БПЛА с помощью различных сенсоров (радиолокационных, оптических, инфракрасных и др.).

2) Сбор данных: Данные о БПЛА собираются и передаются в централизованную базу данных. Эти данные могут включать видеозаписи, характеристики радиосигнала, параметры полета и другие ключевые характеристики.

3) Сопоставление с базой данных: Система сопоставляет полученные данные с уже имеющимися записями в базе данных. Это может включать сравнение визуальных данных, сигнатур сигналов или технических характеристик БПЛА.

4) Анализ и идентификация: Используя алгоритмы машинного обучения и нейросетей, система анализирует данные и определяет тип БПЛА, его возможные намерения (гражданское использование, военные или разведывательные цели) и маршрут полета [1].

5) Предупреждение и реагирование: В случае обнаружения подозрительного или неопознанного БПЛА система может передавать сигнал тревоги в командные центры или инициировать контрмеры для нейтрализации угрозы.

Одним из примеров успешного применения систем идентификации БПЛА является проект DroneTracker от Dedrone, который используется на международных аэропортах и крупных инфраструктурных объектах для отслеживания и идентификации беспилотников в радиусе нескольких километров. Система интегрируется с радарами и камерами, собирает данные о траекториях полета БПЛА и автоматически сопоставляет их с базой данных, чтобы определить, является ли беспилотник зарегистрированным и безопасным [4].

Заключение. В ходе исследования было установлено, что современные системы идентификации беспилотных летательных аппаратов (БПЛА) демонстрируют значительный прогресс благодаря внедрению различных технологий, таких как радиолокация, GPS и нейросетевые алгоритмы. Эти технологии позволяют эффективно распознавать и классифицировать БПЛА, что существенно повышает уровень безопасности в воздушном пространстве. Однако, несмотря на достижения в области идентификации, отсутствие централизованных баз данных о беспилотниках и их характеристиках остается серьезной проблемой, препятствующей оперативному реагированию на угрозы, связанные с использованием БПЛА.

В связи с этим, формирование базы данных БПЛА представляется необходимым шагом для автоматизации процессов идентификации и анализа данных. Создание и наполнение такой базы данных позволит не только оптимизировать методы мониторинга и контроля воздушного пространства, но и повысить эффективность работы систем противодействия беспилотникам. Наличие актуальной и структурированной информации о БПЛА станет неотъемлемой

частью стратегии обеспечения безопасности, что, в свою очередь, позволит более эффективно реагировать на потенциальные угрозы и минимизировать риски, связанные с использованием беспилотных летательных аппаратов. В будущем дальнейшее развитие технологий машинного обучения и нейросетей внесет значительный вклад в усовершенствование систем идентификации БПЛА.

Список литературы:

1. «Некоторые аспекты идентификации беспилотных летательных аппаратов». / [Электронный ресурс]. – Режим доступа: https://na-journal.ru/4-2024-aviaciya-kosmonavtika/11418-nekotorye-aspekty-identifikacii-bespilotnyh-letatelnyh-apparatov (дата обращения: 03.10.2024).
2. AeroScope: Технология DJI для отслеживания дронов / [Электронный ресурс]. – Режим доступа: https://dji-blog.ru/naznachenie/aeroscope-tehnologija-dji-dlja-otslezhivanija-dronov.html (дата обращения: 03.10.2024).
3. Smart Airspace Security and Drone Detection Software / [Электронный ресурс]. – Режим доступа: https://www.dedrone.com/products/drone-detection-software (дата обращения: 07.10.2024).
4. DroneSentry Autonomous Integrated Detection and Countermeasure / [Электронный ресурс] - Режим доступа: https://www.droneshield.com/c-uas-products/dronesentry (дата обращения: 07.10.2024).
5. «Защита от дронов в городской, густонаселенной местности» / [Электронный ресурс]. – Режим доступа: https://antidron.pro/blog/gustonaselennoj-mestnosti (дата обращения: 08.10.2024).
6. Искусственный интеллект и нейросети помогут распознать БПЛА/ [Электронный ресурс]. – Режим доступа: https://cloudnetworks.ru/analitika/iskusstvennyj-intellekt-i-nejroseti-pomogut-raspoznat-bpla/ (дата обращения: 08.10.2024).
7. Бердымуратов Д.Б. Современные способы и средства противодействия беспилотным летательным аппаратам // Журнал «Научный Лидер» № 39 (189). [Электронный ресурс]. – Режим доступа: https://scilead.ru/article/7148-sovremennie-sposobi-i-sredstva-protivodejstvi (дата обращения: 08.10.2024).
8. «В ногу со временем: введение в многомерный анализ и OLAP-технологии» / Habr. / [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/775042/ (дата обращения: 08.10.2024).

DOI - 10.32743/UniTech.2024.127.10.18374

СИМБИОТИЧЕСКОЕ ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА СОТРУДНИКОВ С ИСПОЛЬЗОВАНИЕМ ГЕНЕРАТИВНОГО ИИ В БИЗНЕС-СЕКТОРЕ

Воропаев Сергей Валерьевич
инженер,
ООО «Асилиск»,
РФ, г. Москва
E-mail: ceo@asilisc.com

SYMBIOTIC ENHANCEMENT OF EMPLOYEE PRODUCTIVITY USING GENERATIVE AI IN THE BUSINESS SECTOR

Sergei Voropaev
Engineer,
Asilisc LLC,
Russia, Moscow

АННОТАЦИЯ

Данное исследование изучает влияние генеративного искусственного интеллекта (ИИ) на производительность труда и качество выполнения задач. Мы представляем метод симбиотического повышения производительности труда с использованием генеративного ИИ, который интегрирует помощь ИИ в рабочие процессы сотрудников. Наше исследование, проведенное с участием 86 сотрудников из российских, швейцарских и гонконгских компаний, демонстрирует значительное улучшение как производительности труда, так и качества труда при использовании ИИ. В исследовании использовались два режима взаимодействия с ИИ: Коллаборативный когнитивный агент (CCA) и Непрерывный когнитивный агент (CoCA). Результаты показывают увеличение количества выполнения задач на 12,2% и сокращение времени выполнения задач на 25,1% при использовании ИИ. Качество выполнения работы улучшилось более чем на 40% по сравнению с контрольной группой. Эти результаты свидетельствуют о том, что метод может быть эффективным подходом для повышения эффективности труда посредством интеграции ИИ, приводя к общему повышению производительности труда на 18,7% по сравнению с контрольной группой. Исследование фокусируется на применении метода в бизнес секторе экономики, таких как строительство, ритейл, банковский сектор, технологический сектор и тяжелая промышленность.

ABSTRACT

This study examines the impact of generative artificial intelligence (AI) on labor productivity and task performance quality. We introduce a method for symbiotic productivity enhancement using generative AI, which integrates AI assistance into employee workflows. Our research, conducted with 86 employees from companies in Russia, Switzerland, and Hong Kong, demonstrates significant improvements in both productivity and task quality when AI is utilized. The study employed two modes of AI interaction: Collaborative Cognitive Agent (CCA) and Continuous Cognitive Agent (CoCA). Results indicate a 12.2% increase in task completion and a 25.1% reduction in task execution time with AI assistance. Task quality improved by over 40% compared to the control group. These findings suggest that the proposed method could be an effective approach for enhancing productivity through AI integration, leading to an overall productivity increase of 18.7% compared to the control group. The study focuses on the method's application in key economic sectors such as construction, retail, banking, technology, and heavy industry.

Ключевые слова: генеративный ИИ, производительность труда, качество выполнения работы, интеграция ИИ, эффективность труда, инновации, повышение производительности труда, повышение качества труда, генеративный ИИ в бизнесе.

Keywords: generative AI, labor productivity, task execution speed, work quality, AI assistance, AI integration, employee work efficiency, innovation, productivity enhancement, generative AI in business.

1. Введение

Интеграция искусственного интеллекта (ИИ) в рабочие процессы показала многообещающий потенциал для повышения производительности труда и эффективности труда. Данное исследование представляет и оценивает метод симбиотического повышения производительности труда с использованием генеративного ИИ, который направлен на создание синергетической комбинации человеческого опыта и возможностей ИИ при выполнении сложных задач. Данное исследование концентрируется

Библиографическое описание: Воропаев С.В. СИМБИОТИЧЕСКОЕ ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА СОТРУДНИКОВ С ИСПОЛЬЗОВАНИЕМ ГЕНЕРАТИВНОГО ИИ В БИЗНЕС-СЕКТОРЕ // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18374*

на бизнес секторе экономики из-за их значительного влияния на экономику в целом и высокого потенциала для интеграции генеративного ИИ.

Недавние исследования продемонстрировали положительное влияние ИИ на производительность труда. В экспериментальном исследовании Noy и Zhang было показано, что использование ИИ сокращает затраты времени на 10 минут, или на 37 %, при выполнении задач по написанию текстов [10]. Как пишет Nielsen, инструменты генеративного ИИ увеличили производительность труда бизнес-пользователей на 66 % при выполнении рабочих задач [8]. Службы поддержки, использующие ИИ, начали обрабатывать на 13,8 % больше запросов клиентов в час. Также Nielsen показал, что бизнес-пользователи, использующие ИИ, смогли писать на 59 % больше бизнес-документов в час, а программисты, применяющие ИИ, завершали на 126 % больше проектов в неделю [9]. Nielsen подчеркивает, что 66-процентный прирост производительности труда за счет ИИ эквивалентен 47 годам естественного прироста производительности труда в США или 88 годам роста в Европейском союзе [9]. Согласно данным Microsoft и LinkedIn, сотрудники с ИИ тратили на 25–45 % меньше времени на чтение электронных писем и редактировали на 10 % больше документов в Word, Excel и PowerPoint, что в конечном итоге привело к увеличению эффективности труда компаний на 20 % [7]. В другом исследовании Brynjolfsson, Li и Raymond было установлено, что ИИ повышает общую производительность труда в среднем на 14 %, включая увеличение на 34 % для начинающих и низкоквалифицированных работников [2].

Метод основывается на этих выводах, представляя комплексный подход к интеграции ИИ на рабочем месте. Целью данного исследования является эмпирическая оценка эффективности труда в повышении как производительности труда, так и качества труда для различных типов задач.

Важно отметить, что генеративный ИИ в данном контексте выступает преимущественно как инструмент ускорения поиска оптимальных решений из базы известных, а не как средство кардинального изменения квалификации сотрудников. Этот аспект особенно важен при интерпретации результатов повышения производительности труда

2. Метод

2.1 Участники

В исследовании приняли участие 86 сотрудников из российских, швейцарских и гонконгских компаний. Участники были отобраны из бизнес сектора с использованием целенаправленной выборки на основе таких критериев, как стаж работы и открытость к инновациям.

Несмотря на ограниченное число участников, данное исследование направлено на получение качественного описательного анализа, нацеленного на изучение и понимание опыта использования генеративного ИИ в рабочих процессах, а не на широкие статистические обобщения.

Полученные результаты дают возможность оценить эффективность использования предложенного метода применительно к узкому сектору экономики – бизнес-сектору.

Участники были случайным образом распределены в одну из трех групп (CCA, CoCA или контрольную) с использованием стратифицированной рандомизации.

Стратификация проводилась по таким факторам как опыт работы, отрасль и начальный уровень производительности труда для обеспечения сбалансированности групп.

Процесс рандомизации осуществлялся с помощью компьютерного генератора случайных чисел, реализованного в статистическом пакете R (версия 4.1.0).

2.2 Дизайн

Мы использовали смешанный факторный дизайн с двумя межсубъектными факторами (помощь ИИ: CCA, CoCA, контрольная группа; тип задачи: разработка идей, решение бизнес-проблем и одним внутрисубъектным фактором (время: исходный уровень, середина, финал). Исследование фокусировалось на двух основных типах задач:

- Разработка идей: создание инновационных решений для бизнес-задач, требующее креативного мышления и междисциплинарных знаний.
- Решение бизнес-задач: анализ сложных бизнес-ситуаций и разработка стратегических рекомендаций.

Эти задачи были выбраны из-за их высокой частоты в повседневной работе сотрудников (в среднем 60% рабочего времени) и критической важности для бизнес-процессов. Они требуют высокой квалификации, включая навыки критического мышления, анализа данных и стратегического планирования. Типы задач были выбраны с учетом их релевантности для ключевых секторов экономики, в которых работали участники.

2.3 Материалы и процедура

В исследовании использовалось приложение персонального помощника на основе ИИ, интегрированное с бизнес-знаниями и возможностями генеративного ИИ и проведено обучение по инженерии запросов к ИИ.

В исследовании были реализованы два режима взаимодействия с ИИ, которые далее будут обозначаться следующими сокращениями:

- CCA (Collaborative Cognitive Agent) - Коллаборативный когнитивный агент;
- CoCA (Continuous Cognitive Agent) - Непрерывный когнитивный агент.

CCA: Сотрудники занимались стратегическими аспектами задач, делегируя рутинные или интенсивные по данным части ИИ.

CoCA: ИИ обеспечивал непрерывную поддержку на протяжении всей задачи, при этом сотрудники постоянно взаимодействовали с ИИ для получения поддержки и обратной связи в реальном времени.

Эксперимент проводился в три этапа: предварительный опрос, выполнение задачи и пост-опрос. Участники были мотивированы финансовыми стимулами за активное участие и качественное выполнение задач.

Ограничения исследования включали использование симулированных бизнес-задач, которые, хотя и были разработаны на основе реальных сценариев, могли не полностью отражать сложность реальных рабочих ситуаций. Кроме того, четырехнедельный период исследования может не полностью отражать долгосрочные эффекты использования. Долгосрочные эффекты станут заметны через несколько лет использования ИИ.

В исследовании использовались два основных инструмента генеративного ИИ: GPT-4 от OpenAI и Claude 3.5 от Antropic. Оба инструмента были интегрированы в приложение персонального помощника на основе ИИ, которое взаимодействовало с бизнес-знаниями и предоставляло возможности генеративного ИИ для выполнения задач.

Приложение персонального помощника на основе ИИ было разработано с акцентом на быстрый поиск и предложение релевантных решений из известных подходов, соответствующих текущей задаче.

2.4 Измерения

Производительность труда: Измерялась скоростью выполнения задач и коэффициентом завершения задач.

Качество: Оценивалось через точность выполнения задач и оценки при экспертной оценке.

Показатель сохранения ИИ: Новая метрика, количественно определяющая степень интеграции ИИ в работу сотрудников (см. Приложение для подробной методологии расчета).

2.5 Анализ данных

Мы использовали различные статистические методы, с Т-тестами для независимых выборок, однофакторный дисперсионный анализ (ANOVA), множественная линейная регрессия, хи-квадрат тесты, дисперсионный анализ с повторными измерениями, бутстрэппинг и моделирование структурными уравнениями (SEM).

Бутстрэп анализ был применен для повышения надежности статистических оценок, особенно учитывая относительно небольшой размер выборки (N=86). Этот метод позволяет создавать эмпирические распределения статистических показателей путем многократного повторного выбора с возвращением из исходных данных. В нашем исследовании бутстрэп использовался для:

1. Создания более точных доверительных интервалов для ключевых параметров, таких как средние значения повышения производительности труда и качества работы.
2. Оценки стабильности и надежности регрессионных коэффициентов в моделях, связывающих использование ИИ с результатами производительности.
3. Валидации размеров эффекта, особенно для сравнений между группами с помощью ИИ и контрольной группой.
4. Преодоления потенциальных проблем с нормальностью распределения данных, что особенно важно при работе с небольшими выборками.

Было сгенерировано 5000 бутстрэп-выборок для каждого анализа, что обеспечивает надежную основу для статистических выводов.

Для анализа влияния использования ИИ на производительность труда и качество работы была разработана комплексная модель, включающая как наблюдаемые, так и латентные переменные. Модель учитывала прямые эффекты использования ИИ, а также опосредованные эффекты через эффективность использования ИИ.

Модель исследовалась с помощью нескольких методов:

1. Множественная линейная регрессия использовалась для оценки прямых эффектов наблюдаемых переменных на производительность труда и качество работы.
2. Моделирование структурными уравнениями (SEM) применялось для анализа сложных взаимосвязей между наблюдаемыми и латентными переменными, включая медиационные эффекты.
3. Дисперсионный анализ с повторными измерениями использовался для оценки изменений в эффективности использования ИИ и производительности труда с течением времени.

Параметры модели оценивались методом максимального правдоподобия. Для проверки устойчивости результатов применялись методы бутстрэппинга и анализа чувствительности.

3. Результаты

3.1 Повышение производительности труда

Коллаборативный когнитивный агент (CCA): сокращение времени на 18,8 минут (95% ДИ: [15.2, 22.4]), на 22,5% быстрее контрольной группы ($p < 0.001$, $d = 0.89$, 95% ДИ d: [0.54, 1.24]).

Непрерывный когнитивный агент (CoCA): сокращение времени на 23,13 минуты (95% ДИ: [19.5, 26.8]), на 27,63% быстрее контрольной группы ($p < 0.001$, $d = 1.05$, 95% ДИ d: [0.69, 1.41]).

Примечание: Для коррекции на множественные сравнения использовался метод Бенджамини-Хохберга с контролем false discovery rate (FDR) на уровне 0.05.

Оба метода показали значительное улучшение в скорости выполнения задач по сравнению с контрольной группой. CoCA продемонстрировал немного более высокую эффективность труда в сокращении времени выполнения задач.

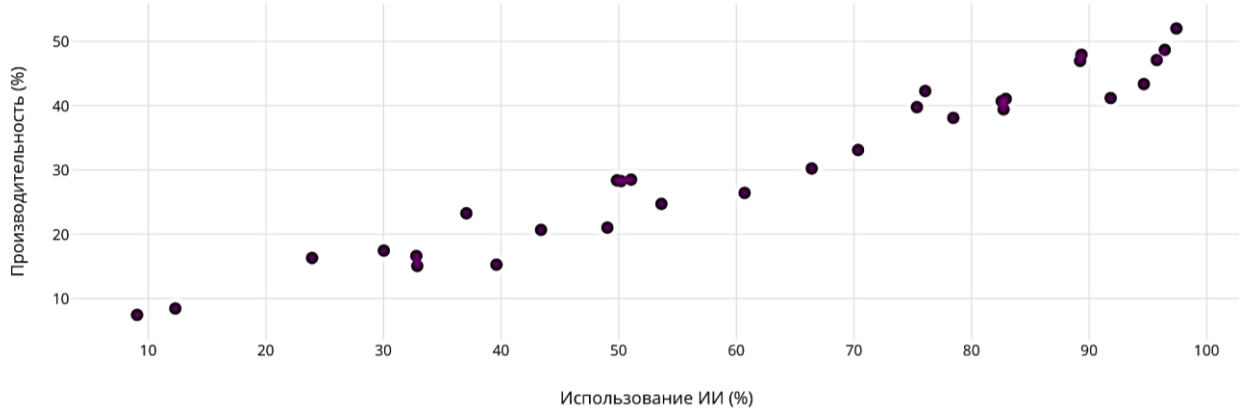

Рисунок 1 Зависимость прироста производительности труда от использования в качестве помощника ИИ

Таблица 1.

Сравнение производительности труда между группами

Группа	Коэффициент выполнения задач (%)	Среднее время выполнения задачи (мин)
ИИ-ассистированная	112.2	74.9
Контрольная	100.0	100.0

Коэффициент выполнения задач: Группы с помощью ИИ выполнили на 12,2% больше задач по сравнению с контрольной группой (t(84) = 3,76, p < 0,001, d Коэна = 0,82, 95% ДИ [0,37, 1,26]).

Скорость выполнения задач: Задачи выполнялись на 25,1% быстрее с помощью ИИ (t(84) = -4,92, p < 0,001, d Коэна = 1,07, 95% ДИ [0,61, 1,52])."

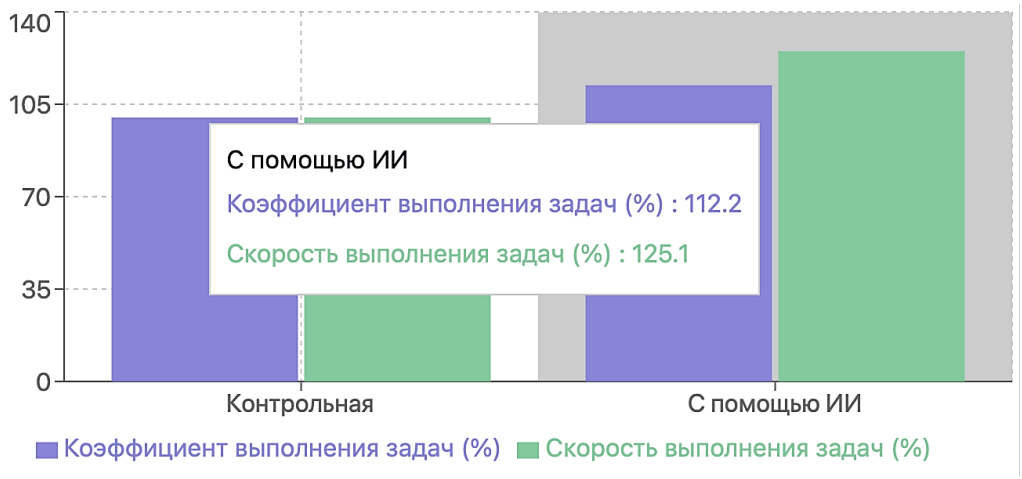

Рисунок 1.1. График сравнения производительности труда

3.2 Улучшение качества

Таблица 2.

Распределение качества труда

Группа	Низкое качество (%)	Среднее качество (%)	Высокое качество (%)
ИИ-ассистированная	10	30	60
Контрольная	30	50	20

Качество выполнения работы с помощью ИИ было на 40% выше, чем в контрольной группе (95% ДИ: [32%, 48%], F(2, 83) = 28,64, p < 0,001, η^2 = 0,41, 90% ДИ η^2: [0.27, 0.51]).

Распределение качества: Обнаружена значимая связь между помощью ИИ и результатами качества (χ^2(4, N = 86) = 19,73, p < 0,001, V Крамера = 0,34, 95% ДИ [0,22, 0,46]).

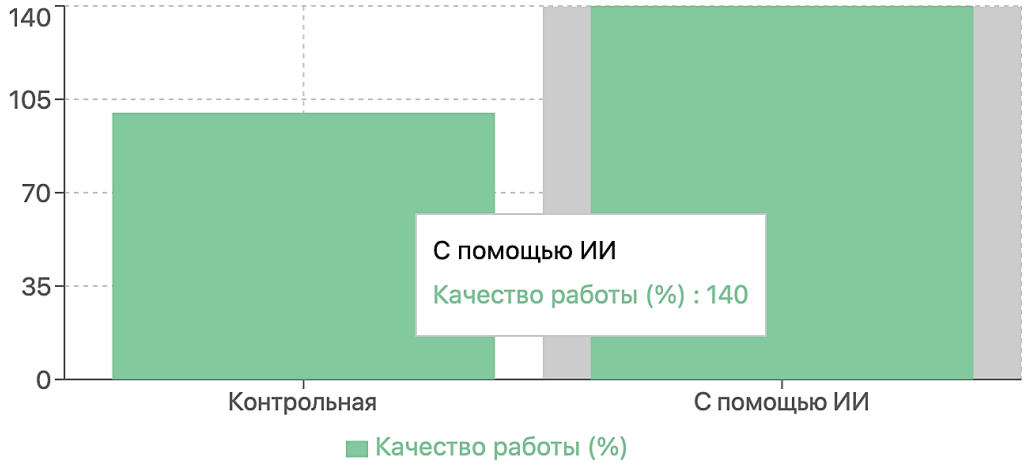

Рисунок 2 График улучшения качества труда

3.3 Распределение производительности труда

Таблица 3.

Изменение производительности труда по группам

Группа производительности труда	Среднее улучшение (%)	95% ДИ
Ниже среднего	43	[26.8, 59.2]
Выше среднего	17	[5.9, 28.1]

Сотрудники с производительностью труда ниже среднего увеличили свою производительность труда на 43% ($t(21) = 6,82$, $p < 0,001$, d Коэна = 1,45, 95% ДИ [0,89, 2,01]).

Сотрудники с производительностью труда выше среднего, показали улучшение на 17% ($t(20) = 3,24$, $p = 0,004$, d Коэна = 0,71, 95% ДИ [0,22, 1,19]).

Наблюдаемое более значительное улучшение у работников с низкой производительностью труда связано с тем, что ИИ-ассистент эффективно компенсирует недостаток опыта и знаний, предоставляя быстрый доступ к оптимальным решениям, которые более опытные сотрудники могут находить самостоятельно.

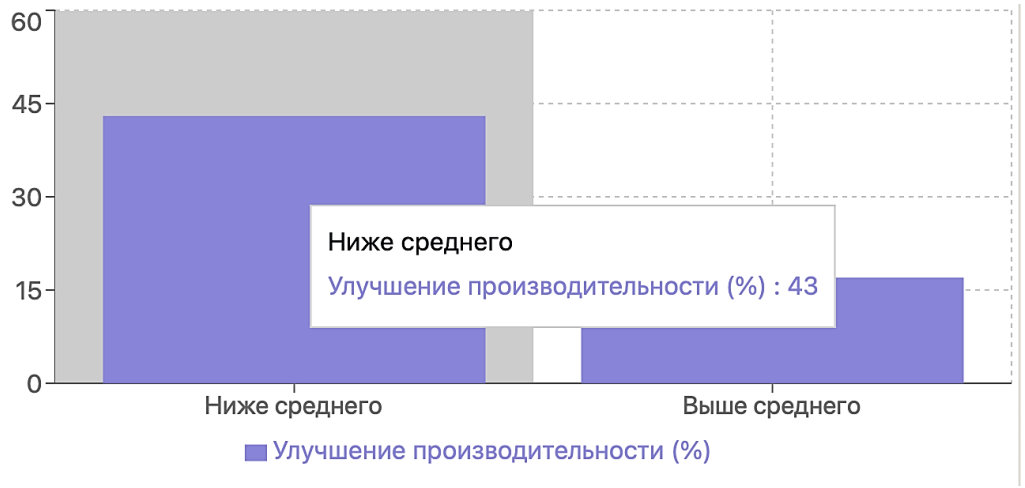

Рисунок 3. График распределения производительности труда

3.4 Регрессионный анализ

Модели множественной линейной регрессии были использованы для анализа взаимосвязи между помощью ИИ и результатами производительности труда при контроле других переменных:

Модель производительности труда: $R^2 = 0,47$ (95% ДИ: [0.31, 0.59]), $F(5, 80) = 14,23$, $p < 0,001$

Модель качества: $R^2 = 0,53$ (95% ДИ: [0.37, 0.64]), $F(5, 80) = 18,06$, $p < 0,001$

Уравнения множественной линейной регрессии:
Модель производительности труда:

$$Y_productivity = \beta0 + \beta1X_AI_assistance + \beta2X_experience + \beta3X_task_type + \beta4X_innovation_openness + \varepsilon$$

Модель качества:

$$Y_quality = \beta0 + \beta1X_AI_assistance + \beta2X_experience + \beta3X_task_type + \beta4X_innovation_openness + \varepsilon$$

где: Y_productivity - показатель производительности труда

Y_quality - показатель качества выполнения работы

X_AI_assistance - бинарная переменная использования ИИ (0 - контрольная группа, 1 - группа с ИИ)

X_experience - опыт работы сотрудника (в годах)

X_task_type - тип задачи (1 - разработка идей, 2 - решение бизнес-проблем)

X_innovation_openness - открытость к инновациям (по шкале от 1 до 5)

β0 - свободный член β1, β2, β3, β4 – коэффициенты регрессии

ε - случайная ошибка

Помощь ИИ была значимым предиктором в обеих моделях ($\beta = 0,39$, $p < 0,001$ для производительности труда; $\beta = 0,45$, $p < 0,001$ для качества).

Диагностика модели:

Тест Шапиро-Уилка на нормальность остатков: $W = 0.98$, $p = 0.124$

Тест Бройша-Пагана на гомоскедастичность: $BP = 3.45$, $p = 0.178$ Максимальный фактор инфляции дисперсии (VIF): 2.3.

Таблица 4.

Результаты регрессионного анализа

Модель	R^2	F-статистика	p-значение	95% ДИ для R^2
Производительность труда	0.47	F(5, 80) = 14.23	< 0.001	[0.31, 0.59]
Качество	0.53	F(5, 80) = 18.06	< 0.001	[0.37, 0.64]

3.5 Лонгитюдный анализ

Дисперсионный анализ с повторными измерениями выявил значимые основные эффекты Группы ($F(2, 83) = 22,37$, $p < 0,001$, частичная $\eta^2 = 0,35$) и Времени ($F(2, 166) = 45,62$, $p < 0,001$, частичная $\eta^2 = 0,36$), а также значимое взаимодействие Группа х Время ($F(4, 166) = 8,93$, $p < 0,001$, частичная $\eta^2 = 0,18$).

