

МОЛОДОЙ УЧЁНЫЙ

ISSN 2072-0297

МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ



25 2026
ЧАСТЬ I

16+

Молодой ученый

Международный научный журнал

№ 25 (628) / 2026

Издается с декабря 2008 г.

Выходит еженедельно

Главный редактор: Ахметов Ильдар Геннадьевич, кандидат технических наук

Редакционная коллегия:

Жураев Хусниддин Олтинбоевич, доктор педагогических наук (Узбекистан)
Иванова Юлия Валентиновна, доктор философских наук
Каленский Александр Васильевич, доктор физико-математических наук
Кошербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)
Куташов Вячеслав Анатольевич, доктор медицинских наук
Лактионов Константин Станиславович, доктор биологических наук
Сараева Надежда Михайловна, доктор психологических наук
Абдрасилов Турганбай Курманбаевич, доктор философии (PhD) по философским наукам (Казахстан)
Авдеюк Оксана Алексеевна, кандидат технических наук
Айдаров Оразхан Турсункожаевич, кандидат географических наук (Казахстан)
Алиева Тарана Ибрагим кызы, кандидат химических наук (Азербайджан)
Ахметова Валерия Валерьевна, кандидат медицинских наук
Бердиев Эргаш Абдуллаевич, кандидат медицинских наук (Узбекистан)
Брезгин Вячеслав Сергеевич, кандидат экономических наук
Данилов Олег Евгеньевич, кандидат педагогических наук
Дёмин Александр Викторович, кандидат биологических наук
Дядюн Кристина Владимировна, кандидат юридических наук
Желнова Кристина Владимировна, кандидат экономических наук
Жуйкова Тамара Павловна, кандидат педагогических наук
Игнатова Мария Александровна, кандидат искусствоведения
Искаков Руслан Маратбекович, кандидат технических наук (Казахстан)
Калдыбай Кайнар Калдыбайулы, доктор философии (PhD) по философским наукам (Казахстан)
Кенесов Асхат Алмасович, кандидат политических наук
Коварда Владимир Васильевич, кандидат физико-математических наук
Комогорцев Максим Геннадьевич, кандидат технических наук
Котляров Алексей Васильевич, кандидат геолого-минералогических наук
Кузьмина Виолетта Михайловна, кандидат исторических наук, кандидат психологических наук
Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)
Кучерявенко Светлана Алексеевна, кандидат экономических наук
Лескова Екатерина Викторовна, кандидат физико-математических наук
Макеева Ирина Александровна, кандидат педагогических наук
Матвиенко Евгений Владимирович, кандидат биологических наук
Матроскина Татьяна Викторовна, кандидат экономических наук
Матусевич Марина Степановна, кандидат педагогических наук
Мусаева Ума Алиевна, кандидат технических наук
Насимов Мурат Орленбаевич, кандидат политических наук (Казахстан)
Паридинова Ботагоз Жаппаровна, магистр философии (Казахстан)
Прончев Геннадий Борисович, кандидат физико-математических наук
Рахмонов Азизхон Боситхонович, доктор педагогических наук (Узбекистан)
Семахин Андрей Михайлович, кандидат технических наук
Сенцов Аркадий Эдуардович, кандидат политических наук
Сенюшкин Николай Сергеевич, кандидат технических наук
Султанова Дилшода Намозовна, доктор архитектурных наук (Узбекистан)
Титова Елена Ивановна, кандидат педагогических наук
Ткаченко Ирина Георгиевна, кандидат филологических наук
Федорова Мария Сергеевна, кандидат архитектуры
Фозилов Садриддин Файзуллаевич, кандидат химических наук (Узбекистан)
Яхина Асия Сергеевна, кандидат технических наук
Ячинова Светлана Николаевна, кандидат педагогических наук

Международный редакционный совет:

Айрян Заруи Геворковна, кандидат филологических наук, доцент (Армения)
Арошидзе Паата Леонидович, доктор экономических наук, ассоциированный профессор (Грузия)
Атаев Загир Вагитович, кандидат географических наук, профессор (Россия)
Ахмеденов Кажмурат Максutowич, кандидат географических наук, ассоциированный профессор (Казахстан)
Бидова Бэла Бертовна, доктор юридических наук, доцент (Россия)
Борисов Вячеслав Викторович, доктор педагогических наук, профессор (Украина)
Буриев Хасан Чутбаевич, доктор биологических наук, профессор (Узбекистан)
Велковска Гена Цветкова, доктор экономических наук, доцент (Болгария)
Гайич Тамара, доктор экономических наук (Сербия)
Данатаров Агахан, кандидат технических наук (Туркменистан)
Данилов Александр Максимович, доктор технических наук, профессор (Россия)
Демидов Алексей Александрович, доктор медицинских наук, профессор (Россия)
Досманбетов Динар Бакбергенович, доктор философии (PhD), проректор по развитию и экономическим вопросам (Казахстан)
Ешиев Абдыракман Молдоалиевич, доктор медицинских наук, доцент, зав. отделением (Кыргызстан)
Жолдошев Сапарбай Тезекбаевич, доктор медицинских наук, профессор (Кыргызстан)
Игисинов Нурбек Сагинбекович, доктор медицинских наук, профессор (Казахстан)
Кадыров Кутлуг-Бек Бекмурадович, доктор педагогических наук, и.о. профессора, декан (Узбекистан)
Каленский Александр Васильевич, доктор физико-математических наук, профессор (Россия)
Козырева Ольга Анатольевна, кандидат педагогических наук, доцент (Россия)
Колпак Евгений Петрович, доктор физико-математических наук, профессор (Россия)
Кошербаева Айгерим Нуралиевна, доктор педагогических наук, профессор (Казахстан)
Курпаяниди Константин Иванович, доктор философии (PhD) по экономическим наукам (Узбекистан)
Куташов Вячеслав Анатольевич, доктор медицинских наук, профессор (Россия)
Кыят Эмине Лейла, доктор экономических наук (Турция)
Лю Цзюань, доктор филологических наук, профессор (Китай)
Малес Людмила Владимировна, доктор социологических наук, доцент (Украина)
Нагервадзе Марина Алиевна, доктор биологических наук, профессор (Грузия)
Нурмамедли Фазиль Алигусейн оглы, кандидат геолого-минералогических наук (Азербайджан)
Прокопьев Николай Яковлевич, доктор медицинских наук, профессор (Россия)
Прокофьева Марина Анатольевна, кандидат педагогических наук, доцент (Казахстан)
Рахматуллин Рафаэль Юсупович, доктор философских наук, профессор (Россия)
Ребезов Максим Борисович, доктор сельскохозяйственных наук, профессор (Россия)
Сорока Юлия Георгиевна, доктор социологических наук, доцент (Украина)
Султанова Дилшода Намозовна, доктор архитектурных наук (Узбекистан)
Узаков Гулом Норбоевич, доктор технических наук, доцент (Узбекистан)
Федорова Мария Сергеевна, кандидат архитектуры (Россия)
Хоналиев Назарали Хоналиевич, доктор экономических наук, старший научный сотрудник (Таджикистан)
Хоссейни Амир, доктор филологических наук (Иран)
Шарипов Аскар Калиевич, доктор экономических наук, доцент (Казахстан)
Шуклина Зинаида Николаевна, доктор экономических наук (Россия)

На обложке изображен Тадао Андо (1941), японский архитектор.

Тадао Андо появился на свет в Осаке 13 сентября 1941 года, на пару минут раньше своего брата-близнеца. Когда мальчишкам исполнилось два года, Андо забрала на воспитание прабабушка. От нее он унаследовал любовь к искусству и трепетное отношение к природе. А еще, прежде чем окончательно посвятить жизнь архитектуре, Андо на протяжении нескольких лет занимался боксом.

Андо — архитектор-самоучка. Искусство проектирования он постигал на практике, изучал по книгам и самостоятельно, посещая памятники в Киото и Наге. Собственное бюро Tadao Ando Architects and Associates Андо основал в 1968 году в возрасте 28 лет, за короткий срок освоив то, на что у студентов архитектурных вузов уходят годы. Сегодня за его плечами более 300 проектов — частные резиденции, музеи, храмы и реконструкции — для Фонда Франсуа Пино в Венеции и Париже. В 1995 году Андо получил Притцкеровскую премию, самую престижную награду в мире архитектуры. Призовые деньги он передал жертвам землетрясения 1995 года, произошедшего в Кобэ. И да, его дома, построенные в этом городе, выдержали подземные толчки и не пострадали.

Часовня Света в Ибараки, небольшом пригороде Осаки, пожалуй, самая известная работа Андо. Крест в центре бетонного строения — единственный источник света, символ надежды в христианской религии. Это соединение традиций японской архитектуры и одновременная отсылка к архитектуре римского Пантеона, который Андо посетил во время своей поездки в Рим в юности.

Дом Адзума в Сумиёси — одна из ранних работ архитектора, которая вывела его на мировой рынок. Решение построить бетонный блок в окружении традиционных деревянных домов Нагая было достаточно смелым. На фасаде здания нет ни одного окна, оно полностью закрыто со стороны улицы и обращено исключительно во внутренний двор, благодаря чему пространства наполняются светом и воздухом — так Андо подчеркивает ощущение безопасности и отсоединенности от внешнего хаотического мира. Дом Кошино архитектор спроектировал как продолжение природного пейзажа: часть здания уходит под землю, чтобы ничто не нарушало естественный ландшафт. Вписанный в склон холма дом состоит из двух бетонных блоков, а из больших окон открываются живописные виды на сад.

Окулус Пантеона возвращается еще в одной работе Андо 2017 года — молитвенном доме на холме Будды на кладбище в Саппоро. Подземный туннель приводит

в центральный зал, где статую Будды окружает нимб голубого неба, контрастирующий с холодным и грубым бетоном. Здесь Андо вновь демонстрирует виртуозную работу со светом, пространством, перспективой. Последовательность в использовании материалов, поэзия света, сценически выстроенное повествование — все это неотъемлемые части его стиля.

Церковь на воде стала олицетворением еще одной природной стихии. Она представляет собой бетонный параллелепипед, одна из граней которого полностью выполнена из стекла. Обращенное к гладкой поверхности водного бассейна здание будто парит в невесомости, а крест, окруженный водой, кажется одновременно близким и недосягаемым, будто отражая в этом всю суть религии как таковой.

Андо проектирует как режиссер, ориентируясь на новый опыт от пребывания в пространстве, — с каждой точки открывается новая захватывающая перспектива, в его домах и общественных пространствах можно часами наблюдать за тем, как меняется свет и тени ложатся на стены в сложной сценографии. Манера Андо получила название «критический рационализм»: несмотря на однообразие материала, работы архитектора, в которых японская философия тесно сплетена с модернизмом, обращены к культуре и традициям места. Исследуя региональное, Андо воплощает его «душу», и даже бетон, созданный из локальных компонентов, будет каждый раз немного другим. Архитектора завораживает сама текстура и возможности работы с материалом, так что неудивительно, что свой метод работы с бетоном Андо совершенствовал много лет.

В том, что касается интеграции в природу, архитектор, кажется, достиг совершенства. В итоге, кроме вездесущего бетона, его средства — свет и пустота. Он управляет стихиями, будто они — гибкий и податливый, еще не застывший материал, он придает им форму, направляет, заставляет работать вместе с архитектурой.

Дома Андо никогда не были ориентированы на удобство и комфорт: своих состоятельных клиентов он заставляет жить в критических условиях — в холоде и влажности. Когда владелец дома, в котором не предусмотрено отопление, пожаловался на холод, Андо посоветовал ему надеть свитер, а если станет холоднее — еще один. В некоторых помещениях во время дождей вода течет по стенам — все, как задумал архитектор.

*Информацию собрала ответственный редактор
Екатерина Осянина*

СОДЕРЖАНИЕ

ХИМИЯ

Динь Ван Так (Dinh Van Tac)

Синтез композитов NiO/ZnO с использованием экстракта листьев лемонграсса и их применение для фотокаталитической деградации смеси красителей метиленового синего и родамина Б1

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Акиев А. К., Давыдов М. И.

Количественные модели для решения задач управления запасами 6

Белолипецкий В. А.

Методы освещения и теневые карты (Shadow Mapping) в рендеринге реального времени.....10

Белолипецкий В. А.

Сравнение алгоритмов трассировки лучей и растеризации в рендеринге реального времени.....11

Белолипецкий В. А.

Постпроцессинговые эффекты в рендеринге реального времени: bloom, tone mapping и antialiasing13

Белолипецкий В. А.

Физически корректное освещение в реальном времени: BRDF-модели и материалы15

Белолипецкий В. А.

Системы частиц в реальном времени: GPU-симуляция и рендеринг визуальных эффектов17

Воропаев Н. А.

Цифровизация учетно-регистрационной системы в Донецкой Народной Республике: основные направления и первые результаты ...19

Гафурова Е. В.

Разработка мобильного приложения с ИИ-помощником для ухода за комнатными растениями22

Гладких В. В.

Применение технологии ассоциативного анализа к данным биржевых рынков с использованием BI-платформ24

Квилюнас И. М.

Введение в технику Screen Space Ambient Occlusion27

Квилюнас И. М.

Основы рендеринга неба: Skybox и процедурная генерация атмосферы29

Квилюнас И. М.

Техника Parallax Mapping для создания рельефных поверхностей30

Квилюнас И. М.

Методы компрессии текстур для мобильных игр: обзор и сравнение32

Ковалев Е. Ю.

Цифровая герменевтика. Применение технологий искусственного интеллекта для анализа исторических правовых текстов33

Назаров В. И.

Управление безопасностью зависимостей клиентских веб-приложений: анализ риска цепочки поставок прм на примере инцидента с библиотекой axios38

Танатаров А. Р.

Система поддержки управления развитием группы разработки ПО на основе имитационного моделирования41

Чесноков Ю. В.

Угрозы информационной безопасности переходного состояния при миграции автоматизированных банковских систем на отечественные системы управления базами данных.....43

Шамжуров И. А.

Сравнительный анализ методов верификации программного обеспечения в распределенных микросервисных системах ...47

Шамжуров И. А.

Проектирование концептуальной архитектуры и сценариев взаимодействия в системах интеграционного тестирования49

Шамжуров И. А.

Обоснование и формирование
технологического стека автоматизации
интеграционного тестирования
распределенных приложений
на платформе.NET 51

Шамжуров И. А.

Объектно-ориентированное проектирование
и реализация структуры классов для
автоматизированного интеграционного
тестирования.....53

Шамжуров И. А.

Проектирование и автоматизированная
верификация структуры данных
при интеграционном тестировании
распределенных систем55

БИОЛОГИЯ**Журавлева К. Е.**

Особенности правильного питания
в подростковом возрасте57

Кашуро Л. А.

Окислительно-восстановительная система
«медь — нитрат серебра» как физико-
химическая макромодел селективной
импрегнации нейронов по методу Гольджи.....58

МЕДИЦИНА**Пучко С. К., Койшебаева К. Б., Арыкбаева М. К.,
Елгазы С. Д.**

Сравнительный анализ исследования
ликвора у детей и подростков
в Многопрофильной городской детской
больнице № 3 Астаны за 2023- 2024 годы64

ХИМИЯ

Синтез композитов NiO/ZnO с использованием экстракта листьев лемонграсса и их применение для фотокаталитической деградации смеси красителей метиленового синего и родамина Б

Динь Ван Так (Dinh Van Tac), кандидат химических наук, преподаватель
Университет Дананга (The University of Danang — University of Science and Education) (Da Nang, Viet Nam)

Введение

Накопление токсичных гетероциклических органических соединений, в частности синтетических красителей метиленового синего и родамина Б, в результате промышленной деятельности представляет собой серьезную проблему для технологий очистки сточных вод из-за стабильности их ароматической структуры и высокой устойчивости к биodeградации. Среди методов перспективного окисления гетерогенный фотокатализ с использованием широкозонных полупроводников, таких как оксид цинка, привлекает значительное внимание благодаря своей высокой активности и химической стабильности. Однако квантовая эффективность ZnO ограничена большой шириной запрещенной зоны ($E_g = 3.3$ eV), что обуславливает поглощение только в ультрафиолетовой области, а высокая скорость рекомбинации фотогенерированных пар электрон — дырка (e^-/h^+) существенно снижает его реальную фотокаталитическую эффективность [1], [2]. Для преодоления указанных недостатков эффективным подходом является формирование p-n гетероперехода путем модифицирования ZnO полупроводником p-типа с более узкой запрещенной зоной, таким как оксид никеля (NiO, $E_g = 3.6$ eV). Разность энергетических уровней краев зон NiO и ZnO индуцирует внутреннее электрическое поле на интерфейсе, что способствует пространственному разделению зарядов и расширению области оптического поглощения композиционного материала в видимый диапазон света [3], [4].

Тенденция использования растительных экстрактов, выступающих в роли реагентов и стабилизаторов поверхности, привлекает все большее внимание благодаря их экологической безопасности и экономической эффективности. Экстракт листьев лемонграсса содержит высокую концентрацию биологически активных соединений, таких как цитраль, полифенолы и флавоноиды, которые способны координировать и направлять формирование наноструктурированных оксидов металлов в мягких условиях [5], [6].

Несмотря на наличие работ по «зеленому» синтезу индивидуальных оксидов, систематические исследования, направленные на контроль процессов формирования гетероструктур NiO/ZnO с использованием биоагентов из листьев лемонграсса для направленной фотодеструкции органических красителей, остаются неполными.

Целью настоящего исследования является синтез композиционных материалов NiO/ZnO с использованием экстракта листьев лемонграсса и их применение в качестве фотокатализаторов для деградации смеси красителей MB и RhB под действием солнечного излучения.

Экспериментальная часть

Экстракт получили при нагревании 25 г листьев лемонграсса в 200 мл дистиллированной воды при 100 °C в течение 30 мин. Добавили 10 мл экстракта в 50 мл раствора, содержащего 0,025 M $Ni(NO_3)_2$ и 0,125 M $Zn(NO_3)_2$. Перемешивали смесь магнитной мешалкой при 70 °C в течение 30 мин. Затем смесь медленно выпаривали на электрической плите при 100 °C. Твердое вещество после выпаривания нагревали при различных температурах, начиная 500 °C в течение 45 мин. Полученный материал обозначен как NiO/ZnO.

Структуры, морфология, химический состав NiO/ZnO были исследованы методами XRD, SEM, EDX.

Для оценки фотокаталитической деградации красителей добавили 10 мг адсорбента (NiO, ZnO, NiO/ZnO) в 50 мл раствора 10 мг/л MB и 5 мг/л RhB. Смесь тщательно перемешивали и помещали на магнитную мешалку, где её выдерживали в темных условиях в течение 90 мин для установления адсорбционно-десорбционного равновесия. После установления адсорбционно-десорбционного равновесия суспензию экспонировали под прямым солнечным излучением в течение 120

мин (с 10:00 до 12:00). Через каждые 30 мин отсасывали примерно 3 мл раствора и измерили оставшуюся концентрацию красителей в растворе. После измерения исследуемый раствор незамедлительно возвращали в реакционный сосуд.

Обсуждение результатов

Рентгеновская дифрактограмма образца NiO/ZnO представлена на рис. 1. Из рис. 1 видно, что на дифрактограмме образца появляются характерные пики для ZnO в положениях $31,8^\circ$; $34,4^\circ$; $36,3^\circ$; $47,5^\circ$; $56,6^\circ$; $62,9^\circ$; $66,4^\circ$; $68,0^\circ$ и $69,1^\circ$, соответствующие граням кристалла (100), (002), (101), (102), (110), (103), (200), (112) и (201), в соответствии с базой данных JCPDS 36-1451 [7]. На дифрактограмме наблюдается только один характерный пик NiO при $2\theta = 43,2^\circ$, соответствующий отражению от плоскости (200), в соответствии с базой данных JCPDS 47-1049. Небольшой пик при $2\theta = 62,8^\circ$ также может быть отнесён к плоскости (220) оксида никеля, однако он практически перекрывается интенсивным пиком (103) ZnO, расположенным при $2\theta = 62,9^\circ$. Полученные результаты подтверждают успешное формирование гетероструктурного материала NiO/ZnO, в котором частицы NiO диспергированы на кристаллической матрице ZnO.

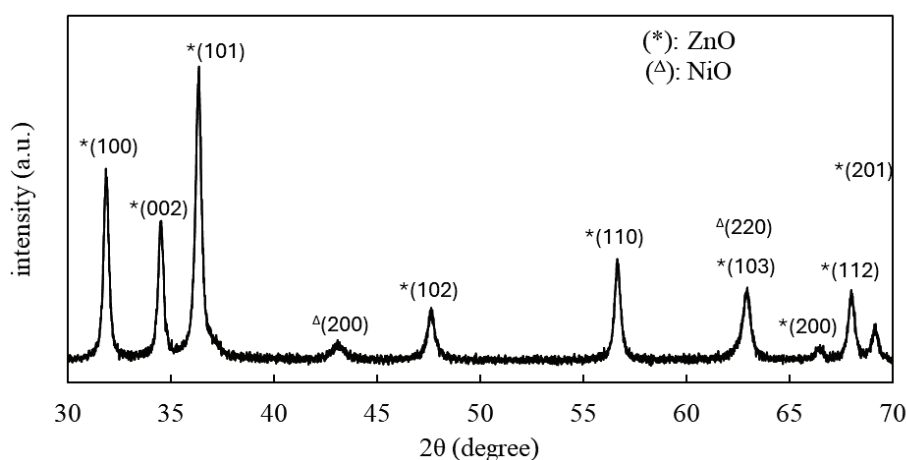


Рис. 1. Дифрактограммы образцов NiO/ZnO

EDX-спектр образца NiO/ZnO, представленный на рис. 2 а, показывает наличие элементов Zn, Ni и O. Особого внимания заслуживает отсутствие в образце значительного содержания углерода. Данный результат служит важным экспериментальным подтверждением эффективности процесса термической обработки, применяемого на стадии синтеза материала.

SEM-изображения показывают, что материал NiO/ZnO состоит из мелких частиц, относительно равномерно распределённых и агрегированных в пористые агломераты. Поверхность материала характеризуется развитым рельефом и наличием многочисленных межчастичных пустот, что создаёт благоприятные условия для адсорбции и увеличения количества активных центров на поверхности. Наблюдаемая агрегация частиц может быть обусловлена высокой поверхностной энергией наночастиц в процессе термической обработки.

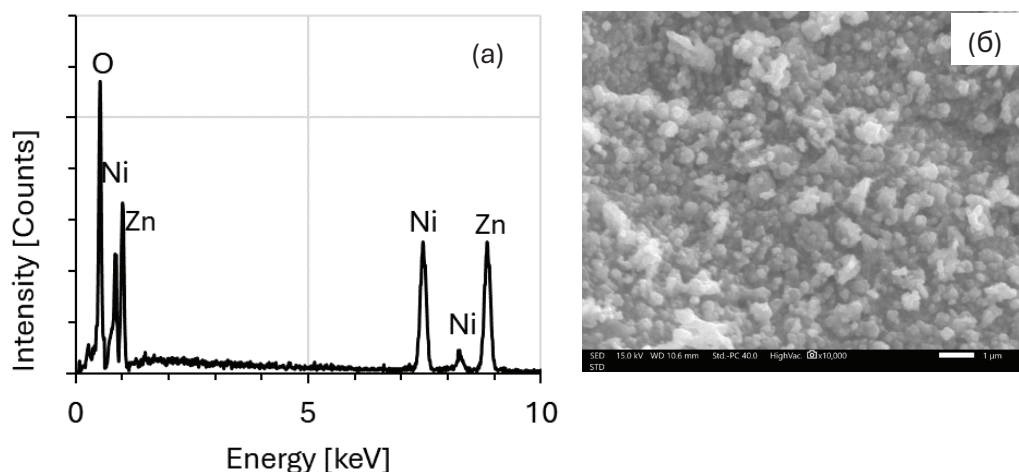


Рис. 2. EDX-спектр (а) и изображения SEM (б) материала NiO/ZnO

Для оценки способности синтезированного материала NiO/ZnO разлагать красители было проведено три параллельных эксперимента в смешанном растворе MB и RhB (рис. 3). Реакционные системы оставляли в темноте для оценки адсорбционной способности, а затем освещали солнечным светом для оценки фотокаталитической активности материала. Результаты эксперимента показали, что в отсутствие катализатора концентрации RhB и MB почти не изменялись со временем при выдерживании в тёмных условиях.

Все три материала (NiO, ZnO и NiO/ZnO) проявляли способность к адсорбции MB и RhB. Однако эффективность адсорбции исследуемых красителей оказалась сравнительно низкой и находилась в диапазоне от 2,05 % (адсорбция RhB на NiO) до 8,18 % (адсорбция MB на NiO/ZnO). Адсорбционное равновесие достигалось через 60 мин.

После экспонирования реакционных смесей под солнечным светом в период с 10:00 до 12:00 концентрации RhB и MB существенно снижались со временем. Наиболее быстрое уменьшение концентрации красителей наблюдалось в присутствии материала NiO/ZnO. После 120 мин облучения солнечным светом эффективность фотодегradации MB и RhB в исследуемых системах представлена в табл. 1.

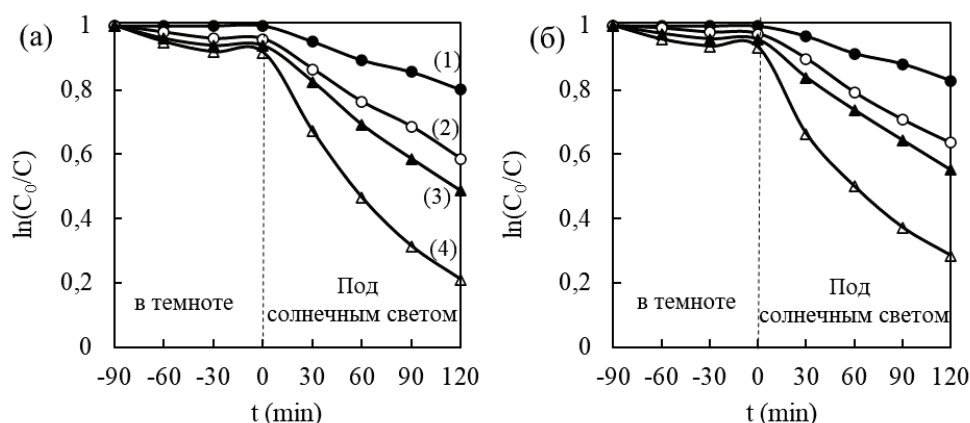


Рис. 3. Зависимость концентрации MB (а) и RhB (б) от времени в отсутствие катализатора (1); и в присутствии NiO (2), ZnO (3) и NiO/ZnO (4)

Таблица 1. Эффективность фотодегradации RhB и MB после 120 мин облучения солнечным светом

Материал	RhB	MB
Без катализатора	17,18 %	19,82 %
NiO	34,82 %	38,95 %
ZnO	42,39 %	48,12 %
NiO/ZnO	69,44 %	77,05 %

График зависимости $\ln(C_0/C)$ (где C_0 — концентрация MB и RhB после достижения адсорбционного равновесия, C — их концентрация в момент времени t) от времени облучения солнечным светом показан на рис. 4.

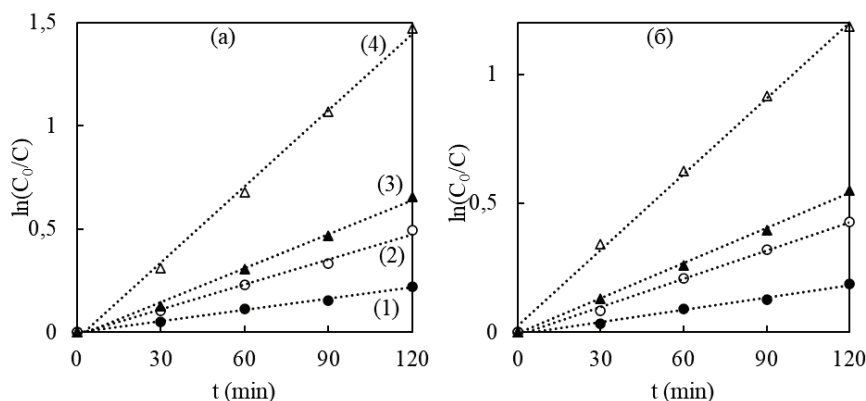


Рис. 4. Зависимость $\ln(C_0/C)$ MB (а) и RhB (б) от времени облучения солнечным светом в отсутствие катализатора (1); и в присутствии NiO (2), ZnO (3) и NiO/ZnO (4)

Зависимости $\ln(C_0/C)$ от времени облучения солнечным светом имеют линейный характер. Это указывает на то, что фотокаталитическая деградация красителей RhB и MB протекает в соответствии с кинетической моделью реакции первого порядка. Константы скорости реакции (k) были определены по тангенсу угла наклона полученных прямых линейной регрессии. Диаграмма, иллюстрирующая константы скорости фотодегradации красителей в отсутствие катализатора, а также в присутствии индивидуальных оксидов NiO, ZnO и композита NiO/ZnO, представлена на рис. 5.

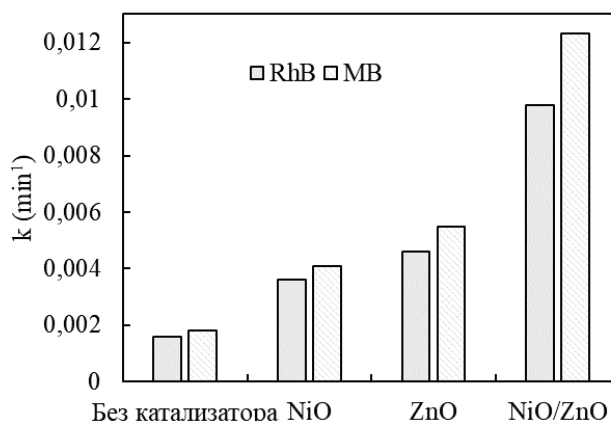


Рис. 5. Константы скорости фотодегradации красителей под солнечным светом без катализатора и в присутствии образцов NiO, ZnO и NiO/ZnO

Присутствие NiO, ZnO и NiO/ZnO, синтезированных методом зелёной химии из экстракта листьев лемонграсса, значительно ускоряет фотодегradацию MB и RhB по сравнению с некаталитическим процессом. Константа скорости без катализатора составляет $0,0016 \text{ мин}^{-1}$ (RhB) и $0,0018 \text{ мин}^{-1}$ (MB). При использовании NiO значения k увеличиваются до $0,0036$ и $0,0041 \text{ мин}^{-1}$ (в $\sim 2,26$ раза), а для ZnO — до $0,0046$ и $0,0055 \text{ мин}^{-1}$ (в ~ 3 раза). Наивысшую активность проявляет композит NiO/ZnO: константа скорости достигает $0,0098 \text{ мин}^{-1}$ (RhB) и $0,0123 \text{ мин}^{-1}$ (MB), что выше холостого опыта в 6,13 и 6,83 раза соответственно.

Высокая эффективность NiO/ZnO обусловлена генерацией электрон-дырочных пар ($e_{\text{CB}}^-/h_{\text{VB}}^+$) под действием света. Благодаря низкому сопротивлению гетероперехода, фотоиндуцированные электроны (e^-) переходят из зоны проводимости ZnO на NiO, а дырки (h^+) концентрируются в валентной зоне ZnO. Формирующаяся область обеднения под действием внутреннего электрического поля эффективно разделяет заряды. Электроны восстанавливают кислород до супероксид-радикалов ($\cdot\text{O}_2^-$), а дырки генерируют гидроксильные радикалы ($\cdot\text{OH}$) и напрямую окисляют красители RhB и MB до безвредных продуктов. Синергетический эффект NiO/ZnO обеспечивает создание активных каталитических центров [8].

Заключение

Материалы NiO/ZnO были успешно получены методом зелёной химии с использованием экстракта листьев лемонграсса и охарактеризованы с помощью различных физико-химических методов. Синтезированные материалы обладают высокой фотокаталитической активностью в отношении красителей MB и RhB при их деградации под действием естественного солнечного света. Наибольшую эффективность продемонстрировал композит NiO/ZnO, для которого константы скорости реакции достигли значений — $0,0098 \text{ мин}^{-1}$ для RhB и $0,0123 \text{ мин}^{-1}$ для MB, что обусловлено синергетическим эффектом разделения зарядов на гетеропереходе. Предложенный подход является перспективным для создания экологически безопасных и эффективных фотокатализаторов.

Литература:

1. Baig, A. A review of visible-light-active zinc oxide photocatalysts for environmental application / A. Baig, M. Siddique, S. Panchal. — Текст: непосредственный // Catalysts. — 2025. — № 15(100). — С. 1–26.
2. Zeid, S. A. Advancements in ZnO-Based Photocatalysts for Water Treatment: A Comprehensive Review / S. A. Zeid, Y. Leprince-Wang. — Текст: непосредственный // Crystals. — 2024. — № 14(611). — С. 1–47.
3. Hassanpour, M. Rapid and eco-friendly synthesis of NiO/ZnO nanocomposite and its application in decolorization of dye / M. Hassanpour, H. Safardoust-Hojaghan, M. Salavati-Niasari. — Текст: непосредственный // Journal of Materials Science: Materials in Electronics. — 2017. — № 28(7). — С. 1–8.

4. ZnO/NiO nanocomposite with enhanced photocatalytic H₂ production / M. Hashim, M. Usman, S. Ahmad [и др.]. — Текст: непосредственный // International Journal of Photoenergy. — 2024. — № 2024. — С. 1–11.
5. Synergistic green synthesis of ZnO nanoparticles using *Cymbopogon citratus* and *Nelumbo nucifera* for enhanced antibacterial activity / R. Prakash, P. Saravanan, P. Kumar [и др.]. — Текст: непосредственный // Sustainable Chemistry One World. — 2025. — № 8.
6. Investigations on biosynthesized nickel oxide nanoparticles using *Cymbopogon citratus* leaf extract for antibacterial activity / S. Thirbika, H. Karthi, R. Premila, M. R. Prabhu. — Текст: непосредственный // Materials Today: Proceedings. — 2022. — № 68(3). — С. 276–281.
7. Biomimetic green synthesis of ZnO nanoparticles using *Cheilocostus speciosus* and *Gardenia gummiifera* with comprehensive characterization and bioactivity assessment / R. Maind, S. Halder, A. R. Bhat [и др.]. — Текст: непосредственный // Scientific Reports. — 2025. — № 15:44323. — С. 1–22.
8. ZnO/NiO Nanocomposite with Enhanced Photocatalytic H₂ Production / M. Hashim, M. Usman, S. Ahmad [и др.]. — Текст: непосредственный // International Journal of Photoenergy. — 2024. — № 2676368. — С. 1–11.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Количественные модели для решения задач управления запасами

Акиев Авраам Казбекович, студент магистратуры;

Давыдов Максим Ильич, студент магистратуры

Научный руководитель: Толоконников Игорь Геннадьевич, кандидат экономических наук, доцент
Северо-Осетинский государственный университет имени К. Л. Хетагурова (г. Владикавказ)

В статье авторы проводят анализ ключевых математических моделей, предназначенных для расчёта оптимального объёма заказа при условии стабильного спроса.

Ключевые слова: заказ поставщику, математическая модель, товарный запас.

Рассмотрим классическую модель определения экономичного размера заказа (EOQ), которая позволяет минимизировать суммарные издержки на хранение и пополнение запасов и модель EOQ с учётом разрывов цен. Этот подход актуален для ситуаций, когда закупочная стоимость единицы товара зависит от объёма партии, что требует корректировки расчётов для поиска наиболее выгодного варианта.

1. Традиционная модель EOQ (экономичного размера заказа)

Допустим, что спрос на продукцию постоянен во времени, а пополнение склада происходит мгновенно и без задержек. Для дальнейшего описания введём следующие обозначения:

x — Объем заказа (количество единиц продукции),

b — Интенсивность спроса,

t — Продолжительность цикла заказ.

Динамика изменения уровня складских запасов описывается ступенчатой функцией, график которой представлен на рис. 1. Процесс выглядит следующим образом: как только запас достигает нулевой отметки, мгновенно размещается и отгружается заказ объёмом x , где длительность цикла оборота запаса $t = x/b$ [1].

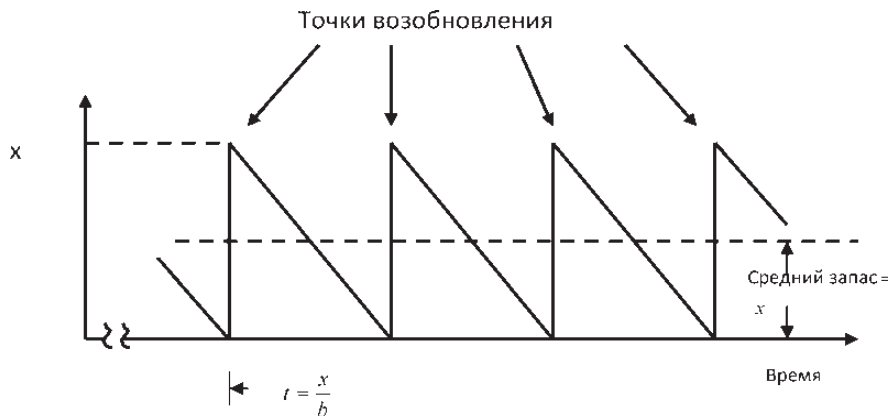


Рис. 1. Характер изменения объёма товаров на складе при использовании традиционной модели

Для определения оптимального объёма заказа x необходимо найти такое значение, при котором целевая функция затрат $y(x)$ достигает своего минимума.

В рамках математической модели мы рассматриваем переменную x как непрерывную величину. Это допущение позволяет применить аппарат дифференциального исчисления. Необходимым условием экстремума (в данном случае — минимума) функции является равенство её первой производной нулю.

Средний уровень запаса определяется соотношением: средний уровень запаса = $x/2$ единиц.

Для построения функции затрат требуется два стоимостных параметра.

c_0 — затраты на оформление, связанные с размещением заказа,

h — затраты на хранение (затраты на единицу складированной продукции в единицу времени).

Суммарные затраты в единицу времени можно представить как функцию $y(x)$ в следующем виде:

$$y(x) = \frac{c_0 b}{x} + h \left(\frac{x}{2} \right) \quad (1)$$

Оптимальное значение объема заказа x определяется путем минимизации по x функции $y(x)$. Предполагая, что x является непрерывной переменной, получаем необходимое условие минимума (в виде уравнения), из которого можно найти оптимальное значение x .

Решая полученное уравнение относительно x , мы находим точку, соответствующую оптимальному размеру заказа.

$$\frac{dy(x)}{dx} = -\frac{c_0 b}{x^2} + \frac{h}{2} \quad (2)$$

Условие равенства производной нулю является необходимым и достаточным для достижения минимума, поскольку функция затрат $y(x)$ обладает свойством выпуклости. Решение этого уравнения позволяет определить экономичный размер заказа хопт.

$$x_{\text{опт}} = \sqrt{\frac{2c_0 b}{h}} \quad (3)$$

В такой ситуации заказ необходимо размещать заранее, не дожидаясь полного обнуления остатков. Критическим моментом для формирования новой заявки становится достижение запасом определённого порогового уровня. Этот порог, или точка возобновления заказа, рассчитывается исходя из ожидаемого расхода за время поставки и составляет Lb единиц.

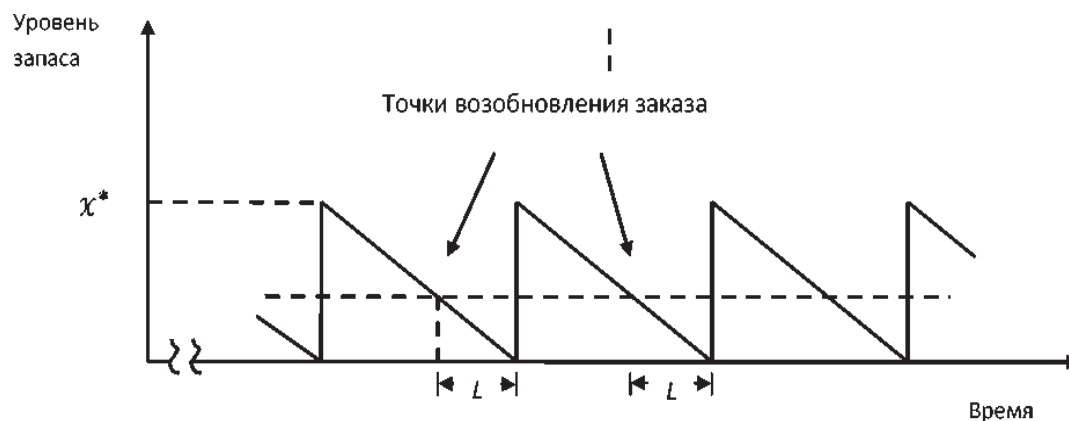


Рис. 2. Моменты возобновления закупки, предусмотренные в традиционной модели управления запасами

Если срок поставки превышает длительность цикла заказа, для расчётов используется эффективный срок выполнения заказа Le , который вычисляется по следующей формуле [2].

$$Le = L - n \cdot \text{тонм} \quad (4)$$

В этой формуле n — обозначает целую часть отношения $\frac{L}{t_{\text{опт}}}$ (наибольшее целое число, не превосходящее результат деления). Такой подход обоснован тем, что по прошествии n полных циклов длиной тонм динамика управления запасами становится эквивалентной ситуации с интервалом между заказом и поставкой, равным Le [3].

2. Расчёт экономичного объёма закупки в условиях ступенчатого ценообразования

Представленная в этом разделе модель управления запасами отличается тем, что производимый товар может быть куплен со скидкой, когда объем x превышает q и стоимость единицы продукции s определяется по формуле:

$$c = \begin{cases} c_1, & \text{если } x \leq q \\ c_2, & \text{если } x > q \end{cases} \quad (5)$$

где $c_1 > c_2$. Таким образом, реальные затраты по закупке равны

$$\begin{cases} \frac{c_1 x}{t} = \frac{c_1}{\left(\frac{x}{b}\right)} = bc_1, & x \leq q \\ \frac{c_2 x}{t} = \frac{c_2}{\left(\frac{x}{b}\right)} = bc_2, & x > q \end{cases} \quad (6)$$

Общие затраты в пересчете на единицу времени рассчитаем по формуле.

$$y(x) = \begin{cases} y_1(x) = bc_1 + \frac{c_0 b}{x} + \frac{h}{2}x, & x \leq q \\ y_2(x) = bc_2 + \frac{c_0 b}{x} + \frac{h}{2}x, & x > q \end{cases} \quad (7)$$

Как показано на рис. 3, функции y_1 и y_2 достигают своего наименьшего значения в одной и той же точке. Это является прямым следствием того, что их значения различаются только на постоянную величину, что не влияет на положение точки минимума.

$$x_m = \sqrt{\frac{2c_0 b}{h}}, \quad (8)$$

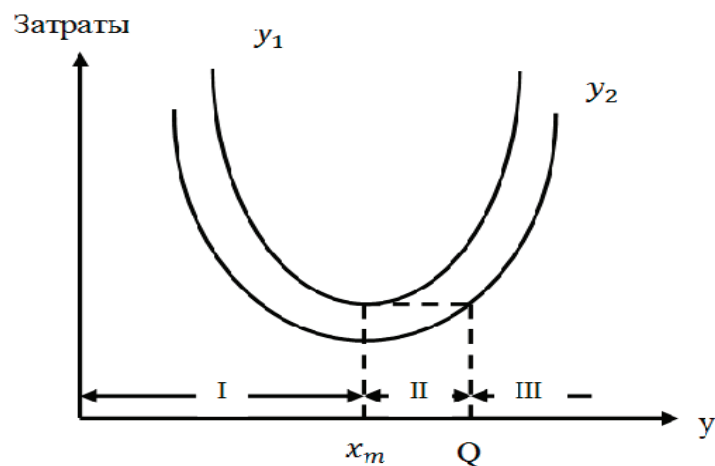


Рис. 3. Кривые, отображающие поведение функции расходов

Функция общих затрат $y(x)$, ведёт себя следующим образом: до определённого момента её график идентичен графику функции $y_1(x)$, но при достижении точки $x = q$, (где меняется цена) он переходит в график функции $y_2(x)$. На рис. 3 проиллюстрировано, что для нахождения оптимального объёма заказа *хопт* необходимо проанализировать, в какую из трёх зон (I, II или III) попадает точка разрыва цены q . Эти зоны — это интервалы $[0, x_m)$, $[x_m, q)$ и $[q, \infty]$.

Величина $q > x_m$ определяется из уравнения:

$$y_2(q) = y_1(x_m) \text{ или } c_2 b + \frac{c_0 b}{q} + \frac{hq}{2} = y_1(x_m) \quad (9)$$

Далее решаем квадратное уравнение относительно Q :

$$q^2 + \left(\frac{2(c_2 b - y_1(x_m))}{h} \right) q + \frac{2c_0 b}{h} = 0 \quad (10)$$

На рисунке 4 показано, как определяется оптимальное значение *хопт*.

$$x_{\text{опт}} = \begin{cases} x_m, & \text{если } q \text{ находится в зоне I или III} \\ q, & \text{если } q \text{ находится в зоне II} \end{cases} \quad (11)$$

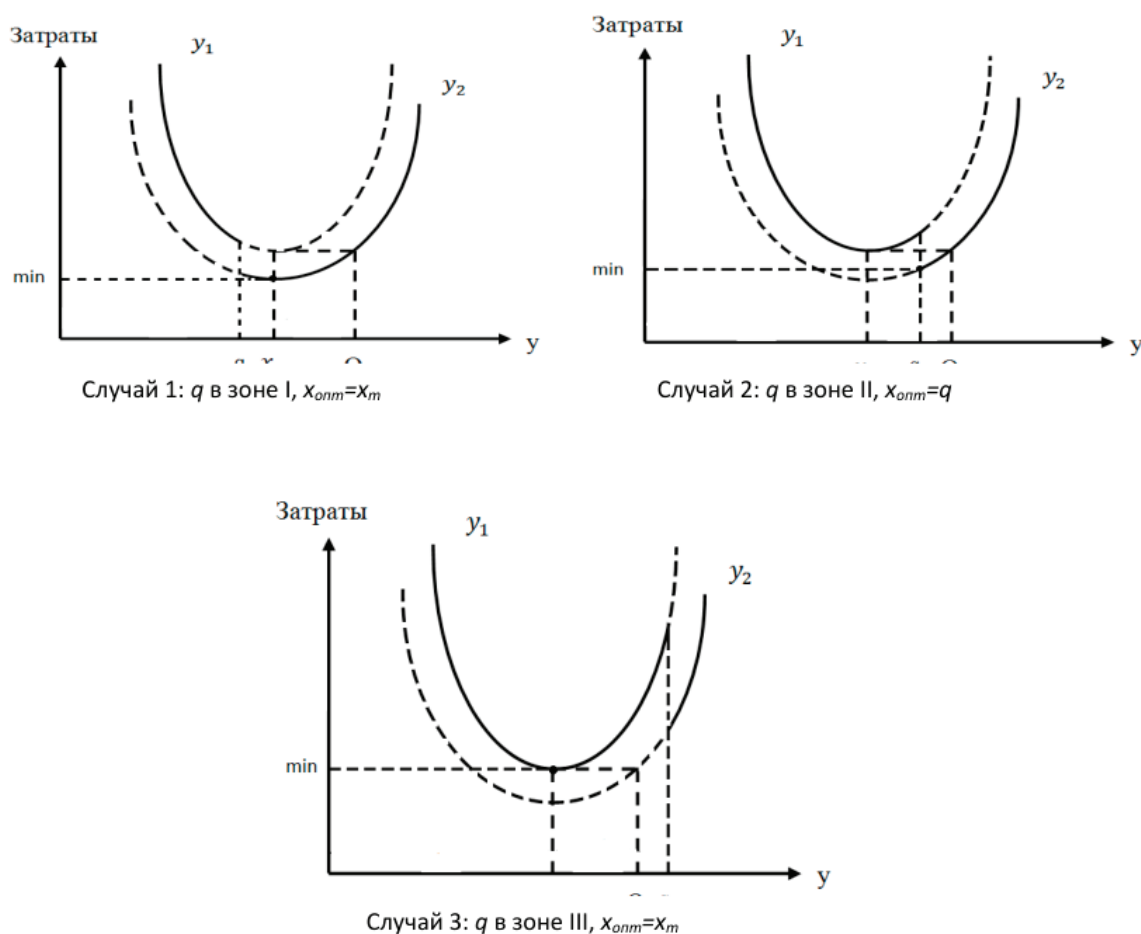


Рис. 4. Три сценария нахождения наилучшего решения

Алгоритм определения *хопт* можно представить в следующем виде.

Этап 1. Вычисляем $x_m = \sqrt{\frac{2c_0b}{h}}$. Если q попадает в зону I, полагаем $хопт=x_m$, в противном случае к этапу 2.

Этап 2. Находим q из уравнения $q^2 + \left(\frac{2(c_2b - y_1(x_m))}{h}\right)q + \frac{2c_0b}{h} = 0$ и определяем зоны II и III. Если q находится в зоне II, полагаем $хопт=q$. Иначе q находится в зоне III, тогда $хопт=x_m$ [4].

Литература:

1. Гусева, Е. Н. Экономика-математическое моделирование: учебное пособие / Е. Н. Гусева. — 4-е изд., стер. — Москва: ФЛИНТА, 2021. — 216 с. — Серия: Информационные технологии. — Доступ с устройства, имеющего доступ к сети Интернет. — URL: <https://biblioclub.ru/index.php?page=book&id=83540>.
2. Джафаров, К. А. Методы оптимальных решений: задачи управления запасами, очередью и конфликтами: учебное пособие / К. А. Джафаров, Л. В. Роева; Новосибирский государственный технический университет. — Новосибирск: Новосибирский государственный технический университет, 2018. — 112 с.: ил., табл. — Доступ с устройства, имеющего доступ к сети Интернет. — URL: <https://biblioclub.ru/index.php?page=book&id=574674>.
3. Волкова, В. Н. Теория информационных процессов и систем: учебник и практикум для вузов / В. Н. Волкова. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2023. — 432 с. — (Высшее образование). — ISBN 978-5-534-05621-1. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511112>.

Методы освещения и теневые карты (Shadow Mapping) в рендеринге реального времени

Белолипецкий Виталий Александрович, студент

Камчатский государственный технический университет (г. Петропавловск-Камчатский)

В статье рассмотрены принципы классического *shadow mapping*, основные артефакты (*shadow acne*, *peter-panning*, алиасинг теней) и методы их устранения — PCF, каскадные теневые карты (CSM) и *variance shadow maps* (VSM). Показано, что CSM с фильтрацией PCF являются наиболее распространённым промышленным решением для открытых сцен, а дальнейшее развитие метода связано с гибридизацией растеризации и трассировки лучей.

Ключевые слова: тени, рендеринг, *shadow mapping*, каскадные теневые карты, PCF, *variance shadow maps*.

Освещение является одним из главных факторов, формирующих визуальное восприятие трёхмерной сцены. Без теней объекты выглядят «плоскими» и оторванными от поверхности, пространственные отношения между ними становятся неочевидными [1, с. 45]. Существует несколько подходов к расчёту теней: трассировка лучей (*ray tracing*), теневые объёмы (*shadow volumes*) и теневые карты (*shadow mapping*). Несмотря на развитие аппаратной трассировки лучей, *shadow mapping* остаётся доминирующим методом из-за предсказуемой производительности и простоты реализации на массовом оборудовании [2]. Целью данной работы является рассмотрение принципов *shadow mapping*, анализ типичных артефактов и обзор методов их устранения.

Принцип работы классического *shadow mapping*

Базовая идея *shadow mapping* — двухпроходный рендеринг сцены [3, с. 112]. На первом проходе сцена рендерится из точки зрения источника света: сохраняется только карта глубины (расстояние от источника до ближайшей поверхности). Полученное изображение называется теневой картой (*shadow map*).

На втором проходе сцена рендерится с точки зрения камеры. Для каждого фрагмента вычисляется его положение в системе координат источника света и полученная глубина сравнивается со значением из теневой карты [4]. Если глубина фрагмента больше — он в тени, иначе — освещён. Главное преимущество: универсальность для произвольной геометрии, нативная поддержка в DirectX, Vulkan, OpenGL [5]. Вместе с тем метод порождает ряд характерных артефактов.

Типичные артефакты и способы их устранения

Shadow acne («теневые угри») — полосатый муаровый узор на освещённых поверхностях [3, с. 118]. Причина: из-за дискретизации глубины поверхность может ошибочно затенять саму себя. Решение — смещение глубины (*depth bias* / *slope-scaled bias*), которое сдвигает сравниваемую глубину, устраняя самозатенение [6].

Избыточное смещение порождает **peter-panning** — тень визуально «отрывается» от объекта [7, с. 54]. Для дополнительного контроля применяют *normal offset bias*, сдвигающий точку сэмплирования вдоль нормали поверхности.

Алиасинг краёв теней («ступенчатость» границ) возникает при недостаточном разрешении теневой карты относительно покрываемой области экрана [8]. Для устранения используется ряд методов, описанных далее.

PCF, каскадные теневые карты и VSM

Percentage Closer Filtering (PCF) вместо одного сравнения выполняет несколько выборок из соседних ячеек теневой карты (например, сетка 3×3 или 5×5) и усредняет результаты [9, с. 67]. Это создаёт плавный переход между освещённой и затенённой областью и частично решает проблему алиасинга.

Для больших открытых сцен одна теневая карта не обеспечивает приемлемое качество как для близких, так и для удалённых объектов. Решение — **Cascaded Shadow Maps (CSM)** [10]: пространство видимости камеры делится на несколько диапазонов глубины, для каждого строится отдельная теневая карта. Ближние каскады имеют высокое разрешение на единицу площади, дальние — пониженное. CSM с PCF — стандарт де-факто для направленного освещения в открытых сценах [7, с. 203].

Variance Shadow Maps (VSM) [11] вместо одного значения глубины хранят среднее и дисперсию, что позволяет аппроксимировать долю закрытых сэмплов через неравенство Чебышева. Метод даёт мягкие тени без множества выборок, как в PCF, однако страдает от *light bleeding* — «протекания» света через тонкие объекты [12, с. 88]. В современных движках теневые карты дополняются контактными тенями и SSAO для мелких деталей [13].

Сравнение рассмотренных методов представлено в таблице 1.

Таблица 1. Сравнение методов формирования теней

Метод	Основное преимущество	Основной недостаток
Shadow mapping (базовый)	Универсальность, поддержка GPU	Shadow acne, peter-panning, алиасинг
PCF	Сглаживание границ теней	Рост числа выборок текстуры
CSM	Качество для больших открытых сцен	Сложность настройки каскадов
VSM	Мягкие тени без множества выборок	Light bleeding

Заключение

Shadow mapping остаётся базовым методом формирования теней в рендеринге реального времени благодаря универсальности и предсказуемой нагрузке на GPU. Классическая реализация порождает характерные артефакты, устраняемые с помощью PCF, CSM и VSM. На практике современные движки комбинируют CSM и PCF, дополняя контактными тенями и ambient occlusion. Дальнейшее развитие связано с гибридизацией shadow mapping и аппаратной трассировки лучей.

Литература:

1. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — 1200 с.
2. Williams L. Casting curved shadows on curved surfaces // ACM SIGGRAPH Computer Graphics. — 1978.
3. Eisemann E., Schwarz M., Assarsson U., Wimmer M. Real-Time Shadows. — CRC Press, 2011. — 284 с.
4. Common Techniques to Improve Shadow Depth Maps [Электронный ресурс] // Microsoft Learn. URL: <https://learn.microsoft.com> (дата обращения: 03.11.2025).
5. Vulkan Shadow Mapping Tutorial [Электронный ресурс] // Vulkan Documentation. URL: <https://docs.vulkan.org> (дата обращения: 03.11.2025).
6. Dou H., Yan Y., Kerzner E. et al. Adaptive Depth Bias for Shadow Maps // Journal of Computer Graphics Techniques. — 2014. — Vol. 3, No. 4.
7. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — Разделы Shadow Techniques, CSM.
8. Scherzer D., Wimmer M., Purgathofer W. A survey of real-time hard shadow mapping methods // Computer Graphics Forum. — 2011.
9. Percentage-Closer Soft Shadows [Электронный ресурс] // NVIDIA GameWorks. URL: <https://developer.nvidia.com> (дата обращения: 04.11.2025).
10. Engel W. Cascaded Shadow Maps // ShaderX. — 2007.
11. Donnelly W., Lauritzen A. Variance Shadow Maps // ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. — 2006.
12. Lauritzen A. Summed-Area Variance Shadow Maps // GPU Gems 3. — NVIDIA, 2008.
13. Screen Space Ambient Occlusion and Contact Shadows [Электронный ресурс] // GDC Vault. URL: <https://www.gdcvault.com> (дата обращения: 05.11.2025).

**Сравнение алгоритмов трассировки лучей и растеризации
в рендеринге реального времени**

Белолипецкий Виталий Александрович, студент
Камчатский государственный технический университет (г. Петропавловск-Камчатский)

В статье рассматриваются два фундаментальных подхода к синтезу изображений в реальном времени — растеризация и трассировка лучей (ray tracing). Проанализированы принципы работы каждого метода, их вычислительная сложность и типичные визуальные эффекты, которые они позволяют корректно воспроизводить. Особое внимание уделено гибриднему рендерингу, совмещающему растеризацию основной геометрии с трассировкой лучей для отражений, теней и глобального освещения, а также аппаратному ускорению трассировки лучей на современных графических процессорах.

Ключевые слова: рендеринг, растеризация, трассировка лучей, ray tracing, гибридный рендеринг, глобальное освещение.

Синтез реалистичного изображения трёхмерной сцены за доли секунды — одна из центральных задач компьютерной графики. Исторически эта задача решалась методом растеризации: геометрия сцены проецируется на плоскость экрана, а видимость и освещение вычисляются приближённо, без точного моделирования распространения света [1, с. 23]. Альтернативный подход — трассировка лучей, при которой для каждого пикселя моделируется путь света от камеры до источников освещения через отражения и преломления [2]. Долгое время трассировка лучей считалась слишком ресурсоёмкой для интерактивных приложений, однако появление аппаратного ускорения изменило это соотношение. Целью статьи является сравнение принципов работы обоих методов, их сильных и слабых сторон, а также анализ современных гибридных решений.

Принцип работы растеризации

Растеризация преобразует трёхмерные примитивы (треугольники) в набор пикселей экрана [3, с. 67]. Каждый треугольник проецируется через матрицы вида и проекции, после чего для каждого покрытого пикселя вычисляется цвет с использованием информации о материале, нормалях и источниках света. Видимость объектов определяется буфером глубины (z-buffer), который хранит расстояние до ближайшей поверхности для каждого пикселя [4].

Главное преимущество растеризации — производительность: алгоритм линейно зависит от числа примитивов и пикселей, а не от сложности светового переноса в сцене, что делает его пригодным для GPU с массовым параллелизмом [5]. Однако растеризация плохо приспособлена для эффектов, требующих глобальной информации о сцене — точных отражений, преломлений и мягких теней, поскольку каждый примитив обрабатывается независимо от остальных.

Принцип работы трассировки лучей

Базовый алгоритм трассировки лучей выпускает из камеры луч через каждый пиксель и находит его пересечение с ближайшей поверхностью сцены [6, с. 41]. В точке пересечения вычисляется освещение, а при необходимости запускаются дополнительные лучи — отражённые, преломлённые или теневые, что позволяет физически корректно моделировать многократные отражения света [2]. Поиск пересечений ускоряется иерархическими структурами данных, такими как BVH (Bounding Volume Hierarchy), которые позволяют избежать перебора всех примитивов сцены [7].

Главное преимущество трассировки лучей — физическая корректность: метод естественно воспроизводит зеркальные отражения, мягкие тени и глобальное освещение без дополнительных приёмов [8, с. 112]. Недостаток — высокая вычислительная стоимость, поскольку для каждого пикселя требуется обход сложной структуры данных, а итоговый рендеринг часто требует множества лучей на пиксель для подавления шума.

Гибридный рендеринг и аппаратное ускорение

Современные игровые движки используют **гибридный рендеринг**, в котором основная геометрия и прямое освещение обрабатываются растеризацией, а трассировка лучей применяется выборочно — для отражений, контактных теней и глобального освещения [9]. Это позволяет сохранить высокую частоту кадров, добавляя физически корректные эффекты только там, где они визуальнo наиболее заметны.

Аппаратное ускорение трассировки лучей реализуется через специализированные блоки GPU, выполняющие обход BVH и проверку пересечений луч-треугольник напрямую в кремнии [10]. Это существенно снижает нагрузку на универсальные вычислительные блоки и делает трассировку лучей применимой в реальном времени даже для сложных сцен. Дополнительно используется **денойзинг** на основе нейронных сетей, который восстанавливает чистое изображение из результата трассировки с малым числом лучей на пиксель, заметно снижая требуемую вычислительную мощность [11, с. 88].

Сравнение рассмотренных подходов представлено в таблице 1.

Таблица 1. Сравнение методов рендеринга

Метод	Основное преимущество	Основной недостаток
Растеризация	Высокая производительность	Слабая поддержка глобальных эффектов
Трассировка лучей	Физическая корректность освещения	Высокая вычислительная стоимость
Гибридный рендеринг	Баланс качества и производительности	Сложность реализации конвейера

Заключение

Растеризация и трассировка лучей решают одну задачу — синтез изображения сцены — принципиально разными способами: первая ориентирована на скорость и предсказуемую нагрузку на GPU, вторая — на физическую корректность светового переноса. Современные системы рендеринга реального времени не выбирают один из подходов, а комбинируют их, применяя трассировку лучей только для эффектов, критичных для восприятия глубины и материала. Развитие аппаратного ускорения и нейросетевого денойзинга продолжает снижать порог вхождения трассировки лучей в массовые интерактивные приложения, что делает гибридный рендеринг основным направлением развития отрасли.

Литература:

1. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — 1200 с.
2. Whitted T. An improved illumination model for shaded display // Communications of the ACM. — 1980. — Vol. 23, No. 6.
3. Shirley P., Marschner S. Fundamentals of Computer Graphics. — 4th ed. — CRC Press, 2015.
4. Akeley K., Hanrahan P. The Graphics Pipeline [Электронный ресурс] // Stanford CS248 Course Notes. URL: <https://graphics.stanford.edu> (дата обращения: 10.11.2025).
5. Blinn J. F. Jim Blinn's Corner: A Trip Down the Graphics Pipeline. — Morgan Kaufmann, 1996.
6. Glassner A. An Introduction to Ray Tracing. — Academic Press, 1989.
7. Wald I., Boulos S., Shirley P. Ray Tracing Deformable Scenes Using Dynamic Bounding Volume Hierarchies // ACM Transactions on Graphics. — 2007.
8. Pharr M., Jakob W., Humphreys G. Physically Based Rendering: From Theory to Implementation. — 3rd ed. — Morgan Kaufmann, 2016.
9. NVIDIA. RTX Hybrid Rendering Pipeline [Электронный ресурс] // NVIDIA Developer. URL: <https://developer.nvidia.com> (дата обращения: 11.11.2025).
10. NVIDIA. RT Cores and Hardware-Accelerated Ray Tracing [Электронный ресурс] // NVIDIA Turing Architecture Whitepaper. URL: <https://www.nvidia.com> (дата обращения: 11.11.2025).
11. Chaitanya C. R. A. et al. Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder // ACM SIGGRAPH. — 2017.