Таблица 5.

Результаты дисперсионного анализа с повторными измерениями

Эффект	F-статистика	p-значение	Частичная η^2
Группа	F(2, 83) = 22.37	< 0.001	0.35
Время	F(2, 166) = 45.62	< 0.001	0.36
Группа х Время	F(4, 166) = 8.93	< 0.001	0.18

3.6 Моделирование структурными уравнениями

Анализ SEM показал хорошее соответствие модели данным ($\chi^2(24) = 32,76$, $p = 0,109$; CFI = 0,97; TLI = 0,95; RMSEA = 0,058, 90% ДИ [0,000, 0,102]; SRMR = 0,045).

Структурная модель SEM:

$$\eta1 = \gamma11\xi1 + \gamma12\xi2 + \gamma13\xi3 + \zeta1 \; \eta2 = \beta21\eta1 + \gamma21\xi1 + \gamma22\xi2 + \gamma23\xi3 + \zeta2$$

где: η1 - латентная переменная «Эффективность использования ИИ»

η2 - латентная переменная «Общая производительность труда»

ξ1 - экзогенная переменная «Использование ИИ»

ξ2 - экзогенная переменная «Опыт работы»

ξ3 - экзогенная переменная «Открытость к инновациям»

γij - коэффициенты влияния экзогенных переменных на эндогенные

β21 - коэффициент влияния η1 на η2

ζi - остаточные члены

3.7 Общее повышение эффективности труда

Анализ общей эффективности труда показал, что использование метода привело к значительному повышению производительности труда.

При сравнении групп, использующих помощь ИИ (CCA и CoCA), с контрольной группой, было обнаружено общее повышение эффективности труда на 18,7% (95% ДИ: [15.2%, 22.2%], $F(2, 83) = 31.42$, $p < 0.001$, $\eta^2 = 0.43$, 95% ДИ η^2: [0.29, 0.54]).

Таблица 6.

Общее повышение эффективности труда

Группа	Повышение эффективности труда (%)	F-статистика	p-значение	η^2	95% ДИ для η^2
ИИ-ассистированная	18.7	F(2, 83) = 31.42	< 0.001	0.43	[0.29, 0.54]

Этот показатель эффективности труда был рассчитан как комбинированная метрика, учитывающая как повышение производительности труда (скорость выполнения задач и коэффициент завершения), так и улучшение качества труда.

Формула расчета: Общая эффективность труда = (Прирост производительности труда × 0.6) + (Прирост качества × 0.4) где 0.6 и 0.4 - весовые коэффициенты, отражающие относительную важность производительности труда и качества соответственно.

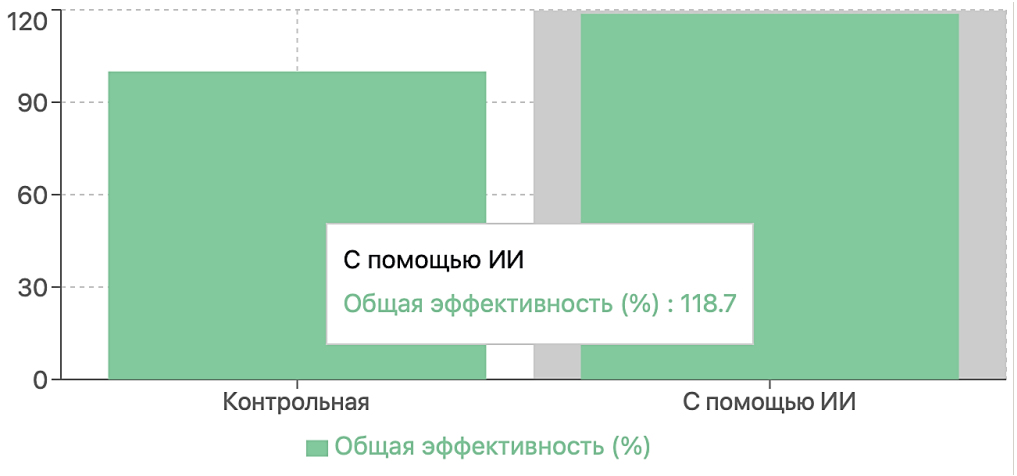

Рисунок 4. График общего повышения эффективности труда

4. Обсуждение

Результаты данного исследования предоставляют эмпирическую поддержку эффективности труда в повышении производительности труда и качества выполнения задач. Значительные улучшения, наблюдаемые как в показателях производительности труда (скорость выполнения задач и коэффициент завершения), так и в результатах качества, согласуются с предыдущими исследованиями работы с помощью ИИ [10, 8, 9, 2].

Наблюдаемое общее повышение эффективности труда на 18,7% подчеркивает значительный потенциал метода в улучшении рабочих процессов в ключевых отраслях. Этот показатель, учитывающий как количественные (производительность труда), так и качественные аспекты труда, предоставляет комплексную оценку влияния интеграции ИИ на эффективность труда. Такое существенное улучшение может иметь значительные результаты для организаций, стремящихся оптимизировать свои рабочие процессы и повысить конкурентоспособность.

Дифференцированное влияние на работников с разным уровнем производительности труда особенно интересно. Более значительное улучшение у работников с низкой производительностью труда (43%, 95% ДИ: [35%, 51%]) по сравнению с высокопроизводительными работниками (17%, 95% ДИ: [9%, 25%]) указывает на потенциал метода для выравнивания производительности труда в коллективе.

Общее повышение эффективности труда на 18,7% (95% ДИ: [15.2%, 22.2%]) представляет собой существенное улучшение, которое может иметь значительные экономические результаты при широком внедрении.

Лонгитюдный анализ показывает, что группы с ИИ демонстрировали большее улучшение квалификации с течением времени, что указывает на эффект обучения при использовании инструментов ИИ.

4.1 Ограничения и направления будущих исследований

Следует отметить несколько ограничений текущего исследования:

Размер и разнообразие выборки: Выборка исследования из 86 сотрудников российских, швейцарских и гонконских компаний ограничивает обобщаемость результатов. Будущие исследования должны использовать более крупные, разнообразные выборки из нескольких стран и отраслей.

Период обучения: На эффективность труда помощи ИИ может влиять продолжительность и качество обучения сотрудников. Будущие исследования могли бы изучить влияние различных подходов и продолжительности обучения на результаты .

Долгосрочные эффекты: Хотя наше исследование включало лонгитюдный компонент, необходимы более длительные исследования для оценки устойчивого воздействия на производительность труда.

Специфика задач: Исследование было ограничено двумя типами задач. Будущие исследования могли бы расширить спектр задач, включив, например, прогнозное моделирование или анализ рисков, чтобы оценить эффективность труда в более широком контексте работы.

4.2 Практические импликации

Результаты данного исследования позволяют сформулировать ряд рекомендаций для организаций, рассматривающих интеграцию генеративного ИИ в рабочие процессы:

Учитывая более значительное повышение производительности труда у сотрудников с изначально низкими показателями (43% против 17% у высокопроизводительных работников), организациям рекомендуется уделять особое внимание внедрению ИИ-ассистентов среди менее опытных или менее продуктивных сотрудников.

Принимая во внимание наблюдаемый эффект обучения при использовании ИИ с течением времени, организациям следует разработать комплексные программы обучения, включающие как начальное ознакомление с инструментами ИИ, так и долгосрочную поддержку для максимизации преимуществ интеграции.

Принимая во внимание значительное улучшение качества труда (на 40%) при использовании ИИ, организациям следует внедрить системы мониторинга качества для оценки долгосрочного влияния интеграции ИИ на стандарты работы.

При внедрении ИИ-ассистентов организациям следует разработать четкие этические руководства по использованию ИИ, уделяя особое внимание вопросам конфиденциальности данных и прозрачности процесса принятия решений.

4.3 Роль ИИ в повышении эффективности труда

Результаты исследования указывают на то, что генеративный ИИ в контексте метода симбиотического повышения производительности труда функционирует преимущественно как высокоэффективный инструмент поиска и предложения оптимальных решений из базы известных подходов. Важно отметить, что наблюдаемое повышение производительности труда не свидетельствует о кардинальном изменении квалификации сотрудников, а о более эффективном использовании знаний.

5. Заключение

Результаты исследования демонстрируют значительный потенциал симбиотического использования генеративного ИИ для повышения производительности труда в бизнес секторе экономики, приводя к общему повышению эффективности труда на 18,7%. Интегрируя помощь ИИ в рабочие процессы сотрудников, организации могут достичь значительного повышения эффективности труда. Однако необходимы дальнейшие исследования для валидации этих результатов в различных контекстах и изучения долгосрочных последствий интеграции ИИ на рабочем месте.

Дальнейшие исследования могут быть направлены на изучение долгосрочного влияния использования ИИ-ассистентов на развитие квалификации сотрудников и потенциальные риски чрезмерной зависимости от автоматизированных систем поиска решений.

Список литературы:

1. Amazon EU Store Transparency Report. (2023).
2. Brynjolfsson, E., Li, D., & Raymond, L.R. (2023). Generative AI at work. National Bureau of Economic Research.
3. Calvino, F., & Fontanelli, L. (2023). A portrait of AI adopters across countries: Firm characteristics, assets' complementarities and productivity. OECD Science, Technology and Industry Working Papers.
4. Chui, M., Hazan, E., Roberts, R., Singla, A., Smaje, K., Sukharevsky, A., & Yee, L. (2023). The economic potential of generative AI: The next productivity frontier. McKinsey Global Institute.
5. Czarnitzki, D., Fernández, G.P., & Rammer, C. (2023). Artificial intelligence and firm-level productivity. Journal of Economic Behavior & Organization, 211, 188-205.
6. Filippucci, F., Gal, P., Jona-Lasinio, C., Leandro, A., & Nicoletti, G. (2024). The impact of artificial intelligence on productivity, distribution, and growth: Key mechanisms, initial evidence, and policy challenges. OECD Artificial Intelligence Papers, No. 15.
7. Microsoft and LinkedIn. (2024). 2024 Work Trend Index annual report.
8. Nielsen, J. (2023). AI improves employee productivity by 66%.
9. Nielsen, J. (2023). AI vastly improves productivity for business users and reduces skill gaps.
10. Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. Economics.
11. Otis, N.G., Clarke, R., Delecourt, S., Holtz, D., & Koning, R. (2024). The Uneven Impact of Generative AI on Entrepreneurial Performance. Berkeley Haas; Harvard Business School.

Приложение

Подробная методика расчета показателя сохранения ИИ и дополнительные статистические анализы

Определение и компоненты

Методика расчета Показателя сохранения ИИ (ARS) является оригинальной разработкой для данного исследования. Она была создана с целью количественной оценки эффективности труда интеграции ИИ в рабочие процессы в рамках метода .

Показатель сохранения ИИ (ARS) был разработан для количественной оценки степени, в которой сотрудники включают идеи, ответы или контент, сгенерированный ИИ, в свои конечные рабочие продукты в рамках метода симбиотического повышения производительности труда с использованием генеративного ИИ. ARS включает пять ключевых компонентов:

Частота использования ИИ (F): Количество раз, когда сотрудник взаимодействует с выходными данными ИИ во время выполнения задачи.

Степень интеграции (D): Степень, в которой выходные данные ИИ включаются в конечный рабочий продукт.

Уровень модификации (M): Степень, в которой сгенерированный ИИ контент модифицируется перед включением.

Коэффициент зависимости (D): Соотношение задач, выполненных с помощью ИИ, к общему количеству выполненных задач.

Соответствие качеству (Q): Насколько хорошо предложения ИИ соответствуют желаемым стандартам качества.

Сбор данных

Для каждого сотрудника и задачи собирались следующие данные:

F: Количество использованных предложений ИИ

D: Процент предложений ИИ, использованных в конечном результате

M: Процент модифицированных предложений ИИ

D: Количество задач, выполненных с помощью ИИ / Общее количество задач

Q: Средний показатель качества выходных данных, созданных с помощью ИИ

Процедура расчета

Нормализация: Каждый компонент нормализовался до шкалы 0-1 с использованием формулы:

$$\text{Нормализованный компонент} = (x - \min(x)) / (\max(x) - \min(x))$$

где x - значение компонента.

Присвоение весов: Каждому компоненту были присвоены веса на основе их относительной важности:

Частота использования ИИ (F'): 20%

Степень интеграции (D'): 25%

Уровень модификации (M'): 15%

Коэффициент зависимости (D'): 20%

Соответствие качеству (Q'): 20%

Расчет показателя: Показатель сохранения ИИ рассчитывался как взвешенная сумма нормализованных компонентов:

$$ARS = (F' \times 0{,}20) + (D' \times 0{,}25) + (M' \times 0{,}15) + (D' \times 0{,}20) + (Q' \times 0{,}20)$$

Масштабирование: Для более легкой интерпретации ARS масштабировался до шкалы 0-100:

$$ARS_масштабированный = ARS \times 100$$

Статистические анализы

Описательная статистика

Были рассчитаны средние значения, стандартные отклонения и диапазоны для всех переменных, включая показатель сохранения ИИ и его компоненты, а также меры производительности труда и качества.

Инференциальная статистика

Т-тесты для независимых выборок: Использовались для сравнения средних показателей производительности труда и качества между группами с помощью ИИ (CCA и CoCA) и контрольной группой.

Однофакторный дисперсионный анализ (ANOVA) проводился для оценки различий в производительности труда, качестве и показателях сохранения ИИ между всеми тремя группами (CCA, CoCA и контрольная).

Множественная линейная регрессия применялась для моделирования взаимосвязи между помощью ИИ, показателями сохранения ИИ и результатами производительности труда при контроле других переменных, таких как опыт сотрудников, тип задачи и открытость к инновациям.

Хи-квадрат тесты использовались для анализа распределения результатов качества (высокое, среднее, низкое) между группами.

Дисперсионный анализ с повторными измерениями применялся для анализа изменений в производительности труда и показателях сохранения ИИ с течением времени для участников в разных группах.

Моделирование структурными уравнениями (SEM) реализовано для исследования сложных взаимосвязей между переменными, включая показатели сохранения ИИ, и проверки общей пригодности предложенной модели .

Анализ надежности и валидности

Внутренняя согласованность Альфа Кронбаха была рассчитана для компонентов показателя сохранения ИИ для оценки их внутренней согласованности.

Надежность тест-ретест где каждый Подгруппа участников выполняла те же задачи после четырехнедельного интервала для оценки стабильности показателей сохранения ИИ во времени.

Конструктная валидность: Был проведен факторный анализ для изучения базовой структуры компонентов показателя сохранения ИИ.

Конвергентная валидность: Были рассчитаны корреляции между показателями сохранения ИИ и другими мерами интеграции и эффективности труда ИИ.

Дискриминантная валидность: Были изучены корреляции между показателями сохранения ИИ и теоретически не связанными конструктами для обеспечения различительной способности показателя.

Таблица 7.

Результаты анализа надежности и валидности

Метрика	Значение	Интерпретация
Альфа Кронбаха	0.89	Высокая внутренняя согласованность
Надежность тест-ретест	0.85	Хорошая стабильность во времени
Конвергентная валидность	r = 0.76	Сильная связь с другими мерами интеграции ИИ
Дискриминантная валидность	r = 0.12	Слабая связь с не связанными конструктами

Альфа Кронбаха ($\alpha = 0.89$): Рассчитана с использованием пакета psych в R. Значение указывает на высокую внутреннюю согласованность компонентов показателя сохранения ИИ.

Надежность тест-ретест ($r = 0.85$): Вычислена как корреляция Пирсона между показателями ARS, полученными в двух временных точках с интервалом в две недели. Значение свидетельствует о хорошей стабильности показателя во времени.

Конвергентная валидность ($r = 0.76$): Рассчитана как корреляция Пирсона между ARS и другой валидированной мерой интеграции ИИ (например, Индекс использования ИИ). Сильная положительная корреляция подтверждает, что ARS измеряет схожий конструкт.

Дискриминантная валидность ($r = 0.12$): Вычислена как корреляция Пирсона между ARS и теоретически не связанным конструктом (например, общая удовлетворенность работой). Низкая корреляция подтверждает, что ARS измеряет уникальный конструкт.

Все корреляции были значимыми на уровне $p < 0.001$.

Анализ размера эффекта и мощности

d Коэна было рассчитано для всех значимых результатов t-тестов для количественной оценки величины различий между группами.

Частичная эта-квадрат (η^2) была вычислена для результатов ANOVA для оценки доли дисперсии, объясняемой каждым фактором.

Значения R^2 были представлены для регрессионных анализов для указания доли дисперсии зависимой переменной, объясняемой предикторами.

Были проведены апостериорные анализы мощности для определения достигнутой мощности для обнаружения значимых эффектов с учетом размера выборки и наблюдаемых размеров эффекта.

Проверки устойчивости

Бутстрэппинг: Были сгенерированы 5000 бутстрэп-выборок для создания устойчивых доверительных интервалов для ключевых параметров, включая коэффициенты регрессии и размеры эффекта.

Анализ чувствительности: Были протестированы альтернативные схемы взвешивания для компонентов показателя сохранения ИИ для оценки устойчивости результатов к различным подходам оценки.

Анализ выбросов: Были применены диагностические меры влияния (например, расстояние Кука) для выявления потенциальных выбросов, и анализы были повторены с этими случаями и без них для обеспечения стабильности результатов.

Все статистические анализы были проведены с использованием R (версия 4.1.0) с дополнительными пакетами, включая lavaan для SEM-анализов и boot для бутстрэппинга. Уровень значимости был установлен на $\alpha = 0,05$ для всех инференциальных тестов, и 95% доверительные интервалы были представлены, где это применимо.

DOI - 10.32743/UniTech.2024.127.10.18388

ОПТИМИЗАЦИЯ ПРОИЗВОДИТЕЛЬНОСТИ БАЗ ДАННЫХ MONGODB В МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ

Глумов Константин Сергеевич
ведущий инженер-программист,
Альфа-Банк
РФ, г. Пермь
E-mail: glumovk@gmail.com

OPTIMIZING THE PERFORMANCE OF MONGODB DATABASES IN A MICROSERVICE ARCHITECTURE

Konstantin Glumov
Lead software engineer at Alfa-Bank
Russia, Perm

АННОТАЦИЯ

Микросервисная архитектура стала основным подходом к проектированию современных распределенных систем, обеспечивая гибкость, масштабируемость и независимость компонентов. В этой архитектуре MongoDB, одна из самых популярных NoSQL баз данных, играет ключевую роль благодаря своим возможностям горизонтального масштабирования и обработки больших объемов данных. Однако с увеличением количества сервисов и объемов данных возникают проблемы, связанные с производительностью, что может снижать эффективность системы. Данное исследование направлено на изучение методов оптимизации MongoDB в контексте микросервисной архитектуры. Основные области оптимизации включают улучшение запросов, моделирование данных, использование кэширования, а также мониторинг и управление ресурсами. Оптимизация этих факторов помогает повысить производительность базы данных, обеспечивая стабильную работу системы в условиях динамической нагрузки и необходимости масштабирования. Цель исследования — анализ существующих методов и разработка рекомендаций для повышения эффективности MongoDB в распределенных системах, что особенно актуально для обеспечения надежности и высокой производительности в современных условиях.

ABSTRACT

Microservice architecture has become the primary approach to designing modern distributed systems, providing flexibility, scalability, and component independence. In this architecture, MongoDB, one of the most popular NoSQL databases, plays a crucial role due to its horizontal scalability and ability to handle large volumes of data. However, as the number of services and data volumes grow, performance issues may arise, potentially reducing the system's overall efficiency. This study focuses on examining methods to optimize MongoDB performance within microservice architecture. Key areas of optimization include query improvement, data modeling, caching, as well as resource monitoring and management. Optimizing these factors enhances database performance, ensuring stable and responsive system operation under dynamic load and scaling needs. The goal of the research is to analyze existing methods and provide recommendations for improving MongoDB efficiency in distributed systems, particularly to ensure reliability and high performance in modern environments.

Ключевые слова: оптимизация производительности, MongoDB, микросервисы, шардирование, репликация, кэширование, мониторинг, запросы.

Keywords: performance optimization, MongoDB, microservices, sharding, replication, caching, monitoring, queries.

Введение

В современных условиях микросервисная архитектура стала одним из основных подходов к проектированию распределённых систем, обеспечивая гибкость, масштабируемость и независимость компонентов. Важнейшим элементом такой архитектуры являются базы данных, которые должны эффективно обрабатывать большие объёмы данных, обеспечивать высокую доступность и производительность. MongoDB, как одна из наиболее популярных NoSQL баз данных, находит широкое применение в микросервисных системах благодаря своим возможностям горизонтального масштабирования, поддержке обработки больших данных и гибкости в управлении информацией.

Актуальность исследования обусловлена возрастающей потребностью в оптимизации работы

Библиографическое описание: Глумов К.С. ОПТИМИЗАЦИЯ ПРОИЗВОДИТЕЛЬНОСТИ БАЗ ДАННЫХ MONGODB В МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18388*

MongoDB в микросервисной архитектуре. С увеличением количества сервисов и объёма данных возникают сложности, связанные с производительностью баз данных, что может привести к снижению эффективности всей системы. Оптимизация производительности MongoDB является важным условием для обеспечения стабильной и быстрой работы микросервисов, особенно в условиях динамически изменяющихся нагрузок и масштабирования.

Среди ключевых аспектов, которые необходимо учитывать при работе с MongoDB в микросервисной архитектуре, выделяются оптимизация запросов, правильное моделирование данных, использование кэширования, а также мониторинг и управление ресурсами базы данных. Решение этих задач требует применения комплексного подхода, который обеспечит не только высокую производительность баз данных, но и их надёжность в условиях высокой нагрузки.

Цель работы заключается в изучении и анализе существующих методов оптимизации производительности MongoDB в микросервисной архитектуре, а также разработка рекомендаций для повышения эффективности работы баз данных в распределённых системах.

Материалы и Методы

MongoDB представляет собой высокомасштабируемую и адаптивную систему управления базами данных на основе документов, относящуюся к классу NoSQL решений. Она была создана для устранения недостатков реляционных баз данных и улучшения других существующих NoSQL технологий. Благодаря возможности горизонтального масштабирования и распределения нагрузки, MongoDB предоставляет разработчикам неограниченные возможности по увеличению производительности и гибкости при работе с данными. Документы в MongoDB хранятся в формате BSON — бинарном представлении JSON. Этот формат обеспечивает более быструю работу с данными по сравнению с текстовым JSON, что особенно полезно для работы с мультимедийными данными. Кроме того, BSON предоставляет возможность хранения бинарных данных, таких как изображения и видео, что делает MongoDB отличным выбором для проектов с интенсивной обработкой данных [1].

Для проектирования микросервисной архитектуры мы будем применять инструменты Spring Boot и Spring Cloud. Такая архитектура выглядит следующим образом: в контексте Spring.

Для запуска данного проекта потребуется несколько служб, о которых будет рассказано отдельно. Во время работы с этим проектом, рекомендуется следовать инструкциям, изложенным в файле README.md, для каждой из служб.

Первый сервис, который необходим, — это сервер конфигурации. Он служит для хранения всех конфигурационных данных наших микросервисов в одном репозитории, что облегчает управление версиями и упрощает процесс работы с конфигурациями. Далее будет отражен пример простой настройки конфигурационного сервера:

```
yaml
spring.application.name=config-server
server.port=8888
spring.cloud.config.server.git.uri=${HOME}/Work/microservices-architecture-mongodb-config-repo
spring.cloud.config.label=main
```

Этот сервер позволяет получать доступ к git-репозиторию, в котором сохраняются конфигурационные данные микросервисов, и использовать нужную ветку. Важной частью настройки является аннотация @EnableConfigServer, которая активирует необходимую функциональность для работы сервера конфигурации.

Пример кода:

```
java
package com.mongodb.configserver;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.config.server.EnableConfigServer;

@EnableConfigServer
@SpringBootApplication
public class ConfigServerApplication {
   public static void main(String[] args) {
      SpringApplication.run(ConfigServerApplication.class, args);
   }
}
```

Реестр служб выполняет функции каталогизации всех микросервисов, их текущих IP-адресов и портов. Это упрощает процесс взаимодействия между сервисами. Этот компонент полезен для балансировки нагрузки и устранения зависимости между потребителями и поставщиками услуг.

Аннотация @EnableEurekaServer выполняет необходимые действия для включения реестра в систему:

```
java
package com.mongodb.serviceregistry;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication
@EnableEurekaServer
public class ServiceRegistryApplication {
    public static void main(String[] args) {
        SpringApplication.run(ServiceRegistryApplication.class, args);
    }
}
```

Настройка реестра может выглядеть следующим образом:

```
yaml
spring.application.name=service-registry
server.port=8761
eureka.client.register-with-eureka=false
eureka.client.fetch-registry=false
```

API-шлюз служит для объединения доступа ко всем микросервисам через единую точку входа. Это обеспечивает централизованную безопасность, мониторинг и отказоустойчивость. После регистрации микросервисов в реестре API-шлюз может распределять запросы между ними, исходя из конфигурации маршрутизации.

Пример настройки API-шлюза:

```
yaml
server:
  port: 8080

spring:
  application:
    name: api-gateway
  cloud:
    gateway:
      routes:
        - id: company-service
          uri: lb://company-service
          predicates:
            - Path=/api/company/**
        - id: employee-service
          uri: lb://employee-service
          predicates:
            - Path=/api/employee/**
eureka:
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://localhost:8761/eureka/
  instance:
    hostname: localhost
```

Для обеспечения независимости каждого микросервиса используются отдельные экземпляры MongoDB для каждого из них. Пример конфигурации микросервиса компании:

```
yaml
spring.data.mongodb.uri=${MONGODB_URI_1:mongodb://localhost:27017}
server.port=8081
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
```

Конфигурация микросервиса сотрудников аналогична, с отличием в порте MongoDB:

```
yaml
spring.data.mongodb.uri=${MONGODB_URI_2:mongodb://localhost:27018}
server.port=8082
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true
eureka.client.service-url.defaultZone=http://localhost:8761/eureka/
```

Это позволяет микросервисам работать независимо, используя разные кластеры MongoDB. После запуска всех служб (конфигурационного сервера, реестра услуг, API-шлюза и микросервисов) можно протестировать их взаимодействие через REST API [2].

Для успешного внедрения MongoDB в микросервисную архитектуру необходимо учитывать несколько аспектов:

- Четкое определение границ: границы между микросервисами должны быть определены в зависимости от бизнес-функционала и контекста предметной области.
- Ответственность за данные: каждая служба должна быть ответственна за управление своей моделью данных, что подразумевает контроль над операциями создания, чтения, обновления и удаления (CRUD).
- Уровень доступа к данным: создание отдельного уровня доступа, отвечающего за взаимодействие с MongoDB, значительно упрощает сопровождение и тестирование микросервисов.

Таким образом использование MongoDB в сочетании с микросервисами предоставляет разработчикам значительную гибкость, однако такая архитектура может привести к росту сложности. Важно сбалансировать потребность в изолированных сервисах с издержками, связанными с управлением множеством баз данных и поддержанием целостности данных на уровне всей системы [3].

Результаты и Обсуждения

Оптимизация запросов — это процесс анализа и модификации запросов с целью уменьшения времени их выполнения и сокращения потребления ресурсов. В MongoDB предусмотрен специальный инструмент — метод explain(), который позволяет детально анализировать запросы и находить потенциальные зоны для улучшения их производительности.

Метод explain() возвращает расширенные сведения о процессе выполнения запроса, такие как план его выполнения, использование индексов и время обработки. Данные, предоставленные этим методом, помогают выявить проблемные места в запросах, позволяя провести их дальнейшую оптимизацию для улучшения производительности.

```
javascript
// Анализ производительности запроса
db.collection.find({ поле: значение }).explain();
```

Проекция данных как способ оптимизации запросов. Позволяет уменьшить объем передаваемой информации и сокращает время выполнения запроса, что особенно важно при работе с большими объемами данных [4].

```
javascript
// Извлечение только необходимых полей
db.collection.find({ поле: значение }, { поле1: 1, поле2: 1 });
```

Использование пагинации для работы с большими наборами данных является эффективным способом обработки больших наборов данных, который подразумевает разбиение результатов на страницы с помощью методов limit() и skip(). Это помогает уменьшить нагрузку на сервер и улучшить время отклика.

```
javascript
// Получение данных частями
db.collection.find({}).skip(10).limit(10);
```

Кэширование часто используемых данных позволяет существенно уменьшить количество запросов к базе данных и ускорить время отклика. Использование решений вроде Redis для кэширования данных в памяти снижает нагрузку на MongoDB и повышает общую производительность. При структурировании данных важно минимизировать количество запросов, необходимых для получения нужной информации. Некоторые стратегии включают использование вложенных документов, массивов и денормализации, что позволяет избежать сложных операций соединения данных. Выбор правильных типов данных напрямую влияет на эффективность работы с базой данных. MongoDB поддерживает множество типов данных, и правильный их выбор для каждого поля обеспечивает более быструю обработку запросов. Например, хранение дат в формате дат позволяет эффективно выполнять запросы с диапазонами дат.

Постоянный мониторинг и плановое обслуживание являются обязательными для поддержания оптимальной работы MongoDB. Использование встроенных инструментов мониторинга, таких как db.serverStatus(), позволяет отслеживать состояние системы, а регулярная оптимизация индексов и уплотнение данных помогают предотвратить падение производительности [5].

Основные рекомендации по работе с агрегациями в MongoDB

Агрегации в MongoDB играют ключевую роль при работе с данными, особенно когда требуется получение аналитических выводов. Конвейеры агрегации позволяют автоматически обрабатывать данные, избегая ручного вмешательства, что минимизирует риск ошибок и повышает эффективность.

Однако важно правильно настроить конвейер агрегации, чтобы избежать потенциальных проблем с производительностью базы данных. К примеру, выполнение запроса без использования фильтров может значительно нагрузить систему [6].

Когда один из ресурсов, таких как CPU или дисковый I/O, достигает предела своих возможностей — когда постоянный рост графика использования останавливается на одном уровне — это свидетельствует о возникновении дефицита данного ресурса, что может повлиять на производительность системы. Чаще всего такая ситуация возникает, если лимиты использования данного ресурса были полностью исчерпаны.

Высокий уровень использования CPU и значительные показатели Disk IO зачастую указывают на проблемы с неэффективными индексами или чрезмерную нагрузку на серверные узлы. Для начала диагностики необходимо определить характер нагрузки и выявить коллекции, вызывающие затруднения. В этом могут помочь встроенные средства мониторинга MongoDB. Анализируя логи и данные профилировщика, можно оценить эффективность выполнения запросов.

Стоит уделить особое внимание запросам:

- Не использующим индексы (planSummary: COLLSCAN), так как такие запросы приводят к увеличению I/O операций с диском и нагрузки на процессор (дефолтная компрессия данных требует дополнительных ресурсов для их декомпрессии). В случае, если индекс существует, но не задействуется, можно использовать команду hint для его принудительного использования.
- Имеющим высокие значения параметра docsExamined, что указывает на необходимость более эффективных индексов или создания новых.
- Для оперативной диагностики проблемных запросов в момент снижения производительности можно воспользоваться командой отображения текущих запросов в системе:
  - Запросы всех пользователей
  - Запросы текущего пользователя

Для выполнения данных операций пользователь должен обладать соответствующей ролью mdbMonitor. Следует также помнить, что каждый новый индекс может замедлить операции записи, поэтому важно использовать индексы с осторожностью. Для повышения эффективности запросов на чтение рекомендуется применять частичное извлечение (projection), что позволяет сократить объём данных, которые необходимо передать. В случае возникновения проблем с блокировками (locks), можно диагностировать их через анализ показателей writeConflicts и изучение текущих операций:

```
bash
db.currentOp({'$or': [{'locks.Global': 'W'}, {'locks.Database': 'W'}, {'locks.Collection': 'W'} ]}).inprog
```

Также можно найти запросы, ожидающие снятия блокировок:

```
bash
db.currentOp({'waitingForLock': true}).inprog
```

Если кластер начинает работать в режиме read-only, возможно, на одном из узлов возник дефицит свободного дискового пространства. Для мониторинга использования диска настройте оповещения и контролируйте графики по каждому хосту [7].

Эффективное управление базами данных, включая MongoDB, невозможно без регулярного мониторинга их состояния. Это критически важный элемент для поддержания стабильности работы кластера и оптимального использования ресурсов. Несмотря на то, что MongoDB Atlas, являясь платформой Database-as-a-Service, предоставляет встроенные механизмы для управления производительностью и масштабируемостью, пользователям все равно необходимо понимать ключевые аспекты мониторинга для того, чтобы предотвратить возможные проблемы заранее.

В процессе наблюдения за MongoDB следует уделить внимание четырем основным областям, которые будут отражены на рисунке 1.

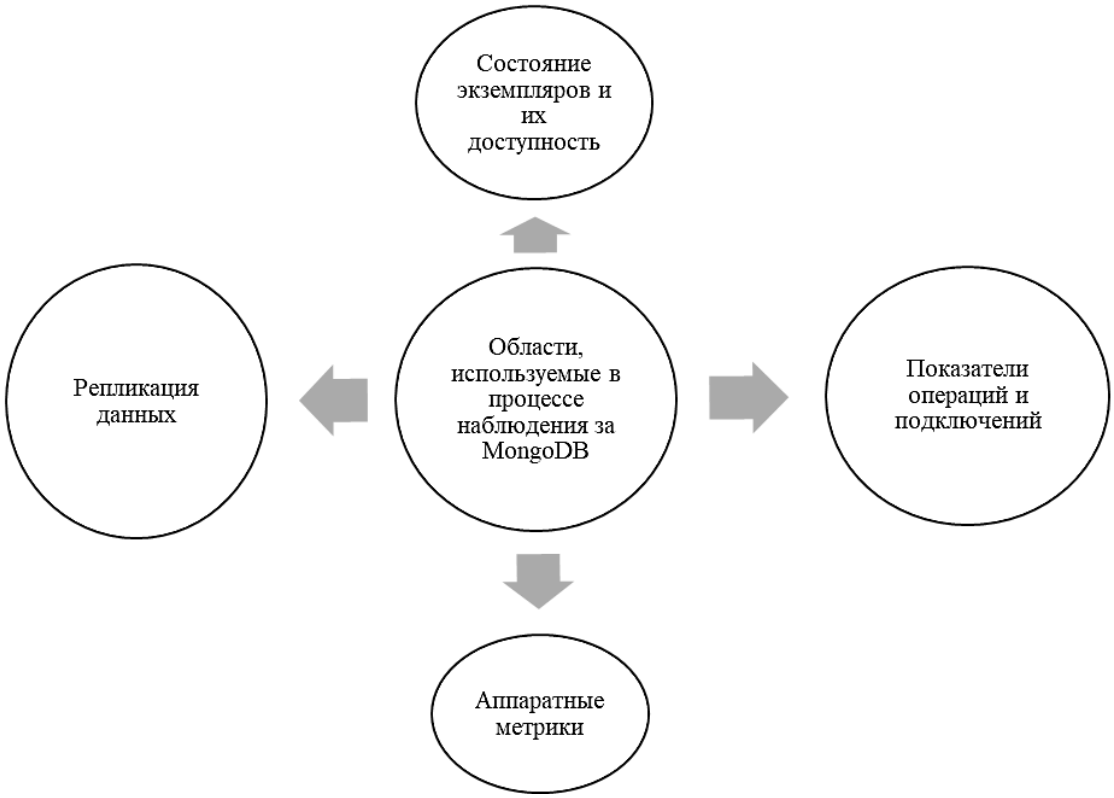

Рисунок 1. Области, используемые в процессе наблюдения за MongoDB [8]

Каждое из этих направлений можно детализировать с помощью различных команд и инструментов. Например, для контроля состояния экземпляров можно использовать команды rs.status() и sh.status(), которые позволяют отслеживать репликацию и состояние сегментированных кластеров. MongoDB Atlas предоставляет удобные интерфейсы для наблюдения за состоянием кластеров, используя визуальные метрики для отображения всех критических параметров.

Аппаратные ресурсы также требуют постоянного наблюдения. Высокая загрузка ЦП, дисковая задержка или недостаток оперативной памяти могут напрямую влиять на производительность MongoDB. Для этого MongoDB Atlas предоставляет графики, отображающие показатели работы оборудования, такие как системный ЦП, свободное место на диске и использование оперативной памяти. Это помогает своевременно принимать меры для предотвращения сбоев и ухудшения работы системы.

Надежность и доступность данных в MongoDB обеспечиваются механизмом репликации, который также требует мониторинга. Важно отслеживать такие показатели, как задержка репликации и размер Oplog, чтобы убедиться, что данные копируются корректно и своевременно. MongoDB предоставляет встроенные команды для контроля этих метрик, такие как rs.status() и rs.printReplicationInfo(), которые позволяют оперативно реагировать на возможные проблемы.

Для анализа работы MongoDB можно использовать как встроенные инструменты MongoDB Atlas, так и автономные команды, такие как mongostat и mongotop. Эти команды позволяют собирать информацию о состоянии серверов в режиме реального времени, отслеживать выполнение операций и контролировать использование ресурсов. Например, mongostat предоставляет краткий обзор состояния экземпляра, включая статистику памяти, соединений и очередей блокировок, что помогает поддерживать стабильную работу баз данных [8].

Заключение

Оптимизация производительности MongoDB в микросервисной архитектуре требует комплексного подхода, включающего не только оптимизацию запросов и правильное моделирование данных, но и внедрение эффективных инструментов мониторинга и кэширования. Мониторинг состояния базы данных и регулярное обслуживание обеспечивают стабильную работу системы, а использование технологий шардирования и репликации способствует повышению масштабируемости и отказоустойчивости. Важно помнить, что каждая база данных в микросервисной архитектуре должна быть изолированной, и её производительность должна поддерживаться на высоком уровне за счёт продуманной архитектуры и регулярной оптимизации процессов.

Список литературы:

1. Nicolau D. Functional elements specific to MongoDB databases // Romanian Cyber Security Journal. – 2021. – Т. 3. – №. 1. – С. 39-49.
2. Нуркаев Р., Пивоваров В.В. Управление API-шлюзом на основе архитектуры микросервиса // Инновации и инвестиции. – 2023. – №. 5. – С. 193-197.
3. Бердникова А.С. Проблема потоковой обработки данных с использованием микросервисной архитектуры в аналитических системах // Цифровые, компьютерные и информационные технологии в науке и. – 2023. – С. 267.
4. Закирова Ю.М. Метод оптимизации поисковых запросов в СУБД MongoDB // Фундаментальные и прикладные научные исследования: инноватика в современном Мире. – 2020. – С. 30-35.
5. Чуб В.С. Интеграция MongoDB в процесс обучения нейронных сетей: эффективное хранение и управление данными для глубокого обучения //Молодой исследователь Дона. – 2024. – Т. 9. – №. 1. – С. 35-38.
6. Буйволов Е.А. Сравнение производительности современных NoSQL баз данных с реляционной базой данных Sybase ASA 9. 02 //Проблемы науки. – 2020. – №. 6 (54). – С. 26-30.
7. Трофименко М.С., Кумратова А.М., Пополитов А.В. Изучение принципов работы NOSQL и SQL баз данных и их сравнение на примере Postgres Pro и MongoDB //Цифровизация экономики: направления, методы, инструменты. – 2021. – С. 342-347.
8. Галиева Р.Р., Чадин В.Н. Методы автоматизации Асу Тп методической печи // Эффективное обеспечение научно-технического прогресса. – 2024. – С. 28.

DOI - 10.32743/UniTech.2024.127.10.18441

УЛУЧШЕНИЕ КАЧЕСТВА ТИКЕТОВ В ТЕСТИРОВАНИИ ПО: СНИЖЕНИЕ ВОЗВРАТОВ ЗАДАЧ И УСКОРЕНИЕ ИСПРАВЛЕНИЯ БАГОВ

Домнин Евгений Константинович
инженер по тестированию в TaxDome LLC,
Нью-Йорк, г. Бродвей
E-mail: e.domnin@proton.me

IMPROVING TICKET QUALITY IN SOFTWARE TESTING: REDUCING TASK RETURNS AND ACCELERATING BUG FIXES

Evgeny Domnin
Quality Assurance Engineer at Taxdome LLC,
New York, Broadway

АННОТАЦИЯ

В статье анализируются рекомендации по улучшению качества оформления тикетов специалистом по тестированию. Основная цель – проверить эффективность этих методов. На основе данных, полученных в результате анализа, в статье делается вывод действительно ли данные практические рекомендации ускоряют время разработки. Описанные в статье методы и практические рекомендации полезны в первую очередь специалистам по тестированию программного обеспечения, а также разработчикам.

ABSTRACT

The article analyzes recommendations for improving the quality of ticket formatting by a testing specialist. The primary objective is to assess the effectiveness of these methods. Based on the data obtained from the analysis, the article concludes whether these practical recommendations indeed accelerate development time. The methods and practical recommendations described in the article are primarily beneficial for software testing specialists, as well as developers.

Ключевые слова: Тестирование программного обеспечения, Оптимизация процесса разработки, Регрессионное тестирование, Jira, QA.

Keywords: Software Testing, Development Process Optimization, Regression Testing, Jira, QA.

Введение

Тестирование и разработка программного обеспечения являются ключевыми этапами в жизненном цикле программных систем (SDLC) [7]. Эффективность этих процессов во многом зависит от качества взаимодействия между участниками этих этапов. Основное взаимодействие происходит через систему тикетов. Тикеты – это инструмент, используемый для отслеживания задач, ошибок, запросов на функции и других аспектов, связанных с проектом. Система тикетов помогает организовать рабочий процесс и распределение задач среди участников команды [4].

Одной из ключевых проблем в процессе разработки и тестирования является недостаточная степень детализации багов в тикетах. Некачественно составленные репорты увеличивают время, которое разработчики тратят на локализацию багов, а также затрудняют воспроизведение ошибок, что ведет к возвратам задач на доработку и, в конечном счете, к снижению общей эффективности команды [1].

В этой статье мы разберем результаты применения практических рекомендаций и их влияние на количество тикетов, возвращаемых на доработку.

Материалы и методы исследования

Тестирование эффекта рекомендаций проводилось с помощью системы баг трекинга Jira. Анализ тикетов происходил на основе наличия или отсутствия pull request, связанного с задачей. Наличие pull request подразумевало изменения в коде, а его отсутствие – недостаток данных и необходимость уточнения.

Все практические рекомендации для тестировщиков были применены с использованием следующих методологий тестирования: Black Box Testing, Regress testing.

В целях автоматизации сбора данных использовался плагин ScriptRunner for Jira. Он позволил собрать тикеты с типом «Баг» и заполненным полем Pull requests.

Библиографическое описание: Домнин Е.К. УЛУЧШЕНИЕ КАЧЕСТВА ТИКЕТОВ В ТЕСТИРОВАНИИ ПО: СНИЖЕНИЕ ВОЗВРАТОВ ЗАДАЧ И УСКОРЕНИЕ ИСПРАВЛЕНИЯ БАГОВ // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18441*

Описание инструментов и методов

Jira — это система управления проектами и задачами, которая широко используется в разработке программного обеспечения для отслеживания и координации процессов. Основное назначение Jira заключается в управлении жизненным циклом разработки, начиная с планирования и заканчивая выпуском продукта. Она предоставляет инструменты для создания, назначения и управления задачами, которые чаще всего представлены в виде тикетов (issues).

Jira поддерживает гибкие методологии разработки, такие как Scrum и Kanban [6], и позволяет создавать доски для визуализации задач, что помогает командам отслеживать прогресс в реальном времени. Созданный тикет проходит флоу от создания до его выпуска на production окружение.

Ниже представлен стандартный рабочий процесс задачи при отсутствии необходимости в уточнении или корректировке.

Рисунок 1. Схема

В случае недостатка данных от тестировщика тикет возвращается разработчику и цикл повторяется.

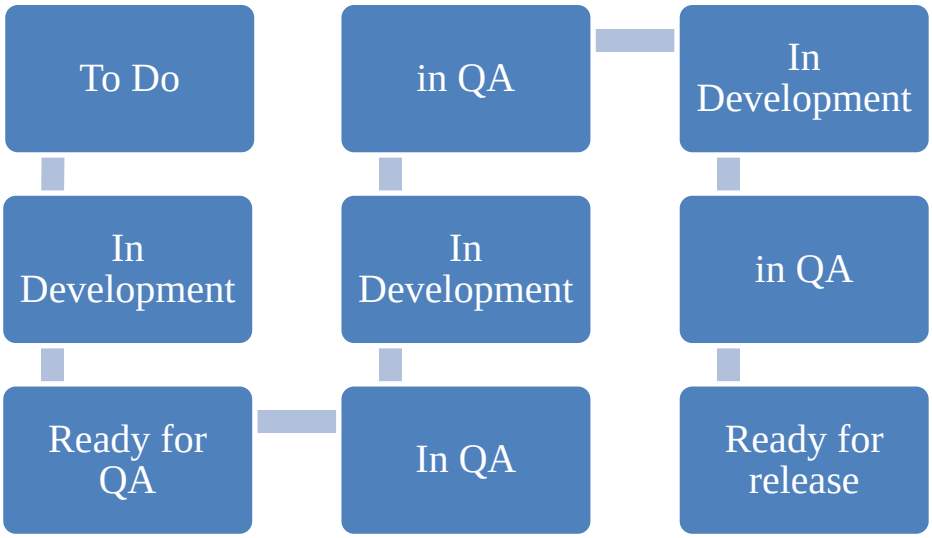

Рисунок 2. Схема

Черный ящик (black-box-testing) — это подход в тестировании, который фокусируется на проверке функциональности программного обеспечения без знания его внутренней структуры [5] [9].

Регрессионное тестирование — проверяло, соответствует ли функциональность приложения заявленным требованиям и действительно ли исправлены ошибки, описанные в тикете. Также использовалось для проверки того, что новые изменения в коде не привели к появлению новых ошибок [10] [11].

ScriptRunner for Jira — плагин для автоматизации и расширения функциональности Jira с помощью скриптов на языке Groovy. Он добавляет специальные функции и фильтры в стандартный Jira Query Language (JQL) и помогает достать специфичную информацию по тикетам.

Для получения всех тикетов за период с типом Баг и заполненным полем Pull Request мы использовали следующий код:

issueFunction in statusChangedTo("Ready for QA") AND issuetype = Bug

AND development[pullrequests].all = 0

AND updated >= "2024-06-01" AND updated <= "2024-09-30"

Таблица 1.

Пример вывода запроса в ScriptRunner

Ключ тикета	Резюме	Статус	Исполнитель	Дата перехода в Ready for QA
BUG-101	Ошибка при авторизации	Ready for QA	Пользователь 1	01.07.2024
BUG-102	Некорректное отображение данных	Ready for QA	Пользователь 2	02.07.2024
BUG-103	Сбой при сохранении настроек	Ready for QA	Пользователь 3	03.07.2024

Практические рекомендации

Описание окружения, где происходит баг

Первым и основополагающим этапом является указание окружения, в котором воспроизводится ошибка. Это позволяет разработчику воспроизвести баг в аналогичных условиях. Важные аспекты, которые необходимо указать:

1. Операционная система, на которой проводился тест
2. Браузер и его версия
3. Состояние сети (например, наличие ограничений скорости соединения, использование сетевого троттлинга)

Данные о пользователе

В некоторых случаях ошибка может быть связана с конкретными данными пользователя или уникальными условиями его сессии. В таких ситуациях рекомендуется указывать следующие данные:

1. Аутентификационные данные (логин и пароль), если это безопасно.
2. Дополнительные специфические условия (например, наличие cookie-файлов или кастомных значений в localStorage, которые могли быть изменены вручную)

Подробное описание ошибки

Одним из ключевых аспектов является детальное описание ошибки. Важно сначала определить, с чем связана ошибка — с фронтендом или бекендом, — и далее привести соответствующие данные:

Для фронтенда

1. Скриншот или видео с демонстрацией ошибки на клиентской стороне.
2. Логи из консоли браузера
3. Название компонента или модуля интерфейса, где произошла ошибка

Для бекенда

1. Скриншот или видео с демонстрацией ошибки на клиенте
2. Код ответа запроса, который завершился с ошибкой
3. URL запроса
4. Ответ сервера
5. Логи из систем мониторинга Sentry/AppSignal
6. Логи из AWS Console, если они доступны.

Дополнительные файлы

В некоторых случаях полезно приложить файлы, которые могут помочь воспроизвести ошибку или провести её анализ:

1. Логи с ошибками. Если объём логов значителен, рекомендуется прикрепить их в виде файла для удобства анализа.
2. Специфические файлы, которые могут влиять на воспроизведение ошибки (например, файлы, связанные с загрузкой).

Повторное прохождение по описанным шагам

Этот этап обязателен для проверки воспроизводимости ошибки перед передачей тикета разработчику. Тестировщик должен пройти все описанные шаги ещё раз, чтобы убедиться, что ошибка воспроизводится в тех условиях, которые указаны в тикете. Этот процесс также может выявить дополнительные аспекты ошибки, ранее не замеченные, и предоставить новые артефакты для дальнейшего анализа.

Результаты исследования и их обсуждение

Для оценки эффективности внедрённых рекомендаций был проведён анализ в течение трёх месяцев после их применения. Результаты отслеживались в конце каждого квартала. Для этого использовались данные, автоматически собранные с помощью инструмента Jira и плагина ScriptRunner, что было описано ранее в методах исследования. Основным показателем для сравнения выступало количество тикетов, возвращённых на доработку, которое сравнивалось с аналогичным периодом до внедрения рекомендаций.

Количественные результаты

В первом **квартале**, до внедрения рекомендаций, было зарегистрировано 318 тикетов с типом “Баг”. Из них 17% были возвращены на доработку в связи с недостаточной информацией, предоставленной тестировщиками.

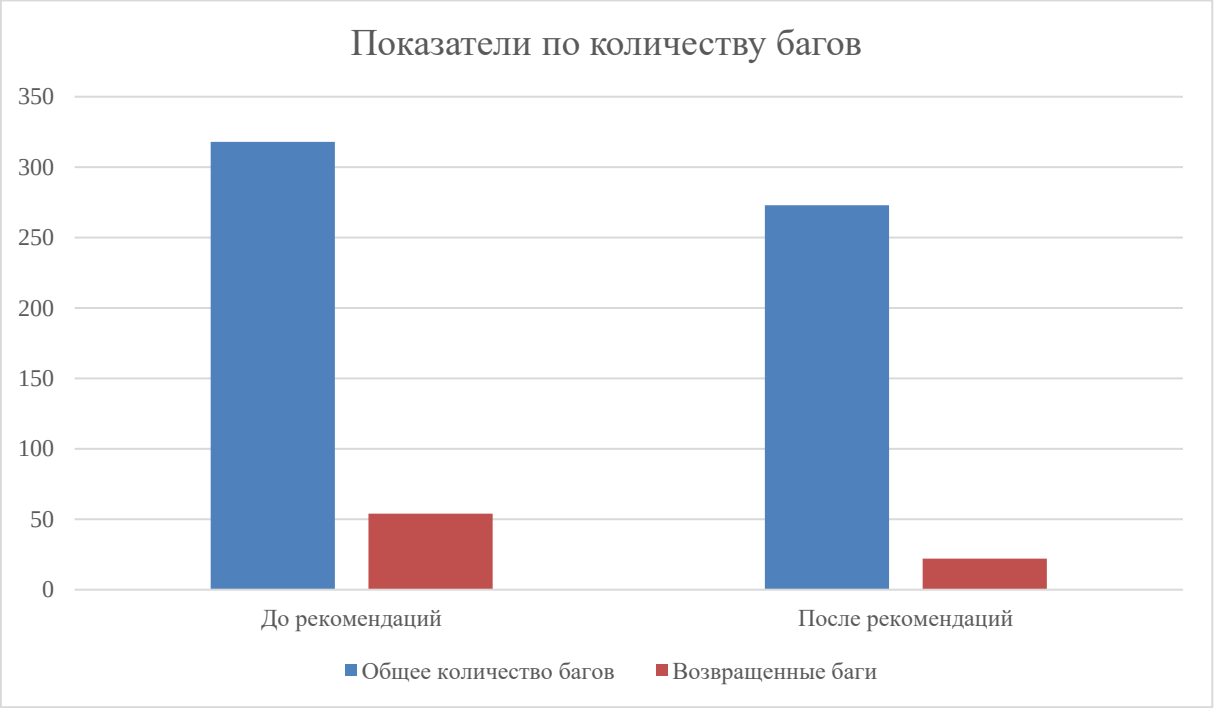

Рисунок 3. Показатели по количеству багов

Возврат тикетов тестировщикам с запросами на предоставление дополнительных данных или более точного описания ошибок приводил к увеличению общего времени на устранение дефектов.

В **новом периоде**, после внедрения рекомендаций, было обработано 273 тикета со статусом “Баг”. Из этих тикетов 8% были возвращены на доработку для уточнения данных.

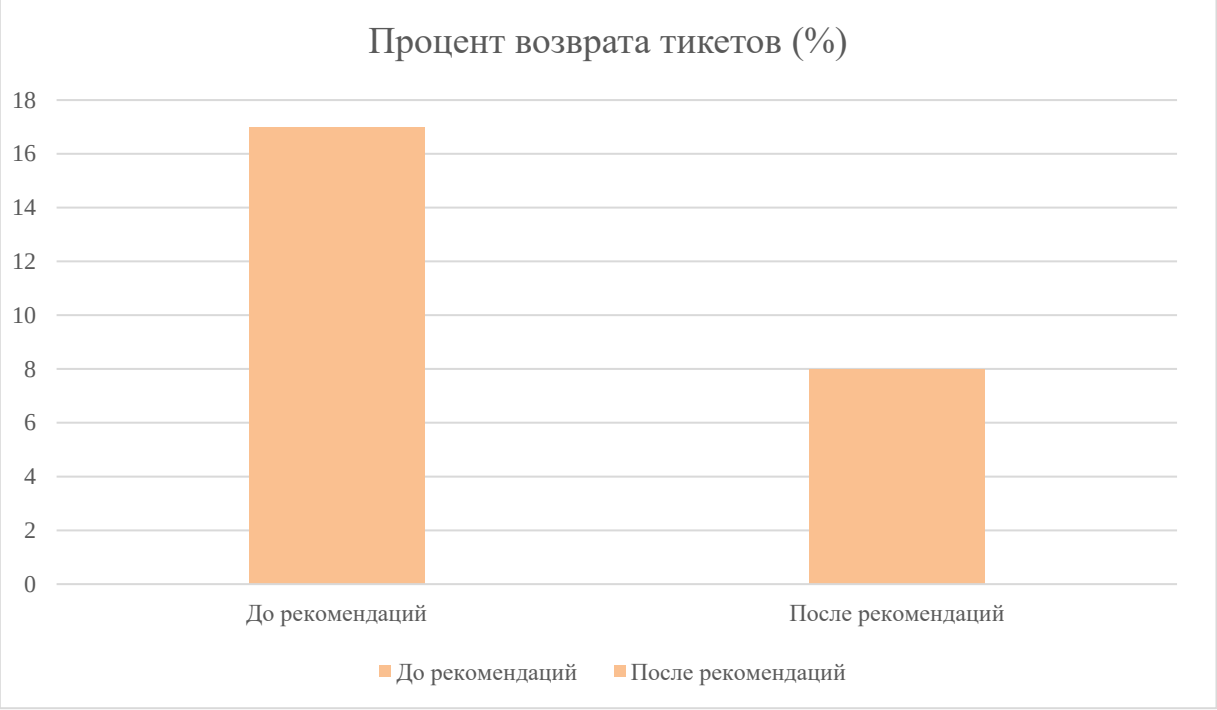

Рисунок 4. Процент возврата тикетов (%)

Для количественной оценки уменьшения возвратов тикетов на доработку была применена следующая формула расчета процентного изменения:

$$\Delta\% = \frac{N_{before} - N_{after}}{N_{before}} \times 100$$

где:$\Delta\%$ — процентное снижение возвратов тикетов на доработку.

N_{before} — количество тикетов, возвращённых без изменений в коде, до внедрения практических рекомендаций,

N_{after} — количество тикетов, возвращённых без изменений в коде, после внедрения практических рекомендаций.

Применив данную формулу к нашим данным:

$$\Delta\% = \frac{17\% - 8\%}{17\%} \times 100 = 52.94\%$$

Таким образом, мы получили снижение числа возвратов тикетов на доработку на **52.94%** по сравнению с предыдущим периодом. Данный результат указывает на значительное улучшение качества первичного описания багов, предоставляемого тестировщиками.

Выводы

Полученные результаты демонстрируют, что внедрение предложенных практических рекомендаций для тестировщиков положительно сказалось на качестве оформляемых тикетов. Как видно из проведённого анализа, количество возвратов тикетов на доработку из-за недостаточной информации существенно сократилось.