Постпроцессинговые эффекты в рендеринге реального времени: bloom, tone mapping и antialiasing

Белолипецкий Виталий Александрович, студент

Камчатский государственный технический университет (г. Петропавловск-Камчатский)

В статье рассматриваются основные постпроцессинговые эффекты, применяемые на финальном этапе конвейера рендеринга реального времени: bloom, tone mapping и сглаживание (antialiasing). Проанализированы принципы работы каждого эффекта, их вычислительная стоимость и роль в общем качестве итогового изображения. Отдельное внимание уделено особенностям рендеринга с расширенным динамическим диапазоном (HDR) и современным методам временного сглаживания, использующим информацию с предыдущих кадров.

Ключевые слова: постпроцессинг, bloom, tone mapping, antialiasing, HDR, рендеринг реального времени.

Изображение, полученное непосредственно из основного конвейера рендеринга, как правило, обладает рядом технических несовершенств: ступенчатыми краями геометрии, неестественно резкими яркостными переходами и узким диапазоном отображаемой яркости [1, с. 142]. Постпроцессинг — этап обработки уже сформированного кадра, на котором эти недостатки устраняются или используются художественно [2]. Поскольку постпроцессинговые эффекты работают с готовым двумерным изображением, а не с геометрией сцены, их вычислительная стоимость предсказуема и не зависит от сложности сцены, что делает их удобным инструментом контроля производительности. Целью данной статьи является рассмотрение трёх ключевых постпроцессинговых эффектов — bloom, tone mapping и antialiasing — и их роли в современном рендеринге реального времени.

Bloom и рендеринг с расширенным динамическим диапазоном

Bloom — эффект свечения вокруг ярких областей изображения, имитирующий рассеивание света в оптической системе глаза или камеры [3, с. 58]. Технически bloom реализуется путём выделения пикселей, яркость которых превышает заданный порог, размытия полученного изображения (обычно гауссовым фильтром) и последующего смешивания результата с исходным кадром [4]. Эффект особенно заметен для источников света, прямые значения яркости которых превышают стандартный диапазон 0–1, что подводит к необходимости рендеринга с расширенным динамическим диапазоном (HDR).

При **HDR-рендеринге** промежуточные вычисления освещения сохраняются в буфере с плавающей точкой, способном представлять яркость значительно выше единицы, в отличие от стандартного 8-битного буфера [5]. Это позволяет физически корректно моделировать яркие источники света и металлические отражения, однако итоговое изображение перед выводом на экран должно быть преобразовано в отображаемый диапазон — для этого используется tone mapping.

Tone mapping

Tone mapping — процесс сжатия широкого динамического диапазона HDR-изображения в диапазон, отображаемый на стандартном мониторе [6, с. 23]. Простейший подход — линейное масштабирование с отсечением (clamping), однако он приводит к потере деталей в ярких областях. Более совершенные операторы, такие как Reinhard tone mapping, сжимают яркость нелинейно, сохраняя при этом контраст в средних тонах и плавно подавляя пересвеченные участки [7].

Современные игровые движки чаще применяют операторы семейства **ACES (Academy Color Encoding System)**, изначально разработанные для киноиндустрии и обеспечивающие предсказуемую и визуально приятную передачу цвета в широком диапазоне условий освещения [8]. Выбор оператора tone mapping напрямую влияет на общее восприятие контраста и цветности сцены, поэтому в современных движках он, как правило, настраивается художником как часть общей цветокоррекции.

Сглаживание (antialiasing)

Растреризация треугольников порождает характерный артефакт — ступенчатые края геометрии (jaggies), возникающие из-за того, что пиксель либо целиком относится к примитиву, либо целиком исключается из него [9, с. 211]. Классическое решение — **MSAA (Multisample Anti-Aliasing)**, при котором для каждого пикселя выполняется несколько выборок видимости геометрии, после чего результаты усредняются [10]. MSAA даёт качественный результат, но пропорционально числу выборок увеличивает нагрузку на память и пропускную способность GPU.

Альтернативный подход — постпроцессинговые методы сглаживания, такие как **FXAA (Fast Approximate Anti-Aliasing)**, которые анализируют уже отрендеренное изображение и размывают резкие контрастные границы, не требуя повторной растреризации геометрии [11]. Современный стандарт — **TAA (Temporal Anti-Aliasing)**, накапливающий информацию с нескольких последовательных кадров со небольшим субпиксельным смещением камеры, что даёт качество, близкое к MSAA с большим числом выборок, при существенно меньшей нагрузке на текущий кадр [12, с. 95]. Недостаток TAA — характерное смазывание (ghosting) при быстром движении объектов, для борьбы с которым применяются методы коррекции на основе векторов движения.

Сравнение рассмотренных эффектов представлено в таблице 1.

Таблица 1. Сравнение постпроцессинговых эффектов

Эффект	Основное преимущество	Основной недостаток
Bloom	Реалистичная передача ярких источников	Требует HDR-конвейера
Tone mapping	Корректное отображение HDR на экране	Влияет на восприятие контраста и цвета
MSAA	Высокое качество сглаживания	Высокая нагрузка на память
TAA	Низкая стоимость на кадр	Смазывание при быстром движении

Заключение

Постпроцессинговые эффекты являются неотъемлемой частью современного конвейера рендеринга реального времени, позволяя устранить технические артефакты растреризации и физически корректно передать широкий диапазон яркости сцены на ограниченных возможностях экрана. Bloom и tone mapping тесно связаны с HDR-рендерингом и фор-

мируют общее восприятие освещения и контраста кадра, а развитие antialiasing от MSAA к временным методам типа ТАА отражает общую тенденцию переноса вычислительной нагрузки с геометрического этапа конвейера на этап обработки готового изображения с использованием информации из предыдущих кадров.

Литература:

1. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — 1200 с.
2. Mitchell J., McTaggart G., Green C. Shading in Valve's Source Engine // ACM SIGGRAPH Courses. — 2006.
3. Kawase M. Frame Buffer Postprocessing Effects in DOUBLE-S. T. E. A. L (Wreckless) // Game Developers Conference Proceedings. — 2003.
4. NVIDIA. HDR Lighting and Bloom [Электронный ресурс] // NVIDIA GPU Gems. URL: <https://developer.nvidia.com> (дата обращения: 14.11.2025).
5. Reinhard E. et al. High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting. — 2nd ed. — Morgan Kaufmann, 2010.
6. Reinhard E., Stark M., Shirley P., Ferwerda J. Photographic Tone Reproduction for Digital Images // ACM Transactions on Graphics. — 2002.
7. Reinhard E. Parameter Estimation for Photographic Tone Reproduction // Journal of Graphics Tools. — 2002.
8. Academy of Motion Picture Arts and Sciences. ACES Documentation [Электронный ресурс]. URL: <https://acescentral.com> (дата обращения: 15.11.2025).
9. Shirley P., Marschner S. Fundamentals of Computer Graphics. — 4th ed. — CRC Press, 2015.
10. Akeley K. Reality Engine Graphics // ACM SIGGRAPH Proceedings. — 1993.
11. Lottes T. FXAA [Электронный ресурс] // NVIDIA White Paper. URL: <https://developer.nvidia.com> (дата обращения: 15.11.2025).
12. Karis B. High Quality Temporal Supersampling // ACM SIGGRAPH Courses, Advances in Real-Time Rendering. — 2014.

Физически корректное освещение в реальном времени: BRDF-модели и материалы

Белолипецкий Виталий Александрович, студент
Камчатский государственный технический университет (г. Петропавловск-Камчатский)

В статье рассматриваются принципы физически корректного освещения (Physically Based Rendering, PBR), ставшего стандартом представления материалов в современных движках реального времени. Проанализированы функция двунаправленного отражения света (BRDF), модель микрофасетного отражения Кука — Торренса и её практическая реализация через параметры шероховатости и металличности. Отдельное внимание уделено принципу сохранения энергии и роли окружающего освещения (image-based lighting), позволяющего корректно отображать материалы в произвольных условиях освещённости.

Ключевые слова: PBR, BRDF, микрофасетная модель, шероховатость, металличность, рендеринг реального времени.

До широкого внедрения физически корректного подхода параметры материалов в играх и интерактивных приложениях задавались художниками произвольно, без привязки к физическим свойствам реальных поверхностей, что затрудняло предсказуемое поведение материала при смене условий освещения [1, с. 257]. Подход **Physically Based Rendering (PBR)** решает эту проблему, описывая взаимодействие света с поверхностью через параметры, имеющие физический смысл, и подчиняя расчёт освещения принципу сохранения энергии [2]. Целью данной статьи является рассмотрение математической основы PBR — функции BRDF, наиболее распространённой микрофасетной модели и её параметризации, используемой в современных игровых движках.

Функция двунаправленного отражения (BRDF)

BRDF (Bidirectional Reflectance Distribution Function) описывает, какая доля света, падающего на поверхность с одного направления, отражается в произвольном направлении наблюдения [3, с. 14]. Формально BRDF — функция четырёх угловых переменных, возвращающая коэффициент отражения для пары направлений «к источнику света» и «к наблюдателю».

Физически корректная BRDF обязана подчиняться двум свойствам: взаимности Гельмгольца (результат не меняется при перестановке направлений) и сохранению энергии — суммарная отражённая энергия не должна превышать падающую [4].

Итоговая BRDF материала обычно представляется как сумма двух составляющих: **диффузной**, описывающей рассеянное переотражение света внутри материала, и **зеркальной (спекулярной)**, описывающей прямое отражение от поверхности [5]. Простейшая диффузная модель — закон Ламберта, согласно которому яркость поверхности не зависит от угла наблюдения. Для зеркальной составляющей в PBR используется более сложная микрофасетная модель.

Микрофасетная модель Кука — Торренса

Микрофасетная теория рассматривает поверхность материала как набор микроскопических плоских зеркальных фасеток, ориентация которых статистически распределена вокруг общей нормали поверхности [6, с. 33]. Чем более гладкая поверхность, тем уже это распределение и тем более чётким и сфокусированным получается зеркальное отражение; для шероховатых поверхностей отражение становится широким и размытым. Модель **Кука — Торренса** объединяет три компонента: функцию распределения нормалей фасеток (NDF, чаще всего GGX/Trowbridge-Reitz), функцию геометрического затенения, описывающую самозатенение и взаимное перекрытие фасеток, и член Френеля, описывающий зависимость доли отражённого света от угла падения [7].

На практике параметры этой модели для художника сводятся к двум интуитивным величинам: **шероховатости (roughness)**, управляющей шириной распределения нормалей фасеток, и **металличности (metalness)**, определяющей, является ли базовый цвет материала диффузным альбедо (для диэлектриков) или цветом зеркального отражения (для металлов) [8, с. 71]. Такая параметризация, известная как *metallic-roughness workflow*, стала фактическим стандартом и поддерживается форматом материалов glTF [9].

Освещение на основе изображений (Image-Based Lighting)

PBR-материалы корректно реагируют не только на точечные источники света, но и на рассеянное окружающее освещение, что требует особого подхода — **image-based lighting (IBL)** [10]. Окружение сцены захватывается в виде кубической карты (cubemap) и предварительно свёртывается с диффузной и зеркальной составляющими BRDF для разных уровней шероховатости, что позволяет в реальном времени получать физически приближённый результат отражения окружения без полного интегрирования по полусфере направлений для каждого пикселя [11, с. 145]. Дополнительно для зеркальной составляющей применяется приближение *split-sum*, разделяющее интеграл освещения на предвычисленную карту окружения и независимую от освещения таблицу интегралов BRDF (BRDF LUT) [12].

Сравнение ключевых параметров PBR-материалов представлено в таблице 1.

Таблица 1. Основные параметры *metallic-roughness* PBR-материала

Параметр	Описание	Влияние на вид материала
Albedo	Базовый цвет поверхности	Цвет диффузного отражения или подложки
Roughness	Ширина распределения нормалей фасеток	Чёткость/размытость зеркального блика
Metalness	Доля металлического отражения	Цветной блик у металлов, бесцветный у диэлектриков

Заключение

Физически корректное освещение перевело описание материалов от произвольных художественных параметров к величинам, согласованным с физическими свойствами реальных поверхностей, что обеспечило предсказуемое поведение материалов при смене условий освещения и единый визуальный язык между различными приложениями и движками. Микрофасетная модель Кука — Торренса с параметризацией *metallic-roughness* стала практическим стандартом отрасли, а методы освещения на основе изображений позволили перенести физически корректный расчёт окружающего освещения в бюджет рендеринга реального времени. Дальнейшее развитие направления связано с более точным моделированием сложных материалов — тканей, кожи и многослойных покрытий, требующих более сложных BRDF-моделей.

Литература:

1. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — 1200 с.
2. Pharr M., Jakob W., Humphreys G. Physically Based Rendering: From Theory to Implementation. — 3rd ed. — Morgan Kaufmann, 2016.

3. Nicodemus F. E. et al. Geometrical Considerations and Nomenclature for Reflectance // National Bureau of Standards Monograph. — 1977.
4. Lagarde S., de Rousiers C. Moving Frostbite to Physically Based Rendering // ACM SIGGRAPH Courses. — 2014.
5. Burley B. Physically Based Shading at Disney // ACM SIGGRAPH Courses. — 2012.
6. Cook R. L., Torrance K. E. A Reflectance Model for Computer Graphics // ACM Transactions on Graphics. — 1982.
7. Walter B., Marschner S. R., Li H., Torrance K. E. Microfacet Models for Refraction through Rough Surfaces // Eurographics Symposium on Rendering. — 2007.
8. Karis B. Real Shading in Unreal Engine 4 // ACM SIGGRAPH Courses, Physically Based Shading in Theory and Practice. — 2013.
9. Khronos Group. glTF 2.0 Specification, Material Model [Электронный ресурс]. URL: <https://www.khronos.org> (дата обращения: 16.11.2025).
10. Ramamoorthi R., Hanrahan P. An Efficient Representation for Irradiance Environment Maps // ACM SIGGRAPH. — 2001.
11. Lazarov D. Getting More Physical in Call of Duty: Black Ops II // ACM SIGGRAPH Courses. — 2013.
12. Karis B. High-Quality Specular Reflections with Roughness // ACM SIGGRAPH Courses, Advances in Real-Time Rendering. — 2013.

Системы частиц в реальном времени: GPU-симуляция и рендеринг визуальных эффектов

Белолипецкий Виталий Александрович, студент

Камчатский государственный технический университет (г. Петропавловск-Камчатский)

В статье рассматриваются принципы построения систем частиц, используемых для визуализации огня, дыма, искр и иных динамических эффектов в рендеринге реального времени. Проанализированы классическая CPU-модель системы частиц и её ограничения, а также современный подход к симуляции на GPU с использованием compute-шейдеров, позволяющий обрабатывать миллионы частиц без участия центрального процессора. Отдельное внимание уделено методам рендеринга частиц через биллборды и проблеме корректной сортировки полупрозрачных частиц по глубине.

Ключевые слова: системы частиц, GPU-симуляция, compute-шейдеры, биллборды, визуальные эффекты, рендеринг реального времени.

Системы частиц — один из основных инструментов визуализации стохастических природных и физических явлений: огня, дыма, пыли, искр, дождя и магических эффектов в играх [1, с. 96]. В отличие от обычной геометрии, частица представляет собой простой объект с минимальным набором атрибутов — позицией, скоростью, временем жизни и размером, — а визуальное богатство эффекта достигается за счёт большого количества таких объектов и случайных вариаций их параметров [2]. Поскольку число частиц в современных эффектах может достигать сотен тысяч и миллионов, выбор архитектуры симуляции существенно влияет на производительность приложения. Целью статьи является сравнение классической CPU-модели систем частиц с подходом на основе GPU-симуляции и рассмотрение особенностей рендеринга больших объёмов частиц.

Классическая модель системы частиц

В классической архитектуре каждая частица описывается структурой данных, хранящей позицию, скорость, ускорение, цвет, размер и оставшееся время жизни [3, с. 211]. На каждом кадре центральный процессор обновляет состояние всех активных частиц по простым физическим уравнениям, создаёт новые частицы взамен умерших и передаёт итоговый набор позиций и атрибутов в графический конвейер для отрисовки [4]. Такая модель проста в реализации и достаточна для эффектов с умеренным количеством частиц, однако обновление на CPU становится узким местом при росте их числа, поскольку вычисления выполняются последовательно или с ограниченным параллелизмом многоядерного процессора.

GPU-симуляция частиц

Современный подход переносит обновление состояния частиц непосредственно на GPU с использованием **compute-шейдеров** [5]. Данные о частицах хранятся в буфере на видеопамяти, и каждая частица обновляется независимым по-

током GPU, что позволяет использовать массовый параллелизм видеокарты для одновременной обработки сотен тысяч частиц без передачи данных между CPU и GPU на каждом кадре [6, с. 58]. Создание и уничтожение частиц при этом реализуется через атомарные операции подсчёта живых частиц или через циклический буфер с фиксированным максимальным числом слотов.

Дополнительное преимущество GPU-симуляции — возможность учитывать взаимодействие частиц со сценой непосредственно в шейдере: например, столкновение со статической геометрией через буфер глубины или влияние силовых полей, заданных текстурой [7]. Это позволяет реализовывать сложное поведение эффектов — взаимодействие дыма с препятствиями, разлёт искр от поверхности — без обращения к физическому движку на CPU.

Рендеринг частиц: биллборды и сортировка по глубине

Большинство систем частиц визуализируются как **биллборды** — плоские прямоугольники, всегда ориентированные на камеру, на которые накладывается текстура частицы (дым, искра, спрайт огня) [8, с. 134]. Поскольку частицы, как правило, полупрозрачны, для корректного смешивания их необходимо отрисовывать в порядке от дальних к ближним относительно камеры — в противном случае возникают визуальные артефакты на пересечениях частиц [9]. Полная сортировка миллионов частиц по глубине на каждом кадре дорога, поэтому на практике применяются приближённые методы: сортировка по ограниченному числу корзин глубины (depth bucketing) или раздельный рендеринг непрозрачных и аддитивных частиц, для которых порядок отрисовки визуально менее критичен [10].

Для повышения детализации без увеличения числа реальных частиц применяются **flipbook-анимации** — текстуры-атласы с последовательностью кадров анимации горения или дыма, проигрываемые на каждой частице в зависимости от её времени жизни [11]. В сочетании с GPU-симуляцией это позволяет добиться визуально сложных эффектов при сохранении высокой производительности, что делает данный подход стандартом для современных движков реального времени [12, с. 87].

Сравнение архитектур системы частиц представлено в таблице 1.

Таблица 1. Сравнение архитектур систем частиц

Архитектура	Основное преимущество	Основной недостаток
CPU-симуляция	Простота реализации и отладки	Ограниченное число частиц
GPU-симуляция	Поддержка миллионов частиц	Сложность реализации и отладки
Биллборды	Низкая стоимость геометрии частицы	Требуют сортировки по глубине

Заключение

Системы частиц остаются основным инструментом визуализации стохастических динамических эффектов в рендеринге реального времени. Перенос симуляции на GPU с использованием compute-шейдеров снял основное ограничение классической CPU-модели — предел по числу одновременно активных частиц, — позволив современным эффектам состоять из сотен тысяч и миллионов элементов. Вместе с тем рендеринг таких объёмов полупрозрачных частиц требует компромиссов при сортировке по глубине, а развитие направления связано с дальнейшей интеграцией частиц с физикой сцены и более совершенными приближёнными методами сортировки и затенения.

Литература:

1. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — 1200 с.
2. Reeves W. T. Particle Systems — A Technique for Modeling a Class of Fuzzy Objects // ACM Transactions on Graphics. — 1983. — Vol. 2, No. 2.
3. Shirley P., Marschner S. Fundamentals of Computer Graphics. — 4th ed. — CRC Press, 2015.
4. Latta L. Building a Million-Particle System // Game Developers Conference Proceedings. — 2004.
5. Microsoft. Compute Shader Overview [Электронный ресурс] // Microsoft Learn, DirectX Graphics. URL: <https://learn.microsoft.com> (дата обращения: 17.11.2025).
6. Green S. Particle Simulation using CUDA // NVIDIA Technical Report. — 2010.
7. Kipfer P., Westermann R. Realistic and Interactive Simulation of Rivers // Graphics Interface Proceedings. — 2006.
8. Akeley K., Hanrahan P. The Graphics Pipeline [Электронный ресурс] // Stanford CS248 Course Notes. URL: <https://graphics.stanford.edu> (дата обращения: 17.11.2025).

9. Bavoil L., Myers K. Order Independent Transparency with Dual Depth Peeling // NVIDIA White Paper. — 2008.
10. Wronski B. GPU Particle Sorting Techniques [Электронный ресурс] // Bart Wronski Blog. URL: <https://bartwronski.com> (дата обращения: 18.11.2025).
11. Thibieroz N. Animated Sprites and Flipbook Textures // ACM SIGGRAPH Courses. — 2011.
12. Lagarde S., de Rousiers C. Moving Frostbite to Physically Based Rendering // ACM SIGGRAPH Courses. — 2014.

Цифровизация учетно-регистрационной системы в Донецкой Народной Республике: основные направления и первые результаты

Воропаев Никита Александрович, студент магистратуры
Донецкий национальный технический университет

Статья посвящена анализу процессов цифровизации государственной регистрации прав и кадастрового учета в Донецкой Народной Республике как новом субъекте Российской Федерации. Рассматриваются ключевые направления внедрения цифровых технологий: переход на электронный документооборот с юридическими лицами, развитие экстерриториального принципа оказания услуг, внедрение Национальной системы пространственных данных (НСПД), использование платформы обратной связи и оптимизация процедур регистрации. На основе количественных данных оцениваются первые результаты цифровой трансформации учетной системы ДНР, а также выявляются основные проблемы и перспективы дальнейшего развития

Ключевые слова: *Донецкая Народная Республика, цифровизация, Росреестр, ЕГРН, электронные услуги, Национальная система пространственных данных, экстерриториальный принцип, кадастровый учет.*

Введение

Включение Донецкой Народной Республики (ДНР) в состав Российской Федерации (РФ) в 2022 году поставило перед государственными органами задачу не только правовой и административной интеграции, но и технологической — внедрения современных цифровых стандартов оказания государственных услуг, действующих на всей территории России. Одной из ключевых сфер, где цифровая трансформация приобрела критическое значение, стала система государственной регистрации прав на недвижимое имущество и кадастрового учета.

На момент вхождения в состав РФ учетная система ДНР находилась в состоянии, далеком от общероссийских стандартов: отсутствовал единый цифровой реестр, значительная часть объектов не была учтена, документооборот велся преимущественно на бумажных носителях. Как отметил руководитель Росреестра Олег Скуфинский, на 1 января 2023 года в реестре было около 300 тысяч объектов — в основном от Донецкой и Луганской Народных Республик. За два с лишним года этот показатель вырос более чем в 20 раз [1].

Цель данной статьи — выявить основные направления цифровизации учетно-регистрационной системы в ДНР, оценить первые количественные результаты этой работы и определить перспективы дальнейшего развития.

1. Предпосылки и необходимость цифровизации учетно-регистрационной системы в ДНР

Цифровизация учетно-регистрационной системы в ДНР была обусловлена несколькими объективными факторами.

Во-первых, масштаб задачи. На территории ДНР расположено значительное количество кадастровых кварталов, требующих инвентаризации и внесения сведений в Единый государственный реестр недвижимости (ЕГРН). По данным Росреестра, на территории ДНР, ЛНР, Запорожской и Херсонской областей расположено более 46 тыс. кадастровых кварталов [2]. Обработка такого объема данных вручную невозможна — требовались цифровые инструменты.

Во-вторых, необходимость интеграции с федеральными системами. Для полноценного функционирования в правовом поле Российской Федерации учетная система ДНР должна быть совместима с общероссийскими информационными ресурсами, прежде всего с ЕГРН и Национальной системой пространственных данных.

В-третьих, потребности граждан и бизнеса. Жители ДНР нуждались в оперативном и удобном доступе к услугам Росреестра — регистрации прав, получении выписок, оформлении наследства. Без цифровизации эти процессы оставались бы длительными и затруднительными.

Как подчеркнул руководитель Управления Росреестра по ДНР Юрий Сироватко, работа велась в рамках стратегической задачи — создать актуальный реестр данных об объектах недвижимости и правах собственности. Многие объекты недвижимости в прошлые годы не были должным образом учтены, имели устаревшие или противоречивые данные, отсутствовали кадастровые номера, а в некоторых случаях — даже технические паспорта [3].

2. Основные направления цифровизации в ДНР

2.1. Переход на обязательный электронный документооборот с юридическими лицами

Одним из ключевых направлений цифровизации стал переход на электронное взаимодействие с юридическими лицами. С 1 марта 2025 года вступили в силу изменения в законодательство РФ, согласно которым юридические лица обязаны представлять заявления о государственном кадастровом учете и (или) государственной регистрации прав, а также прилагаемые к ним документы в орган регистрации прав только в форме электронных документов [4].

Это означает, что все предприятия, организации и застройщики, имеющие недвижимость на территории ДНР, перешли на полностью безбумажный документооборот с Росреестром. Такое решение, уже апробированное в других регионах России, позволило ускорить обработку документов, исключить ошибки, связанные с человеческим фактором, и сократить сроки предоставления услуг.

Кроме того, застройщики теперь обязаны подавать документы за дольщиков, что упрощает процесс регистрации прав участников долевого строительства [4].

2.2. Экстерриториальный принцип оказания услуг

Важным цифровым инструментом, значительно расширяющим доступность услуг Росреестра для жителей ДНР, стал экстерриториальный принцип подачи документов. Суть этого принципа заключается в том, что заявитель может подать документы на регистрацию прав на недвижимость, расположенную в ДНР, не выезжая за пределы своего региона, а обратившись в любой МФЦ на территории России.

По данным Управления Росреестра по ДНР, с начала оказания услуги экстерриториально принято 150 633 заявления для проведения учетно-регистрационных действий, что составляет 5,51 % от общего количества заявлений, принятых в регионе с 2023 года (2 731 886) [5].

С октября 2025 года собственники объектов недвижимости, расположенных на территории ДНР, получили возможность регистрировать права не только в МФЦ Москвы, Московской и Ростовской областей, Санкт-Петербурга, а также ЛНР, Запорожской и Херсонской областей, но еще в 35 регионах России [5]. Таким образом, география доступных МФЦ была существенно расширена,

что особенно важно для граждан, временно покинувших территорию республики.

Новые офисы существенно увеличивают возможности граждан получать услуги Росреестра вблизи места проживания, избавляя их от необходимости выезжать за границы региона [5].

2.3. Внедрение Национальной системы пространственных данных

Наиболее масштабным проектом цифровизации, затрагивающим в том числе ДНР, является создание Национальной системы пространственных данных (НСПД) — единой цифровой платформы, которая объединит федеральные и региональные системы о земле и недвижимости.

НСПД представляет собой, по сути, цифровой двойник страны: с точными границами, рельефом, инфраструктурой и сведениями о недвижимости. Система собирает, обрабатывает и хранит пространственные данные, обеспечивая их точность и доступность.

Внедрение НСПД на воссоединенных территориях идет поэтапно. По состоянию на март 2026 года более 700 сотрудников органов власти всех четырех новых субъектов РФ активно используют в работе НСПД. С помощью платформы они осуществляют планирование развития территорий и определяют локации для приоритетной реализации инвестиционно-строительных проектов [6].

Завершить формирование единой цифровой платформы пространственных данных и единой электронной картографической основы, объединив разрозненные сведения о земле и иных объектах недвижимости на территории всей страны, планируется к 2030 году [6].

2.4. Оптимизация процедур регистрации

Цифровизация коснулась не только форматов взаимодействия, но и самих процедур. Филиал ППК «Роскадастр» по ДНР внедрил новую упрощенную систему регистрации недвижимости. Теперь граждане, получившие результаты кадастровых работ, могут подать заявление на регистрацию прав в ЕГРН напрямую в офисах филиала ППК «Роскадастр» по ДНР, не обращаясь в МФЦ.

Как рассказала заместитель директора — главный технолог филиала ППК «Роскадастр» по ДНР Екатерина Колоденко, это было сделано в целях упрощения и оптимизации процедуры. Подать заявление могут как сами собственники, так и их уполномоченные представители, действующие на основании нотариальной доверенности или других законных оснований, включая наследников [7].

4. Количественные результаты цифровой трансформации

Наиболее наглядным показателем эффективности цифровизации является динамика наполнения ЕГРН сведениями об объектах недвижимости в ДНР.

По состоянию на конец марта 2026 года в ЕГРН внесено 6,7 млн сведений об объектах недвижимости на территории ДНР, ЛНР, Запорожской и Херсонской областей, в том числе с учетом работ по сплошной инвентаризации. Еще год назад этот показатель составлял 1,3 млн.

По данным на июнь 2026 года, в реестре уже чуть более 6,9 млн объектов. Рост за два с половиной года — с 300 тысяч до почти 7 миллионов объектов — демонстрирует колоссальный масштаб проделанной работы.

В рамках сплошной инвентаризации, которая ведется по двум направлениям — изучение архивных документов и обход территорий, — выявлено 251 750 объектов, сведения о которых внесены в ЕГРН. Также на основании архивных документов внесены сведения в отношении 1,4 млн объектов.

По итогам инвентаризации и заявлений граждан в ЕГРН внесены сведения о 6,5 млн объектов недвижимости — 80 % от общего числа. Единый реестр недвижимости на воссоединенных территориях планируется сформировать до конца 2026 года [1].

5. Проблемы и перспективы дальнейшей цифровизации

Несмотря на достигнутые успехи, процесс цифровизации учетно-регистрационной системы в ДНР сталкивается с рядом проблем.

Во-первых, сохраняется цифровое неравенство. Доля электронных сделок в ДНР (16 %) существенно ниже среднероссийского показателя (72 %). Это связано как с недостаточной цифровой грамотностью населения, так и с объективными трудностями переходного периода.

Во-вторых, требуется завершение инвентаризации. Хотя работы близки к завершению, предстоит еще верификация данных и уточнение показателей. До конца 2026 года планируется провести инвентаризацию более 28 тыс. кадастровых кварталов, доступных к обследованию [1,2].

В-третьих, необходима дальнейшая интеграция с НСПД. Передачу данных в ЕЦП НСПД сегодня ведут 64 региона [9]. Задача — обеспечить полноценное подключение ДНР к этой системе и обучение сотрудников работе с платформой.

В-четвертых, требуется развитие кадрового потенциала. Внедрение цифровых технологий требует повышения квалификации как сотрудников Росреестра и Роскадастра, так и работников МФЦ.

Перспективы дальнейшей цифровизации связаны с достижением целевых показателей, установленных государственной программой «Национальная система простран-

ственных данных»: доля электронных услуг — 95 %, сроки регистрации — до 1 дня, полнота данных ЕГРН — 95 % [6]. Для ДНР эти цели отдалены, но вектор движения задан.

Создание отечественной импортозамещенной электронной картографической основы остается важнейшим направлением работы, ее готовность составляет 80 %. Организация работы Росреестра в Донецкой и Луганской народных республиках, а также в Запорожской и Херсонской областях остается приоритетной задачей.

Заключение

Проведенный анализ позволяет сделать следующие выводы:

Цифровизация учетно-регистрационной системы в ДНР является необходимым условием интеграции нового субъекта в общероссийское правовое и информационное пространство. За два с половиной года объем сведений в ЕГРН вырос с 300 тыс. до почти 7 млн объектов, что стало возможным благодаря внедрению цифровых инструментов сбора и обработки данных [1].

Основными направлениями цифровизации стали: переход на обязательный электронный документооборот с юридическими лицами (с 1 марта 2025 года) [4], развитие экстерриториального принципа оказания услуг (охвачено уже 35 регионов РФ) [5], внедрение Национальной системы пространственных данных (более 700 сотрудников используют платформу) [6], запуск цифровых каналов взаимодействия с гражданами (платформа обратной связи, онлайн-консультации) и оптимизация процедур регистрации (прямая подача заявлений в Роскадастр) [7].

Достигнуты первые количественные результаты: в ЕГРН внесено 80 % объектов недвижимости (6,5 млн из общего числа), выявлено более 251 тыс. ранее неучтенных объектов, принято более 150 тыс. заявлений по экстерриториальному принципу, платформой обратной связи воспользовались свыше 10,5 тыс. жителей [5].