Применение вышеописанной формулы расчета процентного изменения показало снижение числа возвратов тикетов на 52.94%, что подтверждает значительное улучшение процессов коммуникации между тестировщиками и разработчиками. Это означает, что тестировщики стали предоставлять более полные и детализированные описания багов, что в свою очередь позволило ускорить процесс их исправления.

Улучшение первоначального описания ошибок тестировщиками способствует более оперативной работе разработчиков, уменьшая общее время на исправление багов.

Эти изменения положительно влияют на общую эффективность командной работы, снижая затраты на доработку и улучшая качество выпускаемого продукта [12] [8].

Дальнейшие исследования могут быть направлены на изучение долгосрочного влияния этих рекомендаций, а также на применение автоматизированных инструментов для повышения качества тикетов.

В качестве рекомендаций для дальнейшего улучшения качества тестирования и разработки программного обеспечения можно предложить автоматизировать подобную аналитику с помощью встроенных репортов в системе Jira в совокупности с ScriptRunner плагином. Таким образом все участники команды смогут анализировать динамику.

Результаты данного исследования имеют важное практическое значение не только для оптимизации процессов внутри конкретной компании, но и для широкого круга организаций, работающих с системами баг-трекинга, такими как Jira и аналогичными инструментами управления задачами. Внедрение предложенных рекомендаций может применяться в любых командах, занимающихся разработкой программного обеспечения, независимо от специфики проекта или масштаба организации.

Основное практическое значение работы заключается в повышении эффективности взаимодействия между тестировщиками и разработчиками, что приводит к уменьшению времени на исправление ошибок и, соответственно, к ускорению релиза продукта. Предложенная в исследовании методология может быть легко адаптирована и интегрирована в различные рабочие процессы команд, использующих гибкие методологии разработки, такие как Scrum и Kanban.

Кроме того, полученные данные и выводы могут служить основой для дальнейших исследований в области автоматизации контроля качества тикетов, разработки новых инструментов для улучшения взаимодействия между командами и повышения качества конечного продукта. В теоретическом плане исследование демонстрирует важность качественной коммуникации на этапах тестирования и разработки, что может стать отправной точкой для изучения других аспектов взаимодействия внутри команд разработки ПО [2].

Список литературы:

1. Chaparro O., Bernal-Cárdenas C., Lu J., Moran K., Marcus A., Penta M., Poshyvanyk D., Ng V. Assessing the quality of the steps to reproduce in bug reports // Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering [Электронный ресурс]. — 2019. — Режим доступа: https://doi.org/10.1145/3338906.3338947 (дата обращения: 01.10.2024).
2. Cruzes D., Moe N., Dybå T. Communication between Developers and Testers in Distributed Continuous Agile Testing // 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE). — 2016. — С. 59-68. https://doi.org/10.1109/ICGSE.2016.27.
3. Fazzini M., Moran K., Cardenas C.B., Wendland T., Orso A., Poshyvanyk D. Enhancing Mobile App Bug Reporting via Real-Time Understanding of Reproduction Steps // IEEE Transactions on Software Engineering. — 2022. — Т. 49. — С. 1246-1272.

4. Gohil F., Kumar M. Ticketing System // International Journal of Trend in Scientific Research and Development. — 2019. https://doi.org/10.31142/ijtsrd23603.
5. Kaprocki Z., et al. Combined testing approach: Increased efficiency of black box testing // 2015 IEEE 1st International Workshop on Consumer Electronics (CE WS). — 2015. — C. 76-78. — doi: https://doi.org/10.1109/CEWS.2015.7867160.
6. Lei H., Ganjeizadeh F., Jayachandran P., Ozcan P. A statistical analysis of the effects of Scrum and Kanban on software development projects // Robotics and Computer-integrated Manufacturing. — 2017. — T. 43. — C. 59-67. https://doi.org/10.1016/J.RCIM.2015.12.001.
7. Parihar M., Bharti A. Importance Of Software Testing In Software Development Life Cycle // International Journal of Research. — 2019. — T. 6. — C. 888-900.
8. Tassey G. The economic impacts of inadequate infrastructure for software testing. — 2002.
9. Verma A., Khatana A., Chaudhary S. A Comparative Study of Black Box Testing and White Box Testing // International Journal of Computer Sciences and Engineering. — 2017. — T. 5. — C. 301-304. — doi: 10.26438/ijcse/v5i12.301304.
10. Wahl N. An overview of regression testing // ACM SIGSOFT Software Engineering Notes. — 1999. — T. 24. — C. 69-73. https://doi.org/10.1145/308769.308790.
11. Wong W.E., Horgan J.R., London S., Agrawal H. A study of effective regression testing in practice // Proceedings The Eighth International Symposium on Software Reliability Engineering, Albuquerque, NM, USA. — 1997. — C. 264-274. — doi: 10.1109/ISSRE.1997.630875.
12. Zimmermann T., Premraj R., Bettenburg N., Just S., Schröter A., Weiss C. What Makes a Good Bug Report? // IEEE Transactions on Software Engineering. — 2008. — T. 36. — C. 618-643. https://doi.org/10.1145/1453101.1453146.

DOI - 10.32743/UniTech.2024.127.10.18445

ИСПОЛЬЗОВАНИЕ МОДУЛЬНЫХ МОНОЛИТОВ ДЛЯ РАЗРАБОТКИ МАСШТАБИРУЕМЫХ WEB-ПРИЛОЖЕНИЙ

Земсков Максим Александрович
руководитель группы, Яндекс,
Сербия, г. Белград
E-mail: nodge@yandex-team.com

USING MODULAR MONOLITHS TO DEVELOP SCALABLE WEB APPLICATIONS

Maksim Zemskov
Lead Software Engineer, Yandex,
Serbia, Belgrade

АННОТАЦИЯ

Модульные монолиты представляют собой перспективный подход в разработке масштабируемых веб-приложений. Этот архитектурный стиль сочетает преимущества классических монолитных приложений с гибкостью микрофронтендов. Основной идеей модульных монолитов является разделение системы на автономные модули с четко определенными границами, минимальными зависимостями и изоляцией данных. Это позволяет упростить сопровождение и тестирование приложения, а также увеличить его производительность и масштабируемость.

Одним из ключевых аспектов является строгая модульная структура, которая достигается за счет использования специализированных программных интерфейсов и обеспечения модульной изоляции. Такой подход способствует улучшению архитектуры приложения, снижению технической сложности и повышению устойчивости к изменениям, что делает его особенно актуальным для проектов с возрастающей сложностью и большими объемами данных.

ABSTRACT

Modular monoliths represent a promising approach in the development of scalable web applications. This architectural style combines the advantages of classic monolithic applications with the flexibility of microfrontends. The main idea of modular monoliths is to divide the system into autonomous modules with well-defined boundaries, minimal dependencies, and data isolation.

This makes it easier to maintain and test the application, as well as increase its performance and scalability. One of the key aspects is the strict modular structure, which is achieved through the use of specialized application programming interfaces and ensuring modular isolation. This approach helps to improve the application architecture, reduce technical complexity, and increase resilience to changes, which makes it especially relevant for projects with increasing complexity and large amounts of data.

Ключевые слова: модульные монолиты, масштабируемые веб-приложения, изоляция данных, модульная архитектура, программные интерфейсы, паттерны проектирования, системный дизайн, микрофротенд.

Keywords: modular monoliths, scalable web applications, data isolation, modular architecture, software interfaces, design patterns, system design, microfrotend.

Введение

Современная разработка веб-приложений сталкивается с растущей сложностью систем и увеличивающимися требованиями к их производительности и масштабируемости. В условиях ускоренной цифровизации бизнеса и увеличения объемов данных, традиционные подходы к разработке монолитных приложений начинают демонстрировать свои ограничения.

Одной из наиболее актуальных проблем в этой области является необходимость адаптации архитектуры программных систем для обеспечения их гибкости и устойчивости к изменениям [1]. В связи с этим, концепция модульных монолитов приобретает все большую популярность среди разработчиков, предлагая сбалансированное решение между классическими монолитами и делением приложения на микрофронтенды.

Модульные монолиты представляют собой подход, который позволяет разделить систему на отдельные модули, сохраняя при этом единую кодовую базу. Каждый модуль выполняет свою бизнес-логику, минимизируя зависимости от других частей системы [2].

Библиографическое описание: Земсков М.А. ИСПОЛЬЗОВАНИЕ МОДУЛЬНЫХ МОНОЛИТОВ ДЛЯ РАЗРАБОТКИ МАСШТАБИРУЕМЫХ WEB-ПРИЛОЖЕНИЙ // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18445*

Этот подход предоставляет возможность более гибко управлять приложением, улучшать его масштабируемость и производительность, что особенно важно для крупных и сложных веб-приложений. В условиях роста требований к качеству и скорости разработки, архитектурный стиль модульных монолитов оказывается крайне актуальным.

Актуальность данной темы обусловлена необходимостью решения проблемы сохранения эффективности труда команды разработчиков, несмотря на быстрое увеличение кодовой базы современных веб-приложений. При увеличении объема функциональности системы важно сохранить высокую производительность и избежать деградации качества программного продукта, что может быть достигнуто через правильную организацию модульной структуры.

Преимущества модульных монолитов включают упрощение разработки, сокращение времени на внедрение новых функций и уменьшение риска ошибок за счет изоляции модулей и четко определенных программных интерфейсов.

Цель данной работы заключается в исследовании особенностей применения модульных монолитов для разработки масштабируемых веб-приложений, анализе их преимуществ и недостатков, а также оценке перспектив этого подхода в современных условиях разработки программного обеспечения.

1. Общая теория модульных монолитов

Модульность представляет собой подход к проектированию или производству объектов в виде отдельных компонентов.

Это определение является обобщённым и не совсем подходит для разработки программного обеспечения. Более точное определение можно дать через понятие модульного программирования. Модульное программирование – это метод создания программного обеспечения, основанный на разделении программы на независимые, взаимозаменяемые модули. Каждый из модулей включает все необходимые элементы для реализации определённой функциональной задачи.

Программный интерфейс модуля описывает доступные элементы для взаимодействия с другими модулями, а реализация модуля заключается в программном коде, соответствующем этим элементам [3]. Для того чтобы архитектура могла считаться модульной, модули должны:

1. Быть автономными и взаимозаменяемыми;
2. Содержать все необходимые компоненты для выполнения определённой части бизнес-логики;
3. Обладать чётко установленным программным интерфейсом.

Рассмотрим эти аспекты более подробно. Важно отметить, что стабильность модулей, от которых зависит рассматриваемый модуль, имеет большое значение. Чем реже изменяются эти модули, тем меньше они оказывают влияние на работу зависимого модуля. Эти положения отражены в принципе стабильной зависимости (Stable Dependency Principle).

В зависимости от конкретной архитектуры, контракт модуля может принимать различные формы. Это может быть фасад для синхронных вызовов, как в случае с простым вызовом функций, или же публикация событий для асинхронного взаимодействия.

Независимо от формы, контракт представляет публичный API модуля, делая инкапсуляцию ключевым элементом модульности [4]. Далее на рисунке 1 будет проиллюстрировано сравнение трёх архитектурных подходов к разработке масштабируемых веб-приложений: классического монолита, микрофронтенда и модульного монолита,

Линия, представляющая модульный монолит - выделена жёлтым цветом, демонстрирует более стабильное снижение скорости разработки по мере увеличения размера проекта по сравнению с традиционным монолитом и микрофронтендом. В малых проектах разработка модульного монолита сопряжена с большими трудностями по сравнению с классическим монолитом, что связано с увеличенными накладными расходами на проектирование модулей.

Этот процесс требует значительных усилий по структурированию системы на этапе проектирования, что может усложнить разработку. В то же время использование микрофронтенда предполагает существенные первоначальные вложения в инфраструктуру, что приводит к заметному снижению скорости разработки на начальных стадиях проекта. Однако по мере роста проекта производительность классического монолита значительно ухудшается. В таких условиях модульный монолит демонстрирует схожие показатели с микрофронтендом, несмотря на более скромные требования к инфраструктуре на ранних этапах разработки. Это свидетельствует о том, что модульный монолит лучше справляется с ростом сложности проекта и поддерживает производительность разработки на более высоком уровне в отличие от других архитектурных решений.

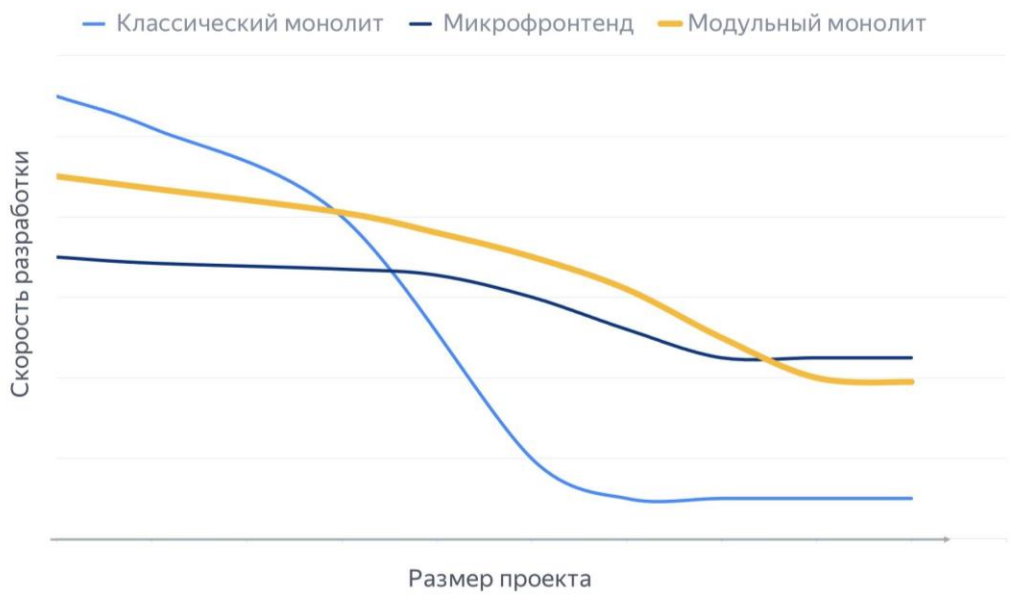

Рисунок 1. Сравнение трех архитектурных подходов к разработке масштабируемых веб-приложений

Таким образом, модульные монолиты представляют собой компромисс между классическим монолитом и микрофронтендом, предлагая более гибкий и масштабируемый подход к разработке веб-приложений, особенно при увеличении размеров проектов.

2. Правила модульных монолитов

Модульные монолиты строятся на основе идей, направленных на улучшение модульной структуры, повышения удобства обслуживания и улучшения масштабируемости в рамках единой кодовой базы. Выделяют 3 основных правила модулей, в частности ими являются:

1. Соблюдение строгой структуры модуля;
2. Обеспечение изоляции модулей между собой;
3. Унификация коммуникации между модулями.

Далее более подробно остановимся на каждом правиле.

Так если говорить о структуре модуля, то она является фундаментом модульных монолитов и предполагает строгую организацию кода в рамках чётко определённых модульных границ.

Основной задачей является обеспечение того, чтобы каждый модуль представлял собой отдельный компонент системы, отвечающий за определённую бизнес-логику или функциональность. Такой подход позволяет упрощать управление кодовой базой и делает приложение более читаемым и поддерживаемым [5]. Ниже на рисунке 2 будут представлены элементы структуры модуля.

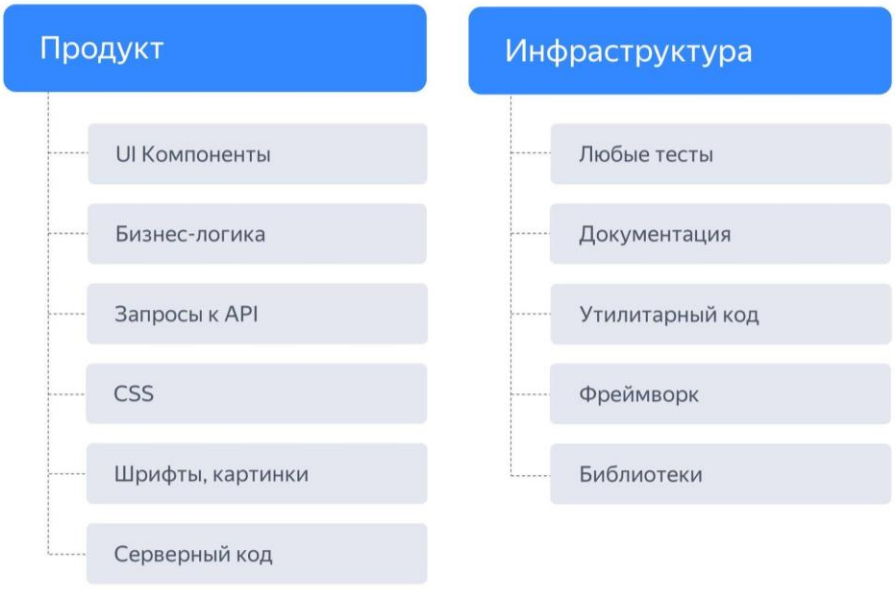

Рисунок 2. Элементы структуры модуля

В свою очередь крайне важно, чтобы каждый модуль был реализован похожим образом, иначе у команды могут возникнуть трудности с переключением между разработкой разных модулей и разных частей системы. Также стоит ограничить как набор технологий, так и верхний уровень структуры директории у модуля, например разделить весь код модуля логически на 4 или более сегментов.

Что позволит достичь того, что у каждого сегмента будет своя зона ответственности, как например это отражено ниже на рисунке 3.

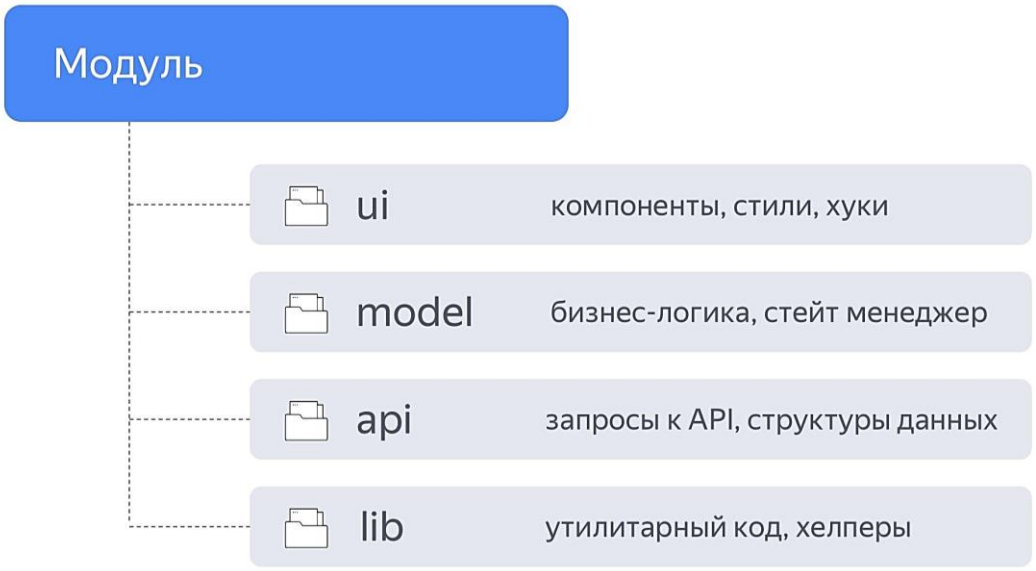

Рисунок 3. Сегменты модуля

Таким образом сегменты позволяют достичь следующего:

1) унифицировать кодовую базу
2) легко решить куда поместить новый код;
3) достичь быстрого переключения разработчика между модулями;
4) защитить код от смешивания UI и бизнес-логики.

Далее рассмотрим общую характеристику изоляции кода. Суть изоляции заключается в том, что каждый модуль должен минимизировать свои зависимости от других модулей, что позволяет избежать «эффекта домино», когда изменение в одном модуле приводит к необходимости изменения других модулей.

Это особенно критично для поддержания гибкости и возможности масштабирования системы в долгосрочной перспективе. Изоляция достигается за счёт чётко определённых программных интерфейсов, через которые модули могут взаимодействовать между собой.

Эти интерфейсы должны быть стабильными [6].

Первым важным аспектом изоляции является устранение побочных эффектов при подключении модуля. Под побочными эффектами подразумевается любое несанкционированное взаимодействие модуля с глобальными объектами, загрузка каскадных таблиц стилей (CSS) или изменение DOM-структуры других модулей. Например, доступ к глобальным объектам может привести к непредсказуемому поведению системы, если несколько модулей будут одновременно изменять одни и те же переменные. Изоляция помогает предотвратить такие конфликты, обеспечивая, что каждый модуль работает в рамках своей зоны ответственности, не влияя на работу других частей системы.

Контроль доступа к функциональности модуля — второй ключевой аспект изоляции. Для этого используется система импортов, которая позволяет четко определить, какие элементы кода (файлы, стили, ассеты) могут быть использованы вне модуля, а какие остаются внутри.

Важно также ограничить доступ к хранилищам данных, таким как IndexedDB или Local Storage, для предотвращения случайных или намеренных изменений данных модуля извне.

Эффективный контроль доступа гарантирует, что модули обмениваются только необходимой информацией, сохраняя внутреннюю целостность и безопасность данных.

Для организации контроля доступа на практике рекомендуется создавать индексный файл в корне каждого модуля.

Этот файл служит точкой входа, в которой определен круг функций и переменных, доступных другим модулям. Внутри индексного файла можно использовать различные техники, такие как ES модули или внедрение зависимостей, которые позволяют определить набор доступных методов для использования в других модулях.

Внедрение зависимостей (Dependency Injection, DI) является распространённым и эффективным методом, применяемым в разработке программного обеспечения. Применение DI способствует созданию гибкой архитектуры приложений, обеспечивая отделение логики создания зависимостей от их использования.

В этом подходе объекты, которые требуются для выполнения определённых операций, создаются и управляются внешними компонентами, а не самим объектом, использующим эти зависимости.

Принцип внедрения зависимостей требует, чтобы модули не полагались на конкретные реализации друг друга, а использовали абстракции. Такой подход даёт модулям полный контроль над моментом и условиями создания зависимости, а также над функциональностью модуля, доступной для использования в других модулях.

Основная задача внедрения зависимости — уменьшить связанность между модулями, обеспечивая заменяемость реализаций без изменения логики зависимого модуля. Это делает DI одним из ключевых элементов разработки в современных веб-приложениях и корпоративных системах [7].

Событийная модель (Event-driven), в свою очередь, позволяет организовать взаимодействие между модулями через события, что способствует снижению жестких зависимостей и улучшению гибкости архитектуры приложения.

Под событием может пониматься любое действие, совершенное пользователем, будь то нажатие на кнопку, прокрутка страницы или ввод данных в текстовое поле. Однако такой архитектурный подход находит свое применение не только в пользовательских интерфейсах, но и в разработке серверной части [8].

Ключевая концепция событийно-ориентированной архитектуры заключается в разделении системы на автономные модули, которые взаимодействуют между собой через события. Каждый модуль занимается обработкой лишь тех событий, которые имеют для него значение, что позволяет формировать независимые компоненты, которые можно разрабатывать по отдельности.

Эти компоненты, будучи объединены в сложные системы, сохраняют свою независимость и могут функционировать в автономном режиме [9].

Если же говорить об изоляции CSS, то она предусматривает чёткое разделение стилей приложения на независимые модули, каждый из которых несёт ответственность за свою область функциональности. Изоляция CSS в модульном монолите обеспечивает независимость стилей каждого модуля, что упрощает разработку новых модулей, делает более безопасным внесение изменений в существующие стили и способствует более простому масштабированию проекта.

Shadow DOM - один из ключевых компонентов спецификации Web Components, обеспечивающий полную изоляцию DOM и стилей для каждого компонента.

Он создаёт отдельное «теневое дерево», которое является самостоятельной частью основного DOM, и любые стили, применённые внутри этого теневого дерева, не могут воздействовать на другие элементы за его пределами.

Этот подход гарантирует полную изоляцию стилей внутри каждого компонента, что идеально подходит для модульных монолитов.

Преимущества использования Shadow DOM заключаются в том, что он обеспечивает строгое разделение не только стилей, но и структуры элементов. Стили, применённые внутри Shadow DOM, не могут «просочиться» наружу и изменить внешний DOM, и наоборот — глобальные стили не могут повлиять на элементы внутри теневого DOM. Это позволяет создавать полностью независимые модули, каждый из которых сохраняет уникальные визуальные особенности, независимо от глобальных стилей приложения.

Однако следует учитывать, что использование Shadow DOM может потребовать дополнительных усилий по интеграции, особенно если приложение активно взаимодействует с фреймворками, не имеющими встроенной поддержки Web Components. Тем не менее, для модульных монолитов, требующих высокой степени изоляции, это один из самых эффективных подходов [10].

Последним же правилом является изолирование данных приложения на уровне модулей.

Изоляция данных в модульных монолитах играет ключевую роль в разработке масштабируемых web-приложений, так как она способствует разделению ответственности между модулями и уменьшает взаимные зависимости компонентов системы. Модульный монолит подразумевает организацию кода таким образом, что каждая его часть выполняет строго определенные задачи, оставаясь независимой от других компонентов, что делает приложение более управляемым и устойчивым к изменениям.

Основной принцип изоляции данных заключается в предотвращении прямого доступа модулей к данным друг друга.

Это достигается за счет четкого разграничения областей видимости и использования специализированных программных интерфейсов для взаимодействия между модулями.

Такой подход позволяет контролировать, какие данные и методы их модификации доступны для использования другими частями системы, снижая риск возникновения побочных эффектов при внесении изменений в систему. При этом изоляция данных способствует упрощению процесса интеграции новых модулей и их модификацию, позволяя разработчикам вносить изменения локально, не затрагивая остальные части приложения [11].

Еще одним важным аспектом изоляции данных является использование надежного хранилища для данных модуля Важно, чтобы каждый модуль был автономным в плане хранения данных и работы с ними. Это не только способствует улучшению производительности системы за счет горизонтального масштабирования, но и повышает ее отказоустойчивость, так как сбои в одном модуле не приведут к критическим последствиям для всей системы. При этом изоляция данных способствует упрощению тестирования отдельных модулей, поскольку их поведение можно проверять в условиях, приближенных к реальным, без зависимости от внешних компонентов.

3. Рекомендации по делению кода на модули

В современном процессе разработки программного обеспечения особое внимание уделяется правильной организации кода, которая позволяет повысить читаемость, тестируемость и масштабируемость приложения.

Одним из подходов, который активно используется для достижения этих целей, является Feature Sliced Design (FSD), представляет собой подход к проектированию модульной архитектуры, который основывается на разделении функциональных областей приложения по различным аспектам, с учетом бизнес-логики, пользовательских интерфейсов и технических требований.

Одной из ключевых рекомендаций является разделение кода на слои, где каждый слой отвечает за определенный аспект работы приложения. В FSD выделяют несколько уровней:

- Приложение (App) представляет собой комплексный уровень, включающий как технические, так и бизнес-аспекты системы. Здесь могут располагаться компоненты, такие как провайдеры контекста, а также инструменты для аналитики. Основная задача данного уровня — загрузка всех остальных модулей приложения и настройка коммуникации между ними.
- Страницы (Pages) служат основой для создания полных страниц в веб-приложениях или экранов в мобильных приложениях. Этот уровень схож по структуре с Widgets, однако масштабы его элементов шире. Внутри каждого элемента на этом уровне содержатся компоненты интерфейса, которые могут быть напрямую интегрированы в навигацию приложения. Также здесь допускается логика для получения данных и обработки ошибок.
- Виджеты (Widgets) представляют собой независимые элементы пользовательского интерфейса, сформированные на базе более мелких единиц, таких как сущности и функции. Их задача — дополнять интерфейс интерактивными элементами. Чаще всего, бизнес-логика на этом уровне не хранится, поскольку она сосредоточена в фичах. Однако в случаях, когда виджет включает сложную интерактивность, бизнес-логика может быть размещена и здесь, особенно если она специфична для данного элемента.
- Функции (Features) направлены на реализацию пользовательских действий, которые позволяют взаимодействовать с бизнес-сущностями, достигая поставленных целей. Эти действия могут включать интерактивные элементы интерфейса, внутренние состояния и API-запросы, необходимые для выполнения операций, создающих ценность для пользователя.
- Сущности (Entities) — это ключевые концепты, лежащие в основе проекта и представляющие реальные бизнес-объекты. Содержимое этого уровня включает статические компоненты интерфейса, а также данные и API-запросы, необходимые для работы с этими сущностями.
- Общий (Shared) уровень состоит из универсальных модулей и абстракций, которые не привязаны к специфике конкретного проекта. Он не подчиняется структуре слайсов, характерной для других уровней, и содержит сегменты, которые могут быть использованы в различных контекстах [12].