Перспективы цифровизации связаны с завершением инвентаризации до конца 2026 года, полным внедрением НСПД к 2030 году и достижением общероссийских стандартов по доле электронных услуг, срокам регистрации и полноте данных ЕГРН [1,6].

Таким образом, цифровизация учетно-регистрационной системы в ДНР находится на начальном, но активно развивающемся этапе. Достигнутые результаты свидетельствуют о правильности выбранного курса, а выявленные проблемы требуют последовательного и системного решения в рамках реализации государственной программы «Национальная система пространственных данных».

Литература:

1. Федеральный конституционный закон от 04.10.2022 № 5-ФКЗ «О принятии в Российскую Федерацию Донецкой Народной Республики и образовании в составе Российской Федерации нового субъекта — Донецкой Народной Республики».
2. Единый реестр недвижимости в новых регионах сформируют до конца 2026 года [Электронный ресурс] // Tavria.tv. — URL: <https://tavria.tv> (дата обращения: 18.06.2026).

3. Руководитель Росреестра ДНР Юрий Сироватко рассказал о значении комплексных кадастровых работ в Республике [Электронный ресурс] // Лента новостей ДНР. — 2026. — 3 марта. — URL: <https://newsdonetsk.ru> (дата обращения: 18.06.2026).
4. Обращаем внимание юридических лиц, имеющих недвижимость на территории ДНР [Электронный ресурс] // Официальный сайт администрации города Докучаевска. — URL: <https://dokuchaevsk-r897.gosweb.gosuslugi.ru> (дата обращения: 18.06.2026).
5. Новые возможности для граждан: регистрация недвижимости теперь проще [Электронный ресурс] // Официальный сайт администрации города Шахтерска. — 2025. — 3 октября. — URL: <https://shahtersk.gosuslugi.ru> (дата обращения: 18.06.2026).
6. Росреестр привёл количественные сведения о Национальной системе пространственных данных [Электронный ресурс] // news.egov.itmo.ru. — 2026. — 13 марта. — URL: <https://news.egov.itmo.ru> (дата обращения: 18.06.2026).
7. Более 10 500 обращений: жители ДНР активно используют платформу обратной связи Росреестра [Электронный ресурс] // Официальный сайт администрации Мангушского муниципального округа. — 2025. — 5 февраля. — URL: <https://mangush.gosuslugi.ru> (дата обращения: 18.06.2026).
8. В ДНР оптимизировали процедуру регистрации недвижимости [Электронный ресурс] // ВсеДНР. — 2025. — 2 июля. — URL: <https://vsednr.ru> (дата обращения: 18.06.2026).
9. Марат Хуснуллин: В новых регионах в рамках сплошной инвентаризации выявлено порядка 252 тыс. объектов недвижимости [Электронный ресурс] // Фонд развития промышленности ДНР. — 2025. — 29 апреля. — URL: <https://frp-dnr.ru> (дата обращения: 18.06.2026).

Разработка мобильного приложения с ИИ-помощником для ухода за комнатными растениями

Гафурова Евгения Владимировна, студент

Научный руководитель: Климова Анжелика Сергеевна, кандидат технических наук, доцент

Казанский национальный исследовательский технологический университет

Введение

Комнатные растения — это не просто элемент декора, а источник кислорода, психологического комфорта и связи с природой в условиях мегаполиса. Для городского жителя они становятся источником кислорода, живым напоминанием о природе и даже способом снизить стресс после насыщенного рабочего дня. Интерес к домашнему садоводству заметно вырос в последние годы: люди всё активнее обустривают зелёные уголки у себя дома и в офисах. Но современный человек, погруженный в рабочие задачи и социальные обязательства, часто забывает о необходимости полить, подкормить или пересадить растение. Цветок, за которым не следят, быстро приходит в упадок, а вместе с ним угасает и мотивация заниматься садоводством вообще.

Актуальность связана с ростом популярности домашнего садоводства и осознанного растениеводства. Новички и опытные энтузиасты сталкиваются с проблемами: неправильный уход, несвоевременное выявление болезней, недостаток структурированных знаний. Существующие приложения для ухода за растениями лишь частично решают эту проблему. Пространство для более умного и персонализированного решения остаётся незанятым.

Функциональные требования для приложения с ИИ-помощником для любителей комнатных растений следующие:

- добавление нового растения в коллекцию и при том для него должна сохраняться информация: прозвище растения, его вид и сорт, дата добавления в коллекцию, фотография растения;
- изменение всех параметров растения;
- отображение списка всех растений;
- возможность создавать график ухода на основе параметров растения;
- push-уведомления о предстоящих процедурах ухода;
- ИИ-помощник, отвечающий на популярные запросы и способный по фотографии определять болезнь растения и рекомендовать решение;
- наличие элементов геймификации для поддержания активности пользователя;
- каталог статей по уходу за растениями, содержащих полезные материалы.

Рассмотрим диаграмму Исикавы для мобильного приложения с ИИ-помощником для ухода за комнатными растениями на рисунке 1.

Данная диаграмма необходима, чтобы обобщить функциональные требования и соответствующие им процессы.

Разработка ИИ-модуля

Подробнее остановимся на реализации ИИ-модуля. В разработке использовалась вторая версия модели

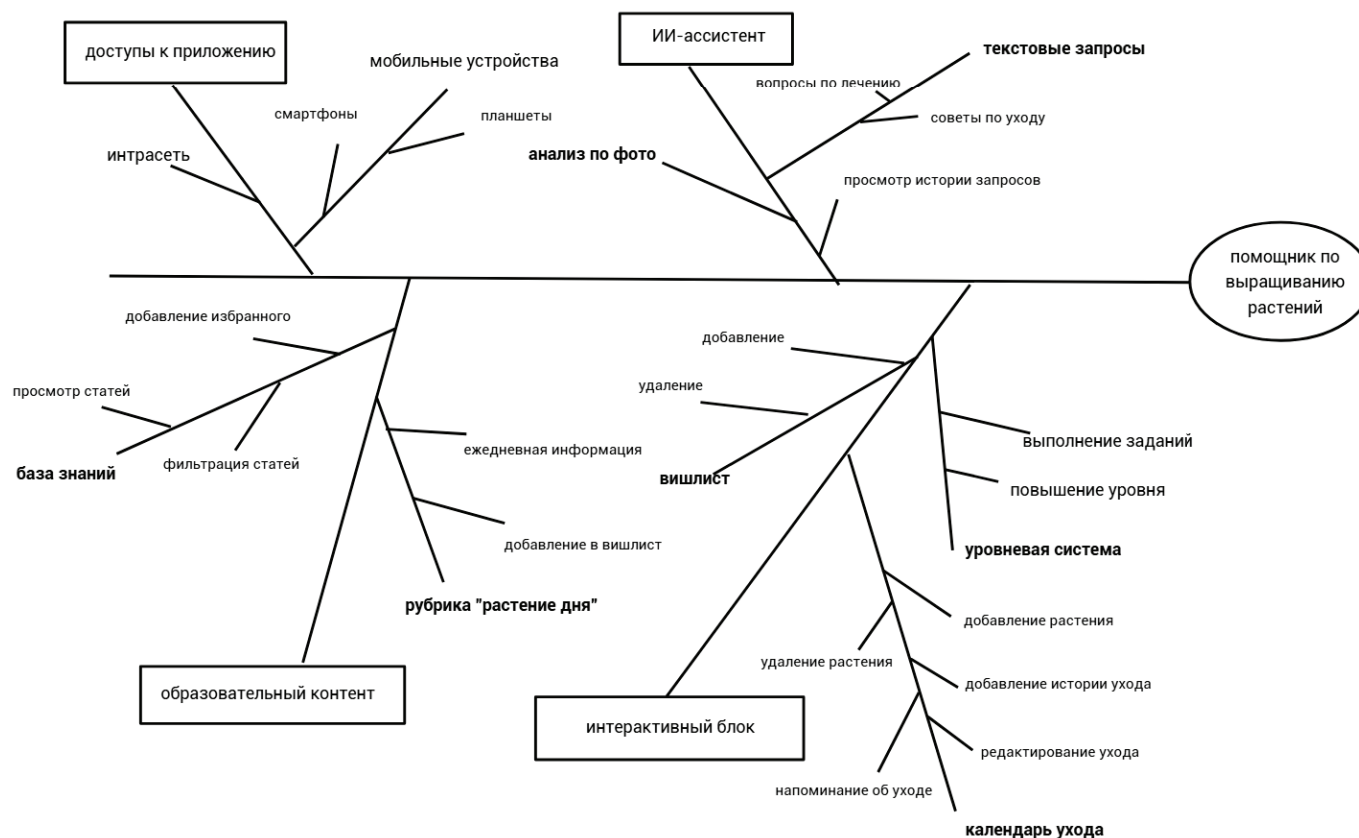


Рис. 1. Диаграмма Исикавы

MobileNet, представленная разработчиками Google в 2018 году. Она стала логическим продолжением идей, лежащих в основе первой версии, однако авторы внесли в неё несколько действительно значимых нововведений. Главной целью было не просто создать облегчённую модель для мобильных устройств, но и сохранить высокую точность, что всегда представляет собой сложный компромисс.

Датасет взят на Kaggle — ведущей платформе для специалистов в области Data Science и машинного обучения. Она хорошо подходит как для начинающих, так и для профи [19]. Огромный репозиторий датасетов и готовые к развертыванию модели являются огромными плюсами платформы, поэтому была выбрана именно она.

Визуальная модель обучена на датасете PlantVillage, который содержит около 50 тысяч изображений листьев различных растений с разными заболеваниями. Датасет разделен на 38 классов, включающих здоровые растения и различные болезни для них.

MobileNetV2 использует специальные блоки с inverted residuals и linear bottlenecks, что позволяет достичь хорошей точности при небольшом размере модели.

В процессе реализации приложения взята MobileNetV2 без верхних полносвязных слоев. Она загружается с весами, предобученными на ImageNet. Затем добавляются собственные слои: GlobalAveragePooling2D для усреднения признаков по пространственным измерениям, Dropout слой с коэффициентом 0.5 для регуляризации и предотвращения переобучения и полносвязный Dense

слой с 38 выходами и функцией активации softmax для классификации по числу классов болезней. Обучение проходило в два этапа.

На первом этапе замораживаются все слои базовой модели MobileNetV2 и обучаются только добавленные слои. Это позволяет быстро адаптировать выходные слои под задачу, не теряя признаки, которые MobileNetV2 научилась извлекать на ImageNet. На этом этапе используется оптимизатор Adam со скоростью обучения 0.001, и модель обучается около 5 эпох.

На втором этапе выполняется тонкая настройка. Размораживаются последние 50 слоев базовой модели, и вся сеть дообучается с маленькой скоростью обучения 0.0001, чтобы не испортить уже выученные признаки. Этот этап длится 10 эпох.

После обучения модель сохраняется в формате Keras H5, а затем конвертируется в формат TensorFlow Lite для использования на мобильных устройствах.

Конвертация выполняется через TFLiteConverter. Поскольку MobileNet V2 содержит некоторые операции, которые не полностью поддерживаются базовым TFLite, при конвертации включается поддержка SELECT_TF_OPS, что позволяет использовать часть операций TensorFlow напрямую.

При инициализации сервиса загружаются три файла из assets приложения: TFLite модель plant_disease_model.tflite, метаданные модели plant_disease_metadata.json, содержащие названия классов на английском, их переводы

на русский и рекомендации по лечению каждого заболевания, а также база вопросов и ответов `qa_database.json`.

Метод `analyzeImage` принимает файл изображения и возвращает результат анализа. Сначала изображение декодируется и масштабируется до размера 224 на 224 пикселя. Затем пиксели нормализуются к диапазону от 0 до 1. Подготовленный тензор подается на вход TFLite интерпретатору, который выполняет инференс и возвращает массив из 38 значений, представляющих вероятности каждого класса.

Результаты сортируются по вероятности, и формируется ответ. Если класс с максимальной вероятностью содержит слово `healthy`, значит растение здорово. В противном случае выводятся топ-3 наиболее вероятных заболевания с их русскими названиями и рекомендациями по лечению.

Если TFLite модель по какой-либо причине не загрузилась, используется `fallback` метод анализа по цветам. Сервис переключается на примитивный анализ, основанный на соотношениях цветов пикселей, что, конечно, менее точно, но лучше, чем полный сбой. Если количество зеленых пикселей более 80 %, то система предполагает, что оно здорово, если более 40 % желтого, то пользователь получит сообщение о признаке пожелтения листьев и базовые рекомендации по уходу, если более 40 % коричневых пикселей, то система оповещает об обнаружении сухих участков или коричневых пятен, которые могут быть признаком грибкового заболевания, если обнаружено более 40 % белого цвета, то это признак мучнистой росы. Если определение по пикселям по каким-либо причинам не получилось, то ассистент попросит сделать фото еще раз крупным планом.

Метод `sendMessage` принимает текстовый вопрос и возвращает ответ. Он перебирает все вопросы в базе

знаний и ищет тот, который содержит максимальное количество общих слов с запросом пользователя. Если подходящий вопрос найден, возвращается соответствующий ответ. Если пользователь ввел всего одно слово, то выдается самый первый ответ, содержащий это слово. Первый ответ для каждого растения — это общие рекомендации по уходу. В противном случае выводится сообщение, что ответ не найден, с предложением переформулировать вопрос.

Файл `plant_disease_metadata.json` содержит всю информацию, необходимую для интерпретации результатов классификации. Поле `class_names` представляет собой словарь, где ключом является индекс класса от 0 до 37 (поскольку язык обучения модели — Python, в нем нумерация начинается с нуля), а значением — английское название класса в формате «Растение_Болезнь». Поле `translations` содержит переводы английских названий на русский. Поле `treatments` содержит рекомендации по лечению для различных ключевых слов. Сервис ищет ключевые слова в английском названии болезни и подбирает соответствующую рекомендацию.

Заключение

В ходе работы спроектировано и разработано мобильное приложение Plantu для ухода за комнатными растениями. Реализованы следующие функции: ведение коллекции растений с графиком ухода, ИИ-помощник, каталог статей и список желаний.

Приложение не имеет бесплатных русскоязычных аналогов и позволит пользователям повысить выживаемость растений за счет персонализированного графика ухода, своевременных напоминаний и оперативной диагностики болезней по фотографии.

Применение технологии ассоциативного анализа к данным биржевых рынков с использованием BI-платформ

Гладких Виктория Владимировна, студент
Донской государственной технической университет (г. Ростов-на-Дону)

В статье автор исследует методологические аспекты применения технологии ассоциативного анализа к данным фондового рынка и предлагает пятиэтапную методологическую основу, реализованную в среде Microsoft Power BI.

Ключевые слова: технология ассоциативного анализа, алгоритм Apriori, анализ фондового рынка, финансовые временные ряды, BI-платформы, Power BI, многомерные зависимости, дискретизация.

Современные финансовые рынки представляют собой сложные многомерные системы, в которых взаимодействие множества факторов создает условия, где традиционные аналитические подходы часто оказываются недостаточно эффективными. Технические индикаторы, скользящие средние и методы регрессионного анализа, со-

ставляющие основу классического технического и фундаментального анализа, преимущественно ориентированы на изучение отдельных активов или парных корреляций, однако многие значимые рыночные явления возникают именно из сложных многомерных взаимодействий, которые остаются невидимыми для таких подходов. Обна-

ружить комбинации событий, таких как одновременное изменение цен на сырьевых рынках, макроэкономические публикации и отраслевые тренды, вручную практически невозможно, что создает потребность в методах, способных автоматически выявлять устойчивые многомерные паттерны.

Технология ассоциативного анализа, изначально разработанная Агравалом и его коллегами для анализа потребительских корзин, представляет значительный потенциал для решения данной проблемы. Ее фундаментальная идея заключается в поиске устойчивых закономерностей вида «если множество A присутствует в транзакции, то с высокой вероятностью в этой же транзакции присутствует множество B ». Классические алгоритмы, такие как Apriori и FP-Growth, позволяют извлекать такие правила из больших наборов дискретных данных, оценивая их значимость с помощью метрик поддержки, достоверности и лифта [1, с. 487]. Однако применение данной технологии к данным финансовых рынков сопряжено с фундаментальными методологическими трудностями, связанными с несоответствием природы исходных данных и требований алгоритмов: биржевые показатели представляют собой непрерывные, нестационарные и зашумленные временные ряды, в то время как алгоритмы ассоциативного анализа предназначены для работы с дискретными категориальными транзакциями [2, с. 298].

Адаптация технологии ассоциативного анализа к данным фондового рынка влечет за собой несколько ключевых проблем, требующих системного решения. Прежде всего, это проблема дискретизации непрерывных финансовых показателей, преобразование числовых значений цен, доходностей и объемов торгов в категориальные события неизбежно, однако выбор метода дискретизации критически влияет на качество извлекаемых правил. Равномерное разбиение на интервалы может уничтожить экономически значимые паттерны, тогда как излишне детализированная дискретизация приводит к потере статистической значимости правил. Кроме того, временная структура данных требует особого учета: финансовые временные ряды характеризуются автокорреляцией, кластерами волатильности и сдвигами режимов, что означает, что правило, обнаруженное на одних данных, может оказаться несостоятельным на других. Классические алгоритмы ассоциативного анализа предполагают независимость транзакций, поэтому временные лаги и последовательности событий должны быть явно включены в модель [3, с. 127]. Также проблема шума и переобучения в финансовых данных особенно остра: низкие пороги поддержки и достоверности генерируют тысячи ложных правил, а слишком высокие могут пропустить ценные, но редкие паттерны. Наконец, интерпретируемость результатов представляет собой самостоятельную задачу, поскольку даже технически корректное правило может быть экономически бессмысленным и неприменимым для принятия инвестиционных решений.

В научной литературе имеются отдельные попытки применения ассоциативного анализа в финансовой сфере. В области обнаружения мошенничества ассоциативные правила помогают выявлять подозрительные транзакционные паттерны [2, с. 299], в кредитном скоринге — обнаруживать правила, связанные с риском дефолта, в управлении портфелем — находить совместные движения среди активов, однако прямое применение к временным рядам цен встречается редко. Известные исследования используют дискретизацию дневных доходностей для поиска взаимосвязей между движениями цен акций или комбинируют ассоциативный анализ с техническими индикаторами для прогнозирования краткосрочных трендов [3, с. 130]. Несмотря на эти усилия, большинство работ остаются изолированными кейс-анализами без обобщенной методологии. При этом современные платформы бизнес-аналитики, такие как Microsoft Power BI, Tableau и Qlik Sense, предлагают мощные средства для интеграции данных, их преобразования и визуализации, однако их применение в связке с алгоритмами ассоциативного анализа для решения финансовых задач методически не систематизировано.

Новизна предлагаемого подхода состоит в комплексной интеграции решения всех выявленных методологических проблем в рамках единой воспроизводимой методологии, реализованной в среде BI-платформы. В отличие от существующих фрагментарных исследований, где ассоциативный анализ применяется изолированно с использованием отдельных скриптов, а результаты вручную импортируются в дашборды, предлагаемая методология объединяет весь аналитический конвейер, от предметно-ориентированной дискретизации и формирования временных транзакций до интерактивной визуализации и интерпретации правил, в единой среде Microsoft Power BI. Это позволяет не только автоматизировать процесс извлечения правил, но и обеспечить их глубокую визуальную валидацию через наложение на временные ряды и интерактивные фильтры, что недостижимо при использовании разрозненных инструментов.

В качестве решения описанных проблем предлагается пятиэтапная методологическая основа, реализованная в среде Microsoft Power BI. Выбор данной платформы обусловлен ее развитыми возможностями интеграции с языками Python и R, что позволяет реализовывать нестандартные алгоритмы анализа непосредственно в среде платформы, гибкими инструментами ETL через Power Query для предобработки исходных данных, а также широким спектром средств интерактивной визуализации, необходимых для интерпретации сложных многомерных зависимостей [4, с. 115].

Первый этап предлагаемой методологии включает сбор и интеграцию данных из различных источников. Исторические данные по ценам акций, включая цены открытия, максимума, минимума, закрытия и объемы торгов, собираются для широкого спектра активов за период не менее пяти лет. Макроэкономические индикаторы

торы, включая ключевые ставки центральных банков, данные по инфляции и индексы деловой активности, синхронизируются с биржевыми данными на ежедневной основе [2, с. 300]. Дополнительно могут быть агрегированы индикаторы новостного фона на основе методов обработки естественного языка. Все данные загружаются в Power BI с использованием Power Query или скриптов Python.

Второй этап посвящен дискретизации непрерывных переменных в категориальные события. Для ценовых изменений предлагается использовать категории на основе относительного изменения за выбранный временной интервал: сильное падение (более -3 %), падение (от -3 % до -1 %), стабильность (от -1 % до +1 %), рост (от +1 % до +3 %) и сильный рост (более +3 %). Режим волатильности рассчитывается как скользящее стандартное отклонение доходностей с последующей дискретизацией в низкую, нормальную и высокую волатильность. Аномалия объема определяется путем сравнения текущего объема с двадцатидневным скользящим средним, что дает категории низкого, нормального и высокого объема. Макроэкономические события также преобразуются в дискретные события с временной привязкой.

На третьем этапе из дискретизированных событий формируются транзакционные наборы. Возможны два принципиальных подхода. Первый предполагает использование скользящего временного окна для каждого актива, где в одну транзакцию собираются все события, произошедшие в пределах этого окна. Второй подход заключается в формировании транзакции состояния рынка, включающей события по всем активам и макропоказателям за конкретный день, что позволяет выявлять кросс-активные зависимости. Второй подход предпочтительнее для обнаружения многомерных взаимодействий между различными секторами рынка и макроэкономическими факторами.

Четвертый этап представляет собой непосредственно извлечение ассоциативных правил с использованием скриптов Python, встроенных в Power BI. Реализуются как классический алгоритм Apriori, обеспечивающий интерпретируемость и простоту настройки параметров [1, с. 490], так и более производительный алгоритм FP-Growth для работы с большими наборами данных [5, с. 4]. Для оценки значимости правил вычисляются ключевые метрики: поддержка как частота одновременного появления антецедента и консеквента правила, достоверность как условная вероятность консеквента при условии антецедента, и лифт, показывающий, насколько более вероятным становится консеквент при наличии антецедента. Минимальные пороговые значения устанавливаются эмпирически с учетом характеристик конкретного набора данных и целей анализа.

Пятый этап обеспечивает интерактивную визуализацию и интерпретацию извлеченных правил. Все обнаруженные правила загружаются обратно в Power BI в виде таблицы с сортируемыми столбцами для всех метрик, ан-

тецедента и консеквента. Сетевой граф, где узлы представляют события, а ребра — правила, дает визуальный обзор наиболее значимых ассоциаций. Слайсеры параметров позволяют фильтровать правила по различным контекстам — классу активов, временному периоду или пороговым значениям метрик. Функциональность наложения временных рядов дает возможность выбрать конкретное правило и подсветить на временной шкале периоды, когда выполнялся его антецедент, что позволяет визуально верифицировать последующую динамику. Такая интерактивная среда превращает ассоциативный анализ из пакетного алгоритма в исследовательский инструмент, позволяющий аналитикам выдвигать и проверять гипотезы в режиме реального времени.

Для иллюстрации практической применимости предложенной методологии рассмотрим гипотетический сценарий ее использования. При формировании транзакций состояния рынка на основе исторических данных за пятилетний период и применении алгоритма Apriori с порогом поддержки 0,02 и достоверности 0,7 могут быть обнаружены правила, отражающие устойчивые взаимосвязи между движениями различных активов и макроэкономическими индикаторами. Например, правило, согласно которому рост цен на энергоносители в сочетании с повышением индекса волатильности VIX выше определенного уровня с высокой вероятностью предшествует росту в технологическом секторе, демонстрирует практическую ценность такого анализа. Аналитик может использовать интерактивные визуализации Power BI для проверки обнаруженных правил, выбирая их в таблице и наблюдая на временной шкале все периоды, когда выполнялись условия антецедента, что позволяет оценить устойчивость и экономическую содержательность выявленных зависимостей. Такой подход дает возможность не только генерировать гипотезы для инвестиционных стратегий, но и проводить их визуальную валидацию на исторических данных.

Практическая значимость предложенной методологии заключается в возможности ее непосредственного применения в деятельности финансовых институтов. В инвестиционных компаниях и хедж-фондах она может служить дополнительным инструментом стратегического и тактического анализа для выявления сложных рыночных аномалий и зависимостей между активами. В риск-менеджменте методология позволяет выявлять комбинации факторов, с высокой вероятностью предшествующих периодам повышенной волатильности, что способствует улучшению моделей управления портфельными рисками. В FinTech-разработках результаты могут быть использованы как основа для создания новых аналитических сервисов, предоставляющих клиентам инсайты о структурных взаимосвязях на рынке. Кроме того, разработанный подход может служить наглядным примером применения методов интеллектуального анализа данных к решению практических экономических задач в образовательном процессе.

Таким образом, в статье предложена методологическая основа для систематического применения технологии ассоциативного анализа к данным фондового рынка с использованием современных BI-платформ. Выявлены и систематизированы ключевые методологические проблемы, включая дискретизацию непрерывных данных, учет временной структуры, фильтрацию шума и интерпретацию результатов. Разработана пятиэтапная методология, объединяющая сбор и интеграцию данных, предметно-ориентированную дискретизацию, формирование временных транзакций, извлечение ассоциативных правил с использованием алгоритмов Apriori и FP-Growth, а также интерактивную визуализацию в среде Microsoft Power BI. В отличие от существующих изолированных кейс-анализов, предлагаемый подход обеспечивает целостность и воспроизводи-

мость аналитического процесса благодаря интеграции всех этапов в единой BI-среде. Представленная работа является первым теоретическим этапом магистерского исследовательского проекта. Последующие этапы будут включать полную программную реализацию предложенной методологии, проведение вычислительных экспериментов на реальных исторических данных, а также валидацию сформулированных гипотез об экономической значимости и предсказательной силе извлекаемых ассоциативных правил. Ожидается, что результаты исследования заполнят существующий пробел между теоретическим потенциалом ассоциативного анализа и его практическим применением в финансовой аналитике, предоставив воспроизводимый инструмент для выявления скрытых многомерных зависимостей на современных финансовых рынках.

Литература:

1. Agrawal R., Srikant R. Fast Algorithms for Mining Association Rules in Large Databases // Proceedings of the 20th International Conference on Very Large Data Bases. Santiago, 1994. P. 487–499.
2. Chen Y., Zhang L. Application of association rule mining in stock market analysis // Journal of Financial Data Science. 2020. Vol. 2. No. 3. P. 45–58.
3. Wang J., Li X. Combining technical indicators with association rules for stock trend prediction // Expert Systems with Applications. 2019. Vol. 128. P. 123–135.
4. Ferrari A., Russo M. Introducing Microsoft Power BI. Redmond: Microsoft Press, 2021. 256 p.
5. Han J., Pei J., Yin Y. Mining frequent patterns without candidate generation // ACM SIGMOD Record. 2000. Vol. 29. No. 2. P. 1–12.

Введение в технику Screen Space Ambient Occlusion

Квилюнас Иван Максимович, студент

Камчатский государственный технический университет (г. Петропавловск-Камчатский)

Статья посвящена технике Screen Space Ambient Occlusion (SSAO) — методу имитации глобального освещения в рендеринге реального времени. Рассмотрены теоретические основы ambient occlusion, принцип работы алгоритма, его этапы, основные артефакты и способы их устранения.

Ключевые слова: рендеринг, ambient occlusion, экранное пространство, затенение, глобальное освещение, G-буфер, шейдер.

An introduction to Screen Space Ambient Occlusion

The article is devoted to the Screen Space Ambient Occlusion (SSAO) technique — a method of simulating global illumination in real-time rendering. The theoretical foundations of ambient occlusion, the algorithm's working principles, its stages, main artifacts and ways to eliminate them are discussed.

Keywords: rendering, ambient occlusion, screen space, shading, global illumination, G-buffer, shader.

Введение

В компьютерных играх и интерактивных приложениях реалистичность изображения во многом определяется корректным моделированием тени от рассеянного освещения. Техника Screen Space Ambient Occlusion (SSAO)

была предложена разработчиками CryEngine 2 в компании Crytek в 2007 году и с тех пор стала стандартным инструментом постобработки в рендеринге реального времени [1, с. 4]. Целью данной статьи является изложение теоретической базы и практических принципов работы SSAO.

1. Понятие Ambient Occlusion

Ambient Occlusion (AO) — метод затенения, аппроксимирующий, какую долю рассеянного света достигает точки поверхности с учётом окружающей геометрии [2, с. 14]. Значение АО в точке соответствует доле полусферы над нормалью, не перекрытой другими объектами: чем сильнее точка «закрыта», тем темнее она выглядит.

В офлайн-рендеринге АО вычисляется методом Монте-Карло — каждый луч трассируется в полусфере, что обеспечивает физическую точность, но требует тысяч лучей на пиксель и неприменимо в реальном времени [2, с. 19]. Визуальный эффект хорошо известен: углы комнат, основания объектов и стыки поверхностей выглядят темнее. Без АО сцены кажутся неестественно «плоскими».

2. История создания SSAO

До 2007 года разработчики либо отказывались от АО, либо запекали его в текстуры на этапе создания ресурсов. Запечённые карты не обновляются в реальном времени и не учитывают динамические объекты [3]. Ведущий разработчик CryEngine 2 Владимир Казаков предложил аппроксимировать АО непосредственно во время рендеринга, используя только информацию из G-буфера. В дальнейшем появились HBAO (NVIDIA, 2008), HBAO+, а также физически точный GTAO (2016). Тем не менее базовый SSAO остаётся популярным благодаря простоте реализации и производительности.

3. Принцип работы алгоритма SSAO

SSAO является постобработочным алгоритмом и применяется после рендеринга сцены в G-буфер. Алгоритм состоит из четырёх этапов.

Подготовка G-буфера. Выполняется проход отложенного рендеринга, формирующий буфер глубины и буфер нормалей. Глубина конвертируется в пространство вида для корректных 3D-вычислений [5, с. 2].

Генерация ядра выборки. Для каждого пикселя формируется набор случайных точек (16–64) в полусфере, ориентированной по нормали. Распределение смещено к началу полусферы, так как ближайшая геометрия вносит наибольший вклад в окклюзию. Небольшая тек-

стура шума (4×4 пикселя), тайлящаяся по экрану, вращает ядро в каждом пикселе, снижая заметность паттернов [5, с. 3].

Тест окклюзии. Каждая точка выборки проецируется в экранное пространство; реальная глубина G-буфера сравнивается с глубиной точки. Если реальная поверхность ближе к камере, точка считается окклюзированной. Итоговая оценка — доля окклюзированных точек. Для устранения самозатенения вводится смещение по глубине и проверка дальности [6, с. 8].

Размытие результата. Зашумлённая карта окклюзии сглаживается фильтром с учётом краёв, сохраняя резкие границы объектов [5, с. 7].

4. Типичные артефакты

Ореолы (halo). Тёмный контур вокруг объектов на контрастном фоне — следствие промаха сэмплов мимо реальной геометрии. Устраняется отсечением сэмплов по нормали и уменьшением радиуса ядра [5, с. 9].

Утечки окклюзии. Геометрия за пределами кадра не учитывается, поэтому объекты у края экрана затеняются некорректно [6, с. 13].

Самозатенение. При недостаточном bias плоские поверхности затеняют сами себя. Для борьбы с зашумлённостью применяется temporal SSAO — ротация ядра выборки между кадрами с усреднением результата [7].

Заключение

Screen Space Ambient Occlusion представляет собой практичный компромисс между физической корректностью и требованиями реалтаймового рендеринга. Стоимость алгоритма определяется разрешением экрана и числом сэмплов, а не сложностью сцены, что делает SSAO применимым даже на мобильных платформах. Главное ограничение метода — работа исключительно в экранном пространстве, порождающая характерные артефакты. Современные варианты — HBAO и GTAO — в значительной мере преодолевают эти ограничения и используются в Unity, Unreal Engine 5 и других ведущих движках. Знание принципов SSAO необходимо любому разработчику в области рендеринга, поскольку метод лежит в основе более современных подходов.