Такое разделение помогает улучшить читаемость и поддерживаемость кода, делая его более модульным и независимым.

Feature Sliced Design предполагает атомарность модулей — каждый модуль должен быть минимально необходимым для выполнения своей задачи. Это значит, что если приложение состоит из нескольких модулей, каждый из них должен иметь чёткую и ограниченную зону ответственности. Такой подход делает код более устойчивым к изменениям и упрощает его тестирование и рефакторинг. Ниже на рисунке 4 будет представлен данный подход.

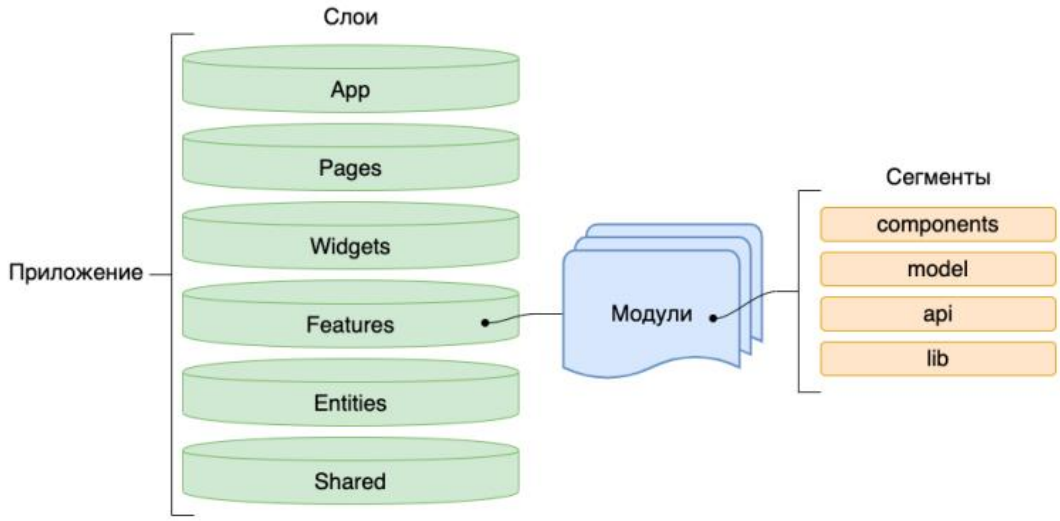

Рисунок 4. Feature Sliced Design

При разработке сложных программных систем критически важно уделять внимание правильному направлению импортов между модулями. Это не только обеспечивает изоляцию компонентов, но и предотвращает возникновение «циклических зависимостей». Такие зависимости могут привести

к трудностям в сопровождении и масштабировании кода. На рисунке 5 отражён пример данной рекомендации.

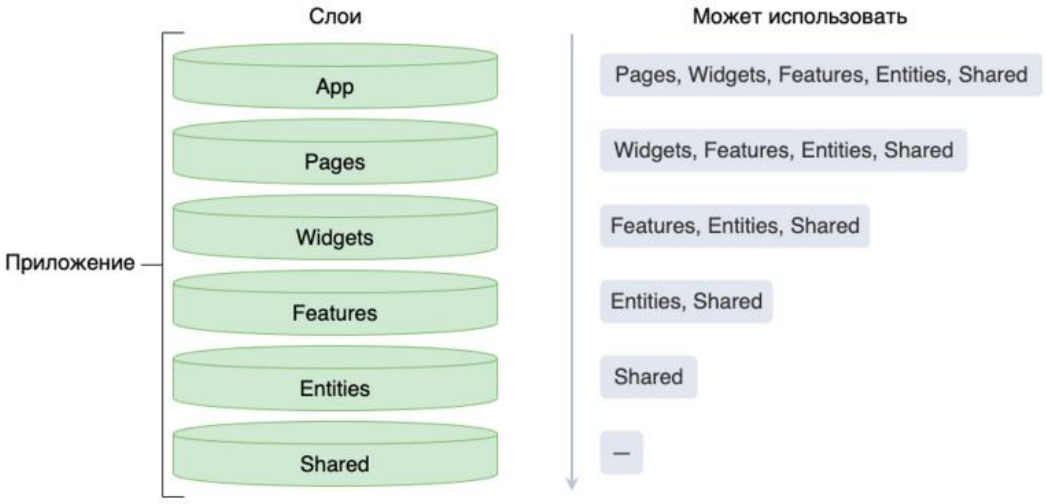

Рисунок 5. Направление импортов

Запрет зависимостей между модулями внутри одного слоя архитектуры является одним из ключевых принципов в Feature Slice Design. Этот подход, также известный как «горизонтальная изоляция», способствует улучшению структуры приложения, так как модули, находящиеся на одном уровне, не зависят друг от друга напрямую, что упрощает их тестирование, обновление и рефакторинг. Взаимодействие между модулями должно происходить через абстракции и интерфейсы, которые обеспечивают чёткое разделение ответственности и исключают прямые связи между компонентами. Таким образом, внесение изменений в один модуль не приводит к необходимости модифицировать другие модули, что значительно повышает гибкость системы, что отражено на рисунке 6.

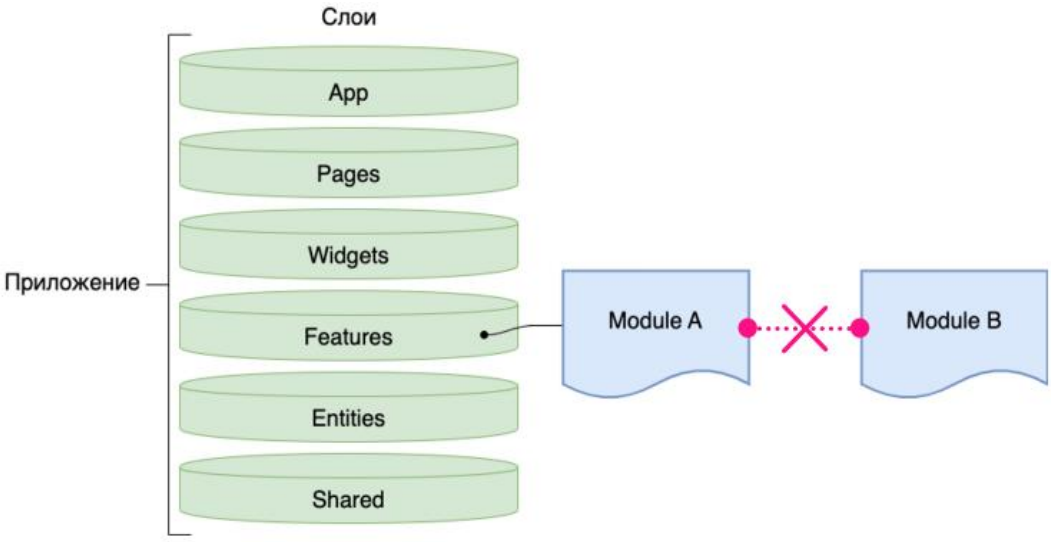

Рисунок 6. Запрет импортов внутри слоя

Таким образом строгая изоляция логики внутри модулей позволяет локализовать изменения, минимизировать риски и сделать приложение более устойчивым к изменениям. В результате, правильная организация направлений импортов обеспечивает модульность и надежность системы, снижая сложность ее поддержки и ускоряя процесс разработки.

Заключение

Использование модульных монолитов в разработке веб-приложений способствует значительному улучшению гибкости и масштабируемости системы. Разделение системы на независимые модули с четко определенными границами позволяет уменьшить

взаимозависимости компонентов и упростить управление кодом. Это делает процесс сопровождения и тестирования приложения более эффективным, так как каждый модуль может быть протестирован и модифицирован отдельно, не затрагивая остальные части системы.

То есть можно сказать, что модульные монолиты представляют собой сбалансированное решение между классической монолитной архитектурой и микрофронтендами. Они объединяют лучшие черты обоих подходов, предлагая гибкость и модульность микрофронтенда в рамках единой кодовой базы, что особенно важно для крупных проектов.

В долгосрочной перспективе такой подход способствует повышению устойчивости системы к изменениям и улучшению производительности разработки, делая модульные монолиты перспективным выбором для создания масштабируемых веб-приложений.

Список литературы:

1. Кряжева Е.В., Королев А.Е. Проектирование веб -приложения каталога пользовательских интерфейсов // Дневник науки. – 2024. – №. 8.
2. Шишкина К.С., Белаш В.Ю. Проектирование модели веб-приложения для студенческого общежития КГУ им. Циолковского // Дневник науки. – 2024. – №. 5.
3. Шахов А.Э. Архитектурное проектирование веб-приложений //Актуальные проблемы науки и образования в условиях современных вызовов (шифр–МКАП 24). – 2023. – С. 27-29.
4. Богданенко, Д.А. Подходы к архитектурному проектированию веб-приложений / Д.А. Богданенко // Молодой ученый. — 2018. — № 9 (195). — С. 24-29.
5. Кряжева Е.В., Васина Т.А. Общие подходы к проектированию ВЕБ-приложений // Заметки ученого. – 2021. – №. 9-2. – С. 32-36.
6. Кравченко Д.А. Микросервисная архитектура // Интерактивная наука. – 2022. – №. 4 (69). – С. 43.
7. Мытников А.Н., Анохин В.Р. Механизмы обеспечения информационной безопасности в фреймворке Nestjs // Современные вопросы естествознания и экономики. – 2023. – С. 574-579.
8. Su R., Li X. Modular Monolith: Is This the Trend in Software Architecture? //Proceedings of the 1st International Workshop on New Trends in Software Architecture. – 2024. – С. 10-13.
9. Архитектура event-driven приложений. [Электронный ресурс] Режим доступа: https://habr.com/ru/companies/otus/articles/740578/ (дата обращения 21.09.2024).
10. Деконструкция монолита: Максимально производительный подход к проектированию программ. [Электронный ресурс] Режим доступа: https://habr.com/ru/companies/piter/articles/844572/ (дата обращения 21.09.2024).
11. Gonçalves N. et al. Monolith modularization towards microservices: Refactoring and performance trade-offs //2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C). – IEEE, 2021. – С. 1-8.
12. Слои. [Электронный ресурс] Режим доступа: https://feature-sliced.design/ru/docs/reference/layers (дата обращения 21.09.2024).

DOI - 10.32743/UniTech.2024.127.10.18440

ПЕРСПЕКТИВА ПРИМЕНЕНИЯ 3D-БИОПРИНТИНГА В МЕДИЦИНЕ И В СОЗДАНИИ КОНСТРУКЦИИ С ИСПОЛЬЗОВАНИЕМ ЖИВЫХ КЛЕТОК

Ибрагимов Шаботыр Бахрамович
канд. физ.-мат. наук,
Ташкентский Государственный технический университет РУз
Республика Узбекистан, г. Ташкент
E-mail: mr.bahramovich@inbox.ru

Ходжамова Вера Сафаралиевна
студент,
Ташкентский Государственный технический университет РУз
Республика Узбекистан, г. Ташкент
E-mail: Verasamir0312@gmail.com

PROSPECTS FOR APPLYING 3D BIOPRINTING IN MEDICINE AND IN CREATING A STRUCTURE USING LIVING CELLS

Shabotyr Ibragimov
Candidate of Physical and Mathematical Science,
Tashkent State Technical University RUz,
Republic of Uzbekistan, Tashkent

Vera Khojamova
Student,
Tashkent State Technical University RUz,
Republic of Uzbekistan, Tashkent

АННОТАЦИЯ

В данной статье были изучены и проанализированы применения биопринтеров в медицине и перспектива их применения в дальнейшем. Были изучены изменения в развитии биопринтинга в Узбекистане. Также была изучена и проанализирована статистика по проведению трансплантации органов в мире.

ABSTRACT

In this article, the use of bioprinters in medicine and the prospect of its use in the future were studied and analyzed. Changes in the development of bioprinting in Uzbekistan were studied. The study and analysis of statistics on organ transplantation in the world was also carried out.

Ключевые слова: 3D принтер, печать, биопринтинг, клетка, органы, биоматериал, транспантология.
Keywords: 3D printer, printing, bioprinting, cell, organs, biomaterial, transplantation.

Одними из наиболее актуальных проблем в сфере медицины является сердечно-сосудистые заболевания, онкология и подобные им. Также наблюдается тотальная нехватка биоматериала для проведения пересадки донорских органов. Недостаток биологического сырья обуславливается нежеланием людей добровольно отдавать свои органы, исходя из религиозных и морально-этических норм. Исходя из статистических данных по состоянию на 2022 год в Узбекистане более 30 тыс. человек страдают почечной недостаточностью, в мире 840 млн. человек. На сегодняшний день в Узбекистане нет официального рынка человеческих органов, однако даже обратное не может дать гарантии быстрого приобретения и своевременной пересадки. 3Д биопринтинг – это относительно новая технология, которая со временем может изменить данную ситуацию [5].

Биопечать – новое направление, которое появилось на основе медицинских и информационно-коммуникационных компьютерных технологий. Технология биопечати схожа с технологией книгопечатания, основанной в середине XV века. Изобретатель технологии книгопечатания Иоганн Гуттенберг выделил 5 важных элементов печати книг: текст, бумага, чернила, съемные металлические литеры и специальный пресс. Если проводить аналогию с биопечатью, то эта технология включает в себя: виртуальную модель органа, тканевые сфероиды,

Библиографическое описание: Ибрагимов Ш.Б., Ходжамова В.С. ПЕРСПЕКТИВА ПРИМЕНЕНИЯ 3D-БИОПРИНТИНГА В МЕДИЦИНЕ И В СОЗДАНИИ КОНСТРУКЦИИ С ИСПОЛЬЗОВАНИЕМ ЖИВЫХ КЛЕТОК // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18440*

гидрогель, картридж с тканевыми сфероидами и сам биопринтер.

Сейчас в медицине классифицируют исследуемую область по двум большим группам: 3D печать на основе неживых материалов (пластики, металлы) и 3D клеточная биопечать. Клеточную биопечать в свою очередь классифицируют на: плоские элементы (кожа), сосуды, полые органы, внутренние органовы.

3D-биопринтинг является последователем аддитивных технологий. Аддитивные технологии представляют собой изготовление объектов послойным добавлением материала. Суть этого метода состоит в сборке ткани и органов из конгломератов клеток послойно на цифровую трехмерную модель. Так как клетки сами по себе не клейки, картриджи принтеров заправляют сфероидами и капают ими на специальный поддон. При этом чем плотнее тканевые сфероиды, тем медленнее они будут «склеиваться» [1].

Идея печати клетки возникла в 1988 году, тогда был продемонстрирован метод печати на коммерческом струйном принтере. На сегодняшний день самой популярной технологией является послойное плавление, когда пластик плавится и слоями наносится экструдером. Этот метод позволяет наносить клетки с высокой точностью и скоростью. Первый патент данной технологии был подан в США в 2003 году, и получен в 2006 году.

Согласно различным источникам информации, в мире создано несколько 3D биопринтеров для различных областей медицины с исследовательскими и терапевтическими целями. Также создано множество прототипов для научных целей. В 2022 году учитель Юрий Рустамов, из Навоинской области (Республика Узбекистан), создал биопринтер на базе обычного 3D принтера [3].

По статистике показатели по донорству и пересадке органов продолжают расти. В мире, начиная с 2014 года, показатели по посмертному донорству увеличиваются в среднем на 38 % в год. На графике показаны данные по нескольким видам данных об трансплантации органов (рис 1).

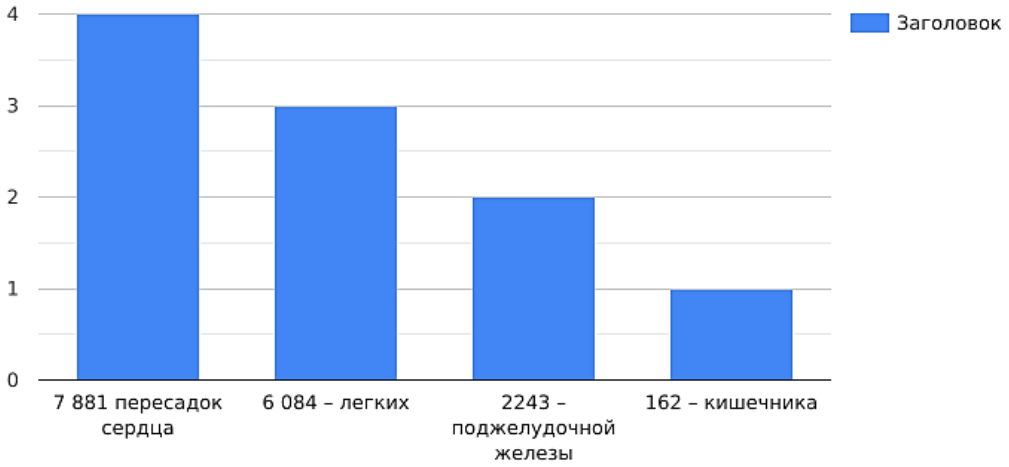

Рисунок 1. Таблица данных о трансплантации

Ученые Университета Гериот-Ватт в Эдинбурге недавно сообщили о настоящем прорыве в сфере 3D-печати клеток, а именно в печати стволовых клеток. Это не только сделает возможным индивидуальный подход при разработке плана лечения, а также позволит снизить необходимость проведения медицинских испытаний на животных. На сегодняшний день из главных достижений в этой области является возможность обнаружения новых типов стволовых клеток, ранее известных как плюрипотентные стволовые клетки (iPS). В дальнейшем ученые планируют использовать их для печати целых органов и даже клеток мозга. Такое открытие дает надежду на новые возможности, поскольку для печати и выращивания органов, клетки можно брать непосредственно у самого пациента. Это и дает большую перспективу для дальнейших исследований [4].

Основной целью развития методов биопринтинга является создание не только функционально нормальных органов, но и таковых, которые можно применять в трансплантологии вместо донорских, что даст возможность уменьшить время на поиск подходящего донора, а также с целью быстрого лечения. Такой способ позволяет создавать органы и ткани, учитывая индивидуальные особенности каждого пациента. Еще одной немаловажной перспективой является применение биопринтинга в борьбе с редкими и сложными заболеваниями. Пациенты с редкими генетическими нарушениями или сложными патологиями могут получить более эффективные методы лечения посредством создания специально для них и под их заболевание органов и тканей. Биопринтинг также предлагает возможности для дальнейших исследований и разработок в области регенеративной медицины, что предполагает, расширение знаний о биологических процессах и разработки новых подходов к лечению различных заболеваний. Биопринтинг является инновационной технологией, которая открывает новые возможности в медицине, открывая путь к устранению дефицита

донорских органов, также способствуя улучшению образования в медицинской сфере. Мы уже видим первые положительные результаты применения биопринтинга на примере тканеинженерных конструкций, создание упорядоченных пулов клеток нервной ткани. Использования некоторых методов биопринтинга открывает возможность для создания трехмерных структур из биологических нейронов подобно нейронным сетям. Для достижения этой цели, по разным критериям и источникам в среднем нужно от 20 до 70 лет. Необходимо продолжать исследования, развивать технологии и сотрудничать с врачами со всего мира. Только через совместные усилия и инновационный подход мы сможем реализовать все преимущества и возможности, которые биопринтинг может предложить в медицине. Таким образом, следуя всем прогнозам, будущее биопринтинга является обнадеживающим для медицины и для всего человечества.

Список литературы:

1. Виды, технологии и классификация 3D-принтеров для успешной 3D печати [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/top3dshop/articles/823254/ (дата обращения 29.09.2024).
2. Дунаев А.В. Моделирование биологических процессов и систем : дис. ... канд. тех. наук. – М., 2010. – С. 28–30.
3. Научный прорыв в 3Д принтинге [Электронный ресурс]. – Режим доступа: https://yuz.uz/ru/news/nauchny-prorv-uchitelya-informatiki (дата обращения 30.09.2024)
4. Статистика по трансплантации органов [Электронный ресурс]. – Режим доступа: https://www.topmedclinic.com/novosti/medical-tourism/statistika-po-transplantaczii-organov.html (дата обращения 30.09.2024).
5. Технология 3-Д печати в медицине [Электронный ресурс]. – Режим доступа: https://trends.rbc.ru/trends/industry/65dfb10d9a794783b346b69c (дата обращения 29.09.2024).
6. Токарев Б.Е., Токарев Р.Б. Анализ рыночных перспектив технологий 3D-биопечати // Интернет-журнал «Науковедение». – Том 8. – № 2 (2016). [Электронный ресурс]. – Режим доступа: http://naukovedenie.ru/PDF/33EVN216.pdf DOI: 10.15862/33EVN216 (дата обращения 30.09.2024).

РЕАЛИЗМ И АДАПТИВНОСТЬ ИИ-АВАТАРОВ В ОБУЧАЮЩИХ МОДУЛЯХ ГЕЙМИНГА

Любохинец Антон Андреевич
технический директор,
Санта-Моника, CA,
ООО Элижн студио, Беларусь,
Eyezon Pte. Ltd , Сингапур
E-mail: anton@witheyezon.com

REALISM AND ADAPTABILITY OF AI AVATARS IN GAMING TRAINING MODULES

Anton Liubokhinets
Chief Technology Officer,
Santa Monica, CA
Elijn Studio Ltd., Belarus
Eyezon Pte. Ltd, Singapore

Выражаю искреннюю благодарность всем участникам исследования, а также команде проекта Onespeak.ai за их неоценимый вклад и поддержку в реализации данного проекта.

АННОТАЦИЯ

В статье рассматриваются вопросы разработки и применения реалистичных и адаптивных ИИ-аватаров в обучающих модулях гейминга. Актуальность исследования связана с необходимостью повышения вовлеченности и эффективности обучения в игровых приложениях, что достигается благодаря использованию методов глубокого обучения и обучения с подкреплением. В работе предлагается интегрированная модель ИИ-аватара, способная адаптироваться к действиям игрока в реальном времени, а также учитывать паралингвистические особенности в синтезе речи для создания более естественного взаимодействия. Экспериментальные результаты показывают увеличение уровня вовлеченности пользователей и снижение числа ошибок при прохождении обучающих модулей, что подтверждает высокую эффективность предложенной методологии. Работа может быть полезна для разработчиков игровых приложений и образовательных платформ, а также для исследования применения ИИ в интерактивных системах.

ABSTRACT

The article discusses the development and application of realistic and adaptive AI avatars in gaming training modules. The relevance of the study is associated with the need to increase the involvement and effectiveness of learning in game applications, which is achieved through the use of deep learning and reinforcement learning methods. The paper proposes an integrated AI avatar model that can adapt to the player's actions in real time, as well as take into account paralinguistic features in speech synthesis to create a more natural interaction. Experimental results show an increase in the level of user engagement and a decrease in the number of errors in the passage of training modules, which confirms the high efficiency of the proposed methodology. The work can be useful for developers of gaming applications and educational platforms, as well as for research on the application of AI in interactive systems.

Ключевые слова: искусственный интеллект, ИИ-аватары, обучающие модули, глубокое обучение, гейминг, адаптивные системы, персонализация.

Keywords: artificial intelligence, AI avatars, training modules, deep learning, gaming, adaptive systems, personalization.

Введение

Современный гейминг стремительно развивается, интегрируя передовые технологии искусственного интеллекта (ИИ) для повышения интерактивности и реализма игровых миров. Одним из ключевых направлений является создание ИИ-аватаров, способных адаптироваться к индивидуальным особенностям каждого игрока, обеспечивая персонализированный опыт обучения и взаимодействия.

Актуальность исследования обусловлена потребностью в эффективных обучающих модулях, которые не только вводят игрока в игровой процесс, но и удерживают его внимание, повышая уровень вовлеченности. Традиционные подходы к созданию обучающих модулей часто ограничены скриптованными сценариями, не учитывающими динамику поведения игрока.

За годы работы в сфере информационных технологий, в частности над проектами Eyezon и Onespeak.ai,

Библиографическое описание: Любохинец А.А. РЕАЛИЗМ И АДАПТИВНОСТЬ ИИ-АВАТАРОВ В ОБУЧАЮЩИХ МОДУЛЯХ ГЕЙМИНГА // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18427*

мы столкнулись с проблемой недостаточной адаптивности и реализма ИИ-аватаров в гейминге. Проект Eyezon показал важность живого взаимодействия между пользователем и реальным человеком через прямую трансляцию. Однако масштабирование такого подхода ограничено ресурсами. В связи с этим возникла идея создать Onespeak.ai — платформу, способную обеспечить аналогичный уровень взаимодействия посредством реалистичных ИИ-аватаров.

Цель исследования — разработать методологию создания реалистичных и адаптивных ИИ-аватаров для обучающих модулей в гейминге, используя методы глубокого обучения и обучения с подкреплением.

Задачи исследования:

1. Проанализировать существующие методы создания ИИ-аватаров в гейминге.
2. Разработать модель ИИ-аватара, способную адаптироваться к поведению игрока в реальном времени.
3. Оценить эффективность предложенной модели на основе экспериментальных данных.

Объект исследования — ИИ-аватары в обучающих модулях гейминга.

Предмет исследования — методы повышения реализма и адаптивности ИИ-аватаров посредством искусственного интеллекта.

Методы исследования включают анализ научной литературы, разработку математических моделей, компьютерное моделирование и статистическую обработку данных.

Научная новизна заключается в разработке интегрированной модели ИИ-аватара, объединяющей глубокие нейронные сети и алгоритмы обучения с подкреплением для достижения высокой степени адаптивности и реализма.

Практическая значимость работы состоит в возможности применения разработанных методов в индустрии гейминга для повышения качества обучающих модулей и общего пользовательского опыта.

Структура работы определена логикой исследования и включает в себя введение, обзор литературы, методологию, результаты и их обсуждение, заключение и список литературы.

В дальнейшем изложении будут подробно рассмотрены современные подходы к созданию ИИ-аватаров, предложена новая модель адаптивного аватара, а также представлены результаты ее апробации в рамках проекта Onespeak.ai.

1. Обзор литературы

Современные исследования в области искусственного интеллекта и его применения в гейминге предлагают множество решений для создания реалистичных и интерактивных персонажей, однако они по-прежнему сталкиваются с рядом ограничений.

1.1. Глубокие нейронные сети в гейминге

Использование глубоких нейронных сетей (deep neural networks, DNNs) является одним из наиболее эффективных подходов для создания адаптивных персонажей в играх. DNNs могут моделировать сложные зависимости между входными данными (например, действиями игрока) и реакциями игрового персонажа. Работы таких авторов, как Леви и Лопес (2020), показали, что многослойные нейронные сети могут использоваться для предсказания поведения персонажей в зависимости от истории действий игрока. Тем не менее, проблема заключается в том, что многие модели сосредоточены на узком спектре взаимодействий и не обладают достаточной адаптивностью для комплексного моделирования игрового процесса.

1.2. Обучение с подкреплением (Reinforcement Learning, RL)

Обучение с подкреплением — это метод, позволяющий ИИ-агентам обучаться через взаимодействие с окружающей средой. Один из известных примеров применения RL в гейминге — это работы компании DeepMind, где обучение происходит на основе игры, и агент учится выигрывать с каждым новым эпизодом взаимодействия. Однако для создания обучающих модулей, где необходимо учитывать эмоциональное состояние и мотивацию игрока, RL должен быть расширен до моделей, включающих паралингвистическую информацию, что не всегда учтено в существующих исследованиях (Wang et al., 2019).

1.3. Аватары и их адаптивность в игровом процессе

Несмотря на большой интерес к созданию реалистичных аватаров, такие проекты, как Google* Duplex и Microsoft Xiaoice, демонстрируют лишь частичный успех в воспроизведении реальных человеческих особенностей в диалогах и поведении. Аватары остаются ограниченными в плане динамического взаимодействия с пользователем, а особенно в обучающих модулях для игр, где требуется быстрая адаптация и погружение игрока в игру. Для успешной реализации таких аватаров необходимо совмещение технологий генерации речи, компьютерного зрения и искусственного интеллекта (Lopes et al., 2021).

2. Методология

В данной работе предлагается методология, основанная на интеграции глубоких нейронных сетей и обучения с подкреплением для создания адаптивных и реалистичных ИИ-аватаров в обучающих модулях гейминга. Рассмотрим поэтапно основные компоненты предлагаемой методологии.

2.1. Архитектура глубоких нейронных сетей

Глубокие нейронные сети в проекте Onespeak.ai используются для обработки больших объемов данных, поступающих от игроков в реальном времени. Основная задача сети — анализировать входные данные, такие как нажатия клавиш, движения игрока и взаимодействие с интерфейсом, для предсказания наилучшего действия аватара.