Литература:

1. Kazakov V. Screen Space Ambient Occlusion. — CryEngine SDK Documentation, Crytek GmbH, 2008.
2. Akenine-Moller T., Haines E. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — ISBN 978-1-1386-2700-0.
3. Shirley P., Chiu K. A Low Distortion Map Between Disk and Square // Journal of Graphics Tools. — 1997. — Vol. 2, № 3. — С. 45–52.
4. Bavoil L., Sainz M. Screen Space Ambient Occlusion [Электронный ресурс] // NVIDIA Developer. — 2008. — URL: <https://developer.download.nvidia.com/SDK/10.5/direct3d/Source/ScreenSpaceAO/doc/ScreenSpaceAO.pdf> (дата обращения: 15.05.2025).
5. Mittring M. Finding Next Gen: CryEngine 2 // Proceedings of ACM SIGGRAPH 2007 Courses. — New York: ACM, 2007. — С. 1–17.

6. de Vries J. SSAO [Электронный ресурс] // Learn OpenGL. — URL: <https://learnopengl.com/Advanced-Lighting/SSAO> (дата обращения: 10.05.2025).
7. Stalker J. Ground Truth Ambient Occlusion [Электронный ресурс] // Epic Games Developer Documentation. — URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/ambient-occlusion-in-unreal-engine> (дата обращения: 12.05.2025).

Основы рендеринга неба: Skybox и процедурная генерация атмосферы

Квилюнас Иван Максимович, студент

Камчатский государственный технический университет (г. Петропавловск-Камчатский)

В статье рассматриваются основные подходы к визуализации неба в рендеринге реального времени: техника Skybox, основанная на кубической карте текстур, и методы процедурной генерации атмосферы. Описаны принципы работы каждого подхода, их сравнительные характеристики и области применения в современных игровых движках.

Ключевые слова: рендеринг неба, skybox, cubemap, атмосферное рассеяние, процедурная генерация, шейдер.

The basics of sky rendering: Skyboxes and procedural atmosphere generation

The article discusses the main approaches to sky rendering in real-time graphics: the Skybox technique based on a cubic texture map, and procedural atmosphere generation methods. The operating principles of each approach, their comparative characteristics, and areas of application in modern game engines are described.

Keywords: sky rendering, skybox, cubemap, atmospheric scattering, procedural generation, shader.

Введение

Небо является одним из наиболее визуально значимых элементов трёхмерной сцены: оно задаёт общий тон освещения, влияет на восприятие пространства и в значительной мере определяет атмосферу происходящего. Несмотря на кажущуюся простоту, реалистичная визуализация неба представляет собой нетривиальную задачу, поскольку требует воспроизведения сложных оптических явлений — рассеяния света в атмосфере, изменения цвета горизонта в зависимости от положения солнца и погодных условий [1, с. 512]. В рендеринге реального времени для решения этой задачи применяются два принципиально различных подхода: статический Skybox и процедурная генерация атмосферы. Цель статьи — систематически изложить принципы работы обоих методов и провести их сравнительный анализ.

1. Техника Skybox

Skybox — это простейший и наиболее распространённый метод отрисовки фона сцены. Концептуально он представляет собой куб бесконечно большого размера, помещённый вокруг камеры, на внутренние грани которого натянута специальная текстура — кубическая карта (cubemap) [2, с. 47]. Камера всегда находится в центре куба, поэтому при повороте создаётся иллюзия бесконечного окружающего пространства.

Cubemap состоит из шести отдельных изображений, каждое из которых соответствует одной грани куба: правой, левой, верхней, нижней, передней и задней. При рендеринге используется специальная проективная выборка текстуры по трёхмерному вектору направления, что обеспечивает бесшовное соединение граней. Альтернативой кубической карте является сферическая панорамная проекция (equirectangular map, HDRI), которая проще в создании, но несколько уступает cubemap по скорости выборки на графическом процессоре.

Главное достоинство Skybox — минимальная стоимость рендеринга: куб состоит из 12 треугольников, а текстура выбирается за один проход. Критическим ограничением является статичность: небо не реагирует на изменение времени суток, погоды или положения солнца без замены текстуры. В динамических сценах эта проблема частично решается переходом между несколькими заранее подготовленными cubemap [3, с. 8].

Широкое применение Skybox нашёл в мобильных играх, инди-проектах и сценах с фиксированным временем суток. В качестве источников HDRI-текстур активно используются фотографии реального неба или специализированные библиотеки, такие как Poly Haven.

2. Процедурная генерация атмосферы

Процедурный подход к рендерингу неба основан на математическом моделировании физических процессов

рассеяния света в атмосфере, что позволяет динамически воспроизводить любое время суток, погодные условия и высоту наблюдателя.

2.1. Алгоритм трассировки лучей через атмосферу

Базовый алгоритм процедурного рендеринга атмосферы основан на численном интегрировании уравнения переноса излучения вдоль луча зрения. Для каждого пикселя трассируется луч от камеры через атмосферу; луч разбивается на N равномерных отрезков (на практике $N = 8-16$). В каждой точке интегрирования вычисляются: плотность атмосферы в зависимости от высоты, коэффициенты рассеяния Рэлея и Ми, а также оптическая глубина — интеграл поглощения от данной точки до Солнца [5, с. 3].

Суммирование вкладов всех точек даёт итоговый цвет пикселя с учётом многократного рассеяния. Для снижения стоимости вычислений результаты предварительно запекаются в трёхмерные lookup-таблицы (LUT),

что позволяет выполнять точный расчёт атмосферы в реальном времени — подход, реализованный в системе Sky Atmosphere движка Unreal Engine 5 [6].

Заключение

Skybox и процедурная генерация атмосферы представляют собой два полюса в пространстве компромиссов между производительностью и физической корректностью. Первый метод отличается минимальными вычислительными затратами и полным художественным контролем, второй — воспроизводит реальные оптические явления и поддерживает динамическое освещение сцены. Современные движки, как правило, предоставляют оба инструмента, позволяя разработчику выбирать наиболее подходящий в зависимости от целевой платформы и требований к визуальному качеству. Понимание принципов алгоритма интегрирования через атмосферу является необходимой базой для работы с продвинутыми системами рендеринга неба.

Литература:

1. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — 1198 с.
2. de Vries J. Cubemaps [Электронный ресурс] // Learn OpenGL. — URL: <https://learnopengl.com/Advanced-OpenGL/Cubemaps> (дата обращения: 20.05.2026).
3. Zucconi A. Volumetric Atmosphere Scattering [Электронный ресурс] // Alan Zucconi Blog. — 2022. — URL: <https://www.alanzucconi.com/2017/10/10/atmospheric-scattering-1> (дата обращения: 16.05.2026).
4. Wilkie A., Hošek L. An Analytic Model for Full Spectral Sky-Dome Radiance // ACM Transactions on Graphics. — 2012. — Vol. 31, № 4. — С. 1–9.
5. Hillaire S. A Scalable and Production Ready Sky and Atmosphere Rendering Technique // Computer Graphics Forum. — 2020. — Vol. 39, № 4. — С. 13–22.
6. Epic Games. Sky Atmosphere [Электронный ресурс] // Unreal Engine Documentation. — URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/sky-atmosphere-component-in-unreal-engine> (дата обращения: 11.05.2026).
7. Unity Technologies. Physically Based Sky [Электронный ресурс] // Unity HDRP Docs. — URL: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@14.0/manual/Override-Physically-Based-Sky.html> (дата обращения: 12.05.2026).

Техника Parallax Mapping для создания рельефных поверхностей

Квилюнас Иван Максимович, студент

Камчатский государственный технический университет (г. Петропавловск-Камчатский)

В статье рассматривается семейство техник Parallax Mapping, применяемых для создания иллюзии объёмного рельефа поверхности без увеличения числа полигонов. Описаны предпосылки появления метода, принципы работы базового Parallax Mapping и Parallax Occlusion Mapping (POM), их характерные артефакты и способы устранения.

Ключевые слова: parallax mapping, рельефное текстурирование, height map, tangent space, POM, bump mapping, шейдер.

The Parallax Mapping technique for creating textured surfaces

The article discusses the Parallax Mapping family of techniques used to create the illusion of volumetric surface relief without increasing the polygon count. The background of the method, the operating principles of basic Parallax Mapping and Parallax Occlusion Mapping (POM) are described, along with their characteristic artifacts and mitigation strategies.

Keywords: parallax mapping, relief texturing, height map, tangent space, POM, bump mapping, shader.

Введение

Детализированная геометрия поверхностей — один из ключевых факторов визуального реализма трёхмерной сцены. Однако увеличение числа полигонов для передачи мелкого рельефа — трещин, выбоин, тиснения, каменной кладки — резко повышает нагрузку на конвейер обработки вершин и приводит к высокому расходу видеопамети. Для решения этой проблемы в реалтаймовом рендеринге применяется ряд техник, имитирующих объём поверхности средствами фрагментного шейдера [1, с. 484]. Техника Parallax Mapping стала важным шагом на пути от примитивного Normal Mapping к полноценной имитации рельефа с самозатенением и параллаксным смещением [2]. Цель данной статьи — последовательно изложить эволюцию методов семейства Parallax Mapping и провести их сравнительный анализ.

1. Предшественники: Normal Mapping и Bump Mapping

Bump Mapping, предложенный Джеймсом Блинном в 1978 году, стал первым методом имитации рельефа без изменения геометрии [3, с. 2]. Метод возмущает нормаль поверхности перед вычислением освещения, используя карту высот (height map). Это создаёт убедительную иллюзию рельефа при фронтальном взгляде, однако силуэт объекта остаётся плоским, а боковое скольжение взгляда немедленно разрушает иллюзию.

Normal Mapping — хранит возмущённые нормали непосредственно в текстуре в пространстве касательных (tangent space), что позволяет эффективно использовать данные, запечённые в высокополигональной модели [1, с. 486]. Normal Mapping стал индустриальным стандартом с начала 2000-х годов и по-прежнему широко используется, однако смещение текстурных координат при косом взгляде отсутствует, вследствие чего рельеф «скользит» вместе с плоской поверхностью.

2. Базовый Parallax Mapping

Ключевая идея Parallax Mapping состоит в смещении текстурных координат в зависимости от угла взгляда и высоты поверхности в данной точке, что создаёт эффект параллакса — кажущегося смещения рельефных деталей при изменении ракурса, характерного для реальных объёмных поверхностей [2].

Алгоритм работает следующим образом. В фрагментном шейдере вычисляется вектор взгляда V в пространстве касательных. Из карты высот считывается значение h в исходных координатах текстуры. Затем ко-

ординаты смещаются по формуле: $UV' = UV + (V.x / V.z) \times h \times \text{scale}$, где scale — масштабный коэффициент глубины рельефа [4, с. 3]. Все последующие операции — выборка диффузной текстуры, нормалей, расчёт освещения — выполняются по смещённым координатам UV' .

Метод крайне дешёв в реализации (добавляет лишь одну выборку текстуры и несколько арифметических операций) и хорошо работает при небольшой глубине рельефа и умеренных углах взгляда. Однако при больших значениях scale и пологих углах смещение становится чрезмерным и текстура «уплывает», что является главным артефактом базового метода.

3. Parallax Occlusion Mapping

Parallax Occlusion Mapping (POM) устраняет ступенчатость с помощью линейной интерполяции между последним «воздушным» и первым «заглублённым» слоями, что даёт субпиксельно точную оценку точки пересечения [4, с. 7]. Дополнительно POM реализует самозатенение: из найденной точки трассируется вторичный луч в направлении источника света; если на его пути встречается рельеф, точка находится в тени. Это кардинально повышает реалистичность, поскольку глубокие борозды и выступы отбрасывают тени непосредственно на прилегающую поверхность.

POM является наиболее качественным методом семейства и применяется в высокобюджетных проектах для текстурирования каменной кладки, кирпича, кожи, гравия и других материалов с выраженным рельефом. Стоимость метода существенно выше базового Parallax Mapping, однако значительно ниже, чем у тесселяции с displacement mapping при сопоставимом визуальном результате [1, с. 491].

Заключение

Семейство техник Parallax Mapping представляет собой эффективный инструмент имитации объёмного рельефа поверхностей в реалтаймовом рендеринге. Эволюция от базового смещения текстурных координат к итеративному поиску пересечения луча и субпиксельной интерполяции в POM позволила многократно повысить качество эффекта при умеренном росте вычислительных затрат. Техника нашла широкое применение в современных игровых движках: Unreal Engine поддерживает POM через параметр Parallax Occlusion Mapping Depth в материальном редакторе, Unity — через пакет HDRP с опцией Pixel Displacement. Понимание принципов работы методов семейства является необходимой базой для специалистов в области технического художественного оформления и разработки шейдеров.

Литература:

1. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — 1198 с.
2. Kaneko T. et al. Detailed Shape Representation with Parallax Mapping // Proceedings of ICAT. — 2001. — С. 205–208.

3. Blinn J. F. Simulation of Wrinkled Surfaces // ACM SIGGRAPH Computer Graphics. — 1978. — Vol. 12, № 3. — С. 286–292.
4. de Vries J. Parallax Mapping [Электронный ресурс] // Learn OpenGL. — URL: <https://learnopengl.com/Advanced-Lighting/Parallax-Mapping> (дата обращения: 10.05.2025).
1. McBain M. Steep Parallax Mapping [Электронный ресурс] // NVIDIA Developer. — 2005. — URL: <https://developer.nvidia.com/gpugems/gpugems3/part-i-geometry/chapter-4-next-generation-speedtree-rendering> (дата обращения: 12.05.2026).
2. Welsh T. Parallax Mapping with Offset Limiting // Infiscap Technical Report. — 2004. — С. 1–9.
3. Epic Games. Parallax Occlusion Mapping [Электронный ресурс] // Unreal Engine Documentation. — URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/parallax-occlusion-mapping-in-unreal-engine> (дата обращения: 09.05.2026).

Методы компрессии текстур для мобильных игр: обзор и сравнение

Квилюнас Иван Максимович, студент

Камчатский государственный технический университет (г. Петропавловск-Камчатский)

В статье рассматриваются основные форматы аппаратной компрессии текстур, применяемые в мобильных играх: ETC1/ETC2, PVRTC, ASTC и BC (DXT). Описаны характеристики каждого формата и его поддержка на распространённых мобильных GPU.

Ключевые слова: компрессия текстур, ASTC, ETC2, PVRTC, блочное кодирование, мобильный GPU, видеопамять.

Texture compression methods for mobile games: an overview and comparison

The article discusses the main hardware texture compression formats used in mobile games: ETC1/ETC2, PVRTC, ASTC, and BC (DXT). The characteristics of each format and its support on common mobile GPUs are described.

Keywords: texture compression, ASTC, ETC2, PVRTC, block coding, mobile GPU, video memory.

Введение

Пропускная способность памяти является одним из наиболее острых узких мест в мобильном рендеринге. В отличие от настольных GPU, оснащённых выделенной видеопамью с высокой пропускной способностью, мобильные системы на кристалле (SoC) используют единую оперативную память, совместно потребляемую процессором и GPU. Типичная 2D-текстура с разрешением 1024×1024 пикселей в несжатом формате RGBA8 занимает 4 МБ; с учётом MIP-цепочки объём возрастает примерно до 5,3 МБ. При наличии в сцене сотен текстур суммарный расход памяти становится неприемлемым [1, с. 621].

Аппаратные форматы компрессии текстур решают эту проблему: они позволяют хранить текстуру в видеопамети в сжатом виде и декомпрессировать её непосредственно в блоке выборки текстур (TMU) за один такт без привлечения общего вычислительного конвейера. Это одновременно сокращает занимаемую память и снижает нагрузку на шину памяти [2, с. 14]. Цель данной статьи — систематизировать и сравнить основные форматы компрессии текстур, актуальные для мобильных игровых проектов.

1. Обзор основных форматов

1.1. ETC1 и ETC2

ETC1 (Ericsson Texture Compression) — обязательный формат для OpenGL ES 2.0, поддерживаемый практически всеми мобильными GPU. Блок 4×4 кодируется в 8 байт (4 бита на пиксель). Каждый блок делится на два подблока 2×4 или 4×2; для каждого подблока задаётся базовый цвет и одна из четырёх предустановленных таблиц яркостного смешения. Главное ограничение ETC1 — отсутствие поддержки альфа-канала [3, с. 4].

ETC2, введённый в OpenGL ES 3.0 и обязательный для всех Vulkan-совместимых устройств Android, расширяет ETC1 поддержкой альфа-канала (форматы RGB8A1 и RGBA8) и улучшенным режимом кодирования Т- и Н-блоков, снижающим артефакты на резких цветовых границах [4, с. 2]. ETC2 обратно совместим с ETC1 для RGB-текстур.

1.2. PVRTC

PVRTC (PowerVR Texture Compression) разработан компанией Imagination Technologies и аппаратно поддерживается исключительно на GPU серии PowerVR, которые использовались в iPhone и iPad вплоть до перехода на Apple Silicon. Принципиальное отличие PVRTC от других форматов: изображение кодируется двумя низкочастотными картами опорных цветов, интерполируемыми по всей текстуре одновременно, а не независимо по блокам. Это даёт хорошие результаты на плавных градиентах, но порождает артефакты «утечки» между несмежными деталями [1, с. 625]. PVRTC2, поддерживаемый только на PowerVR Series6 и новее, устраняет ряд этих ограничений, однако широкого распространения за пределами экосистемы Apple не получил.

1.3. ASTC

ASTC (Adaptive Scalable Texture Compression), разработанный ARM совместно с AMD и принятый в качестве расширения OpenGL ES 3.1 и обязательного компонента Vulkan на Android 8.0+, является наиболее гибким современным форматом [5, с. 1]. Его ключевая особенность — переменный размер блока от 4×4 до 12×12 пикселей при фиксированном размере закодированного блока 16 байт, что позволяет выбирать битрейт от 8 до 0,89 бит на пиксель. ASTC поддерживает HDR-текстуры, 3D-текстуры и текстуры с несколькими слоями, что делает его универсальным решением для широкого класса задач.

Заключение

Аппаратная компрессия текстур является необходимым инструментом разработки мобильных игр, позволяя сократить расход видеопамати в 4–8 раз без снижения производительности выборки. ETC2 обеспечивает надёжную базовую совместимость на платформе Android; ASTC представляет собой перспективный универсальный стандарт с гибким управлением качеством и поддержкой HDR. PVRTC утрачивает актуальность по мере перехода Apple на собственные GPU серии M и A, также поддерживающие ASTC. Современные игровые движки — Unity и Unreal Engine — предоставляют инструменты автоматического выбора и транскодирования форматов, однако осознанный выбор целевого формата остаётся задачей разработчика и напрямую влияет на объём дистрибутива, потребление памяти и качество визуального контента.

Литература:

1. Akenine-Möller T., Haines E., Hoffman N. Real-Time Rendering. — 4th ed. — CRC Press, 2018. — 1198 с.
2. Wicke M., Hachisuka T. GPU-Based Texture Compression // GPU Pro 4. — CRC Press, 2013. — С. 13–28.
3. Akkouchi S., Galin E. Adaptive Implicit Surface Polygonization Using Marching Triangles // Computer Graphics Forum. — 2001. — Vol. 20, № 2. — С. 67–80.
4. Khronos Group. ETC2 Texture Compression [Электронный ресурс] // OpenGL ES 3.0 Specification. — 2012. — URL: https://registry.khronos.org/OpenGL/extensions/OES/OES_compressed_ETC2_RGB8_texture.txt (дата обращения: 28.05.2026).
5. ARMv8 Architecture Reference Manual. ASTC Compressed Texture Image Formats [Электронный ресурс] // ARM Developer. — URL: <https://developer.arm.com/documentation/102162/0100> (дата обращения: 27.05.2026).
6. Unity Technologies. Texture Compression [Электронный ресурс] // Unity Manual. — URL: <https://docs.unity3d.com/Manual/class-TextureImporterOverride.html> (дата обращения: 29.05.2026).
7. Google. Android GPU Inspector — Texture Compression Guide [Электронный ресурс] // Android Developers. — URL: <https://developer.android.com/games/optimize/texture-compression> (дата обращения: 30.05.2026).

Цифровая герменевтика. Применение технологий искусственного интеллекта для анализа исторических правовых текстов

Ковалев Егор Юрьевич, студент
Херсонский государственный педагогический университет

В статье рассматриваются возможности и ограничения применения технологий искусственного интеллекта (ИИ), прежде всего больших языковых моделей (LLM), для анализа историко-правовых документов. Анализируется, как «цифровой поворот» трансформирует методологию историко-правовых исследований, а также ключевые проблемы: «гал-

люцинации», культурные искажения, контекстная ограниченность. Обосновывается использование LLM как вспомогательного инструмента для первичной обработки источников и выявления скрытых паттернов. Уделяется внимание технологии RAG (Retrieval-Augmented Generation) и значению методологии «обучаемости». В качестве методологического вклада предлагается конкретный алгоритм (пайплайн) применения RAG для историко-правового анализа. Делается вывод о формировании нового направления — цифровой герменевтики исторического источника — и предлагаются практические рекомендации. Работа носит обзорно-методический характер.

Ключевые слова: искусственный интеллект, большие языковые модели, цифровая герменевтика, историко-правовой анализ, RAG, валидация.

Digital hermeneutics: the application of artificial intelligence technologies for the analysis of historical legal texts

The article examines the possibilities and limitations of applying artificial intelligence (AI) technologies, primarily large language models (LLMs), to the analysis of historical legal documents. It analyzes how the «digital turn» is transforming the methodology of historical and legal research, addressing key issues such as «hallucinations», cultural bias, and context limitation. The use of LLMs as auxiliary tools for primary source processing and hidden pattern identification is substantiated. Attention is given to RAG (Retrieval-Augmented Generation) technology and the role of learnability methodology. As a methodological contribution, a specific pipeline for applying RAG to historical-legal analysis is proposed. The conclusion is drawn about the formation of a new direction — digital hermeneutics of the historical source — and practical recommendations are offered. The work is of a review and methodological nature.

Keywords: artificial intelligence, large language models, digital hermeneutics, historical-legal analysis, RAG, validation.

Введение

Актуальность исследования. Стремительное развитие технологий искусственного интеллекта (ИИ), особенно больших языковых моделей (Large Language Models, LLM), в 2025–2026 гг. переводит их из области футурологических дискуссий в актуальную методологическую проблему [1, с. 87]. Способность LLM обрабатывать массивы текстов, реконструировать повреждённые фрагменты и генерировать нарративы ставит перед историками, в том числе в области истории государства и права, принципиально новые вызовы: эти технологии требуют переосмысления традиционных практик работы с историческими источниками [2, с. 88].

Для историка права внедрение ИИ — это эволюция от простого инструментария к фундаментальной смене парадигмы. [5, с. 308]. Как отмечает А. Ю. Володин, процесс формализации исторических данных не является нейтральной технической операцией; это форма цифровой герменевтики, требующая критической рефлексии, с неизбежными потерями нюансов и рисками предвзятости алгоритмов [1, с. 90].

Цель и новизна исследования. Целью настоящей работы является не просто обзор существующих практик, а разработка и обоснование конкретного методологического подхода — «цифровой герменевтики» — для анализа историко-правовых текстов с использованием LLM. **Научная новизна** заключается в том, что в отличие от существующих обзорных работ, предлагается структурированный алгоритм (пайплайн) интеграции LLM в исследовательский процесс, а также формулируются критерии верификации результатов.

Задачи исследования:

- 1) классифицировать методы работы LLM с историческими правовыми текстами;
- 2) выявить ключевые методологические проблемы (галлюцинации, предвзятость, контекстная ограниченность);
- 3) предложить и детализировать применимый для историка права RAG-алгоритм;
- 4) сформулировать практические рекомендации по интеграции.

1. Технологические основы и современные практики

1.1. Классификация LLM-инструментов

Современные LLM могут применяться в историко-правовых исследованиях для решения следующих задач [4, с. 5]:

- 1. Извлечение структурированной информации** (именованные сущности, даты, названия законов).
- 2. Базовая обработка текста:** распознавание символов (OCR), нормализация, лемматизация [3].
- 3. Семантический анализ:** отслеживание изменения значения правовых понятий, выявление интертекстуальных связей [2, с. 11].
- 4. Реконструкция и перевод:** восстановление повреждённых фрагментов, перевод с архаичных языков [10, с. 8].

Особого внимания заслуживают RAG-системы (Retrieval-Augmented Generation). Они не опираются исключительно на внутренние «знания», а динамически извлекают релевантные фрагменты из заданного набора документов [10,

с. 12]. RAG-подход позволяет преодолеть ключевые ограничения генеративных моделей: галлюцинации, «чёрный ящик» и оторванность от источников [1, с. 94].

1.2. Кейс: дообучение больших языковых моделей, работающих с текстом, для исторических баз данных

Примером использования LLM в исторических исследованиях служит работа Г. В. Кнутсена, опубликованная в *Digital Scholarship in the Humanities* (2026) [2, с. 3]. На примере биографической базы данных исследователь демонстрирует, как дообученные модели LLM могут выполнять работу квалифицированного ассистента. [2, с. 5]. Ключевой вывод: open-source модели обеспечивают стабильность и воспроизводимость, критичные для долгосрочных исторических проектов [2, с. 9].

1.3. Технологии AI-распознавания

AI-распознавание — перспективное направление для работы с плохо сохранившимися источниками (берестяные грамоты, древние списки законов) [3]. В 2025 г. был представлен проект оцифровки «Дигест» Юстиниана с использованием AI-OCR. На семинаре в Институте Макса Планка в 2026 г. обсуждалось использование LLM для прослеживания рецепции римского права в японском праве [3].

2. Методологические проблемы и ограничения

2.1. Проблема галлюцинаций

«Галлюцинациями» называют генерацию правдоподобной, но ложной информации [7, с. 2]. Исследование, посвящённое выявлению галлюцинаций в юридических задачах, показало, что у ChatGPT-4 он составляет 58 %, а у LLaMA 2—88 % [7, с. 5]. Следовательно, ни один результат, полученный с помощью LLM, не может быть принят без проверки по первоисточнику.

2.2. Методологии верификации

Для борьбы с галлюцинациями предложены следующие подходы [8, с. 3]:

- 1. Методология «обучаемости» (learnability framework):** систематическая проверка надёжности извлечения информации (на корпусе османских правовых текстов) [8, с. 12].
- 2. Технология RAG:** модель опирается на конкретные документы, позволяя проверять источники [10, с. 22].
- 3. Использование open-source моделей:** обеспечивают стабильность и воспроизводимость [2, с. 8].

2.3. Культурная предвзятость и контекстная ограниченность

LLM обучаются на текстах, отражающих западные правовые нарративы, что может приводить к искажению при

анализе российских правовых институтов [11, с. 3]. Например, институт «кормления» в Московском государстве может быть ошибочно интерпретирован как «коррупция» без учёта исторического контекста. Кроме того, ограниченный размер контекстного окна LLM (у GPT-4 — около 200–300 страниц) делает невозможным анализ больших законодательных корпусов без применения RAG [2, с. 6].

2.4. Методология исследования: предлагаемый RAG-алгоритм

В отличие от обзорных работ, настоящая статья предлагает конкретный методологический алгоритм (пайплайн) для применения RAG в историко-правовых исследованиях. Алгоритм включает следующие этапы:

- 1. Формирование корпуса:** отбор и оцифровка (OCR) релевантных историко-правовых документов (законы, судебные решения).
- 2. Индексация и векторизация:** разбиение документов на фрагменты (chunks) и создание векторных представлений (embeddings) для последующего поиска.
- 3. Формулирование запроса:** исследователь задаёт вопрос на естественном языке (например, «Как менялось наказание за кражу по Соборному уложению 1649 года?»).
- 4. Поиск и генерация:** RAG-система находит наиболее релевантные фрагменты и передаёт их LLM для генерации ответа с цитированием источников.
- 5. Верификация:** исследователь проверяет все цитаты по оригинальным документам и корректирует запрос при необходимости.

Предлагаемый алгоритм (пайплайн) позволяет сохранить критическую роль человека-исследователя, используя LLM как ускоритель поиска и первичного синтеза. Этот алгоритм может быть реализован с использованием открытых инструментов (LangChain, LlamaIndex, открытые модели) и адаптирован к конкретным исследовательским задачам.

3. Перспективы интеграции и практические рекомендации

3.1. Приоритетные технологии

RAG-системы, по мнению А. Ю. Володина, выступают как мощные «цифровые лупы», берущие на себя трудоёмкие задачи обработки данных и освобождающие время для критической интерпретации [1, с. 95].

3.2. Области применения

На основе анализа выделены перспективные направления: анализ эволюции правовых понятий [1, с. 90]; сравнительно-правовой анализ рецепции институтов [3]; создание поисковых RAG-систем по оцифрованным источникам [10, с. 25].

3.3. Рекомендации для исследователя. [6, с. 509; 9, с. 17]:

1. Использовать RAG, ограничивая корпус проверенными источниками.
2. Всегда проверять ссылки по первоисточникам.
3. Документировать версию модели и параметры.
4. Отдавать предпочтение open-source моделям.
5. Не делегировать ИИ задачи интерпретации.

Заключение

Подводя итоги проведённого исследования, можно сформулировать несколько развёрнутых выводов, которые отражают как теоретическую значимость работы, так и её практические ограничения.

1. О трансформации методологии историко-правовых исследований

Технологии искусственного интеллекта, особенно большие языковые модели, уже сегодня оказывают заметное влияние на методы работы с историческими правовыми источниками. Этот процесс не сводится к простой автоматизации рутинных операций; он затрагивает эпистемологические основания дисциплины. Как показал анализ, формализация текстов, их векторизация и последующая обработка нейросетями превращают источниковедческую работу в особый вид цифровой герменевтики, где интерпретация неотделима от алгоритмических решений, принимаемых на этапе подготовки данных. Это требует от исследователя не только традиционных навыков палеографии и текстологии, но и понимания логики работы LLM, осознания их ограничений и предвзятостей. Таким образом, мы являемся свидетелями формирования нового междисциплинарного направления, которое можно условно назвать «цифровой герменевтикой исторического источника» [1, с. 87].

2. О двойственной природе LLM как исследовательского инструмента

Проведённый обзор показывает, что LLM обладают значительным потенциалом для ускорения вспомогательных операций: первичного поиска, извлечения именованных сущностей, тематического моделирования, реконструкции повреждённых фрагментов. В то же время этот потенциал может быть реализован только при строгом соблюдении двух условий: во-первых, критического отношения к результатам генерации (обязательная верификация каждой ссылки по первоисточнику), во-вторых, применения специальных методик валидации, таких как RAG, методология «обучаемости» или заземление цитат. В отсутствие обозначенных превентивных мер внедрение больших языковых моделей в исследовательский процесс чревато лавинообразной генерацией неверифицированных данных, недопустимых в сфере юриспруденции

и историко-правовых исследований, где любая интерпретационная неточность ставит под сомнение достоверность ретроспективного анализа правовых институтов.

3. О ключевых рисках и способах их минимизации

Наиболее серьёзные риски, выявленные в ходе анализа, включают:

- *галлюцинации* (достигающие 58–88 % на юридических запросах), которые требуют обязательной постгенерационной проверки;
- *культурную предвзятость* (модели, обученные на западных корпусах, навязывают чуждые правовые категории), что особенно актуально для российских исследований;
- *контекстную ограниченность* (невозможность охватить большие законодательные корпуса целиком), которая частично преодолевается с помощью RAG, но не полностью;
- *размывание ответственности* (кто отвечает за ошибки — исследователь или модель), что требует прозрачного документирования всех этапов работы с LLM.

Для минимизации этих рисков предложен комплекс мер: использование RAG-систем, предпочтение open-source моделей, фиксация версий и параметров, а также сохранение за исследователем финальной интерпретации и ответственности.