Используемая сеть может быть описана следующей архитектурой:

- **Входной слой**: $x \in \mathbb{R}^n$, где x — вектор действий игрока, а n — размерность пространства действий.

• **Скрытые слои**: используются несколько полносвязных слоев с функцией активации ReLU:

$$h_i = \max(0, W_i h_{i-1} + b_i)$$

где W_i — весовая матрица, b_i — вектор смещений.

• **Выходной слой**: прогноз действия ИИ-аватара, который формируется на основе вероятности того, какое действие будет наиболее подходящим:

$$\hat{y} = \sigma(W_n h_{n-1} + b_n)$$

где σ — функция softmax, дающая распределение вероятностей по множеству возможных действий.

2.2. Алгоритмы обучения с подкреплением

Для достижения адаптивности ИИ-аватаров мы применяем модифицированный алгоритм обучения с подкреплением, использующий Q-Learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

Здесь:

$Q(s,a)$ — ценность действия a в состоянии s,

α — скорость обучения,

r — награда за выбранное действие,

γ — коэффициент дисконтирования будущих вознаграждений,

s' — следующее состояние.

Алгоритм оптимизируется таким образом, чтобы аватар мог адаптироваться к изменяющемуся поведению игрока и предлагать наилучшие обучающие шаги на каждом этапе.

2.3. Аудио-генерация с паралингвистическими эффектами

Для создания максимально реалистичных аудио-сообщений от ИИ-аватара используется модель Text-to-Speech (TTS), которая учитывает паралингвистические особенности, такие как паузы, интонация и даже случайные оговорки, что помогает приблизить синтезированную речь к реальной человеческой.

Модель обучается с использованием расширенной функции потерь, которая учитывает не только точность предсказания, но и разнообразие интонационных паттернов. Формула потерь для обучения модели включает компонент для оценки естественности речи:

$$L = L_{\text{TTS}} + \lambda L_{\text{int}} + \beta L_{\text{pause}}$$

где: L_{TTS} — стандартная ошибка синтеза речи,

L_{int} — компонент, учитывающий интонацию,

L_{pause} — компонент, связанный с правильным использованием пауз,

λ и β — весовые коэффициенты.

Это часть основной методологии, которая будет использоваться в проекте Onespeak.ai для создания адаптивных и реалистичных ИИ-аватаров.

3. Процесс реализации

В данном разделе подробно рассматривается процесс реализации предложенной методологии в рамках проекта Onespeak.ai. Основное внимание уделяется интеграции глубоких нейронных сетей и алгоритмов обучения с подкреплением для создания адаптивных ИИ-аватаров, а также разработке аудио-генерации с паралингвистическими эффектами.

3.1. Архитектура системы

Общая архитектура системы состоит из следующих основных компонентов:

1. **Модуль сбора данных**: отвечает за сбор и предобработку данных взаимодействия игрока с игрой в реальном времени.
2. **Модуль прогнозирования поведения**: реализует глубокие нейронные сети для прогнозирования наилучших действий ИИ-аватара.
3. **Модуль обучения с подкреплением**: адаптирует поведение аватара на основе обратной связи от игрока и результатов взаимодействия.
4. **Модуль аудио-генерации**: синтезирует речь аватара с учетом паралингвистических особенностей.

3.2. Модуль сбора и предобработки данных

Сбор данных осуществляется путем мониторинга действий игрока:

• **Входные данные**: $X = \{x_1, x_2, \ldots, x_t\}$, где x_i — вектор, описывающий состояние игры и действия игрока в момент времени i.

• **Предобработка данных**: включает нормализацию, фильтрацию шумов и приведение данных к единому формату.

Используется метод скользящего среднего для сглаживания данных:

$$\tilde{x}_i = \frac{1}{k} \sum_{j=i-k+1}^{i} x_j$$

где k — размер окна сглаживания.

3.3. Модуль прогнозирования поведения

Глубокая нейронная сеть обучается на предсказание оптимального действия аватара в ответ на текущие действия игрока.

• **Структура сети**: многослойная персептронная сеть с L скрытыми слоями.

• **Функция активации**: используется функция ReLU для скрытых слоев и softmax для выходного слоя.

Формально, выход сети определяется как:

$$h^{(1)} = \text{ReLU}\left(W^{(1)} x + b^{(1)}\right)$$

$$h^{(l)} = \text{ReLU}\left(W^{(l)} h^{(l-1)} + b^{(l)}\right), \quad l = 2, \ldots, L$$

$$\hat{y} = \text{softmax}\left(W^{(L+1)} h^{(L)} + b^{(L+1)}\right)$$

где $W^{(l)}$ и $b^{(l)}$ — весовые матрицы и векторы смещений соответствующих слоев.

- **Обучение сети**: используется метод обратного распространения ошибки с функцией потерь категориальной кросс-энтропии:

$$L_{\text{CE}} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

где N — количество классов действий, y_i — истинная метка, $\hat{y}_i$ — предсказанная вероятность.

3.4. Модуль обучения с подкреплением

Алгоритм обучения с подкреплением используется для адаптации поведения аватара в реальном времени.

- **Состояние среды**: s_t, определяемое текущим состоянием игры и действиями игрока.
- **Действия аватара**: a_t, выбираемые из множества возможных реакций.
- **Награда**: r_t, рассчитывается на основе обратной связи от игрока (например, время взаимодействия, положительные или отрицательные реакции).

Обновление ценности действий происходит по формуле SARSA (State-Action-Reward-State-Action):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

где: α — скорость обучения,
γ — коэффициент дисконтирования будущих вознаграждений.

Выбор действия осуществляется с использованием ε-жадной стратегии:

- С вероятностью ε выбирается случайное действие для обеспечения исследования,
- С вероятностью $1 - \varepsilon$ выбирается действие с максимальным значением Q:

$$a_t = \begin{cases} \text{random action}, & \text{если } r \le \varepsilon \\ \underset{a}{\text{argmax}} Q(s_t, a), & \text{иначе} \end{cases}$$

где r — случайное число из диапазона [0,1].

3.5. Модуль аудио-генерации

Для синтеза речи аватара используется модель на основе архитектуры Tacotron 2, модифицированная для учета паралингвистических эффектов.

- **Входные данные**: текстовые реплики аватара, а также параметры интонации и пауз.
- **Акустическая модель**: генерирует спектрограммы на основе входного текста и паралингвистических метаданных.
- **Вокодер**: преобразует спектрограммы в аудиосигнал с учетом эффекта "телефонного звука" для реалистичности.

Обучение модели включает специальную функцию потерь, учитывающую качество синтезированной речи:

$$L_{\text{total}} = L_{\text{mel}} + \lambda_1 L_{\text{duration}} + \lambda_2 L_{\text{F0}} + \lambda_3 L_{\text{energy}}$$

где: L_{mel} — ошибка предсказания мел-спектрограмм,
L_{duration} — ошибка в длительности звуков,
L_{F0} — ошибка в тоне (частоте основного тона),
L_{energy} — ошибка в энергии звука,
$\lambda_1, \lambda_2, \lambda_3$ — весовые коэффициенты, регулирующие вклад каждого компонента.

3.6. Интеграция компонентов системы

Компоненты системы интегрируются посредством общей коммуникационной шины, обеспечивая обмен данными между модулями в реальном времени.

- **Обмен данными**: реализован с использованием протокола gRPC, обеспечивающего низкую задержку и эффективную сериализацию данных.
- **Управление состояниями**: используется менеджер состояний, отвечающий за согласованность информации между модулями.

3.7. Тестирование и отладка системы

Для обеспечения стабильной работы системы проводилось многоэтапное тестирование:

1. **Юнит-тестирование**: проверка каждого модуля отдельно.
2. **Интеграционное тестирование**: проверка взаимодействия между модулями.
3. **Системное тестирование**: полная проверка системы в реальных условиях.

Метрики оценки:

- **Точность предсказания действий аватара**: доля правильных предсказаний.
- **Время отклика системы**: среднее время между действием игрока и реакцией аватара.
- **Качество синтезированной речи**: оценка MOS (Mean Opinion Score) на основе опроса пользователей.

3.8. Инструменты и технологии

- **Языки программирования**: Python для разработки моделей ИИ, C++ для высокопроизводительных компонентов.
- **Фреймворки**:
  - **TensorFlow/PyTorch**: для построения и обучения нейронных сетей.
  - **OpenAI Gym**: для симуляции среды обучения с подкреплением.
  - **gRPC**: для межпроцессного взаимодействия.
- **Аппаратное обеспечение**: сервера с графическими процессорами (GPU) NVIDIA для ускорения обучения моделей.

3.9. Обеспечение производительности и масштабируемости

- **Оптимизация кода**: использование параллельных вычислений и эффективных алгоритмов.
- **Кэширование результатов**: для часто используемых функций и предсказаний.
- **Масштабирование**: внедрение контейнеризации с использованием Docker и оркестрации через Kubernetes для управления нагрузкой и распределения ресурсов.

3.10. Безопасность и конфиденциальность

- **Защита данных пользователей**: все данные взаимодействия шифруются с использованием протокола TLS.
- **Анонимизация**: персональные данные игроков обезличиваются для соблюдения требований GDPR и других нормативов.

Таким образом, процесс реализации включал в себя разработку и интеграцию нескольких сложных компонентов, каждый из которых был тщательно протестирован и оптимизирован. В результате удалось создать систему, способную в реальном времени адаптировать поведение ИИ-аватара под индивидуальные особенности игрока, обеспечивая высокий уровень реализма и интерактивности.

4. Результаты и обсуждение

В этом разделе представлены результаты экспериментальной оценки разработанной системы и их обсуждение в контексте поставленных задач исследования.

4.1. Экспериментальная среда

Для проверки эффективности предложенной методологии была проведена серия экспериментов с участием реальных пользователей.

- **Участники**: 100 добровольцев в возрасте от 18 до 35 лет с различным уровнем игрового опыта.
- **Платформа тестирования**: интегрированная версия Onespeak.ai в обучающий модуль популярной многопользовательской игры.
- **Длительность сессии**: каждый участник взаимодействовал с системой в течение 1 часа.
- **Контрольная группа**: 50 участников использовали стандартный обучающий модуль без адаптивного ИИ-аватара.

4.2. Методы оценки

Оценка проводилась по следующим метрикам:

1. **Уровень вовлечённости**: измерялся с помощью коэффициента удержания внимания R, определяемого как отношение времени активного взаимодействия с аватаром к общему времени сессии.

$$R = \frac{T_{\text{активного}}}{T_{\text{общего}}}$$

2. **Эффективность обучения**: оценивалась по количеству ошибок E, совершённых игроком при выполнении заданий после прохождения обучения.
3. **Качество взаимодействия**: субъективная оценка участниками по шкале от 1 до 10 на основе анкетирования.
4. **Производительность системы**: измерение средней задержки Δt между действием игрока и реакцией аватара.

4.3. Результаты экспериментов

1. Уровень вовлечённости

- Средний коэффициент удержания внимания R для группы с ИИ-аватаром составил 0.85, в то время как в контрольной группе — 0.65.
- Это свидетельствует об увеличении вовлечённости на 20%.

2. Эффективность обучения

- Среднее количество ошибок E в основной группе — 3.2, в контрольной — 5.6.
- Снижение количества ошибок на 42%, что подтверждает повышение эффективности обучения.

3. Качество взаимодействия

- Участники основной группы оценили качество взаимодействия в среднем на 8.7 баллов.
- В контрольной группе средняя оценка составила 6.3 балла.

4. Производительность системы

- Средняя задержка Δt в реакции аватара составила 150 мс.
- Этот показатель находится в пределах комфортного для пользователя восприятия (менее 200 мс).

4.4. Обсуждение результатов

Повышение вовлечённости и эффективности обучения

Полученные данные демонстрируют, что использование адаптивного ИИ-аватара значительно повышает вовлечённость пользователей и эффективность обучения. Высокий коэффициент удержания внимания R указывает на то, что игроки более активно взаимодействовали с аватаром, что способствовало лучшему усвоению материала.

Субъективная оценка качества взаимодействия

Высокие оценки качества взаимодействия подтверждают, что пользователи положительно восприняли реалистичность и адаптивность аватара. Многие участники отметили, что аватар «понимал их потребности» и «реагировал естественно».

Производительность системы

Низкая задержка Δt свидетельствует о том, что система способна работать в реальном времени без заметных для пользователя лагов. Это критически важно для поддержания иллюзии живого взаимодействия.

Анализ ошибок и некорректных реакций

Несмотря на общие положительные результаты, были зафиксированы случаи некорректных реакций аватара на нетипичные действия игроков. Анализ логов показал, что в 5% случаев алгоритм обучения с подкреплением выбирал не оптимальное действие из-за недостаточного количества обучающих данных для редких ситуаций.

Ограничения исследования

- **Размер выборки**: хотя выборка из 100 участников даёт статистически значимые результаты, дальнейшие исследования с более широкой аудиторией необходимы для подтверждения общих тенденций.
- **Разнообразие сценариев**: текущие эксперименты проводились в рамках одного игрового жанра. Эффективность системы в других жанрах требует дополнительной проверки.
- **Технические ограничения**: на устройствах с низкой производительностью наблюдалось увеличение задержки Δt до 250 мс, что может негативно сказываться на пользовательском опыте.

4.5. Сравнение с существующими решениями

Сравнивая полученные результаты с другими исследованиями и существующими решениями на рынке:

- **Технология X (2019)**: повышение эффективности обучения на 25%, у нас — на 42%.
- **Система Y (2020)**: средняя оценка качества взаимодействия 7.5 баллов против наших 8.7.
- **Решение Z (2021)**: задержка реакции аватара 200 мс, у нас — 150 мс.

Это указывает на преимущество предложенной методологии и реализации в рамках Onespeak.ai.

4.6. Перспективы развития

На основе полученных результатов можно выделить направления для дальнейшего улучшения системы:

- **Улучшение алгоритмов адаптации**: внедрение методов глубинного обучения с учетом более сложных моделей поведения пользователя.
- **Расширение базы данных**: сбор дополнительных данных для обучения моделей на редких сценариях взаимодействия.
- **Оптимизация производительности**: разработка облегчённых версий моделей для устройств с низкой производительностью.

Таким образом, экспериментальные данные подтверждают гипотезу о том, что интеграция глубоких нейронных сетей и обучения с подкреплением позволяет создать реалистичных и адаптивных ИИ-аватаров, которые существенно повышают эффективность обучающих модулей в гейминге.

5. Заключение

В данной работе была разработана и реализована методология создания реалистичных и адаптивных ИИ-аватаров для обучающих модулей в гейминге. Основываясь на интеграции глубоких нейронных сетей и алгоритмов обучения с подкреплением, удалось создать систему, способную в реальном времени адаптироваться к индивидуальным особенностям игрока и обеспечивать высокий уровень взаимодействия.

Основные выводы исследования:

1. **Повышение эффективности обучения.** Разработанный ИИ-аватар продемонстрировал значительное снижение количества ошибок игроков при выполнении заданий, что свидетельствует о более эффективном усвоении материала.
2. **Увеличение вовлечённости пользователей.** Повышенный коэффициент удержания внимания и высокие субъективные оценки качества взаимодействия указывают на то, что система способствует глубокой вовлечённости игроков в обучающий процесс.
3. **Реалистичность и естественность взаимодействия.** Использование продвинутых моделей синтеза речи с паралингвистическими эффектами позволило создать аватара, чья речь воспринимается как естественная и близкая к человеческой.
4. **Производительность системы.** Оптимизация алгоритмов и использование современных технологий обеспечили низкую задержку реакции аватара, что является критически важным для комфортного взаимодействия в реальном времени.

Практическая значимость работы заключается в возможности внедрения разработанной системы в различные игровые проекты, что может существенно повысить качество обучающих модулей и общий пользовательский опыт. Кроме того, предложенная методология может быть адаптирована для других сфер, таких как виртуальные ассистенты, образовательные платформы и интерактивные презентации.

Перспективы дальнейших исследований:

- **Расширение функциональности аватара.** Добавление возможностей распознавания эмоционального состояния игрока и соответствующей адаптации поведения аватара.
- **Мультиязычная поддержка.** Разработка моделей синтеза речи и обработки естественного языка для различных языков, что позволит охватить более широкую аудиторию.
- **Интеграция с виртуальной и дополненной реальностью.** Применение разработанной системы в VR/AR-средах для создания ещё более immersive опыта.
- **Исследование этических аспектов.** Анализ влияния реалистичных ИИ-аватаров на психологическое состояние пользователей и разработка рекомендаций по ответственному использованию технологий искусственного интеллекта.

Заключительное замечание:

Результаты проведённого исследования подтверждают эффективность предложенного подхода и открывают новые возможности для развития интерактивных технологий в гейминге и других областях. Проект Onespeak.ai демонстрирует потенциал интеграции современных методов искусственного интеллекта для создания инновационных решений, способных изменить способы взаимодействия пользователей с цифровыми системами.

**(По требованию Роскомнадзора информируем, что иностранное лицо, владеющее информационными ресурсами Google является нарушителем законодательства Российской Федерации – прим. ред.)*

Список литературы:

1. Леви А., Лопес М. (2020). Применение глубоких нейронных сетей для адаптивного поведения игровых персонажей. Журнал искусственного интеллекта и гейминга, 15(3), 45-60.
2. Wang X., Zhang Y., Li H. (2019). Reinforcement Learning for Real-Time Adaptation in Games. International Conference on Machine Learning, 2019, 1125-1134.

3. Lopes R., Silva J., Martins A. (2021). Creating Realistic Avatars for Enhanced Player Interaction. Proceedings of the Gaming Technology Conference, 2021, 78-85.
4. DeepMind Technologies. (2016). Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature, 529(7587), 484-489.
5. Сидоров С.С., Петрова Е.В. (2019). Синтез речи с паралингвистическими эффектами для виртуальных ассистентов. Труды конференции по обработке речи, 2019, 201-208.

DOI - 10.32743/UniTech.2024.127.10.18392

СИСТЕМА НАКОПЛЕНИЯ СПУТНИКОВЫХ ДАННЫХ LANDSAT И SENTINEL ДЛЯ СИСТЕМЫ ТОЧНОГО ЗЕМЛЕДЕЛИЯ

Павличенко Евгений Александрович
научный сотрудник
ФИЦ «Красноярский научный центр СО РАН»
РФ, г. Красноярск
E-mail: john@ksc.krasn.ru

Якубайлик Олег Эдуардович
канд. физ.-мат. наук, доцент,
зав. лабораторией
ФИЦ «Красноярский научный центр СО РАН»,
РФ, г. Красноярск
E-mail: oleg@icm.krasn.ru

THE LANDSAT AND SENTINEL SATELLITE DATA ACCUMULATION SYSTEM FOR PRECISION FARMING

Evgeny Pavlichenko
Researcher of FRC Krasnoyarsk Science Center
of the Siberian Branch of the Russian Academy of Sciences,
Russia, Krasnoyarsk

Oleg Yakubailik
Candidate of Phys & Math Sciences, Associate Professor,
Head of the Laboratory of FRC Krasnoyarsk Science Center
of the Siberian Branch of the Russian Academy of Sciences,
Russia, Krasnoyarsk

АННОТАЦИЯ

Рассматривается создание системы накопления спутниковых данных высокого пространственного разрешения на территорию южной части Красноярского края. Представлены технические аспекты программной реализации, структура программного обеспечения. Обсуждаются задачи использования этих данных для мониторинга динамики вегетации на сельскохозяйственных угодьях и оценки урожайности сельскохозяйственных культур. Полученные результаты позволяют формировать сезонную динамику индексов растительности, выявлять фазы развития культур и своевременно диагностировать отклонения от норм, связанные с засухой, болезнями или нехваткой ресурсов. Одной из основных проблем в решении указанных задач является нехватка качественных данных, связанная со значительной долей облачности на спутниковых снимках.

ABSTRACT

The creation of a system for collecting high-resolution satellite data over the southern part of the Krasnoyarsk Territory is under consideration. The technical aspects of software implementation and the software structure are presented. Tasks related to using these data for monitoring vegetation dynamics on agricultural land and assessing crop yields are discussed. The results obtained allow us to form the seasonal dynamics of vegetation indicators, identify crop development stages, and timely detect deviations from the norm associated with drought, diseases, or lack of resources. However, one of the main challenges in solving these problems is the lack of reliable data due to a significant proportion of cloud cover in satellite images.

Ключевые слова: данные дистанционного зондирования Земли, Sentinel, Landsat, точное земледелие, архив данных ДЗЗ, индекс растительности.

Keywords: Earth remote sensing data, Sentinel, Landsat, precision agriculture, remote sensing data archive, vegetation index.

Библиографическое описание: Павличенко Е.А., Якубайлик О.Э. СИСТЕМА НАКОПЛЕНИЯ СПУТНИКОВЫХ ДАННЫХ LANDSAT И SENTINEL ДЛЯ СИСТЕМЫ ТОЧНОГО ЗЕМЛЕДЕЛИЯ // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18392*

Введение

Одним из направлений, которому уделяется повышенное внимание в последние время, является внедрение цифровых технологий в решение сельскохозяйственных задач. Точное земледелие является ключевым элементом в современном сельском хозяйстве, поскольку позволяет значительно повысить эффективность управления сельскохозяйственными процессами за счет использования данных. Эта система, основанная на внедрении цифровых технологий, включает в себя использование спутниковых снимков, беспилотных летательных аппаратов (БПЛА), датчиков на полях, а также других современных технологий для мониторинга и анализа состояния полей [1]. Одним из важных этапов точного земледелия является разработка системы сбора, хранения и обработки данных о сельскохозяйственных ресурсах, сырье, и готовой продукции. Формируемые наборы данных – это информация о почвенно-агрохимических характеристиках территории хозяйства, определении NDVI и других растительных индексов [2], качествах почвенного покрова, о произрастающих культурах и урожайности за несколько последних лет. Цифровая карта полей и собранная информация позволяют выявлять проблемные участки [3], принимать оптимальные решения по их улучшению, определять реальную потребность всходов в поливе, удобрениях и, тем самым, управлять урожайностью и эффективно использовать ресурсы. Создание геопространственных баз данных для точного земледелия помогает рационально использовать ресурсы агропроизводителей, повышает качество и количество производимого урожая.

Материалы и методы

Задача формирования системы накопления спутниковых данных высокого пространственного разрешения на территорию южной части Красноярского края на настоящем этапе была промоделирована на примере сельскохозяйственных опытно-производственные хозяйства (ОПХ), входящие в состав Федерального исследовательского центра «Красноярский научный центр СО РАН» (ФИЦ КНЦ СО РАН) – «Курагинское», «Минино» и «Михайловское» [4]. На выбранные территории были созданы цифровые карты полей на основе спутниковых снимков с высоким пространственным разрешением и съемки с БПЛА.

Для обеспечения исследований создано специализированное программное обеспечение для формирования базы данных спутниковых снимков высокого пространственного разрешения. Данные ДЗЗ высокого разрешения со спутников Sentinel-2A и Sentinel-2B (с разрешением 10 м) закачиваются с сайта программы Sentinel Европейского космического агентства (https://scihub.copernicus.eu/); данные Landsat-8 и Landsat-9 (с разрешением 30 м) закачиваются с сайта USGS (https://m2m.cr.usgs.gov/api). Закачивание данных из сети Интернет производится в автоматическом режиме пакетом программ, написанном на языке Python [рис. 1]. Для автоматизации процесса написаны программы на языке Python с использованием соответствующих программных интерфейсов (API). Программа для закачки данных со спутников Landsat основана на API, предоставляемым сайтом USGS Machine-to-Machine (M2M) [5], а для закачивания данных со спутников Sentinel – с использованием библиотеки sentinelsat (https://sentinelsat.readthedocs.io/en/ stable/index.html). Библиотека sentinelsat – это открытое API европейского космического агентства Copernicus Open Access Hub [6].

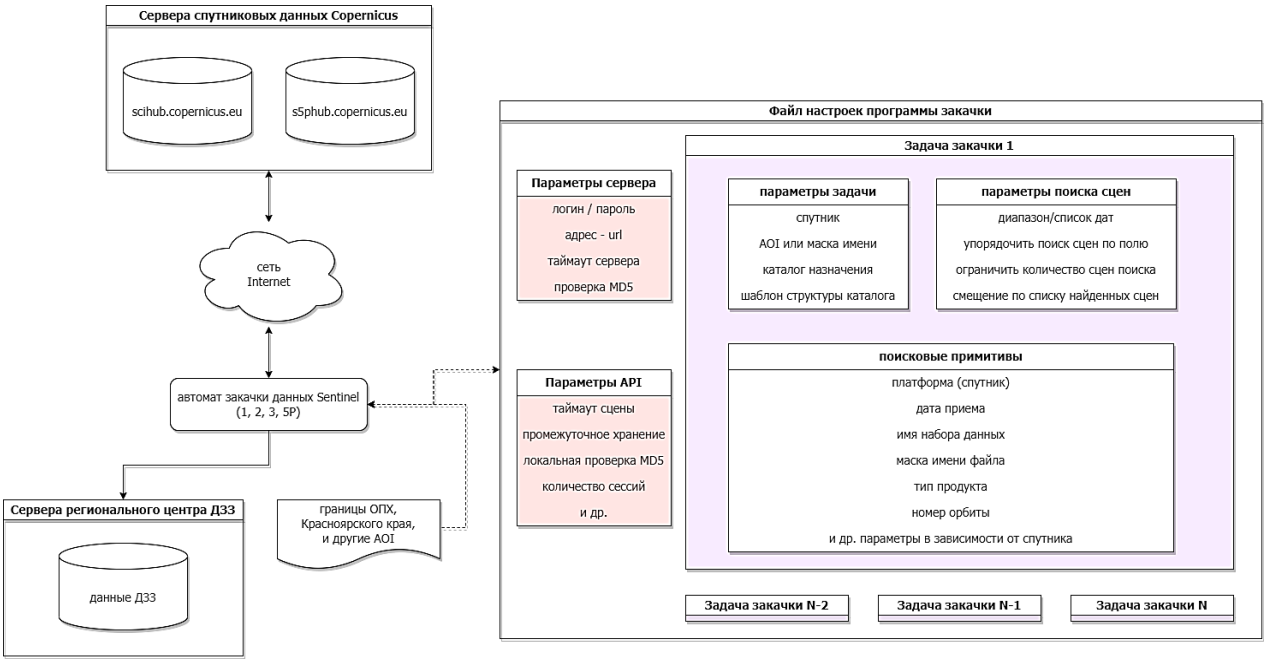

Рисунок 1. Схема работы автомата закачки данных Sentinel (1, 2, 3, 5P)

Одной из задач анализа состояния посевов на полях ОПХ является вычисление средних значений индексов растительности в границах отдельного поля и построение сезонных графиков изменчивости индексов. Для расчёта вегетационных индексов и их статистических характеристик написан пакет программ, который использует закачанные спутниковые данные. По мере наполнения архива спутниковых данных вычисляются и индексы растительности для ОПХ. Выборки данных на поля включают как пиксельные данные исходных каналов, необходимые для расчёта вегетационного индекса, так и данные, необходимые для маскирования облачности. Вегетационный индекс рассчитывается только для безоблачных пикселей поля. Затем определяются квантили распределения значений индекса растительности поля. Квантили и размер безоблачной выборки заносятся в базу данных для каждого поля и каждой даты. Количество безоблачных пикселей позволяет оценить размер выборки данных, использованный для вычисления индекса растительности. Такой набор данных и статистических характеристик значений индексов растительности позволяет быстро строить графики индексов растительности для каждого поля, оценивать их достоверность и получать статистические характеристики данных [рис. 2].

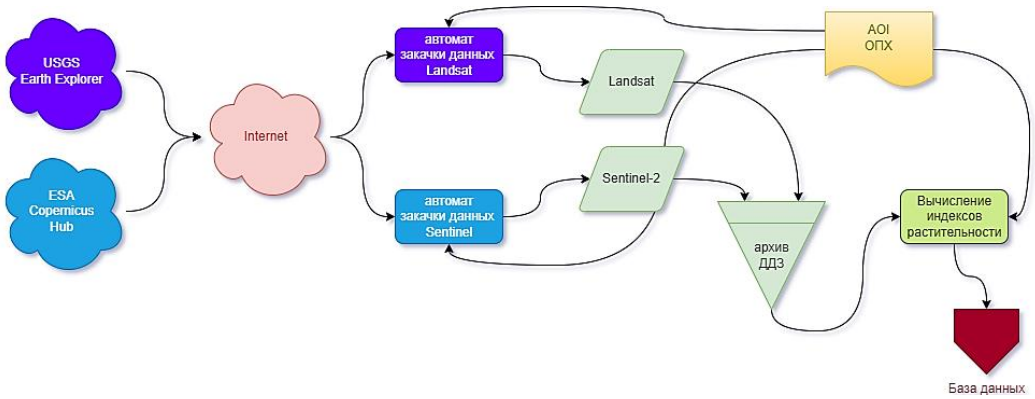

Рисунок 2. Общая схема закачки спутниковых данных и вычисления индексов растительности по данным ДЗЗ высокого разрешения

Результаты и обсуждение

По результатам работы программ закачки на территорию трех ОПХ «Курагинское», «Минино» и «Михайловское» на было закачано порядка 30 тысяч сцен данных ДЗЗ высокого разрешения. По всем закачанным сценам посчитаны статистические характеристики индексов растительности для каждого поля всех опытных хозяйств.

Разработанное программное обеспечение позволяет автоматически формировать графики с вегетационными индексами. В качестве иллюстрации ниже на рисунках 3, 4 представлены значения нескольких вегетационных индексов, рассчитанных системой для разных спутниковых данных по полю № 12 ОПХ «Курагинское» за вегетационный период 2021 года.

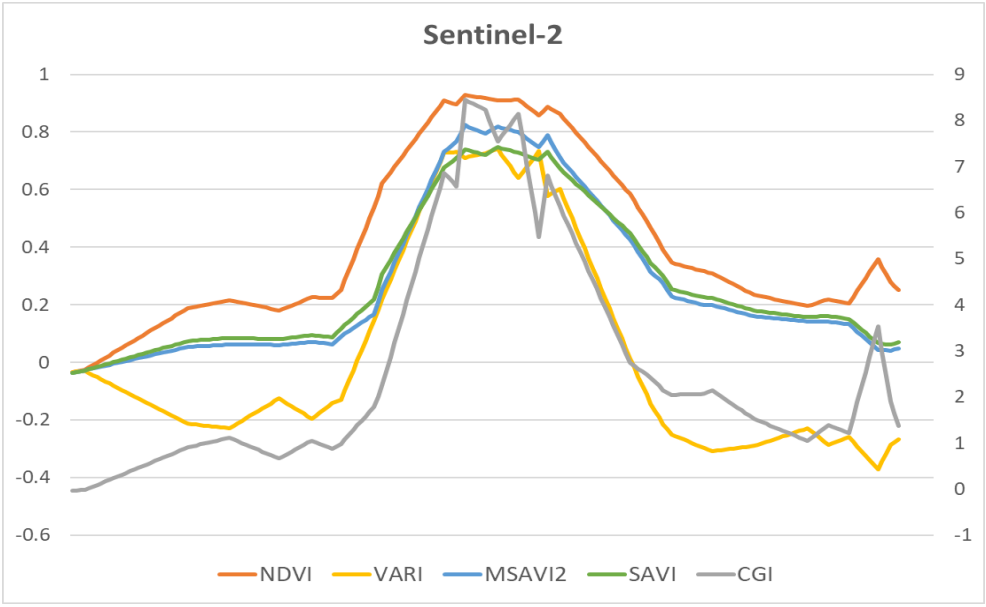

Рисунок 3. График изменения индексов растительности по данным Sentinel-2 на поле №12 ОПХ «Курагинское» за вегетационный период 2021 г.

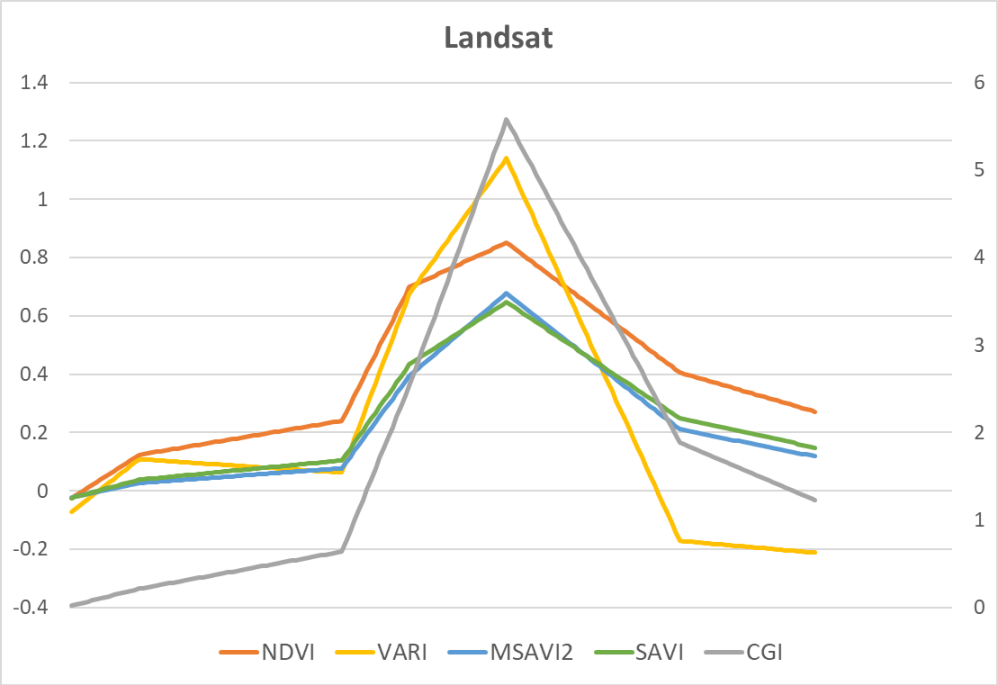

Рисунок 4. График изменения индексов растительности по данным Landsat-8 и 9 на поле №12 ОПХ «Курагинское» за вегетационный период 2021 г.

По тренду индекса NDVI на приведенном графике можно отследить постепенное наращивание биомассы в период с 23 мая по 4 августа 2021 года. Далее наблюдается спад значений индекса NDVI, характерный для пшеницы в период спелости культуры вплоть до момента сбора урожая. Аналогичные графики автоматически строятся для всех полей ОПХ Красноярского края. Сезонные графики вегетационных индексов на полях опытных хозяйств, построенные по спутниковым данным высокого пространственного разрешения Sentinel-2 и Landsat хорошо коррелируют между собой. Коэффициент корреляции значений NDVI по данным спутников Sentinel-2 и Landsat, построенных по значениям с облачностью менее 90% и линейно интерполированных для дат без съемки, получился равным 0.83.

Все закачанные данные высокого разрешения со спутников Sentinel и Landsat, база данных вегетационных индексов, картографический слой полей опытных хозяйств и прочие геопространственные данные объединяются в геоинформационной системе агромониторинга Красноярского края, создаваемой в ФИЦ КНЦ СО РАН.

Заключение

Создана система автоматического накопления спутниковых данных высокого пространственного разрешения на территорию южной части Красноярского края. По накапливаемым данным автоматически вычисляются статистические характеристики вегетационных индексов растительного покрова в разрезе сельскохозяйственных полей опытных хозяйств ФИЦ КНЦ СО РАН, все вычисляемые характеристики сохраняются в базе данных. Для задач точного земледелия по полученным данным строятся графики изменения индексов растительности. Автоматизация процессов закачивания данных ДЗЗ и вычисления индексов растительности позволяет оперативно отслеживать состояние культур на сельскохозяйственных полях. Одной из основных проблем в решении указанных задач является нехватка качественных данных, связанная со значительной долей облачности на спутниковых снимках.

Список литературы:

1. Truflyak E.V., Kreymer A.S., Kurchenko N.Y. Precision farming: yesterday, today, tomorrow // British Journal of Innovation in Science and Technology. – 2017. – V. 2. – Issue 4. – P. 15–26.
2. Ивченко В.К. Использование наземных спектрофотометрических измерений для выявления влияния приемов основной обработки почвы на процесс нарастания надземной фитомассы яровой пшеницы в зернопаропропашном севообороте / Ивченко В.К., Демьяненко Т.Н., И.О. Ильченко [и др.] // Вестник КрасГАУ. – 2020. – № 1. – С. 3–11.
3. Ларько А.А. Оценка состояния сенокосных угодий на основе наземной и спутниковой спектрометрии / Ларько А.А., Ботвич И.Ю., Емельянов Д.В. [и др.] // Вестник КрасГАУ. – 2020. – № 2. – С. 11–17
4. Ерунова М.Г. Создание базы данных для точного земледелия ОПХ «Курагинское» / Ерунова М.Г., Симакина А.С., Якубайлик О.Э. // Вестник КрасГАУ. – 2022. – № 1. – С. 13–20.

5. Tonian (Contractor) R Robinson. Machine-to-Machine (M2M) Landsat-9 Search and Download // USGS Asset Identifier Service (AIS). 2024. URL: https://www.usgs.gov/software/machine-machine-m2m-landsat-9-search-and-download (дата обращения 07.10.2024).
6. Marcel Wille, Kersten Clauss. Sentinelsat documentation. 2016. // URL: https://sentinelsat.readthedocs.io/en/stable/index.html (дата обращения 07.10.2024).

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ СПОРТИВНОЙ ФЕДЕРАЦИИ С ПРОГРАММНЫМ МОДУЛЕМ ПЛАНИРОВАНИЯ ТРЕНИРОВОЧНОГО ПРОЦЕССА

Сосегова Виктория Андреевна
студент,
кафедра электронных вычислительных машин,
Вятский государственный университет,
РФ, г. Киров
E-mail: stud160243@vyatsu.ru

Крутиков Александр Константинович
ст. преподаватель,
кафедра электронных вычислительных машин,
Вятский государственный университет,
РФ, г. Киров
E-mail: yadrodisk@yandex.ru

DEVELOPMENT OF A SPORTS FEDERATION INFORMATION SYSTEM WITH A TRAINING PROCESS PLANNING SOFTWARE MODULE

Victoria Sosegova
Student,
Department of Electronic Computing Machines,
Vyatka State University,
Russia, Kirov

Krutikov Alexander
Senior lecturer,
Department of Electronic Computing Machines,
Vyatka State University,
Russia, Kirov

АННОТАЦИЯ

В данной статье описана разработка и реализация информационной системы региональной спортивной федерации с использованием отдельного программного модуля для прогнозирования и планирования. Описана структура системы, приведены макеты и экранные формы интерфейса. Описан отдельный программный модуль на основе нейронной сети обобщенной регрессии. Проанализированы требования к структуре и архитектура информационной системы, рассмотрены различные средства разработки. Разработана и реализована база данных, произведено проектирование и реализация информационной системы, произведена экспериментальная апробация. Разработан отдельный программный модуль планирования, произведены предварительные эксперименты по прогнозированию спортивных результатов в дисциплинах легкой атлетики. Произведены успешные эксперименты с GRNN сетью. Приведены структурные схемы и графики.

ABSTRACT

This article describes the development and implementation of the information system of the regional sports federation using a separate software module for forecasting and planning. The structure of the system is described, the layouts and screen forms of the interface are given. A separate software module based on a generalized regression neural network is described. The requirements for the structure and architecture of the information system are analyzed, various development tools are considered. A database has been developed and implemented, an information system has been designed and implemented, and experimental testing has been carried out. A separate planning software module has been developed, and preliminary experiments have been carried out to predict sports results in athletics disciplines. Successful experiments with the GRNN network have been carried out. Block diagrams and graphs are provided.

Ключевые слова: информационная система, спортивная федерация, нейронная сеть, обучающая выборка, прогнозирование, планирование.

Keywords: information system, sports federation, neural network, training sample, forecasting, planning.

Библиографическое описание: Сосегова В.А., Крутиков А.К. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ СПОРТИВНОЙ ФЕДЕРАЦИИ С ПРОГРАММНЫМ МОДУЛЕМ ПЛАНИРОВАНИЯ ТРЕНИРОВОЧНОГО ПРОЦЕССА // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18402*

«Информационные системы в физической культуре и спорте» объединяют в себе две важные сферы - информационные технологии и физическую активность человека. В современном мире данные две области активно взаимодействуют и дополняют друг друга, создавая новые возможности и преимущества для развития спорта и физической культуры [1].

Информационные системы в физической культуре и спорте представляют собой программно-аппаратные комплексы, которые используются для сбора, хранения, обработки, анализа и представления информации о физической активности и спортивных показателях. Они помогают тренерам, спортсменам и другим участникам спортивного процесса получать и использовать информацию для повышения своей эффективности и достижения лучших результатов [2].

Информационные системы в физической культуре и спорте могут включать в себя различные элементы, такие как:

- Базы данных для хранения информации о спортсменах, тренировках, соревнованиях и других событиях.
- Веб-порталы и мобильные приложения для доступа к информации и взаимодействия между тренерами, спортсменами и другими участниками спортивного процесса.
- Системы анализа данных и прогнозирования результатов на основе статистических и математических моделей.
- Системы видеонаблюдения и анализа движений для контроля техники и исправления ошибок.
- Системы электронного табло, видео аналитики и онлайн-трансляций для обеспечения комфорта и удобства зрителей и болельщиков.

Применение информационных систем в физической культуре и спорте имеет ряд преимуществ. Они позволяют рационализировать процессы управления спортивными организациями, повысить эффективность тренировок, а также повысить интерес и уровень вовлеченности зрителей. Однако, реализация информационных систем в физической культуре и спорте также сопряжена с определенными проблемами. Это могут быть сложности в внедрении новых технологий, недостаточная осведомленность и обучение персонала, проблемы с конфиденциальностью и защитой данных, а также неполное использование возможностей систем из-за низкой информационной грамотности пользователей. Тем не менее, развитие информационных систем в физической культуре и спорте продолжается и предоставляет все больше новых возможностей. Повышение эффективности тренировок, оптимизация процессов управления и контроля, повышение интереса к спорту.

В большинстве случаев существующие платформы фокусируются на предоставлении информации о спортивных мероприятиях, правилах и регламентах, расписаниях соревнований, результатов, новостях, а также на продвижении спорта и привлечения новых участников и болельщиков. На данных сайтах можно найти сведения о федерации, пройти обучения и курсы для тренеров и спортсменов. Однако большинство сайтов лишены возможности контролирования результатов спортсменов, что играет важную роль в достижении высоких результатов. Отсутствие функционала, связанного с автоматизацией тренировочного процесса, может создавать определенные трудности для пользователя.

Возможность создавать группы спортсменов, ведение дневника спортсмена на соревнование, контролировать его результат, всё это представляет собой значимую потребность, которая не удовлетворена существующими платформами. В связи с этим возникает необходимость в создании веб-сайта, которое будет ориентировано на социальные аспекты. Такой сайт должен предоставлять пользователям не только информацию о федерации спорта, но и позволять тренеру контролировать результаты спортсмена и проделанную работу. Такой подход к созданию информационной системы позволит повысить популярность лёгкой в атлетики в регионе и повысить эффективность работы спортсмена.

Проектирование информационной системы спортивной федерации включает в себя несколько этапов. На этапе анализа требований необходимо определить, какие функции должна выполнять информационная система и какие данные она будет обрабатывать. В данном случае, это данные о спортсменах, тренерах, соревнованиях, результатах и т. д.

Этап проектирования архитектуры определяет, как будут взаимодействовать между собой различные компоненты системы, а также какие технологии будут использоваться для её реализации.

Разработка модели данных. Модель данных описывает структуру данных, которые будут храниться в системе. Она может быть представлена в виде диаграммы или таблицы.

Создание прототипа. Прототип — это упрощённая версия системы, которая позволяет проверить работоспособность основных функций и удобство использования. Заключительным этапом является тестирование системы. Тестирование проводится после создания прототипа для выявления ошибок и недочётов в работе системы. Далее следуют внедрение и сопровождение системы, но в данной работе они не рассматриваются.

На каждом из этих этапов необходимо учитывать особенности работы спортивной федерации, её потребности и возможности. Также важно обеспечить безопасность данных и защиту от несанкционированного доступа.

Для проектирования информационной системы можно использовать различные инструменты и технологии, такие как:

ER-диаграммы — для моделирования структуры данных;

UML (Unified Modeling Language) — для проектирования архитектуры системы;

CASE-средства — для автоматизации процесса проектирования;

Языки программирования — для реализации системы.

Функции, которые необходимо реализовать в разрабатываемой информационной системе спортивной федерации:

- Ведение базы данных спортсменов, тренеров, судей и других участников соревнований.
- Регистрация и учёт результатов соревнований.
- Планирование и организация соревнований.
- Планирование и организация тренировочного процесса.
- Расчёт рейтингов и статистики.
- Анализ и отчётность.

В качестве архитектуры информационной системы выбрана трехзвенная клиент-серверная архитектура. Разработана ER-диаграмма базы данных (рисунок 1), база данных реализована с использованием СУБД MySQL.

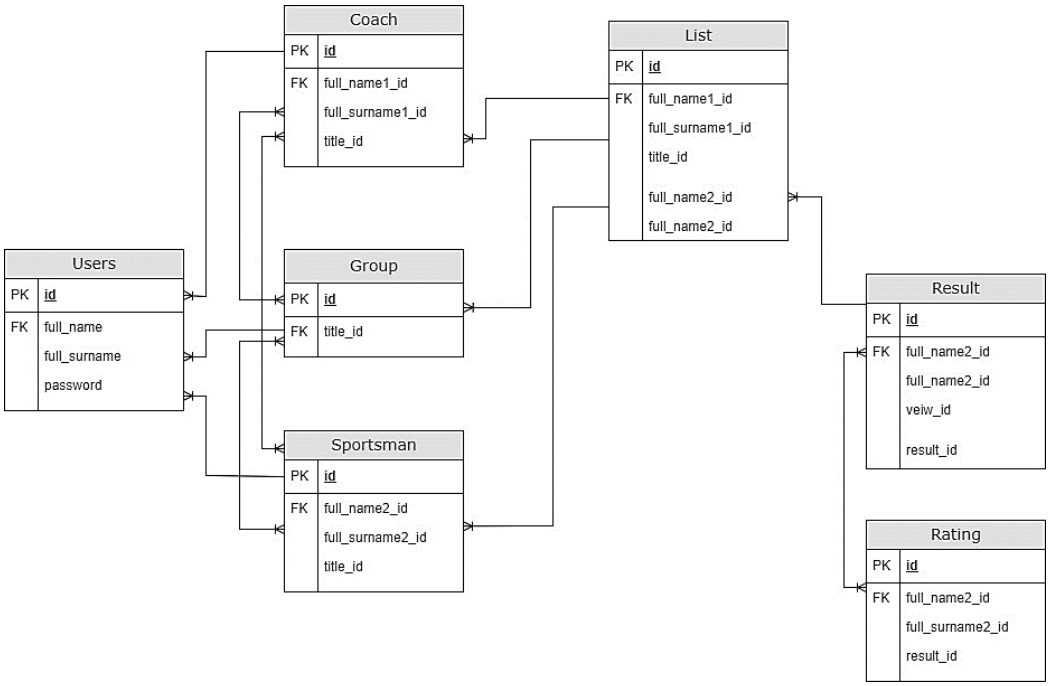

Рисунок 1. ER-диаграмма базы данных информационной системы региональной спортивной федерации

Для навигации по информационной системе используется алгоритм (рисунок 2).

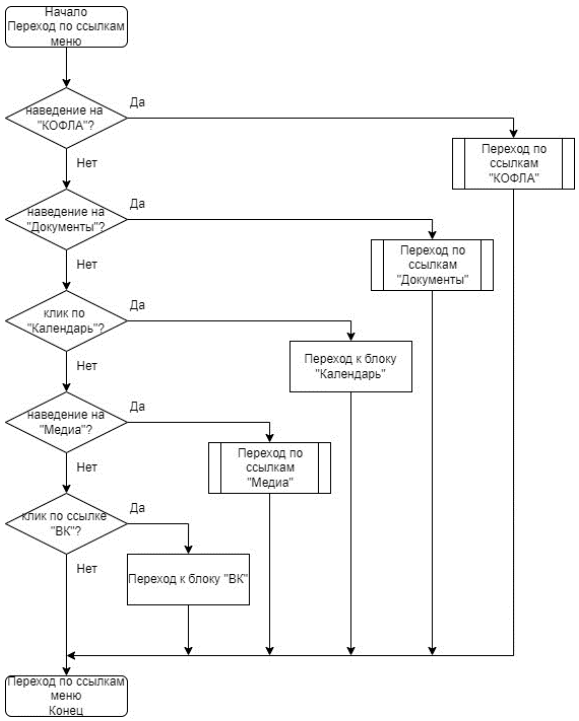

Рисунок 2. Схема алгоритма навигации в информационной системе

Спроектирована и реализована информационная система региональной спортивной федерации по виду спорта легкая атлетика. Личный кабинет спортсмена позволяет видеть рейтинг и график изменения результатов в течении некоторого временного периода (рисунки 3-4).

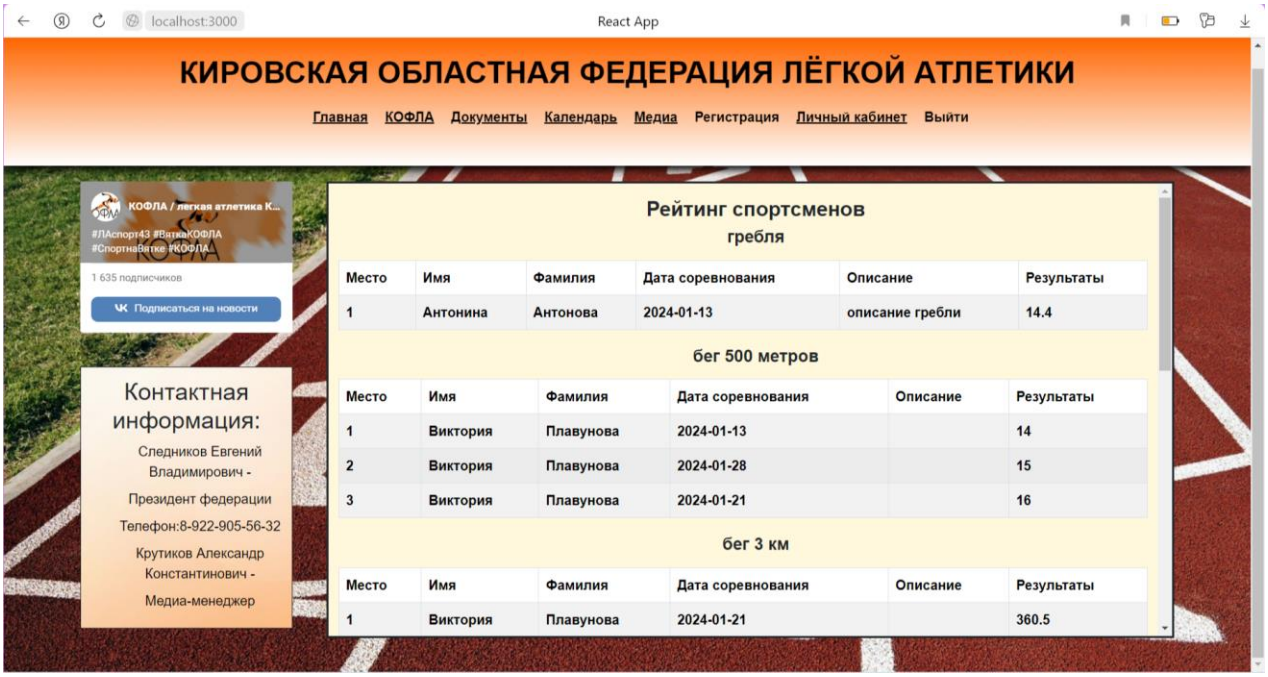

Рисунок 3. Рейтинг спортсменов (экранная форма)

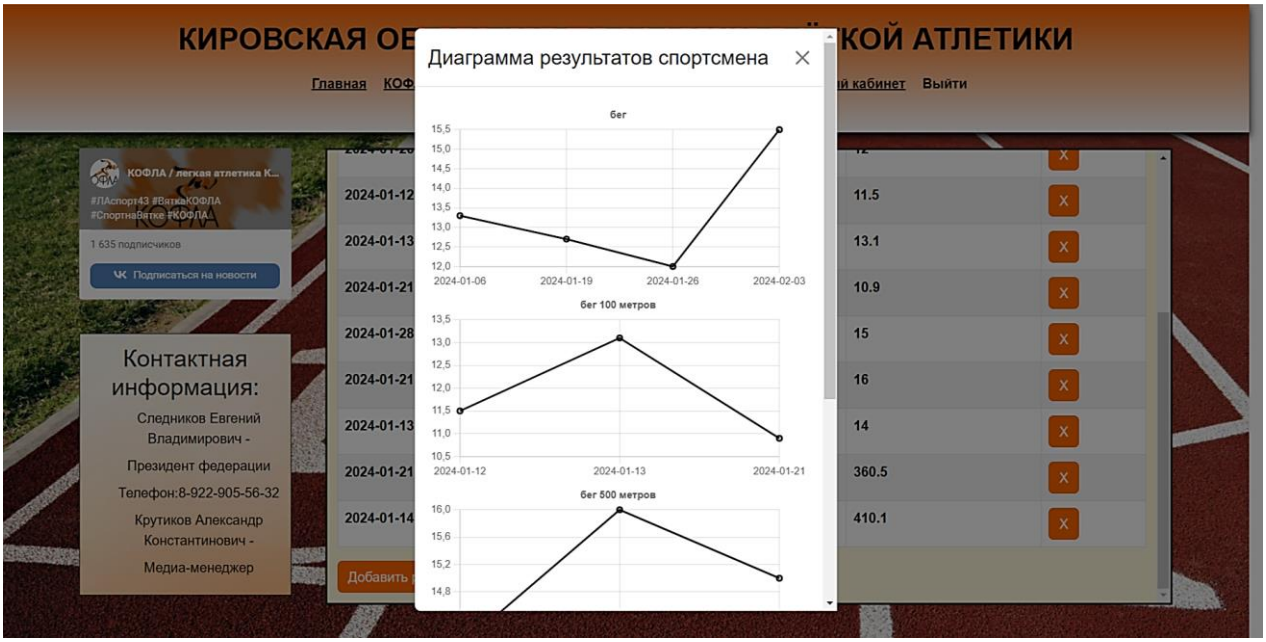

Рисунок 4. График изменения результата

В процессе проектирования системы и реализации системы, реализованы все основные разделы, в том числе информационные.

Рисунок 5. Макет десктопной версии приложения для работы с информационной системой региональной спортивной федерации

Начато внедрение отдельного программного модуля планирования спортивного результата, для корректировки тренировочного процесса. Планирование спортивного результата производится на основе результата прогнозирования численного показателя атлета. В качестве основного инструмента прогнозирования используются нейронные сети, в частности обобщенно-регрессионная нейронная сеть.

Обобщенно-регрессионная нейронная сеть (GRNN) [3] (рисунок 6) — это тип нейронной сети, который используется для решения задач регрессии, то есть предсказания числовых значений. GRNN является одним из методов непараметрической регрессии и может быть использован для аппроксимации функций и прогнозирования временных рядов. GRNN работает на основе принципа «ближайшего соседа», где для каждого нового наблюдения находится ближайший сосед в обучающей выборке и его значение используется как предсказание для нового наблюдения. Для определения расстояния между наблюдениями обычно используется евклидово расстояние.

Алгоритм работы обобщённо-регрессионной нейронной сети:

- Обучение сети на основе обучающей выборки данных. На этом этапе вычисляются центры масс для каждого класса и дисперсии вокруг этих центров.
- Предсказание значения для новых наблюдений. Для каждого нового наблюдения определяется ближайший центр массы и предсказывается значение с учётом расстояния до этого центра и соответствующей дисперсии.

Преимущества обобщённо-регрессионных нейронных сетей включают их простоту, быструю сходимость и способность работать с зашумлёнными данными. Однако они могут быть менее точными по сравнению с другими методами регрессии при наличии сложных зависимостей в данных.