4. О методологическом вкладе работы

Основной вклад настоящей статьи заключается не в обзоре, а в разработке конкретного, воспроизводимого алгоритма (RAG-пайплайна), который переводит дискуссию с абстрактного уровня «можно ли использовать ИИ» на практический уровень «как именно это делать». Предложенный алгоритм состоит из пяти этапов — от формирования корпуса до верификации результатов — и может быть адаптирован к различным типам историко-правовых источников. Его ценность в том, что он сохраняет центральную роль человека-исследователя, превращая LLM из потенциального «соавтора» в надёжный инструмент, подчинённый задачам источниковедческого анализа.

5. Ограничения предложенного подхода

Следует признать, что предлагаемый алгоритм носит концептуальный характер и не был эмпирически апробирован на конкретном материале. Отсутствие количественной оценки его эффективности, сравнительного анализа с традиционными методами и статистической проверки результатов является существенным ограничением. Кроме того, работа не затрагивает вопросы работы с рукописными источниками, плохо поддающимися OCR, и не рассматривает этические аспекты, связанные с возможным искажением исторической памяти при машинной генерации нарративов. Эти пробелы могут быть восполнены в дальнейших исследованиях.

6. Перспективные направления дальнейших исследований

На основе проведённого анализа можно выделить несколько приоритетных направлений для будущих работ:

- создание открытых размеченных корпусов российских историко-правовых документов (законодательных актов, судебных решений, нотариальных книг) для обучения и тестирования LLM;
- разработка стандартизированных протоколов валидации для RAG-систем, учитывающих специфику исторических текстов (вариативность орфографии, наличие архаизмов, разные системы датировки);
- адаптация существующих RAG-подходов (LangChain, LlamaIndex) под задачи российской правовой истории;
- проведение контролируемых экспериментов по сравнению эффективности LLM и человека-исследователя на задачах разного уровня сложности (от поиска до интерпретации);
- изучение этических аспектов использования ИИ в историко-правовой науке, включая вопросы авторства, объективности и ответственности.

7. Практическая значимость для студента-исследователя

Для студентов, работающих над курсовыми и дипломными проектами по истории государства и права, предложенные рекомендации могут быть полезны на этапах сбора и первичной обработки источников, особенно если объём материала велик. Однако важно подчеркнуть, что LLM не заменяют традиционное источниковедение: самостоятельное прочтение первоисточников, критический анализ контекста и синтез выводов остаются фундаментом научной работы. ИИ может выступать лишь как ускоритель вспомогательных операций, но не как замена мышлению исследователя.

В целом, статья демонстрирует, что интеграция LLM в историко-правовую науку — это не вопрос выбора, а вопрос методологической грамотности. Если использовать эти технологии осознанно, с соблюдением всех процедур валидации и сохранением критической рефлексии, они могут существенно расширить исследовательские горизонты, не ставя под угрозу научную строгость и достоверность выводов.

Литература:

1. Володин А. Ю. Цифровая герменевтика исторического источника: формализация как толкование // Вестник Пермского университета. Серия «История». — 2025. — № 2. — С. 87–100. — DOI: 10.17072/2219-3111-2025-2-87-100. — Текст: непосредственный.
2. Knutsen G. W. Fine-tuning LLAMA models for historical databases: methods, challenges, and long-term implications // Digital Scholarship in the Humanities. — 2026. — DOI: 10.1093/lhc/fqag057.
3. Kawamoto Y., Iwata N., Sano T. From Roman Jurisprudence to Modern Japanese Statutes: Tracing the Reception of Law via LLMs and Generative AI [Электронный ресурс]: доклад на онлайн-семинаре Max Planck Institute for Legal History and Legal Theory. — 2026. — URL: <https://www.lhlt.mpg.de> (дата обращения: 16.06.2026).
1. Dehghani F., Dehghani R., Naderzadeh Ardebili Y., Rahnamayan S. Large Language Models in Legal Systems: A Survey // Humanities and Social Sciences Communications. — 2025. — Vol. 12, Article 1977. — DOI: 10.1057/s41599-025-05924-3.
2. Водин Д. В. Применение искусственного интеллекта как основного элемента создания информационного ресурса в исторических исследованиях // Молодой учёный. — 2025. — № 5 (556). — С. 308–311. — URL: <https://moluch.ru/archive/556/> — Текст: непосредственный.
3. Чунькова Н. А. Правовой статус искусственного интеллекта: от электронного лица до деликтоспособности // Молодой учёный. — 2026. — № 19 (622). — С. 507–510. — URL: <https://moluch.ru/archive/622/> — Текст: непосредственный.
4. Ovcharov V. Citation Grounding: Detecting and Reducing LLM Citation Hallucinations via Legal Citation Graphs // arXiv:2606.00898. — 2026. — P. 1–15.
5. Çetinkaya A. A Systems Approach to Validating Large Language Model Information Extraction: The Learnability Framework Applied to Historical Legal Texts // Information (MDPI). — 2025. — Vol. 16 (11). — P. 960. — DOI: 10.3390/info16110960.
6. Павленко Б. В. Искусственный интеллект как актуальный тренд содержания обучения информатике в условиях цифровизации // Молодой учёный. — 2023. — № 51 (498). — С. 15–18. — URL: <https://moluch.ru/archive/498/> — Текст: непосредственный.
7. Gao Y., Xiong Y., Gao X. et al. Retrieval-Augmented Generation for Large Language Models: A Survey // arXiv:2312.10997v5. — 2024. — P. 1–45.
8. Celli F., Spathulas G. Cultural Biases of Large Language Models and Humans in Historical Interpretation // arXiv:2504.02572. — 2025. — P. 1–12.

Управление безопасностью зависимостей клиентских веб-приложений: анализ риска цепочки поставок npm на примере инцидента с библиотекой axios

Назаров Владимир Игоревич, студент

Московский авиационный институт (национальный исследовательский университет)

В статье рассматривается проблема безопасности программных зависимостей клиентских веб-приложений на основе экосистемы npm. На примере инцидента с компрометацией HTTP-библиотеки axios в марте 2026 года показано, что угроза цепочки поставок для фронтенд-разработки носит системный характер. Описана методика аудита зависимостей инструментом rnpm audit, применённая к трём промышленным веб-системам. Особое внимание уделено противоречию между требованием статического аудита обновить уязвимый пакет и риском обновления внутри скомпрометированной линии релизов. Предложен набор практических рекомендаций по управлению зависимостями.

Ключевые слова: безопасность цепочки поставок, npm, rnpm, аудит зависимостей, фронтенд-разработка, axios, управление уязвимостями.

Введение

Современная фронтенд-разработка немыслима без переиспользования стороннего кода: типовое клиентское приложение на стеке React/TypeScript включает десятки прямых и сотни транзитивных зависимостей из публичного реестра npm. Безопасность приложения тем самым становится производной от безопасности всех используемых пакетов — эта проблема получила название угрозы цепочки поставок программного обеспечения.

Актуальность темы возросла в связи с серией атак на экосистему npm, в ходе которых злоумышленники получали контроль над учётными записями сопровождающих и публиковали вредоносные версии. Показателен инцидент марта 2026 года с библиотекой axios — одним из самых распространённых HTTP-клиентов для браузера и Node.js: в npm были опубликованы вредоносные версии, доставлявшие на машины разработчиков и в конвейеры непрерывной интеграции троян удалённого доступа [1, 2]. Хотя вредоносные версии находились в реестре лишь несколько часов, при объёме порядка ста миллионов ежедневных загрузок и более чем ста семидесяти тысячах зависимых пакетов радиус поражения оказался крайне велик [2].

Объект исследования

Объектом исследования являются программные зависимости клиентских веб-приложений, построенных на экосистеме npm. В качестве эмпирической базы рассматриваются три промышленных клиентских веб-приложения на стеке React/TypeScript, использующих менеджер пакетов rnpm.

Под угрозой цепочки поставок понимают компрометацию приложения через доверенные сторонние компоненты, а не через собственный код разработчика. Применительно к экосистеме npm можно выделить несколько типовых классов угроз.

Компрометация сопровождающего. Злоумышленник получает доступ к учётной записи автора пакета (через

утечку токена, фишинг или повторно используемый пароль) и публикует вредоносную версию под легитимным именем. Именно к этому классу относится инцидент с axios: вредоносный код был опубликован под официальным именем пакета, что делает атаку особенно опасной, поскольку версия проходит штатную установку без видимых признаков подмены.

Тайпсквоттинг. Публикация пакета с именем, близким к популярному, в расчёте на ошибку разработчика при установке.

Известные уязвимости (CVE). Наиболее массовый класс — это не вредоносный код, а ошибки в легитимных версиях пакетов, которым присвоены идентификаторы CVE. Для библиотеки axios, в частности, актуальны уязвимости класса подделки запроса на стороне сервера (SSRF) и инъекции в HTTP-заголовки, устранённые в более поздних версиях [3, 4].

Цель и задачи исследования

Цель работы — на основе практического опыта аудита трёх промышленных веб-приложений сформулировать методику управления безопасностью зависимостей клиентского приложения и обосновать набор практических рекомендаций. Для достижения цели решаются следующие задачи:

- 1) классифицировать основные виды угроз цепочки поставок в экосистеме npm;
- 2) описать инструменты аудита зависимостей и их ограничения;
- 3) проанализировать инцидент с axios как пример конфликта между требованием обновления и риском компрометации линии релизов;
- 4) сформулировать практические рекомендации по снижению риска.

Методы исследования

Базовым средством статического анализа зависимостей в рассматриваемой экосистеме является команда

аудита, сопоставляющая дерево установленных пакетов с базой данных известных уязвимостей. В проектах, использующих менеджер пакетов `pnpm`, применяется команда `pnpm audit`, которая по файлу блокировок (`pnpm-lock.yaml`) строит полное дерево зависимостей, включая транзитивные, и для каждого пакета проверяет наличие зарегистрированных уязвимостей с указанием уровня критичности и рекомендуемой безопасной версии [5].

Инструмент аудита обладает рядом ограничений, которые необходимо учитывать на практике. Во-первых, он выявляет только известные уязвимости, уже попавшие в базу данных, и бессилён против ранее неизвестных закладок в момент их появления. Во-вторых, рекомендация «обновиться до версии X» носит механический характер и не учитывает ни обратной совместимости, ни того, что рекомендуемая линия релизов могла быть скомпрометирована. В-третьих, значительная доля находок приходится на транзитивные зависимости, которые невозможно обновить напрямую без вмешательства в дерево разрешения версий.

Для управления версиями транзитивных зависимостей в `pnpm` предусмотрен механизм переопределений (поле `overrides` в `package.json`), позволяющий принудительно зафиксировать версию пакета во всём дереве, независимо от того, какую версию запрашивают промежуточные зависимости. Этот механизм является ключевым инструментом ручного управления риском.

Аудит каждого из трёх веб-приложений выполнялся по единой последовательности действий: фиксация исходного состояния дерева зависимостей; запуск `pnpm audit` и выгрузка отчёта с уровнями критичности и рекомендуемыми версиями; классификация находок; устранение (обновление прямых зависимостей, переопределение транзитивных через `overrides`, повторный аудит); документирование решений по уязвимостям, обновление которых сознательно отложено.

Результаты исследования

В результате аудита трёх веб-приложений выявлены уязвимости различных уровней критичности; сводные данные приведены в таблице 1.

Таблица 1. Результаты аудита зависимостей трёх веб-приложений

Проект	Найдено (до)	Critical / High / Moderate / Low	Устранено	Отложено
Веб-приложение № 1	48	1 / 22 / 22 / 3	44	4
Веб-приложение № 2	31	1 / 13 / 16 / 1	29	2
Веб-приложение № 3	19	0 / 8 / 10 / 1	19	0

Наиболее показательным результатом аудита стала ситуация вокруг библиотеки `axios`, иллюстрирующая нетривиальный характер управления риском цепочки поставок.

В ходе инцидента марта 2026 года в реестр `npm` под официальным именем `axios` были опубликованы вредоносные версии, содержавшие троян удалённого доступа [1]. Последней безопасной версией в соответствующей минорной линии оказалась версия, предшествующая скомпрометированной; именно на неё рекомендовалось откатиться, а также зафиксировать её в файле блокировок и отключить автоматическое обновление пакета, поскольку вредоносная нагрузка содержала механизм повторной попытки самообновления [1, 2].

Одновременно с этим в той же библиотеке существовали известные уязвимости (класса `SSRF` и инъекции в заголовки), устранённые лишь в более поздней минорной линии [3, 4]. Таким образом, статический аудит формирует рекомендацию обновиться вперёд (для закрытия CVE), тогда как анализ инцидента предписывает остаться на более ранней безопасной. Возникает прямое противоречие между двумя корректными по отдельности требованиями безопасности (рис. 1).

В рассматриваемых проектах было принято решение зафиксировать безопасную версию `axios`, не затронутую компрометацией, и сознательно подавить соответствующие

предупреждения аудита о CVE, задокументировав это решение. Обоснование: риск исполнения вредоносного кода при попадании в скомпрометированную линию релизов оценивался как существенно более высокий и немедленный, чем риск эксплуатации известных уязвимостей в контролируемом окружении клиентского приложения. Решение носит временный характер и подлежит пересмотру по мере стабилизации линии релизов библиотеки.

Данный пример демонстрирует ключевой тезис работы: автоматический аудит является необходимым, но недостаточным инструментом. Его рекомендации не могут применяться механически — итоговое решение требует экспертной оценки контекста, в том числе истории релизов конкретного пакета.

На основе проведённого анализа сформулированы следующие рекомендации по управлению безопасностью зависимостей клиентского веб-приложения.

Закрепление версий и контроль файла блокировок. Отказ от диапазонной записи версий для критичных зависимостей и фиксация точных версий в файле блокировок снижают риск автоматического получения скомпрометированной версии при установке. Файл блокировок должен находиться под контролем версий и проходить ревью при изменении.

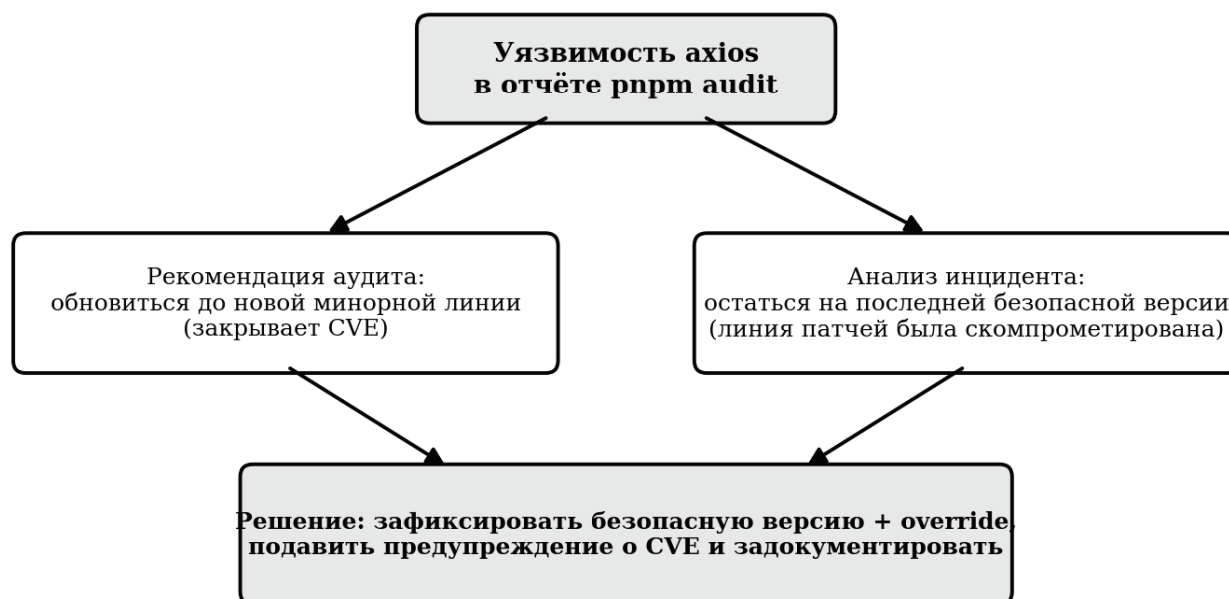


Рис. 1. Конфликт требований при управлении уязвимостью axios

Использование переопределений (overrides). Для транзитивных зависимостей, которые невозможно обновить напрямую, следует применять механизм overrides, принудительно фиксируя безопасную версию во всём дереве.

Регулярный, но осмысленный аудит. Команду аудита целесообразно встраивать в конвейер непрерывной интеграции, однако её результаты должны проходить экспертную классификацию, а не применяться автоматически. Решения о сознательном откладывании обновления необходимо документировать с указанием причины и срока пересмотра.

Отказ от автоматического обновления и реагирование на инциденты. Для пакетов, затронутых инцидентами компрометации, следует отключать автообновление и при необходимости ротировать секреты и учётные данные, которые могли быть скомпрометированы [1].

Минимизация поверхности атаки. Снижение числа прямых зависимостей и периодический пересмотр их необходимости уменьшают общий объём доверенного стороннего кода.

Вывод

В работе показано, что безопасность зависимостей является неотъемлемой частью инженерии клиентских веб-приложений, а угроза цепочки поставок носит системный характер. На примере инцидента с библиотекой axios продемонстрировано, что управление риском не сводится к механическому исполнению рекомендаций статического аудита: в ряде случаев требование закрыть известную уязвимость путём обновления вступает в прямое противоречие с риском попадания в скомпрометированную линию релизов. Предложенная методика аудита и набор практических рекомендаций (закрепление версий, переопределения, контроль файла блокировок, осмысленный аудит и отказ от автообновления) позволяют снизить риск без потери управляемости проекта. Дальнейшие направления исследования связаны с автоматизацией экспертной классификации находок аудита и интеграцией данных об инцидентах компрометации в инструменты управления зависимостями.

Литература:

1. Mitigating the axios npm supply-chain compromise. — Текст: электронный // microsoft.com: [сайт]. — URL: <https://www.microsoft.com/en-us/security/blog/2026/04/01/mitigating-the-axios-npm-supply-chain-compromise> (дата обращения: 13.06.2026).
2. Axios Supply Chain Attack. — Текст: электронный // unit42.paloaltonetworks.com: [сайт]. — URL: <https://unit42.paloaltonetworks.com/axios-supply-chain-attack/> (дата обращения: 13.06.2026).
3. CVE-2026-40175: Axios Unrestricted Cloud Metadata Exfiltration via Header Injection Chain. — Текст: электронный // GitHub Security Advisory: [сайт]. — URL: <https://github.com/axios/axios/security/advisories/GHSA-fvcv-3m26-rscq> (дата обращения: 14.06.2026).
4. CVE-2025-62718: Axios Server-Side Request Forgery and proxy bypass. — Текст: электронный // nvd.nist.gov: [сайт]. — URL: <https://nvd.nist.gov/vuln/detail/CVE-2025-62718> (дата обращения: 14.06.2026).
5. pnpm audit. — Текст: электронный // pnpm.io: [сайт]. — URL: <https://pnpm.io/cli/audit> (дата обращения: 14.06.2026).

Система поддержки управления развитием группы разработки ПО на основе имитационного моделирования

Танатаров Александр Романович, студент
Поволжский государственный технологический университет (г. Йошкар-Ола)

В статье представлен анализ системы поддержки управления развитием группы разработки программного обеспечения на основе имитационного моделирования.

Ключевые слова: СППР, разработка, имитационная модель.

Simulation-based software development group development management support system

Tanatarov Aleksandr Romanovich, student
Volga State University of Technology (Yoshkar-Ola)

The article presents an analysis of the support system for managing the development of a software development group based on simulation modeling.

Keywords: DSS, development, simulation model.

Управление развитием группы разработки программного обеспечения представляет собой сложную задачу с множеством параметров. Она характеризуется высокой степенью неопределенности и наличием множества обратных связей между техническими, организационными и человеческими факторами. Множество проектов не завершаются в срок, при этом основной причиной неудач является неэффективное управление качеством на всех этапах жизненного цикла разработки.

В отличие от традиционных проектов, ИТ-проект обладает рядом особенностей, осложняющих управление: нематериальность промежуточных результатов, высокая динамика требований, нелинейный характер зависимости качества продукта от навыков и мотивации команды. Эти факторы обуславливают необходимость применения методов моделирования для прогнозирования последствий управленческих решений и перехода от реактивного управления к проактивному.

Цель настоящей работы — анализ возможностей имитационного моделирования как инструмента поддержки принятия решений при управлении группой разработки ПО и обоснование выбора архитектуры соответствующей системы поддержки управления.

Имитационное моделирование позволяет воспроизводить поведение сложных систем во времени, учитывая их стохастическую природу и нелинейные взаимосвязи. В контексте управления ИТ-проектами выделяются три основных подхода к имитационному моделированию: системная динамика, дискретно-событийное моделирование и агентное моделирование.

Системная динамика оперирует агрегированными показателями и потоками, позволяя анализировать поведение системы на макроуровне. Этот подход эффективен для исследования долгосрочных тенденций, таких как на-

копление технического долга или влияние инвестиций в обучение на общую производительность команды. Однако системная динамика не учитывает индивидуальные особенности членов команды и их локальные взаимодействия.

Дискретно-событийное моделирование фокусируется на последовательности операций и использовании ресурсов, что делает его применимым для оптимизации процессов разработки, например, планирования спринтов или распределения задач между исполнителями.

Агентное моделирование представляет наиболее детализированный подход, позволяющий задавать индивидуальное поведение для каждого участника процесса разработки. Агенты (разработчики, тестировщики, менеджеры) могут иметь собственные состояния, память, навыки и правила взаимодействия. Это дает возможность моделировать такие феномены, различие в производительности опытных разработчиков и стажеров, процессы наставничества и передачи знаний, влияние коммуникаций внутри команды на эффективность работы, индивидуальную мотивацию и ее изменение под влиянием внешних факторов.

Как отмечается в исследовании, посвященном агентному моделированию процессов разработки ПО, «агенты в имитационной модели — это сущности, которые могут иметь поведение, память и свой набор требуемых характеристик». Это свойство делает агентное моделирование наиболее адекватным инструментом для воспроизведения социально-технической природы группы разработки.

Система поддержки принятия решений (СППР) для управления развитием группы разработки на основе имитационного моделирования должна обеспечивать:

Прогнозирование последствий управленческих решений — оценку влияния изменений в составе команды,

методологии работы или приоритетах задач на ключевые показатели проекта (сроки, бюджет, качество).

Анализ сценариев «что-если» — возможность проигрывания альтернативных стратегий управления для выбора оптимальной.

Интеграцию с реальными данными проекта — использование актуальной информации о ходе разработки для калибровки модели и повышения точности прогнозов.

Визуализацию результатов — представление данных в форме, доступной для понимания лицами, принимающими решения.

Концептуальная архитектура такой системы может быть представлена следующими компонентами:

Модуль имитационного моделирования — ядро системы, реализующее агентную модель команды разработки. Агенты обладают характеристиками: уровень квалификации, опыт, специализация, текущая загрузка, состояние (разработка, обучение, коммуникация). Взаимодействие агентов определяется правилами, отражающими реальные процессы: распределение задач, взаимопомощь, наставничество, проведение совещаний.

Модуль сбора и обработки данных — обеспечивает интеграцию с системами управления проектами (JIRA, GitLab и др.) для получения информации о реальном ходе разработки. Эти данные используются для калибровки модели и верификации прогнозов.

Модуль сценарного анализа — позволяет задавать управляющие воздействия (найм сотрудников, изменение методологии, корректировка графика) и оценивать их влияние на модельные показатели.

Модуль поддержки принятия решений — интерпретирует результаты моделирования и формирует рекомендации для руководителя проекта.

Опыт применения имитационного моделирования в управлении проектами показывает, что наиболее эффективным является гибридный подход, сочетающий агентное моделирование с элементами системной динамики. Агентное моделирование обеспечивает детали-

зацию на уровне отдельных разработчиков, а системная динамика позволяет учитывать агрегированные показатели, такие как накопление технического долга или общий объем выполненных задач.

Для реализации агентных моделей разработки ПО могут использоваться специализированные платформы, такие как AnyLogic, поддерживающая все три основных подхода к имитационному моделированию, или NetLogo, ориентированная преимущественно на агентное моделирование. Выбор платформы определяется требуемой детализацией модели, масштабом проекта и доступными вычислительными ресурсами.

Важным направлением развития является интеграция имитационных моделей с нейросетевыми технологиями для повышения точности прогнозирования и адаптации модели к изменяющимся условиям. Нейросетевые модели могут обучаться на исторических данных проекта, выявлять сложные паттерны и формировать рекомендации по управленческим воздействиям.

В заключении можно сказать, имитационное моделирование представляет собой эффективный инструмент поддержки управления развитием группы разработки ПО, позволяющий перейти от реактивного управления к проактивному прогнозированию последствий решений. Агентный подход, благодаря своей способности моделировать индивидуальное поведение и социальные взаимодействия, является наиболее адекватным для воспроизведения процессов разработки программного обеспечения.

Разрабатываемая система поддержки управления должна интегрировать имитационную модель с реальными данными проекта, обеспечивать возможность сценарного анализа и визуализацию результатов для лица, принимающего решения. Дальнейшие исследования могут быть направлены на совершенствование моделей поведения агентов, интеграцию с нейросетевыми алгоритмами и разработку методов автоматической калибровки модели на основе проектных данных.

Литература:

1. Isakov D. Агентная модель процесса разработки программного обеспечения Блог AnyLogic. — 2022. — URL: <https://www.anylogic.ru/blog/agentnaya-model-protssessa-razrabotki-programmnogo-obespecheniya/>
2. About NetLogo // NetLogo Official Website. — URL: <https://www.netlogo.org/about/>
3. ИНФОСТАРТ Анализ&Управление 1С СППР, как инструмент по внедрению, разработке и сопровождению информационных систем. URL: <https://infostart.ru/pm/1177144/>
4. Вестник Евразийской науки. Методология разработки и внедрения систем поддержки принятия управленческих решений на основе технологий искусственного интеллекта. Незамайкин В. Н.

Угрозы информационной безопасности переходного состояния при миграции автоматизированных банковских систем на отечественные системы управления базами данных

Чесноков Юлиан Валериевич, студент магистратуры

Научный руководитель: Черкасов Александр Николаевич, доктор технических наук, доцент

Кубанский государственный технологический университет (г. Краснодар)

В статье рассматриваются угрозы информационной безопасности, возникающие в процессе миграции автоматизированных банковских систем на отечественные системы управления базами данных в условиях обязательного импортозамещения программного обеспечения на значимых объектах критической информационной инфраструктуры. Показано, что наибольшую сложность и концентрацию рисков представляют АБС, построенные на двухзвенной архитектуре, в которой бизнес-логика реализована непосредственно на стороне СУБД и требует переработки значительного массива серверного кода при смене платформы. Обосновывается, что специфические риски миграции относятся не к стабильно функционирующей системе, а к её переходному состоянию, и потому не охватываются в полной мере действующим инструментарием оценки угроз, ориентированным на стабильные системы. Угрозы систематизированы по трём категориям (технологические, организационные и операционные, регуляторные) с привязкой к затрагиваемым нормативным требованиям. Установлено, что стандарты серии ГОСТ Р 57580 формируют общую методологическую основу управления рисками, однако не содержат специализированной модели для переходного состояния системы, что определяет направление дальнейшего исследования.

Ключевые слова: информационная безопасность, импортозамещение СУБД, автоматизированные банковские системы, критическая информационная инфраструктура, двухзвенная архитектура, переходное состояние, управление рисками информационной безопасности, миграция СУБД, ГОСТ Р 57580.

Технологический суверенитет финансового сектора превратился из отвлечённой стратегической установки в практическую задачу с конкретными сроками. После 2022 года ведущие зарубежные поставщики корпоративного программного обеспечения свернули работу в Российской Федерации: о прекращении деятельности заявили Oracle и SAP [1], приостановила продажи Microsoft [2], закрыла бизнес IBM [3]. Для банков это означало не просто смену поставщика, а утрату технической поддержки, прекращение поставки обновлений безопасности и невозможность легального продления лицензий на эксплуатируемые продукты. При этом одним из наиболее уязвимых звеньев оказались системы управления базами данных (СУБД): даже там, где прикладная часть автоматизированной банковской системы (АБС) была отечественной, в её основании нередко работала иностранная СУБД.

Обязательность перехода на отечественное программное обеспечение закреплена последовательным пакетом нормативных актов. Указ Президента Российской Федерации от 30.03.2022 № 166 установил запрет на использование иностранного программного обеспечения на значимых объектах критической информационной инфраструктуры (КИИ) с 1 января 2025 года [4]. Контроль за переходом кредитных организаций отнесён к полномочиям Банка России статьёй 57.5–1 Федерального закона от 10.07.2002 № 86-ФЗ [5]. Положение Банка России от 12.01.2022 № 787-П включило технологическую зависимость от иностранных вендоров в периметр управления операционными рисками [6], а Письмо Банка России от

05.02.2025 № 56–21/949 дополнительно разъяснило необходимость перехода в рамках обеспечения операционной надёжности [7]. Завершил конструкцию Федеральный закон от 07.04.2025 № 58-ФЗ, обязавший субъектов КИИ применять на значимых объектах только программное обеспечение из Единого реестра отечественного ПО [8]. В совокупности эти документы не оставляют банковскому сектору возможности отложить миграцию на неопределённый срок.

Практическая трудоёмкость перехода, однако, определяется не столько нормативными требованиями, сколько архитектурой конкретной АБС. В трёхзвенных системах прикладная логика вынесена на сервер приложений, что ослабляет привязку к конкретной СУБД и упрощает её замену. В двухзвенных системах ситуация принципиально иная: бизнес-логика размещена непосредственно в СУБД (в хранимых процедурах, функциях и триггерах) из-за чего система оказывается жёстко связанной со средой исполнения. Опыт миграции сильно связанных систем такого типа показывает, что перенос данных составляет лишь малую часть работы, тогда как основной объём приходится на переработку серверного кода под иной диалект SQL и иной процедурный язык [9; 10]. Именно поэтому двухзвенные АБС представляют собой наиболее сложный и одновременно показательный случай с точки зрения рисков информационной безопасности, и настоящая работа сосредоточена на этом классе систем.

Часть кредитных организаций в качестве целевой платформы выбрала PostgreSQL и построенные на его основе отечественные продукты. Такой выбор обеспечил бы-

стрый старт, но привнёс собственные риски: применение программного обеспечения с открытым исходным кодом в системах банковского уровня сопряжено с возможностью наследования уязвимостей базового ядра. Цель настоящей статьи — выделить и охарактеризовать угрозы информационной безопасности, специфичные именно для переходного состояния двухзвенной АБС, и показать, что действующий методический инструментарий не охватывает их в полной мере. Методическую основу составили анализ нормативно-правовых актов в области импортозамещения и защиты информации, а также систематизация угроз по источнику их возникновения.

Ключевая особенность рассматриваемой задачи состоит в том, что риски миграции относятся не к стабильно работающей системе, а к процессу её преобразования. На всём протяжении перехода АБС не тождественна ни исходной, ни целевой конфигурации: она пребывает в промежуточном, неустойчивом состоянии, в котором меры защиты реализованы лишь частично, а контуры безопасности старой и новой платформ частично перекрываются. Конфигурация системы при этом меняется по ходу самого проекта, целевая платформа на ранних этапах эксплуатируется в условиях ограниченного опыта, а часть рисков носит временный характер и исчезает по завершении перехода. Эти свойства принципиально отличают переходное состояние от стабильного функционирования и не позволяют описать его однократно построенной моделью угроз.