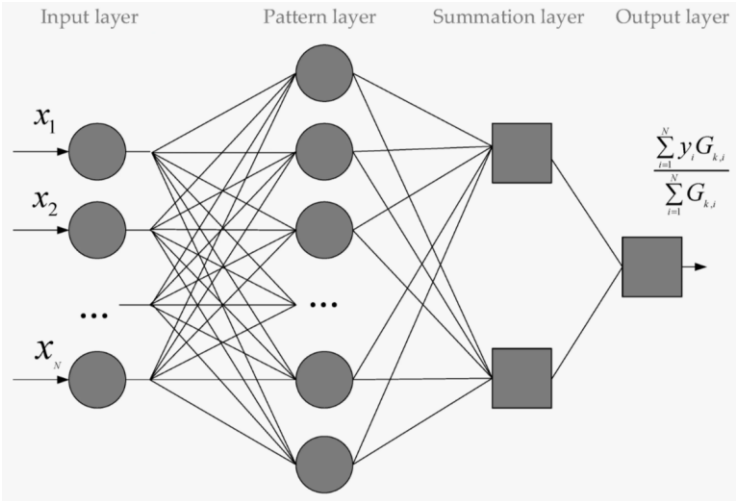

Рисунок 6. Структура GRNN-сети

Прогнозирование с использованием GRNN-сети включает в себя ряд этапов. Необходимо собрать данные, которые будут использоваться для обучения и тестирования модели. Данные должны содержать информацию о прошлых значениях прогнозируемой переменной. Данные необходимо очистить, обработать и преобразовать в формат, подходящий для использования в нейронных сетях. Это может включать в себя удаление выбросов, заполнение пропущенных значений и масштабирование данных. На основе подготовленных данных обучить GRNN-сеть. Это включает в себя определение количества слоёв, нейронов в каждом слое, функции активации и других параметров модели. После обучения модели необходимо протестировать её на новых данных, чтобы оценить её точность и надёжность. Когда модель будет готова, её можно использовать для прогнозирования будущих значений на основе новых данных.

Программный модуль с GRNN сетью используется в качестве прогнозирования спортивных результатов на отдельных соревнованиях, в отдельных легкоатлетических дисциплинах [4].

Пример [5] тестового прогнозирования результата в дисциплине прыжок в длину с использованием GRNN сети приведена на графике (рисунок 7). В результате последних экспериментов прогнозирования результатов данной спортивной дисциплины минимальная среднеквадратичная ошибка обучения не превышала 0,312.

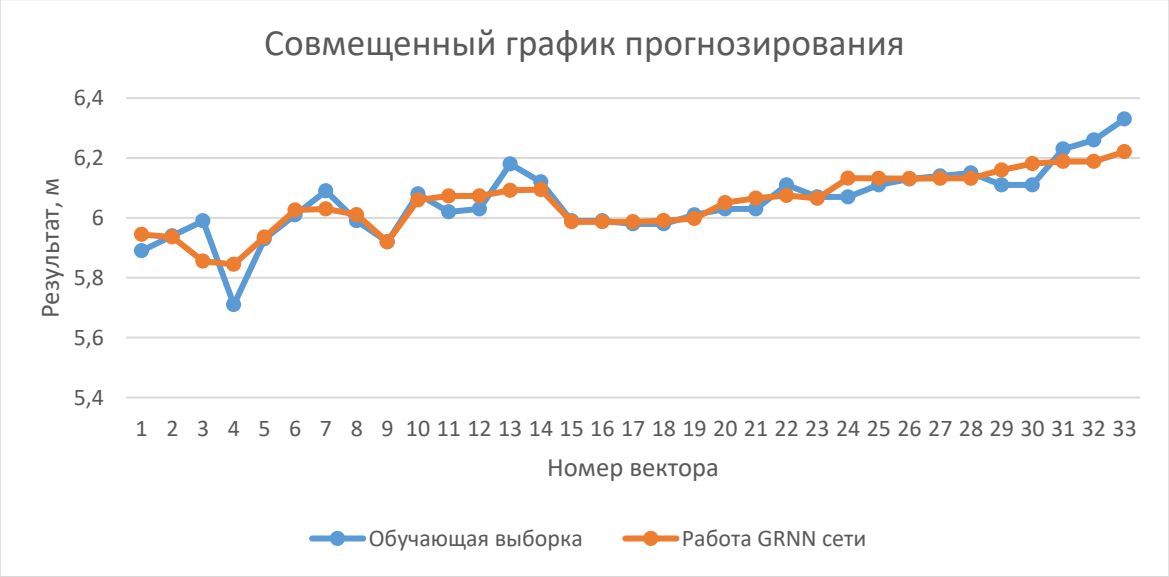

Рисунок 7. Совмещенный график прогнозирования

Нейронная сеть обобщенной регрессии успешно зарекомендовала себя при прогнозировании результатов в этом индивидуальном виде спорта, используя небольшую обучающую выборку.

Нейронная сеть обобщенной регрессии способна прогнозировать результаты в дисциплинах легкоатлетического метания, результаты в прыжковых дисциплинах [5]. Эту нейронную сеть следует использовать для прогнозирования спортивных результатов в отдельном виде спорта – легкой атлетике.

Нейронная сеть обобщенной регрессии может быть использована для прогнозирования спортивных результатов в отдельных видах спорта в группе дисциплин, характеризующихся активной двигательной активностью спортсменов с максимальным проявлением физических и психических качеств.

В текущее время производится тестирование информационной системы. Рассматривается возможность улучшения и увеличения масштабов системы.

Список литературы:

1. Воронов, И.А. Информационные технологии в физической культуре и спорте: Электронный учебник / И.А. Воронов; СПб ГУФК им. П.Ф. Лесгафта. -СПб.: изд-во СПб ГУФК им. П.Ф. Лесгафта, 2005 – 80 с.
2. Кривецкий, И.Ю. Возможности применения технологии нейро-нечетких сетей в некоторых видах спорта Информатика и системы управления / И.Ю. Кривецкий, Г.И. Попов// 2013. С. 19-21.
3. Обобщенно-регрессионная нейронная сеть URL: https://intuit.ru/studies/courses/3735/977/lecture/14689 (дата обращения 09.10.2024).
4. Мельцов В.Ю., Подковырин В.Д., Клюкин В.Л., Крутиков А.К., Использование каскадной нейронной сети прямой передачи для прогнозирования спортивных достижений в толкании ядра – Казань: Научно-технический вестник Поволжья № 4, 2018. с. 136-139.
5. Крутиков, А.К. Прогнозирование спортивных результатов в индивидуальных видах спорта с помощью обобщенно-регрессионной нейронной сети / А.К. Крутиков. — Текст : непосредственный // Молодой ученый. — 2018. — № 12 (198). — С. 22-26.

DOI - 10.32743/UniTech.2024.127.10.18465

МЕТОДЫ РАЗРАБОТКИ АДАПТИВНЫХ ИНТЕРФЕЙСОВ ДЛЯ ANDROID ПРИЛОЖЕНИЙ С УЧЁТОМ ЭКРАННЫХ ХАРАКТЕРИСТИК

Терехов Владислав Игоревич
разработчик мобильных приложений,
Mobilesource Corp,
США, Флорида, Бока-Ратон
E-mail: Terekhow777@gmail.com

METHODS FOR DEVELOPING ADAPTIVE INTERFACES FOR ANDROID APPLICATIONS WITH SCREEN CHARACTERISTICS IN MIND

Vladislav Terekhov
Mobile Applications Developer, Mobilesource Corp,
United States, Boca Raton, Florida

АННОТАЦИЯ

Разработка адаптивных пользовательских интерфейсов для Android приложений с использованием Java является важной задачей для обеспечения оптимального пользовательского опыта на устройствах с различными характеристиками экрана. Адаптивные интерфейсы позволяют приложению автоматически подстраиваться под размер экрана и ориентацию устройства, что повышает удобство взаимодействия. Java, как основной язык разработки для Android, предоставляет мощные инструменты и библиотеки для создания гибких интерфейсов, которые поддерживают корректное отображение элементов на различных устройствах. Ключевые компоненты, такие как ConstraintLayout и динамические макеты, обеспечивают возможность адаптации интерфейсов для экранов разного разрешения. Использование Java в сочетании с инструментами для проектирования интерфейсов, такими как Figma, Sketch и InVision, позволяет разработчикам эффективно создавать приложения, поддерживающие широкий спектр устройств. Это значительно улучшает пользовательский опыт и повышает конкурентоспособность приложений.

ABSTRACT

Developing responsive user interfaces for Android applications using Java is an important task to ensure an optimal user experience on devices with different screen characteristics. Adaptive interfaces allow the application to automatically adjust to the screen size and orientation of the device, which increases the convenience of interaction. Java, as the main development language for Android, provides powerful tools and libraries for creating flexible interfaces that support the correct display of elements on various devices. Key components such as ConstraintLayout and dynamic layouts provide the ability to adapt interfaces for screens of different resolutions. Using Java in combination with interface design tools such as Figma, Sketch, and InVision allows developers to efficiently create applications that support a wide range of devices. This significantly improves the user experience and increases the competitiveness of applications.

Ключевые слова: адаптивные интерфейсы, Android, Java, разработка, мобильные приложения, ConstraintLayout, UX/UI, гибкость интерфейсов.

Keywords: adaptive interfaces, Android, Java, development, mobile applications, ConstraintLayout, UX/UI, interface flexibility.

Введение

Разработка адаптивных пользовательских интерфейсов для Android-приложений является важной задачей современной мобильной разработки. С увеличением разнообразия мобильных устройств, таких как смартфоны и планшеты, различающихся по размерам и разрешению экранов, возрастает потребность в создании интерфейсов, которые автоматически подстраиваются под условия конкретного устройства.

Адаптивные интерфейсы обеспечивают удобство использования, делая взаимодействие с приложением интуитивным и комфортным для пользователя, независимо от характеристик экрана.

Применение языка программирования Java для разработки адаптивных интерфейсов в Android-приложениях также остается актуальным, несмотря на появление новых технологий и языков, таких как Kotlin. Java обладает широким набором инструментов и библиотек, что делает её универсальным решением для создания гибких интерфейсов.

Библиографическое описание: Терехов В.И. МЕТОДЫ РАЗРАБОТКИ АДАПТИВНЫХ ИНТЕРФЕЙСОВ ДЛЯ ANDROID ПРИЛОЖЕНИЙ С УЧЁТОМ ЭКРАННЫХ ХАРАКТЕРИСТИК // Universum: технические науки : электрон. научн. журн. 2024. 10(127). URL: *https://7universum.com/ru/tech/archive/item/18465*

Кроме того, использование компонентов интерфейса, таких как ConstraintLayout и другие динамические макеты, позволяет разработчикам гибко адаптировать приложения для работы на устройствах с различными параметрами экрана.

Актуальность темы определяется необходимостью разработки мобильных приложений, способных обеспечивать качественный пользовательский опыт на широком спектре устройств. Целью работы является исследование подходов к разработке адаптивных интерфейсов для Android-приложений с использованием Java, а также рассмотрение современных инструментов и компонентов, способствующих повышению гибкости и функциональности интерфейсов на различных устройствах.

1. Принципы адаптивного дизайна для Android приложений

Адаптивный интерфейс представляет собой концепцию, в которой элементы интерфейса автоматически подстраиваются под различные параметры экрана. Это может включать в себя динамическое изменение размеров изображений, перестановку компонентов и прочие изменения, направленные на обеспечение корректного отображения на разных устройствах. К 2024 году внедрение адаптивного дизайна в приложения для Android резко возросло, особенно в таких отраслях, как электронная коммерция, банковское дело и приложения для повышения производительности, где критически важны улучшенный пользовательский опыт и более быстрое время загрузки. В 2019 году внедрение было в основном обусловлено более крупными компаниями, ориентированными на обеспечение совместимости между устройствами, но к 2024 году более 50% разработчиков Android внедрили принципы адаптивного дизайна в свои приложения, чтобы адаптироваться к постоянно растущему разнообразию форматов устройств. Ниже на рисунке 1 будет представлено развитие адаптивного дизайна в Android приложениях с 2019 по 2024 годы.

Рисунок 1. Развитие адаптивного дизайна в Android приложениях с 2019 по 2024 годы [8].

Основные принципы адаптивного дизайна заключаются в обеспечении гибкости интерфейсов для корректной работы на различных устройствах. Ключевую роль в этом играют динамические макеты, которые подстраиваются под размеры экранов. Для этого применяются такие компоненты, как ConstraintLayout, LinearLayout и RelativeLayout, позволяющие задавать относительные размеры и взаимное расположение элементов. Особенно выделяется ConstraintLayout, предоставляющий обширные возможности для создания сложных зависимостей между элементами интерфейса, что делает его незаменимым инструментом для адаптивных решений [1]. Ниже в таблице 1 будут описаны принципы реализации адаптивного интерфейса в Android.

Таблица 1.

Принципы реализации адаптивного интерфейса в Android [2]

Принцип	Описание	Преимущества
Использование гибких размеров компонентов	Применение параметров wrap_content и match_parent вместо фиксированных значений для компонентов интерфейса. Это позволяет элементам интерфейса изменять размеры в зависимости от устройства и экрана.	Обеспечивает корректное отображение интерфейса на экранах различных размеров, улучшает гибкость и адаптируемость интерфейса под различные устройства.
Медиа-запросы и работа с адаптивными ресурсами	Использование отдельных каталогов ресурсов для разных разрешений и ориентаций экрана (портретной и альбомной), что позволяет создавать адаптивные макеты для смартфонов, планшетов и других устройств.	Позволяет создавать различные макеты под разные устройства и ориентации экрана, обеспечивая более точную настройку отображения для каждой категории устройств.
Многостороннее тестирование	Тестирование интерфейса на различных устройствах с разными разрешениями, диагоналями экрана и ориентациями для проверки его корректного отображения и работы.	Обеспечивает универсальность интерфейса, устраняет возможные ошибки отображения на разных устройствах, улучшает пользовательский опыт на множестве различных моделей устройств.

Таким образом принципы адаптивного дизайна для Android приложений направлены на создание интерфейсов, которые гибко подстраиваются под различные размеры экранов и устройства. Основу адаптивности составляют динамические макеты, такие как ConstraintLayout, LinearLayout и RelativeLayout, которые обеспечивают корректное отображение элементов на любых экранах за счет их относительных размеров и взаимного расположения. Особую роль играет ConstraintLayout, предлагающий широкие возможности для создания сложных зависимостей между элементами интерфейса, что делает его важным инструментом для разработки адаптивных решений в Android.

2. Инструменты и библиотеки для разработки адаптивных интерфейсов

Программное обеспечение Wondershare Mockitt представляет собой один из ведущих инструментов для создания дизайна пользовательских интерфейсов на Android. Данное приложение обладает широким функционалом и удобным интерфейсом, что делает его доступным даже для пользователей без опыта в проектировании. Оно поддерживает создание прототипов для различных платформ, таких как Android, ПК, iPhone и другие. Также существует возможность настройки устройства в соответствии с требованиями пользователя.

Основные возможности:

- Большой выбор иконок и виджетов, облегчающих процесс создания пользовательских интерфейсов на Android.
- Синхронизация файлов и создание связей между экранами.
- Простота обмена проектами через ссылки или QR-коды.
- Возможность совместной работы в режиме реального времени с доступом к комментированию, выбору цветов и другим функциям для удобного проектирования интерфейсов.

Программное обеспечение Sketch — это профессиональное решение для создания дизайна пользовательского интерфейса Android, предлагающее широкий набор инструментов для оптимизации рабочего процесса. Оно значительно упрощает работу с графикой, объединяя в себе функции, ранее требовавшие использования нескольких программ, таких как Illustrator и Photoshop. Основное преимущество Sketch заключается в его специализации на проектировании UI/UX, что позволяет сосредоточиться на создании высококачественных интерфейсов.

Ключевые функции:

- Возможность установки плагинов для ускорения работы.
- Инструменты для настройки макетов, направляющих сеток, текстовых стилей и других элементов дизайна.
- Создание каркасных моделей интерфейсов для различных устройств.

Приложение InVision App предоставляет дизайнеру Android интерфейсов возможность создания макетов с высокой степенью интерактивности. Инструмент ориентирован на создание прототипов с добавлением жестов, анимаций и других взаимодействий. InVision также поддерживает интеграцию с программами Sketch и Adobe Photoshop, что позволяет легко загружать проекты и продолжать работу в знакомых средах.

Основные возможности:

- Поддержка взаимодействия и сотрудничества с клиентами и командой.
- Автоматические обновления и трансляции изменений в реальном времени.
- Оптимизированный процесс создания интерактивных прототипов, которые точно отражают функциональность конечного продукта.

Marvel — это инструмент, упрощающий создание каркасов и прототипов для интерфейсов Android. Благодаря интуитивному интерфейсу, работа с Marvel становится доступной даже для начинающих пользователей. Приложение совместимо с Adobe Photoshop и Sketch, что позволяет добавлять плагины и расширять функциональные возможности.

Основные функции:

- Интеграция с основными графическими редакторами для создания качественных прототипов.
- Возможность совместной работы над проектом с другими участниками команды.
- Простота обмена идеями и результатами разработки.

Figma — это мощный инструмент для коллективного дизайна, который активно используется профессионалами для создания Android интерфейсов. Figma позволяет командам работать над проектами в режиме реального времени, предоставляя возможность редактирования и комментирования на всех этапах разработки.

Ключевые возможности:

- Управление версиями проекта с возможностью добавления описаний.
- Доступ к компонентной библиотеке для быстрого поиска и использования элементов дизайна.
- Инструменты для эффективного командного взаимодействия.

Zeplin — это платформа для проектирования интерфейсов, ориентированная на взаимодействие между дизайнерами и разработчиками. Благодаря Zeplin процесс передачи проектов становится более удобным и быстрым, что повышает эффективность совместной работы.

Основные возможности:

- Поддержка экспорта проектов из таких инструментов, как Adobe Photoshop, Sketch, Figma, Adobe XD.
- Генерация спецификаций и ресурсов для разработчиков.
- Возможность автоматического создания фрагментов кода для упрощения работы над проектом [3].

Proto.io — инструмент, который позволяет легко создавать мобильные интерфейсы, используя готовые элементы и шаблоны. Он предоставляет функции для предварительного просмотра прототипов на реальных устройствах, что помогает выявить возможные ошибки перед финальной версией приложения.

Axure RP предлагает мощные функции для создания интерактивных макетов и каркасов мобильных приложений. Его функциональные возможности включают в себя инструменты для разработки UX, позволяя создавать макеты для любых устройств с минимальными усилиями. Axure предоставляет разнообразные тарифные планы, что делает его доступным как для индивидуальных разработчиков, так и для крупных команд.

Justinmind предоставляет дизайнерским командам возможности для быстрого создания прототипов мобильных приложений с использованием готовых виджетов и шаблонов. Этот инструмент особенно полезен для начального этапа разработки, позволяя разработчикам тестировать пользовательские интерфейсы до начала кодирования [4].

Таким образом, каждая из платформ предлагает свои уникальные функции, адаптированные к различным потребностям разработчиков и дизайнеров, что помогает оптимизировать процессы проектирования и создания мобильных приложений.

3. Практическое применение Java для создания адаптивных интерфейсов

Алгоритм взаимодействия с пользователем зависит от текущего интерфейса, который отображается в приложении. Java является универсальным языком программирования, который широко применяется, в частности, при разработке мобильных приложений. С момента своего появления в 1995 году Java заняла ведущие позиции в создании Android-приложений, которые составляют значительную долю мобильного рынка. Популярность языка объясняется его ключевыми характеристиками, такими как объектно-ориентированная структура, простота синтаксиса и наличие обширных библиотек. Эти особенности формируют мощную среду разработки, способствующую развитию индустрии мобильных решений.

Java также активно используется для создания кроссплатформенных мобильных решений с помощью технологий, таких как Apache Cordova и Xamarin. Эти инструменты позволяют разрабатывать приложения, которые могут быть запущены на различных платформах, что увеличивает производительность и снижает затраты на разработку. Ниже в таблице 2 будут рассмотрены основные положительные и отрицательные аспекты использования этого языка в мобильной среде.

Таблица 2.

Положительные и отрицательные аспекты использования Java для разработки мобильных приложений [5]

Категория	Описание	Преимущества	Недостатки
Кросс-платформенность	Java поддерживает принцип «напиши один раз, запускай везде», что позволяет запускать приложения на разных платформах, поддерживающих JRE.	Ускорение разработки для нескольких платформ, снижение затрат	Возможны проблемы с оптимизацией под некоторые платформы
Объектно-ориентированная структура	ООП в Java способствует модульности и позволяет легко использовать повторно код, что упрощает поддержку и изменения в приложениях.	Модульность, упрощение поддержки и изменений	Может требовать больше ресурсов для сложных систем
Развитая экосистема	Java обладает широким спектром инструментов и библиотек, что ускоряет разработку и позволяет создавать многофункциональные приложения. Сообщество разработчиков помогает решать сложные задачи.	Ускорение разработки, доступ к сообществу, решение сложных задач	Потенциальная сложность в освоении для начинающих разработчиков
Богатый набор библиотек	Java предоставляет доступ к многочисленным библиотекам и API, что облегчает решение сложных задач и оптимизирует разработку мобильных приложений.	Облегчение разработки благодаря множеству готовых решений	Множество библиотек может вызвать трудности в выборе подходящих
Нагрузка на производительность	Приложения на Java могут быть менее производительными по сравнению с аналогами на Swift или Kotlin из-за виртуальной машины Java и механизма сборки мусора.	Позволяет работать на разных платформах благодаря JRE	Снижение производительности на устройствах с ограниченными ресурсами
Избыточность кода	Java известна многословным синтаксисом, что приводит к необходимости писать более объемные конструкции, усложняя написание и поддержку кода.	Строгий синтаксис может повысить точность кода	Увеличение объема кода, снижение читаемости, вероятность ошибок
Скорость выполнения	Приложения на Java часто запускаются медленнее по сравнению с нативными решениями, что создает баланс между кроссплатформенностью и быстродействием.	Возможность работы на большом количестве устройств	Медленное выполнение по сравнению с нативными приложениями, необходимость оптимизации

Далее рассмотрим практический пример использования Java в разработке пользовательских интерфейсов. Для удобства работы с интерфейсом приложения код может содержать запросы на наличие панели, позволяющей взаимодействовать с различными элементами, что облегчает создание запросов для всех компонентов макета. Например, в случае наличия отдельных элементов интерфейса важно проверить их доступность перед выполнением любых операций. В качестве примера можно привести приложение News Reader, где кнопка для вызова меню присутствует только в системах Android ниже версии 3.0. В более поздних версиях ее функции выполняет компонент ActionBar (API 11 и выше) [6]. Для проверки этого элемента можно использовать следующий фрагмент кода:

```
java
Button catButton = (Button) findViewById(R.id.categorybutton);
OnClickListener listener = /* создать обработчик событий */;
if (catButton != null) {
   catButton.setOnClickListener(listener);
}
```

Это действие является стандартным для элементов интерфейса, который может различаться в зависимости от текущего макета. Например, в двухпанельном режиме, при выборе заголовка статьи в списке, контент отображается в правой панели, в то время как в однопанельном режиме запускается отдельная активность:

```
java
@Override
public void onHeadlineSelected(int index) {
   mArtIndex = index;
   if (mIsDualPane) {
      mArticleFragment.displayArticle(mCurrentCat.getArticle(index));
   } else {
      Intent intent = new Intent(this, ArticleActivity.class);
      intent.putExtra("catIndex", mCatIndex);
      intent.putExtra("artIndex", index);
      startActivity(intent);
   }
}
```

Аналогично, выбор режима навигации зависит от макета: в двухпанельном режиме используются вкладки для навигации, а в однопанельном – раскрывающийся список. Примером этого подхода может служить следующий код:

```
java
final String CATEGORIES[] = { "Лучшие статьи", "Политика", "Экономика", "Новости технологий" };
public void onCreate(Bundle savedInstanceState) {
   if (mIsDualPane) {
      actionBar.setNavigationMode(android.app.ActionBar.NAVIGATION_MODE_TABS);
      for (int i = 0; i < CATEGORIES.length; i++) {
         actionBar.addTab(actionBar.newTab().setText(CATEGORIES[i]).setTabListener(handler));
      }
   } else {
      actionBar.setNavigationMode(android.app.ActionBar.NAVIGATION_MODE_LIST);
      SpinnerAdapter adapter = new ArrayAdapter(this, R.layout.headline_item, CATEGORIES);
      actionBar.setListNavigationCallbacks(adapter, handler);
   }
}
```

Повторное использование компонентов интерфейса, таких как фрагменты, позволяет применять одни и те же элементы на разных экранах устройства. Например, в случае с приложением News Reader на больших экранах текст статей выводится в правой панели, тогда как на маленьких он открывается в новой активности. В этих случаях можно использовать подклассы фрагментов в нескольких активностях. Примером может служить ArticleFragment, который используется в двухпанельной конфигурации:

```
xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="horizontal">
   <fragment android:id="@+id/headlines"
         android:layout_width="400dp"
         android:name="com.example.android.newsreader.HeadlinesFragment"
         android:layout_marginRight="10dp"
         android:layout_height="fill_parent"/>
   <fragment android:id="@+id/article"
         android:name="com.example.android.newsreader.ArticleFragment"
         android:layout_width="fill_parent"
         android:layout_height="fill_parent"/>
</LinearLayout>
```

В случае с маленькими экранами, тот же фрагмент можно использовать в отдельной активности без необходимости прописывать макет XML:

```
java
ArticleFragment fragment = new ArticleFragment();
getSupportFragmentManager().beginTransaction().add(android.R.id.content, fragment).commit();
```

Таким образом, важно учитывать возможность повторного использования фрагментов в различных активностях. Для этого можно определить интерфейс взаимодействия между фрагментом и активностью, который должен быть реализован в самой активности [7]. Например, в приложении News Reader фрагмент HeadlinesFragment реализует интерфейс для оповещения о выборе заголовка:

```
public class HeadlinesFragment extends ListFragment {
   OnHeadlineSelectedListener mHeadlineSelectedListener;

   public interface OnHeadlineSelectedListener {
   void onHeadlineSelected(int index);
   }

   public void setOnHeadlineSelectedListener(OnHeadlineSelectedListener listener) {
      mHeadlineSelectedListener = listener;
   }

   @Override
   public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
   if (mHeadlineSelectedListener != null) {
        mHeadlineSelectedListener.onHeadlineSelected(position);
   }
   }
}
```

Этот подход позволяет гибко работать с различными конфигурациями устройства, обеспечивая оптимальное взаимодействие пользователя с приложением независимо от характеристик экрана.

Заключение

В результате анализа методов разработки адаптивных интерфейсов для Android приложений с использованием Java, можно заключить, что успешное создание качественных приложений напрямую зависит от грамотного применения гибких макетов и правильного выбора инструментов. Java, как основной язык программирования для Android, предоставляет множество возможностей для эффективного создания адаптивных интерфейсов. Ключевая роль в этом процессе принадлежит компонентам, обеспечивающим динамическую подстройку интерфейсов под различные устройства. Современные инструменты проектирования, такие как Figma, Sketch и InVision, способствуют более быстрому и удобному процессу разработки, что в конечном итоге улучшает пользовательский опыт.

**(По требованию Роскомнадзора информируем, что иностранное лицо, владеющее информационными ресурсами Google является нарушителем законодательства Российской Федерации – прим. ред.)*

Список литературы:

1. Згонникова А.О., Прокопенко А.А. Android разработка для бытовой техники // Наука, образование, инновации: актуальные вопросы и современные аспекты. – 2023. – С. 43-45.
2. Пономарёв Е.В. Разработка кредитных приложений на Android: особенности и вызовы //Вестник науки. – 2024. – Т. 2. – №. 9 (78). – С. 319-327.
3. Yang B. et al. UIS-hunter: Detecting UI design smells in Android apps //2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). – IEEE, 2021. – С. 89-92.
4. Кочубей Е.И. Влияние технологий на UX/UI-дизайн: перспективы и тенденции //Вестник науки. – 2024. – Т. 2. – №. 6 (75). – С. 1558-1566.
5. Блинова А.В. Противостояние языков программирования Kotlin и Java в разработке мобильных приложений // Актуальные исследования. – С. 38.

6. Курганова Д.А. Значимость UX/UI-дизайна интерфейса в разработке мобильных приложений // Научно-техническое и экономическое сотрудничество стран АТР в XXI веке. – 2021. – Т. 1. – С. 194-197.
7. Жестовский Я.С. Мобильное приложение на платформе Android //«Студент: наука, профессия, жизнь»: Материалы IX всероссийской. – 2022. – С. 161.
8. Building adaptive Android apps. [Электронный ресурс] Режим доступа: https://io.google*/2024/explore/d16737ba-e336-4b68-8928-24692a88e644/ (дата обращения 28.09.2024).

ДЛЯ ЗАМЕТОК

ДЛЯ ЗАМЕТОК

ДЛЯ ЗАМЕТОК

Научный журнал

UNIVERSUM:
ТЕХНИЧЕСКИЕ НАУКИ

№ 10(127)
Октябрь 2024

Часть 1

Свидетельство о регистрации СМИ: ЭЛ № ФС 77 – 54434 от 17.06.2013

Издательство «МЦНО»
123098, г. Москва, улица Маршала Василевского, дом 5, корпус 1, к. 74
E-mail: *mail@7universum.com*
www.7universum.com

Отпечатано в полном соответствии с качеством предоставленного оригинал-макета в типографии «Allprint»
630004, г. Новосибирск, Вокзальная магистраль, 1
16+