Действующий инструментарий оценки угроз ориентирован на иную ситуацию. Методический документ ФСТЭК России по оценке угроз безопасности информации регламентирует построение модели угроз для систем с определённой, устойчивой конфигурацией [11] и не предусматривает отдельного порядка анализа угроз, возникающих в самом процессе смены платформы. Между тем импортозамещение не устраняет риски информационной безопасности, а перестраивает их структуру: возникают новые уязвимости, сложности интеграции и ошибки, характерные именно для переходного периода [12]. Масштаб задачи усугубляет дефицит апробированных практик, миграция корпоративных систем такого класса на отечественные СУБД оценивается отраслевыми специалистами как беспрецедентная по сложности, а массовый практический опыт её решения фактически отсутствует [13]. Совокупность этих обстоятельств делает обоснованным рассмотрение переходного состояния как самостоятельного объекта анализа. Ниже специфические угрозы рассмотрены по трём группам — технологические, организационные и операционные, регуляторные.

Технологические угрозы порождаются различиями между исходной и целевой платформами на уровне архитектуры, диалекта SQL и механизмов исполнения кода. В двухзвенной АБС основным их носителем выступает серверный код, в котором реализована бизнес-логика финансовых операций. При переходе на PostgreSQL этот код подлежит конвертации из процедурного языка Oracle

в PL/pgSQL, причём автоматизированные средства переноса обеспечивают синтаксическую корректность результата, но не гарантируют его смысловую эквивалентности. Сконвертированная процедура может выполняться без ошибок и при этом возвращать значения, отличные от исходных. Применительно к расчётным операциям банка подобное расхождение означает прямую угрозу искажения финансовых результатов, способную долго оставаться незамеченной.

Конкретные источники смысловых расхождений хорошо известны из практики переноса. Объекты, выполняющиеся с правами владельца схемы, при конвертации требуют явного воспроизведения этого режима и фиксации пути поиска схем. Пропуск любого из этих шагов либо открывает клиентскому коду прямой доступ к таблицам в обход контролируемого интерфейса, либо создаёт условия для подмены схемы. Автономные транзакции, широко используемые в банковских системах для журналирования и фиксации аудит-записей, не имеют прямого аналога в ряде редакций PostgreSQL, и их некорректная замена приводит к откату аудит-записи вместе с основной операцией, нарушая целостность журнала. Переменные уровня пакета, хранящие состояние в пределах сессии, при переносе требуют воспроизведения альтернативными средствами, а ошибка в реализации способна привести к разделению состояния между сессиями разных клиентов. Свой вклад вносят и менее очевидные различия: в арифметике дат, в поведении логического типа, в обработке неопределённых значений при соединениях таблиц, — каждое из которых в финансовых расчётах обобщается ошибкой результата.

Помимо угроз целостности логики, переход сопряжён с риском снижения производительности. Замена зрелого коммерческого решения на платформу с иными планировщиком запросов и механизмами блокировок может привести к деградации отклика под нагрузкой, прежде всего в пиковые периоды банковских операций, что напрямую угрожает непрерывности обслуживания. Наконец, происхождение целевой платформы от открытого ядра формирует риск унаследованных уязвимостей. Поддержание минимального уровня защиты на протяжении всего перехода, включая контроль подобных уязвимостей, относится к базовым требованиям к защите информации финансовых организаций [14].

Эта группа угроз связана с человеческим фактором, зрелостью эксплуатационных процессов и необходимостью сохранять непрерывность деятельности банка. Первым источником риска выступает недостаточная готовность организации к эксплуатации новой СУБД: нехватка специалистов с релевантным опытом, неполнота эксплуатационной документации, неотработанность процедур настройки безопасности и восстановления после сбоев. На ранних этапах перехода персонал вынужден действовать в условиях ограниченного опыта, что повышает вероятность ошибок конфигурации. Самостоятельную угрозу образует и процесс переноса данных, в ходе которого воз-

можны нарушение целостности, потеря записей и некорректное применение защитных механизмов.

Период параллельной эксплуатации заслуживает отдельного внимания как наиболее концентрированное выражение переходного состояния. Невозможность одноментного перевода монолитной АБС на новую платформу вынуждает банк эксплуатировать обе СУБД одновременно, и именно в это время контур защиты оказывается удвоенным и максимально размытым. Одни и те же пользователи работают с обеими платформами, модели разграничения доступа которых построены на различных принципах, что создаёт риск расхождения политик: учётная запись, строго ограниченная в исходной системе, может получить избыточные права в целевой вследствие ошибки переноса. Поддержание согласованности пересекающихся наборов данных через механизмы репликации или двойной записи порождает риск рассинхронизации: сбой фиксации на одной платформе при успешной фиксации на другой ведёт к расхождению состояний, а в худшем случае — к двойному проведению или потере транзакции. Раздельные журналы аудита двух платформ с различными форматами и уровнями детализации затрудняют восстановление полной последовательности событий при расследовании инцидента, а сами каналы межплатформенного взаимодействия расширяют поверхность атаки и, будучи временными, нередко проектируются с меньшим вниманием к защите, чем основные компоненты.

Снизить остроту перечисленных угроз позволяет вынесение предмиграционного аудита и оценки рисков перехода в самостоятельный этап, предшествующий активной фазе. Предварительная инвентаризация критически важных компонентов и оценка операционных, финансовых и регуляторных последствий перехода рассматриваются как необходимое условие управляемости проекта [15], поскольку ошибки планирования, допущенные на этом этапе, проявляются позднее, когда стоимость их устранения существенно возрастает.

Регуляторные угрозы реализуются не через прямое воздействие на систему, а через разрыв между декларируемым и фактическим уровнем защищённости. Применение СУБД в сертифицированной редакции является необходимым, но недостаточным условием соответствия требованиям регулятора: сертификат подтверждает свойства самого продукта, но не корректность его применения в конкретной системе [16]. Ошибки конфигурации защитных механизмов, отсутствие актуальных обновлений безопасности или отклонение от эксплуатационной документации создают ситуацию формального соответствия при фактической незащищённости — положение тем более опасное, что оно не выявляется стандартными процедурами формального контроля.

Вторая регуляторная угроза связана с аттестацией. Смена программно-технической платформы значимого объекта КИИ влечёт необходимость повторной оценки соответствия и аттестации информационной системы [17]. Этот процесс требует времени и ресурсов, а период

между завершением технической миграции и получением документального подтверждения соответствия, в течение которого система уже обрабатывает реальные данные, представляет собой самостоятельную зону регуляторного риска.

Управление перечисленными угрозами опирается на отраслевую нормативную базу, прежде всего на стандарты серии ГОСТ Р 57580. ГОСТ Р 57580.1–2017 задаёт базовый состав организационных и технических мер защиты, образующий нижнюю границу требований, которую недопустимо нарушать даже в переходный период [14]. ГОСТ Р 57580.3–2022 формирует методологию управления риском реализации информационных угроз, описывая его как непрерывный цикл из стадий планирования, реализации, контроля и совершенствования [18]; эта цикличность концептуально согласуется с проектным характером миграции и допускает проекцию стадий стандарта на этапы перехода. ГОСТ Р 57580.4–2022 конкретизирует меры обеспечения операционной надёжности и устанавливает требования к повторной аттестации после изменения системы [17].

Теоретическую опору такого подхода составляет ряд исследований. Моделирование угроз рассматривается как основа выбора адекватных мер защиты и минимизации рисков [19], а управление информационной безопасностью обоснованно строится на системной оценке рисков, а не на простом перечислении защитных мероприятий [20]. Применительно к импортозамещению подчёркивается, что переход на отечественные решения следует трактовать не как замену отдельных продуктов, а как изменение всей модели обеспечения безопасности бизнеса [21]; при этом известны и примеры системных моделей информационной безопасности, разработанных для объектов кредитно-финансовой сферы [22]. Эти работы формируют общую методологическую рамку, однако ни одна из них не адаптирована к специфике переходного состояния системы.

В этом и состоит методологический пробел. Все рассмотренные стандарты и подходы ориентированы на защиту стабильно функционирующих систем с устойчивым контуром безопасности. Ни один из них не предлагает специализированной модели управления рисками для динамического переходного состояния — периода, в котором конфигурация защиты нестабильна, контуры старой и новой среды пересекаются, а часть рисков существует лишь временно. Применение положений серии ГОСТ Р 57580 обеспечивает нормативное соответствие и общую основу управления рисками, но требует содержательной адаптации к процессу перехода.

Миграция автоматизированных банковских систем на отечественные СУБД представляет собой не технический, а стратегический процесс, в котором тесно переплетены вопросы информационной безопасности, нормативного соответствия и непрерывности критически важных операций. Сложность задачи существенно зависит от архитектуры: для двухзвенных систем с бизнес-логикой на

стороне СУБД миграция оказывается принципиально более трудоёмкой и сопряжена с повышенной концентрацией рисков в переходный период.

Проведённая систематизация показывает, что специфические угрозы миграции относятся к процессу преобразования системы, а не к её стабильному состоянию, и потому не охватываются в полной мере стандартным инструментарием оценки угроз. Действующие стандарты серии

ГОСТ Р 57580 задают общую модель управления рисками и базовый состав мер защиты, но не содержат специализированного подхода к управлению рисками переходного состояния. Устранение этого пробела — разработка адаптированной, структурированной по фазам миграции модели обеспечения информационной безопасности АБС — представляет собой самостоятельную научную задачу и составляет предмет дальнейшего исследования автора.

Литература:

1. SAP и Oracle заявили об уходе с российского рынка. — Текст: электронный // i-ias: [сайт]. — URL: <https://i-ias.ru/press/publikatsii-v-smi/a5rczzo3a1-sap-i-oracle-zayavili-ob-uhode-s-rossiis/> (дата обращения: 19.06.2026).
2. ТАСС Microsoft приостанавливает продажи в России / ТАСС. — Текст: электронный // tass.ru: [сайт]. — URL: <https://tass.ru/ekonomika/18874539> (дата обращения: 19.06.2026).
3. IBM уничтожила свой бизнес в России. — Текст: электронный // CNews: [сайт]. — URL: https://www.cnews.ru/news/top/2022-06-07_ibm_unichtozhila_svoj_biznes (дата обращения: 19.06.2026).
4. О мерах по обеспечению технологической независимости и безопасности критической информационной инфраструктуры Российской Федерации: Указ Президента Российской Федерации от 30.03.2022 № 166. — Текст: электронный // Гарант: [сайт]. — URL: <https://www.garant.ru/products/ipo/prime/doc/404395544/> (дата обращения: 19.06.2026).
5. О Центральном банке Российской Федерации (Банке России): Федеральный закон от 10.07.2002 № 86-ФЗ. — Текст: электронный // КонсультантПлюс: [сайт]. — URL: https://www.consultant.ru/document/cons_doc_LAW_37570/ (дата обращения: 19.06.2026).
6. Об обязательных для кредитных организаций требованиях к операционной надёжности при осуществлении банковской деятельности в целях обеспечения непрерывности оказания банковских услуг: Положение Банка России от 12.01.2022 № 787-П. — Текст: Гарант // garant.ru: [сайт]. — URL: <https://www.garant.ru/products/ipo/prime/doc/411381203/> (дата обращения: 19.06.2026).
7. О предоставлении разъяснений требований по переходу на отечественное ПО: Письмо Банка России от 05.02.2025 № 56–21/949. — Текст: электронный // КонсультантПлюс: [сайт]. — URL: https://www.consultant.ru/document/cons_doc_LAW_502581/ (дата обращения: 19.06.2026).
8. Федеральный закон от 07.04.2025 № 58-ФЗ «О внесении изменений в Федеральный закон «О безопасности критической информационной инфраструктуры Российской Федерации»» [Электронный ресурс]. — URL: https://www.consultant.ru/document/cons_doc_LAW_413177/ (дата обращения: 19.06.2026).
9. Малых, В. Л. Миграция с oracle на postgresql сильно связанной МИС Интерин / В. Л. Малых, А. Н. Калинин. — Текст: непосредственный // Программные системы: теория и приложения. — 2022. — № 4 (55).
10. Бельшев, Д. В. Опыт импортозамещения в медицинской информационной системе «Интерин Promis Alpha» / Д. В. Бельшев. — Текст: непосредственный // Программные системы: теория и приложения. — 2022. — № 4 (55).
11. Методика оценки угроз безопасности информации: методический документ от 05.02.2021. — Текст: электронный // fstec.ru: [сайт]. — URL: <https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-5-fevralya-2021-g> (дата обращения: 19.06.2026).
12. Верещагин, И. Ю. Современные угрозы и риски информационной безопасности корпоративных систем в условиях импортозамещения / И. Ю. Верещагин. — Текст: непосредственный // Вестник евразийской науки. — 2024. — № 4S.
13. Как банки готовятся к переходу на отечественные СУБД для хранилищ данных? / Lab Intersoft. — Текст: электронный // iso.ru: [сайт]. — URL: <http://iso.ru/ru/press-center/publications/03683-Kak-banki-gotoviyatsya-k-perehodu-na-otechestvennye-SUBD-dlya-hra.phtml> (дата обращения: 19.06.2026).
14. ГОСТ Р 57580.1–2017. Безопасность финансовых (банковских) операций. Защита информации финансовых организаций. Базовый состав организационных и технических мер. — Текст: электронный // cntd.ru: [сайт]. — URL: <https://docs.cntd.ru/document/1200146534> (дата обращения: 19.06.2026).
15. Николашин М. В., Белаш В. Ю. О реализации требований информационной безопасности в условиях импортозамещения. [Электронный ресурс] URL: <https://www.dnevniknauki.ru> (дата обращения: 19.05.2026).
16. Реестр сертифицированных средств защиты информации. — Текст: электронный // fstec.ru: [сайт]. — URL: <https://reestr.fstec.ru/reg3> (дата обращения: 19.06.2026).
17. ГОСТ Р 57580.4–2022. Безопасность финансовых (банковских) операций. Обеспечение операционной надёжности. Базовый состав организационных и технических мер. — Текст: электронный // cntd.ru: [сайт]. — URL: <https://docs.cntd.ru/document/1200194765> (дата обращения: 19.06.2026).

18. ГОСТ Р 57580.3–2022. Безопасность финансовых (банковских) операций. Управление риском реализации информационных угроз и обеспечение операционной надёжности. Общие положения. — Текст: электронный // cntd.ru: [сайт]. — URL: <https://docs.cntd.ru/document/1200194766> (дата обращения: 19.06.2026).
1. Баранкова, И. И. Минимизация рисков информационной безопасности на основе моделирования угроз безопасности / И. И. Баранкова, У. В. Михайлова, М. В. Афанасьева. — Текст: непосредственный // ОмГТУ. — 2019. — № 4.
2. Максименко, В. Н. Основные подходы к анализу и оценке рисков информационной безопасности / В. Н. Максименко, Е. В. Ясюк. — Текст: непосредственный // Экономика и качество систем связи. — 2017. — № 2 (4).
3. Витязев Г. Г. Информационная безопасность бизнеса в контексте импортозамещения // Вестник науки и образования. 2018. № 9 (45). URL: <https://cyberleninka.ru/article/n/informatsionnaya-bezopasnost-biznesa-v-kontekste-importozamescheniya> (дата обращения: 19.06.2026).
4. Козьминых, С. И. Моделирование обеспечения информационной безопасности объекта кредитно-финансовой сферы / С. И. Козьминых. — Текст: непосредственный // Финансы: теория и практика. — 2018. — № 5.

Сравнительный анализ методов верификации программного обеспечения в распределенных микросервисных системах

Шамжуров Иван Александрович, студент
Московский государственный технологический университет «СТАНКИН»

В статье рассматриваются проблемы верификации распределенного программного обеспечения, построенного на базе микросервисной архитектуры. Автор проводит детальный сравнительный анализ модульного, интеграционного и сквозного уровней тестирования по ключевым критериям эффективности. На основе предложенного многокритериального подхода научно обосновывается выбор интеграционного уровня тестирования как наиболее сбалансированного решения для снижения рисков интеграционных дефектов при сохранении высокой скорости непрерывной поставки программного обеспечения (CI/CD).

Ключевые слова: автоматизация тестирования, микросервисная архитектура, интеграционные тесты, верификация программного обеспечения, многокритериальный анализ, CI/CD.

1. Введение

Массовый переход ИТ-индустрии от монолитных архитектурных решений к микросервисным обеспечивает гибкость, масштабируемость и отказоустойчивость систем. Однако распределенный характер приложений существенно усложняет процессы их верификации. В микросервисной парадигме бизнес-логика оказывается рассредоточенной по изолированным узлам, взаимодействующим посредством сетевых протоколов и независимых хранилищ данных.

Практика разработки распределенных систем на отечественных ИТ-предприятиях показывает, что большинство критических дефектов локализуется не внутри изолированных компонентов, а в межсервисных контрактах, схемах СУБД и сетевых интерфейсах. Это обуславливает необходимость формирования обоснованного подхода к выбору стратегии автоматизированного тестирования.

2. Уровни верификации распределенного ПО

Для контроля качества программных систем традиционно рассматриваются три уровня автоматизированного тестирования:

Модульное (Unit) тестирование. Направлено на проверку изолированных единиц кода (классов, методов) в полной изоляции от внешнего окружения. Сетевые вызовы и обращения к базам данных подменяются виртуальными эмуляторами («заглушками»).

Интеграционное (Integration) тестирование. Фокусируется на проверке корректности взаимодействия между несколькими компонентами. Интеграционный тест проверяет способность сервиса корректно обрабатывать запросы к своему API и работать с реальной физической СУБД.

Сквозное (End-to-End, E2E) тестирование. Проверяет работоспособность всей распределенной системы целиком, полностью имитируя поведение конечного пользователя.

Каждый уровень обладает уникальными характеристиками стоимости разработки, скорости выполнения и достоверности результатов.

3. Методология оценки и критерии сравнения

Для обоснования выбора уровня верификации системы кадрового учета ИТ-предприятия был определен набор из пяти ключевых критериев:

- Критерий 1 (Скорость): быстродействие обратной связи (время прогона всего пакета тестов).
- Критерий 2 (Стабильность): отсутствие ложных срабатываний из-за сетевых задержек и внешней инфраструктуры.
- Критерий 3 (Простота): минимальная стоимость поддержки и написания сценариев.
- Критерий 4 (Покрытие интеграции): глубина выявления дефектов сетевого взаимодействия и схем СУБД.
- Критерий 5 (Реалистичность): степень приближенности к продуктовой среде эксплуатации.

Каждый метод оценивался по трехбалльной шкале: 0 — не соответствует критерию, 1 — частично соответствует, 2 — полностью соответствует. Общая оценка рассчитывалась путем прямого суммирования баллов по всем критериям.

Таблица 1. Сравнительный анализ уровней тестирования

Критерий	Модульное тестирование	Интеграционное тестирование	Сквозное тестирование
К1. Скорость выполнения	2	1	0
К2. Надежность результатов	2	1	0
К3. Стоимость поддержки	2	1	0
К4. Обнаружение дефектов интеграции	0	2	2
К5. Приближенность к реальной среде	0	1	2
Итог	6	6	4

4. Обсуждение результатов

Анализ таблицы показывает, что модульное и интеграционное тестирование набрали равное итоговое количество баллов (по 6 баллов), однако структура их полезности принципиально отличается.

Модульные тесты идеальны по скорости, стабильности и стоимости поддержки, но они абсолютно неэффективны для обнаружения интеграционных ошибок (0 баллов). Они не могут гарантировать, что SQL-запрос успешно выполнится на реальной СУБД, или что сетевой JSON-контракт не вызовет падения веб-сервера.

Сквозные тесты (E2E) обладают максимальной достоверностью, но из-за медлительности, нестабильности внешних сетевых интерфейсов и высокой стоимости сопровождения они не могут эффективно применяться на каждом этапе непрерывной сборки ПО (CI/CD).

Интеграционный уровень тестирования является наиболее сбалансированным решением. Он частично уступает модульным тестам в скорости, но полностью закрывает риски межкомпонентных сбоев. Средние показатели скорости и реалистичности интеграционных тестов могут быть существенно улучшены программным путем — за счет применения современных средств контейнеризации (Docker) и библиотек динамического управления инфраструктурой (Testcontainers). Это позволяет разворачивать изолированные экземпляры баз данных PostgreSQL на лету прямо в процессе выполнения тестов.

5. Заключение

В ходе исследования проведен сравнительный анализ уровней верификации распределенного программного обеспечения. Доказано, что для микросервисов, критически зависящих от сохранности и согласованности данных, интеграционное тестирование является единственным подходом, гарантирующим защиту от инфраструктурных дефектов и поломок API-контрактов. На основе проведенного анализа интеграционный уровень верификации выбран как оптимальный для программной реализации на ИТ-предприятии.

Литература:

1. Басс Л., Клементс П., Казман Р. Архитектура программного обеспечения на практике. — СПб.: Питер, 2022. — 576 с.
2. Мартин Р. Чистый код: создание, анализ и рефакторинг. — СПб.: Питер, 2019. — 464 с.

3. Фаулер М. Шаблоны корпоративных приложений. — М.: Вильямс, 2018. — 544 с.
4. Ларичев О. И. Теория и методы принятия решений. — М.: Логос, 2002. — 392 с.
5. Хориков В. Принципы юнит-тестирования. — СПб.: Питер, 2021. — 320 с.

Проектирование концептуальной архитектуры и сценариев взаимодействия в системах интеграционного тестирования

Шамжуров Иван Александрович, студент
Московский государственный технологический университет «СТАНКИН»

В данной статье рассматриваются вопросы системного анализа и проектирования автоматизированных комплексов для интеграционного тестирования микросервисных приложений. Автор детально описывает прецеденты использования системы для различных ролей участников процесса разработки программного обеспечения, а также предлагает платформу-независимую компонентную архитектуру тестового окружения. Описано моделирование жизненного цикла тестовых сценариев с помощью диаграмм состояний.

Ключевые слова: системный анализ, проектирование ПО, концептуальная архитектура, диаграмма прецедентов, автоматизированное тестирование, жизненный цикл теста.

1. Введение

Разработка надежного программного обеспечения в условиях микросервисной архитектуры требует создания гибких и легко масштабируемых систем автоматического контроля качества. Написание автотестов без предварительного проектирования их архитектуры приводит к возникновению «хрупких» тестов, которые жестко привязаны к конкретному окружению, часто ломаются и требуют высоких затрат на поддержку.

Целью настоящей работы является разработка концептуальной платформу-независимой модели комплекса интеграционного тестирования, формализация сценариев взаимодействия пользователей с системой и построение модели состояний жизненного цикла интеграционного теста.

2. Анализ ролей и прецедентов использования

Проектируемый комплекс интеграционного тестирования является междисциплинарным инструментом, объединяющим задачи различных участников команды разработки программного обеспечения. В рамках системного анализа были выделены три ключевые роли (актера) и прецеденты их взаимодействия с системой:

Разработчик (Developer). Основная задача — написание интеграционных сценариев параллельно с созданием бизнес-логики. Разработчик осуществляет локальный запуск тестов для мгновенной проверки работоспособности своего кода перед отправкой изменений в общий репозиторий, а также анализирует детальные отчеты и логи в случае обнаружения дефектов.

Специалист по контролю качества (QA Engineer). Отвечает за расширение тестового покрытия, параметризацию сценариев, подготовку тестовых наборов данных и анализ регрессионных ошибок на системном уровне.

Инженер по автоматизации инфраструктуры (DevOps/Infrastructure Engineer). Интегрирует разработанный комплекс в общий конвейер непрерывной доставки (CI/CD), настраивает автоматические триггеры запуска и обеспечивает выделение вычислительных ресурсов под временные тестовые контейнеры.

Разработанная модель прецедентов гарантирует, что система предоставляет удобные интерфейсы управления для каждого участника жизненного цикла разработки.

3. Концептуальная компонентная архитектура

Для обеспечения гибкости и независимости от конкретных языков программирования и баз данных была спроектирована концептуальная платформу-независимая архитектура тестового комплекса. Система структурно разделена на два изолированных контура:

Тестовое окружение (Test Environment). Содержит модули управления выполнением тестов:

Модуль управления жизненным циклом — координирует запуск всей сессии тестирования и сбор результатов.

Модуль оркестрации окружения — автоматически инициализирует и останавливает инфраструктурные зависимости.

Механизм очистки данных — отвечает за мгновенное приведение баз данных в исходное состояние перед каждым тестом.

Тестовый клиент — имитирует вызовы к прикладному программному интерфейсу (API).

Изолированная среда выполнения (Execution Environment). Представляет собой временную песочницу, в которой разворачиваются:

Контур приложения — тестируемый микросервис со всеми своими конфигурациями.

Контур данных — полноценная тестовая база данных, изолированная от продуктовых хранилищ.

Такое разделение позволяет легко масштабировать систему и заменять отдельные модули (например, переходить на другую СУБД) без переработки ядра тестового комплекса.



Рис. 1. Диаграмма компонентов

4. Моделирование состояний и жизненного цикла теста

Для обеспечения предсказуемости и стабильности результатов прогона автотестов был детально смоделирован их жизненный цикл. С помощью диаграммы состояний выделены следующие ключевые фазы системы:

Ожидание. Тестовая сборка помещается в очередь непрерывной интеграции и ожидает выделения свободного сборочного агента.

Подготовка. Происходит автоматическое развертывание изолированных контейнеров с базой данных и запуск веб-сервера с тестируемым микросервисом. Настраиваются необходимые заглушки для внешних систем.

Выполнение. Тестовый клиент последовательно отправляет запросы к микросервису, а база данных очищается перед каждым шагом для соблюдения принципа независимости тестов.

Ветвление (Успех / Неудача). В зависимости от прохождения проверок формируется либо успешный статус, либо подробный отчет с описанием ошибок и трассировкой стека для разработчиков.

Отчет и завершение. Генерируются итоговые файлы результатов тестирования в стандартных форматах, после чего все временные контейнеры утилизируются для освобождения ресурсов сервера.

5. Заключение

Проектирование концептуальных моделей на платформо-независимом уровне позволило заложить надежный фундамент для последующей программной реализации автотестов. Разработанная компонентная архитектура и формализованный жизненный цикл тестов позволяют исключить взаимное влияние тестов друг на друга, гарантируют стабильность результатов и обеспечивают легкую интеграцию комплекса в конвейеры непрерывной поставки ИТ-предприятия.

Литература:

1. Коберн А. Современные методы описания функциональных требований к системам. — М.: Лори, 2019. — 263 с.
2. Ньюмен С. Создание микросервисов. — СПб.: Питер, 2021. — 416 с.
3. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб.: Питер, 2020. — 368 с.

Обоснование и формирование технологического стека автоматизации интеграционного тестирования распределенных приложений на платформе.NET

Шамжуров Иван Александрович, студент
Московский государственный технологический университет «СТАНКИН»

В статье рассматриваются вопросы выбора и обоснования технологических решений для создания стабильной и производительной инфраструктуры интеграционного тестирования. Автор проводит сравнительный анализ популярных тестовых фреймворков для платформы.NET, а также исследует применение современных инструментов контейнеризации баз данных на лету и механизмов интеллектуальной очистки состояния хранилищ данных. Представлен научно обоснованный стек технологий, минимизирующий накладные расходы на инфраструктуру.

Ключевые слова: автоматизированное тестирование, тест-раннер, контейнеризация, Testcontainers, Respawn, PostgreSQL, .NET.

1. Введение

Развертывание эффективного конвейера непрерывной интеграции (CI/CD) требует от разработчиков подбора инструментов тестирования, способных работать быстро, стабильно и без ложных срабатываний. Традиционные подходы к тестированию распределенных систем часто сталкиваются с дилеммой: использовать медленные и нестабильные общие тестовые базы данных («стейджинг») или применять нереалистичные эмуляторы баз данных в оперативной памяти («In-Memory»).

Для построения надежного комплекса верификации микросервиса кадрового учета крупного ИТ-предприятия потребовалось сформировать технологический стек, обеспечивающий стопроцентную реалистичность окружения при минимальном времени выполнения тестов.

2. Сравнительный анализ тестовых фреймворков

Первым этапом формирования технологического стека является выбор оркестратора тестовых сценариев (тест-раннера). В экосистеме.NET исторически доминируют три решения: MSTest, xUnit и NUnit. Сравнение проводилось по пятибалльной шкале по пяти критериям, важным для промышленной разработки:

- Гибкость параметризации: возможность легкой передачи сложных наборов данных в тестовые методы.
- Управление жизненным циклом: наличие удобных механизмов подготовки и утилизации ресурсов (фикстур).
- Совместимость с Docker-инфраструктурой: стабильность работы в изолированных агентах сборки.
- Простота конфигурирования: отсутствие избыточного шаблонного кода.
- Популярность и поддержка: зрелость сообщества и частота обновлений.

По результатам анализа был выбран фреймворк NUnit. Он предоставляет наиболее мощные встроенные атрибуты для параметризации тест-кейсов (например, [TestCaseSource]), а также гибкую иерархию фикстур ([SetUp], [OneTimeSetUp]), что критически важно для управления контейнеризованными зависимостями.

3. Решение проблемы изоляции данных через Testcontainers

Ключевым требованием к интеграционным тестам является их изолированность — выполнение одного теста не должно влиять на результаты других. Использование баз данных в оперативной памяти (например, EF Core In-Memory или SQLite) не позволяет проверить специфичные

Таблица 1. Сравнение фреймворков тестирования для .NET

Критерий	NUnit	xUnit	MSTest
Интеграция с .NET-экосистемой	4	4	5
Поддержка асинхронных тестов	4	5	3
Параметризация тестов	5	4	3
Интеграция с CI/CD	5	4	4
Сообщество и документация	5	4	3
Итог	23	21	18

для целевой СУБД (PostgreSQL) особенности: триггеры, типы данных (например, UUID, JSONB), каскадные удаления и транзакционную логику.

Для решения этой проблемы было обосновано внедрение технологии Testcontainers для .NET. Данная библиотека предоставляет программный интерфейс для управления Docker-контейнерами непосредственно из кода тестов.

В процессе инициализации тестового класса библиотека Testcontainers обращается к локальному Docker-демону сборочного агента, скачивает официальный легковесный образ PostgreSQL и запускает чистый экземпляр СУБД на динамически выделенном свободном порту. По завершении выполнения всех тестов контейнер автоматически уничтожается. Это гарантирует стопроцентную идентичность продуктовой базе данных при полной изоляции среды тестирования.

4. Оптимизация времени прогона с помощью библиотеки Respawn

Несмотря на преимущества использования Docker, запуск контейнеров и применение миграций базы данных для каждого отдельного теста занимают значительное время (от нескольких секунд до минуты), что делает запуск полного пакета тестов неприемлемо долгим для контейнеров CI/CD.

Для оптимизации быстродействия была внедрена библиотека Respawn. Ее ключевая особенность заключается в том, что контейнер СУБД и схема таблиц разворачи-

ваются всего один раз перед запуском всего тестового класса.

Перед выполнением каждого индивидуального тестового метода библиотека Respawn осуществляет интеллектуальный сброс данных:

- Она считывает системный каталог PostgreSQL для построения графа зависимостей и внешних ключей.
- Отключает проверки ограничений целостности на время очистки.
- Выполняет быструю очистку пользовательских таблиц с помощью команд DELETE или TRUNCATE.
- Восстанавливает проверки ограничений.

Этот процесс занимает от 5 до 15 миллисекунд, сохраняя при этом структуру таблиц и примененные миграции нетронутыми. В результате обеспечивается высокое быстродействие тестов без потери их изоляции.

5. Заключение

Сформированный технологический стек на базе целевой платформы .NET 8, тест-раннера NUnit, инструментов контейнеризации Testcontainers и библиотеки оптимизации Respawn представляет собой высокоэффективное синергетическое решение. Оно позволяет проводить интеграционное тестирование микросервисов в абсолютно реалистичном окружении реальной СУБД PostgreSQL, затрачивая на очистку данных сотые доли секунды. Данный стек рекомендован к внедрению на профильных ИТ-предприятиях.

Литература:

1. Дейт К. Дж. Введение в системы баз данных. — М.: Вильямс, 2019. — 1328 с.
2. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. — М.: Вильямс, 2017. — 1440 с.
3. Ньюмен С. Создание микросервисов. — СПб.: Питер, 2021. — 416 с.
4. Хориков В. Принципы юнит-тестирования. — СПб.: Питер, 2021. — 320 с.

Объектно-ориентированное проектирование и реализация структуры классов для автоматизированного интеграционного тестирования

Шамжуров Иван Александрович, студент
Московский государственный технологический университет «СТАНКИН»

В статье рассматриваются вопросы практической программной реализации объектной модели автоматизированного тестового комплекса на платформе .NET 8. Автор исследует архитектурные шаблоны проектирования, используемые для построения гибкой, масштабируемой и изолированной структуры классов интеграционных автотестов. В работе подробно описывается назначение базовых классов инфраструктуры, специализированных фабрик веб-приложений и прикладных тестовых классов. Рассмотрены аспекты оптимизации управления тестовыми данными.

Ключевые слова: объектно-ориентированное проектирование, интеграционное тестирование, паттерны проектирования, WebApplicationFactory, .NET 8, C#, автоматизация тестирования.

1. Введение

При разработке автоматизированных тестов для промышленных программных систем к коду тестов должны предъявляться столь же строгие требования качества, как и к основному коду приложения. Отсутствие объектной структуры и хаотичное написание сценариев верификации быстро приводят к накоплению технического долга, дублированию кода инициализации инфраструктуры и усложнению поддержки тестов при изменении бизнес-логики.

Особенно остро данная проблема стоит при интеграционном тестировании микросервисов, где каждый тест требует сложной предварительной настройки: запуска базы данных, применения миграций, конфигурации HTTP-клиентов и последующего сброса состояния таблиц. Целью данной статьи является разработка и описание объектно-ориентированной структуры классов тестового комплекса на языке C# для платформы .NET 8, минимизирующей дублирование кода и повышающей стабильность прогона автотестов.

2. Архитектурный паттерн внутрипроцессного тестирования

Для реализации реалистичного интеграционного тестирования в экосистеме .NET традиционно применяется паттерн внутрипроцессного хостинга. Вместо развертывания веб-приложения на внешнем сервере, тестируемый микросервис запускается в оперативной памяти сборочного агента с использованием специального класса-фабрики.

В стандартной библиотеке .NET для этих целей предназначен класс WebApplicationFactory. Он позволяет:

- Сконфигурировать и запустить виртуальный веб-сервер (Kestrel) в памяти тестового процесса.
- Переопределить зависимости в контейнере внедрения зависимостей (Dependency Injection Container), подменив внешние сетевые сервисы фиктивными эмуляторами.

— Изменить строку подключения к базе данных, перенаправив запросы микросервиса на временную тестовую СУБД PostgreSQL, развернутую в Docker.

— Предоставить готовый экземпляр класса HttpClient для отправки HTTP-запросов к API тестируемого микросервиса.

Использование данного паттерна позволяет проводить полноценное тестирование всего стека приложения (от контроллеров API до физических таблиц базы данных) в изолированном пространстве памяти с высокой скоростью выполнения.

3. Иерархия и взаимодействие классов тестового комплекса

Для реализации спроектированной концепции была разработана строго иерархическая объектно-ориентированная структура программных классов на языке C#, представленная тремя основными уровнями:

3.1. Глобальная конфигурация среды (GlobalSetup)

Класс GlobalSetup отвечает за однократную предварительную подготовку инфраструктуры перед запуском всего пакета тестов. Он инициализирует статический объект Docker-контейнера СУБД PostgreSQL, запускает его и применяет необходимые миграции базы данных. Использование данного механизма избавляет систему от необходимости повторного поднятия базы данных для каждого отдельного класса, существенно экономя общее время прохождения CI/CD пайплайна.

3.2. Базовый инфраструктурный класс (IntegrationTestBase)

Класс IntegrationTestBase служит фундаментом для всех прикладных тестов. Он инкапсулирует в себе:

- Доступ к экземпляру веб-фабрики приложения.
- Настройку вспомогательных утилит для отправки HTTP-запросов (ApiRequestHelper).

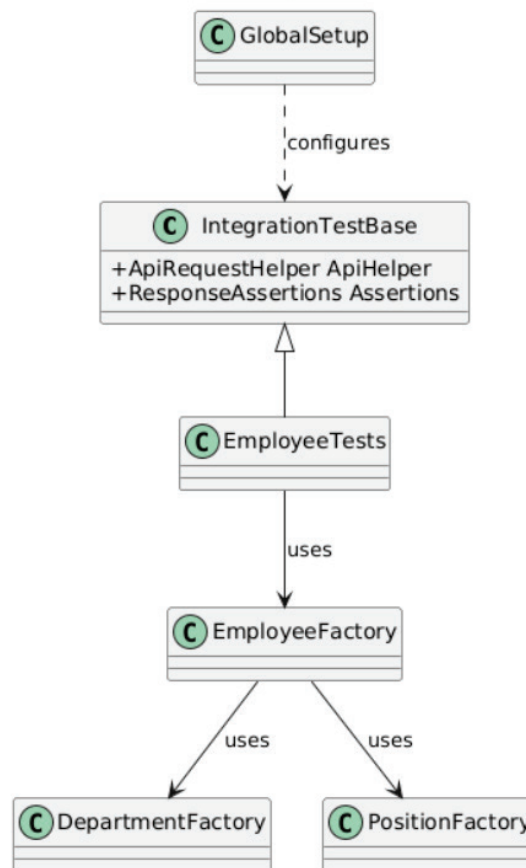


Рис. 1. Диаграмма классов тестового комплекса

— Инициализацию очистителя таблиц базы данных Resawner (библиотека Respawn).

— Реализацию интерфейса очистки ресурсов после прогона тестов (IDisposable).

Каждый прикладной тестовый класс наследуется от *IntegrationTestBase*, благодаря чему разработчику не требуется повторно писать код подключения к БД или создания HTTP-клиента.

3.3. Прикладной тестовый класс (*EmployeeTests*)

Класс *EmployeeTests* содержит непосредственные сценарии верификации бизнес-логики микросервиса управления сотрудниками. В нем сгруппированы параметризованные методы, проверяющие создание, чтение, обновление и удаление учетных записей сотрудников (CRUD-операции), а также валидацию граничных значений и бизнес-правил. Для отправки запросов он использует защищенные методы базового класса.

4. Паттерны генерации тестовых данных

Важным аспектом объектного проектирования является управление тестовыми сущностями. Создание сотрудников в базе данных требует предварительного наличия связанных объектов (например, отделов компании или должностей). Прямое наполнение базы данных

внутри тестовых методов приводит к замусориванию кода тестов.

Для решения этой проблемы в структуре классов были применены паттерны Test Data Builder (Строитель тестовых данных) и Object Mother (Объектная фабрика):

— *EmployeeFactory*, *DepartmentFactory*, *PositionFactory*. Данные классы инкапсулируют логику сборки валидных моделей данных по умолчанию. Они генерируют случайные, но реалистичные имена, адреса электронной почты и даты приема на работу.

— Метод построения цепочек (Fluent API). Фабрики позволяют гибко изменять только те свойства, которые критичны для конкретного теста, оставляя остальные значения заполненными по умолчанию. Например, вызов *EmployeeFactory.Create().WithSalary(50000).Build()* создаст сотрудника со всеми валидными полями, но с фиксированным размером заработной платы.

Это позволяет держать тестовые методы лаконичными (не более 10–15 строк кода), фокусируя внимание читающего код разработчика исключительно на проверяемом бизнес-сценарии.

5. Заключение

Разработанная объектно-ориентированная структура классов тестового комплекса на языке C# позволила успешно разделить инфраструктурный код и при-

кладную логику верификации. Использование паттернов WebApplicationFactory, Test Data Builder и наследование от базового класса IntegrationTestBase обеспечивает высокую скорость написания новых автотестов, исклю-

чает дублирование конфигурационного кода и гарантирует простоту поддержки тестовой системы при долгосрочном развитии микросервисной архитектуры предприятия.

Литература:

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб.: Питер, 2020. — 368 с.
2. Фаулер М. Шаблоны корпоративных приложений. — М.: Вильямс, 2018. — 544 с.
3. Хориков В. Принципы юнит-тестирования. — СПб.: Питер, 2021. — 320 с.
4. Мартин Р. Чистый код: создание, анализ и рефакторинг. — СПб.: Питер, 2019. — 464 с.

Проектирование и автоматизированная верификация структуры данных при интеграционном тестировании распределенных систем

Шамжуров Иван Александрович, студент
Московский государственный технологический университет «СТАНКИН»

В статье исследуются методы проектирования физической структуры реляционных баз данных и способы их автоматизированной проверки с использованием современных средств интеграционного тестирования. На примере реляционной модели данных микросервиса кадрового учета рассматривается структура таблиц базы данных PostgreSQL, связи между ними и физические ограничения целостности. Описывается методика написания автотестов для верификации каскадных удалений, уникальных индексов и проверок на пустые значения.

Ключевые слова: проектирование баз данных, физическая структура СУБД, PostgreSQL, интеграционные тесты, целостность данных, каскадное удаление.

1. Введение

В современных распределенных и микросервисных программных комплексах база данных перестает быть просто пассивным хранилищем и берет на себя значительную часть логики контроля целостности данных. Использование ограничений внешних ключей, уникальных индексов, каскадных операций и проверок на пустые значения (NOT NULL) позволяет защитить систему от логических ошибок на физическом уровне хранения.

Однако при внесении изменений в кодовую базу или при накате новых миграций структуры СУБД существует высокий риск нарушения существующих ограничений. Классические модульные тесты не способны верифицировать физическое поведение базы данных, так как работают с эмуляторами. Целью данной работы является описание спроектированной схемы данных системы кадрового учета и разработка методики автоматизированной верификации ограничений СУБД PostgreSQL с помощью интеграционных автотестов.

2. Физическая структура СУБД PostgreSQL

Для автоматизации кадрового учета на профильном ИТ-предприятии была спроектирована и реализована

физическая схема сущностей базы данных PostgreSQL. Схема отражает ключевое бизнес-требование системы: четкую иерархическую структуру организации, в которой каждый сотрудник должен быть привязан к определённому подразделению.

Модель данных состоит из двух основных сущностей, связанных между собой:

— Таблица отделов (departments). Хранит структуру компании. Первичным ключом является поле идентификатора (id), типизированное как системный глобальный уникальный идентификатор (UUID) для предотвращения коллизий при синхронизации распределенных узлов. Таблица также включает наименование отдела, описание и метки времени создания.

— Таблица сотрудников (employees). Содержит персональные и профессиональные атрибуты работников (имя, фамилия, уникальный адрес электронной почты, телефон, оклад, дата приема). Идентификатор сотрудника также имеет тип UUID.

Связь между таблицами реализуется по принципу «один ко многим»: один отдел может содержать множество сотрудников, но каждый сотрудник привязан строго к одному отделу. Данное отношение поддерживается внешним ключом department_id в таблице сотрудников, ссылающимся на первичный ключ таблицы отделов.

3. Верификация уникальности и ограничений целостности

Первым направлением интеграционного тестирования базы данных является проверка соблюдения ограничений на уникальность и запрет пустых значений (NOT NULL). Если бизнес-логика требует, чтобы у каждого сотрудника был уникальный рабочий email, автотест должен гарантировать, что СУБД аппаратно заблокирует попытку создания дубликата.

Методика интеграционной проверки уникальности состоит из следующих шагов:

- Этап подготовки: через тестовый HTTP-клиент отправляется валидный запрос на создание сотрудника с определенным адресом почты. Запись успешно фиксируется в PostgreSQL.

- Этап действия: отправляется второй запрос на создание другого сотрудника, но с тем же самым почтовым адресом.

- Этап верификации: тест перехватывает ответ системы и проверяет, что веб-сервер вернул статус-код ошибки конфликта, а база данных не создала вторую физическую запись, выбросив исключение нарушения уникального индекса СУБД.

Аналогичным образом проверяются ограничения NOT NULL: автотесты отправляют запросы с пропущенными обязательными полями и верифицируют, что база данных отклоняет транзакции, сохраняя согласованность схемы данных.

4. Тестирование каскадных операций и внешних ключей

Особое значение при проектировании реляционных баз данных имеет управление зависимыми сущностями при их удалении. В спроектированной системе кадрового учета для внешнего ключа department_id было настроено правило каскадного удаления (ON DELETE CASCADE).

Это означает, что при ликвидации отдела из базы данных должны автоматически удаляться и все связанные с ним учетные записи сотрудников.

Ручная проверка таких сценариев трудоемка, поэтому была разработана автоматизированная процедура тестирования каскадных связей:

- Тестовый метод инициирует создание нового тестового отдела и привязывает к нему нескольких сгенерированных сотрудников.

- С помощью прямого SQL-запроса автотест подтверждает, что записи физически появились в таблицах departments и employees в контейнере PostgreSQL.

- Отправляется HTTP-запрос на удаление ранее созданного отдела через API микросервиса.

Автотест выполняет прямое обращение к тестовой СУБД для подсчета количества сотрудников, ссылающихся на удаленный отдел. Если возвращенное количество записей равно нулю — СУБД успешно отработала триггер каскадного удаления. Если записи остались — тест фиксирует критический дефект целостности.

Использование реального контейнера PostgreSQL в процессе выполнения тестов гарантирует, что поведение СУБД в тестовой среде абсолютно идентично поведению в продуктивном кластере Kubernetes.

5. Заключение

Проектирование строгой реляционной схемы данных и её последующее покрытие интеграционными автотестами позволило гарантировать целостность информации на физическом уровне СУБД. Написанные сценарии автоматизированной верификации уникальных индексов, каскадных удалений и ограничений внешних ключей обеспечивают надежную защиту системы от порчи данных при рефакторинге кода или обновлении версий баз данных, минимизируя ручной контроль со стороны администраторов баз данных.

Литература:

1. Дейт К. Дж. Введение в системы баз данных. — М.: Вильямс, 2019. — 1328 с.
2. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. — М.: Вильямс, 2017. — 1440 с.
3. Ньюмен С. Создание микросервисов. — СПб.: Питер, 2021. — 416 с.
4. Хориков В. Принципы юнит-тестирования. — СПб.: Питер, 2021. — 320 с.

БИОЛОГИЯ

Особенности правильного питания в подростковом возрасте

Журавлева Ксения Евгеньевна, учитель биологии
МБОУ «Школа № 51» г. Прокопьевска (Кемеровская область)

В настоящее время в подростковом возрасте наблюдается характерное явление — акселерация. От динамики увеличения роста и массы тела отстает функциональная и морфологическая зрелость отдельных органов и систем, что увеличивает неустойчивость организма к действию различных факторов внутренней и внешней среды.

Система пищеварения у детей и подростков несовершенна, у них не отрегулирован ферментативный фон, а эпителий не достигает функциональной зрелости.

Величина основного обмена у детей в 1,5-2 раза превышает основной обмен взрослого человека. Относительная величина основного обмена (в килокалориях на 1 кг массы тела) с возрастом уменьшается: у детей 2–3 лет — 55; 6–7 лет — 42; 10–11 лет — 33; 12–13 лет — 34; у взрослых — 24.

Примерная калорийность рациона школьника должна быть следующей:

- 7–10 лет — 2400 ккал
- 11–13 лет — 2400–2600 ккал
- 14–17 лет — 2600–3000 ккал

Если ребенок занимается спортом, он должен получать на 300–500 ккал больше.

Обеспечение пластических процессов организма и энергетических функций наиболее полно осуществляется при сбалансированном питании.

Рациональное соотношение белков и жиров в питании детей 1:1. Приблизительное содержание белков, жиров и углеводов в пище 1:1:3 для детей младшего возраста и 1:1:4 старшего возраста.

В период роста и развития важна пластическая функция минеральных элементов, являющихся составной частью клеток и тканей организма, а так же биокатализаторами обменных процессов. Особого внимания заслуживает кальций, являющийся структурным элементом костной ткани. Установлено, что обмен и усвоение кальция в организме зависят от содержания фосфора и магния.

Оптимальное для усвоения организма соотношение кальция и фосфора в пищевых продуктах с 3 лет — 1:1;2. Оптимальное соотношение магния 1:0,7.

Большая роль воды в питании детей. Это обусловлено тем, что составляющей клеток и тканей является вода, на

ее долю приходится 65 процентов массы тела человека. Вода необходима для выведения из организма конечных продуктов обмена. Дети теряют в сутки 1,5–2 литра воды, и зачастую многие люди не заботятся о восполнении запасов воды в должной мере.

Режим питания — это характеристика питания, включающая кратность, время приема пищи и распределение ее по калорийности и химическому составу, а также поведение человека во время еды.

Правильный режим питания обеспечивает эффективность работы пищеварительной системы, нормальное усвоение пищи и течение обмена веществ, позволяет обеспечить своевременное восстановление израсходованных энергетических резервов организма, оптимально использовать пищевые ингредиенты, поддерживать высокую работоспособность человека.

Первым принципом правильного режима питания является регулярность питания, т. е. приемы пищи в одно и то же время суток. В этих случаях вырабатывается условный рефлекс, к привычному времени приема пищи пищеварительные органы усиливают секрецию и пища попадает в желудок, подготовленный к ее восприятию. При нерегулярном питании, наоборот, ритмичность работы желудочно-кишечного тракта нарушается, что с течением времени может повлечь за собой расстройство функции пищеварительных органов.

Вторым принципом правильного режима питания является дробность питания в течение суток. Одно- или двухразовое питание нецелесообразно и опасно для здоровья. Исследования показали, что при двухразовом питании острые панкреатиты встречаются значительно чаще, чем при трех- и четырехразовом питании, и это объясняется именно обилием потребляемой пищи за один прием. Практически здоровому человеку рекомендуется трех- или четырехразовое питание, а именно: завтрак, обед, ужин и стакан молока или кефира перед сном. Когда позволяют условия, то можно вводить в режим питания один или два дополнительных приема пищи: между завтраком и обедом и между обедом и ужином. Естественно, что дополнительные приемы пищи отнюдь не

предполагают увеличения общего количества потребляемых пищевых продуктов за день.

Третьим принципом правильного режима питания является максимальное соблюдение рационального питания при каждом приеме пищи. Это значит, что набор продуктов при каждом приеме пищи (завтрак, обед, ужин) должен быть продуман с точки зрения поставки организму человека белков, жиров, углеводов, а также витаминов и минеральных веществ в наиболее благоприятном (рациональном) соотношении.

Четвертым принципом правильного режима питания является наиболее физиологическое распределение количества пищи по ее приемам в течение дня. Многочисленными наблюдениями подтверждается, что наиболее полезен для человека такой режим, при котором за завтраком и обедом он получает более двух третей общего количества калорий суточного рациона, а за ужином — менее одной трети. Время суток для завтрака, обеда и ужина, естественно, может варьировать в довольно широких пределах в зависимости от производственной дея-

тельности человека. Однако важно, чтобы время между завтраком и обедом составляло 5–6 ч и время между обедом и ужином также составляло 5–6 ч. Время от последнего приема пищи до сна должно составлять 3–4 ч.

Еще одна составляющая рационального режима питания — это правильное поведение за столом. Во время еды нельзя отвлекаться на внешние раздражители, такие как книга, телевизор, компьютер, поскольку сосредоточенность на процессе приема пищи способствует оптимальному пищеварению и усвоению пищи. Важной составляющей правильного поведения во время еды является тщательное пережевывание, когда кусочки пищи измельчаются до жидкого состояния, где отсутствуют какие-либо неоднородности. С точки зрения физиологии, это исключительно важно, поскольку расщепление питательных веществ происходит лишь в растворенном состоянии, а не в виде комка, увеличивается коэффициент полезного действия пищеварительного тракта. Комочки тормозят процесс переваривания, вызывая риск гнилостного брожения.

Таблица 1. Основные отклонения от принципов рационального питания современных дошкольников и школьников

Отклонения в потреблении продуктов	Нехватка или увеличение питательных компонентов	Риск заболеваний
Избыточное потребление продуктов с высоким содержанием ненасыщенных жиров (жирное мясо, кондитерские изделия)	Повышенное потребление ненасыщенных жиров и холестерина	Риск возникновения атеросклероза
Ограниченное потребление кисломолочных и пребиотических продуктов	Недостаточное потребление пребиотиков	Риск возникновения патологий ЖКТ и нарушение иммунного ответа
Недостаточное потребление рыбы	Недостаточное потребление W-3 ПНЖК	Риск снижения эффективности обучения

Литература:

1. Гигиена питания подростков. — Текст: электронный // infourok.ru: [сайт]. — URL: <https://infourok.ru/statya-gigiena-pitaniya-podrostkov-6561217.html> (дата обращения: 15.06.2026).
2. Павлоцкая Л. Ф., Дуденко Н. В., Эйдельман М. М. Физиология питания. М.: Высшая школа 1989–367 с;
3. Рязанский государственный медицинский университет им. акад. И. П. Павлова. Основы законодательства по охране труда подростков; sgm.rzgmu.ru
4. https://medn.ru/zdorovi_obraz_gizni/4.htm
5. <http://megabook.ru>

Окислительно-восстановительная система «медь — нитрат серебра» как физико-химическая макромоделль селективной импрегнации нейронов по методу Гольджи

Кашуро Лидия Александровна, инженер
АСУТП ООО «Трибониан» (г. Новомосковск, Тульская область)

В работе предложена и экспериментально обоснована физико-химическая макромоделль процесса избирательного серебрения нейронов, реализованная в виде пространственно-ограниченной гетерогенной системы $\text{Cu}^0\text{--AgNO}_3$ на предметном стекле. Методами оптической микроскопии установлено, что формирование горизонтально-ориентированных фрак-

тальных кристаллов серебра подчиняется законам диффузионно-ограниченной агрегации (DLA) в условиях геометрического стеснения мениском жидкости. Показано, что автокаталитический рост дендритов серебра и всех его боковых ветвей направлен строго наружу — навстречу градиенту концентрации, что приводит к формированию локальных зон истощения реагирующих ионов. Установлено, что сопряженное выделение монооксида азота (NO) генерирует эффект локального газлифта, обеспечивающий динамическую доставку реагентов. Обосновано введение в модель искусственного «серного щита» в виде автогенных затравок сульфида серебра (Ag_2S). Проведенные аналогии позволяют объяснить феномен стохастической селективности метода Гольджи и идеальную чистоту межклеточного пространства за счет кинетического доминирования единичных центров кристаллизации, блокирующих нуклеацию на соседних биологических структурах.

Ключевые слова: метод Гольджи, импрегнация серебром, серный щит, фрактальный рост, диффузионно-ограниченная агрегация, DLA, центры кристаллизации, газлифт, макромодель нейрона.

1. Введение

Классический метод «черной реакции» Камилло Гольджи (1873 г.) [1], модификации и гистологические протоколы которого детально описаны в фундаментальных руководствах Б. Ромейса [2] и современных пособиях по гистологической технике [3], до сих пор остается одной из интригующих загадок нейрогистологии. Метод демонстрирует стохастическую селективность, окрашивая лишь 1–5 % нейронов, но выявляя их морфологию с абсолютной полнотой — от сомы до дендритных шипиков [4]. Физико-химические механизмы, удерживающие фронт кристаллизации серебра строго в границах геометрии отдельного нейрона и оставляющие межклеточный матрикс прозрачным, активно обсуждаются в академической среде [5].

В ряде современных нейрогистологических модификаций для стабилизации осадка серебра и предотвращения его деформации применяется метод сульфидирования — формирование так называемого «серного щита» из химически стойкого сульфида серебра (Ag_2S). В данной работе автором разработана аналоговая макромодель на предметном стекле, воспроизводящая геометрию нейрона и позволяющая визуализировать пространственно-ограниченный рост кристаллов серебра под совместным действием сил капиллярного стеснения, локального газлифта и искусственно введенного «серного щита».

2. Описание экспериментальной установки и макромодели

Модельный «нейрон» был сконструирован на поверхности предметного стекла в виде открытой медной ячейки (Cu^0), имеющей геометрию вулканического кратера. В качестве заготовки использовался цилиндрический медный элемент высокой чистоты, имевший технологическую вмятину. Заготовка дорабатывалась вручную: с помощью механической обработки высота была равномерно занижена до 0.2 мм, при этом глубина исходной вмятины на верхнем основании была сохранена.

Для устранения микродефектов и заусенцев медная заготовка подвергалась анизотропному химическому полированию методом погружения в водный раствор пероксида водорода (H_2O_2) и лимонной кислоты ($\text{C}_6\text{H}_8\text{O}_7$). Данная процедура обеспечила селективное растворение высокоэнергетических микровыступов шероховатости и финишную очистку поверхности Cu^0 при строгом сохранении заданной геометрии и остроты внутреннего бортика вмятины. Это также позволило минимизировать количество паразитных центров нуклеации на дне ячейки, создав оптимальный оптический контраст для последующей микроскопии. Перед началом эксперимента деталь обезжиривалась этиловым спиртом согласно общим протоколам подготовки подложек [3].

К предметному стеклу конструкция фиксировалась в центральной точке основания при помощи микроколичеств расплавленного парафина, выступающего в роли химически инертного гидрофобного диэлектрического адгезива. Использование парафина позволило жестко локализовать зону контакта и полностью исключить затекание фиксирующей массы в рабочую область капиллярного канала.

Финальные параметры макромодели составили:

- Диаметр сомы (основания корытца): 2.0 мм
- Максимальная высота внешнего бортика («пик кратера»): 0.2 мм
- Высота локальной заниженной вмятины (аналог зоны отхождения крупного дендрита): 0.1 мм

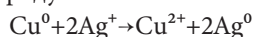
В качестве внешнего пула ионов («межклеточной среды») использовался водный раствор фармакопейного (аптечного) нитрата серебра (AgNO_3). Выбор данной субстанции вместо химически чистого лабораторного реагента был обусловлен необходимостью кинетического регулирования скорости процесса. Технологический избыток свободной азотной кислоты (HNO_3), присутствующий в аптечной форме для стабилизации раствора, выступает в роли естественного регулятора скорости ОВР за счет параллельного конкурирующего подрастворения первичных метастабильных зародышей. Это позволило снизить избыточную начальную скорость нуклеации на меди и предотвратить мгновенную сплошную пассивацию подложки.

С целью моделирования «серного щита», характерного для прецизионных гистологических протоколов [4], в рабочую среду были инкорпорированы гетерогенные центры кристаллизации сульфида серебра (Ag_2S). Данная ультрадисперсная взвесь формировалась в растворе *ex tempore* за счет контролируемого контакта внешнего стабилизированного кислотой нитрата серебра с серосодержащими вулканизирующими компонентами резиновой пробки флакона. Соединение внешней капли раствора с вмятиной корытца осуществлялось формированием капиллярного канала с помощью химически инертного стеклянного микрозонда.

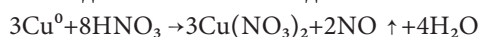
3. Результаты и обсуждение

3.1. Химизм сопряженных реакций и инициация процесса

Раствор нитрата серебра из капли под действием сил поверхностного натяжения омыл внешнее основание корытца, проникнув в капиллярный зазор. На границе раздела фаз инициировался сопряженный электрохимический процесс. Наряду с основным восстановлением серебра:



протекала параллельная реакция взаимодействия меди с избыточной свободной азотной кислотой, сопровождающаяся выделением монооксида азота:



За счет протекания ОВР и локального экзотермического эффекта у подножия модели также активизировался процесс термической дегазации растворенных в воде атмосферных газов. Благодаря эффекту катодной защиты со стороны избыточного массива металлической меди (Cu^0) корытца, формирующийся осадок металлического серебра (Ag^0) был надежно защищен от деструктивного растворения кислотой.

3.2. Роль локального газлифта и кинетика массопереноса

Выделяющиеся у наружного подножия модели микропузырьки газа NO и атмосферного воздуха, устремляясь вверх вдоль вертикальной стенки корытца, создавали локальный эффект микро-газлифта. Данный гидродинамический фактор обеспечивал динамическое перемешивание приповерхностных слоев и непрерывную доставку свежих порций ионов Ag^+ к вершине бортика. Интенсивные микропотоки на внешнем склоне корытца препятствовали закреплению случайных зародышей серебра на внешнем фоне, обеспечивая его оптическую чистоту.

Экспериментально установлено, что ограничение высоты профиля величиной 0.2 мм необходимо для преодоления капиллярного барьера и формирования устойчивого мениска жидкого мостика. Кинематика встречных диффузионных потоков ионов и капиллярных сил на предметном стекле представлена на теоретической схеме (Рис. 1).

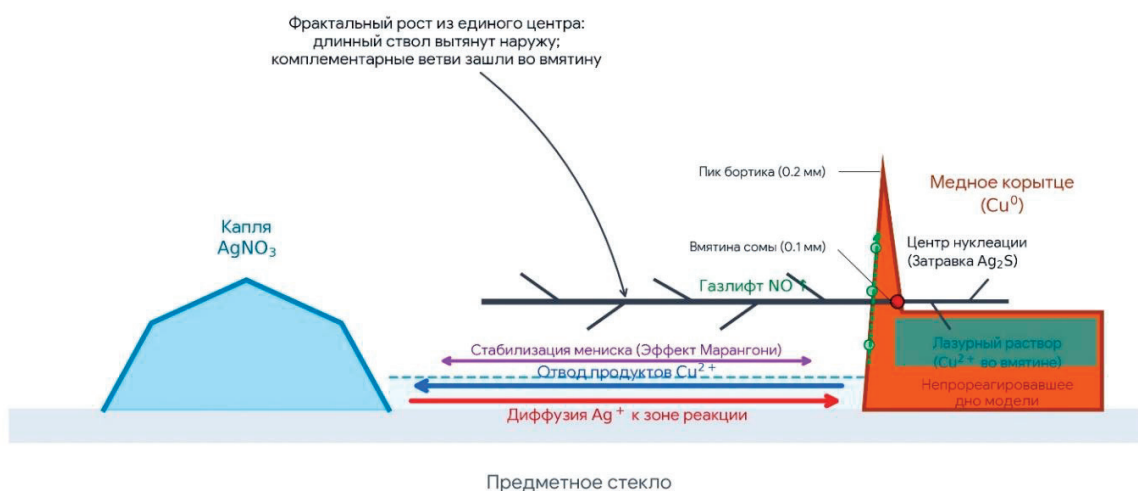


Рис. 1. Кинематика встречных диффузионных потоков и геометрия DLA-роста

3.3. Фиксация «серного щита» и направленный рост дендритов

Введенные в систему твердые мелкодисперсные частицы сульфида серебра (Ag_2S), выполняющие роль «серного щита», транспортировались восходящими потоками газлифта и за счет сил адгезии фиксировались непосредственно

на внутреннем наклонном бортике вмятины. Обладая в полной мере полупроводниковыми свойствами и находясь в прямом электрическом контакте с ювенильной поверхностью меди, активированной свободной кислотой, частицы Ag_2S сформировали устойчивые структурированные матрицы (гетерогенные центры кристаллизации).

Ионы Ag^+ , непрерывно поступающие из внешнего пула, начали автокаталитически восстанавливаться на сульфидных матрицах. Кристаллизация пошла не хаотично, а в виде направленного фрактального дендрита [6], развивающегося строго горизонтально. Данная пространственная ориентация обусловлена ограничением фронта кристаллизации в ультратонком слое раствора, зажатою между поверхностью стекла и мениском жидкости, что соответствует математической модели диффузионно-ограниченной агрегации (DLA).

3.4. Визуализация и морфогенез «шипикиков»

Эксперимент продемонстрировал формирование полностью зрелых дендритов на границе внешнего и внутреннего пространств медного корытца (Рис. 2). Основной фронт роста, включая ствол и все комплементарные боковые ветви и игольчатые микроструктуры, вытянулся длинным массивом **строго наружу в канал** — навстречу диффузионному притоку ионов Ag^+ из внешней капли (в соответствии со схемой на Рис. 1). Рост внутрь «сомы» корытца отсутствовал вследствие истощения серебра во внутренней полости. Раствор внутри сомы приобрел стабильный лазурно-голубой цвет за счет накопления ионов Cu^{2+} в чаше вмятины. Нижние слои массивного основания медной модели при этом оставались временно непрореагировавшими, что подтверждает отсутствие сквозного конвективного перемешивания всего объема ячейки.

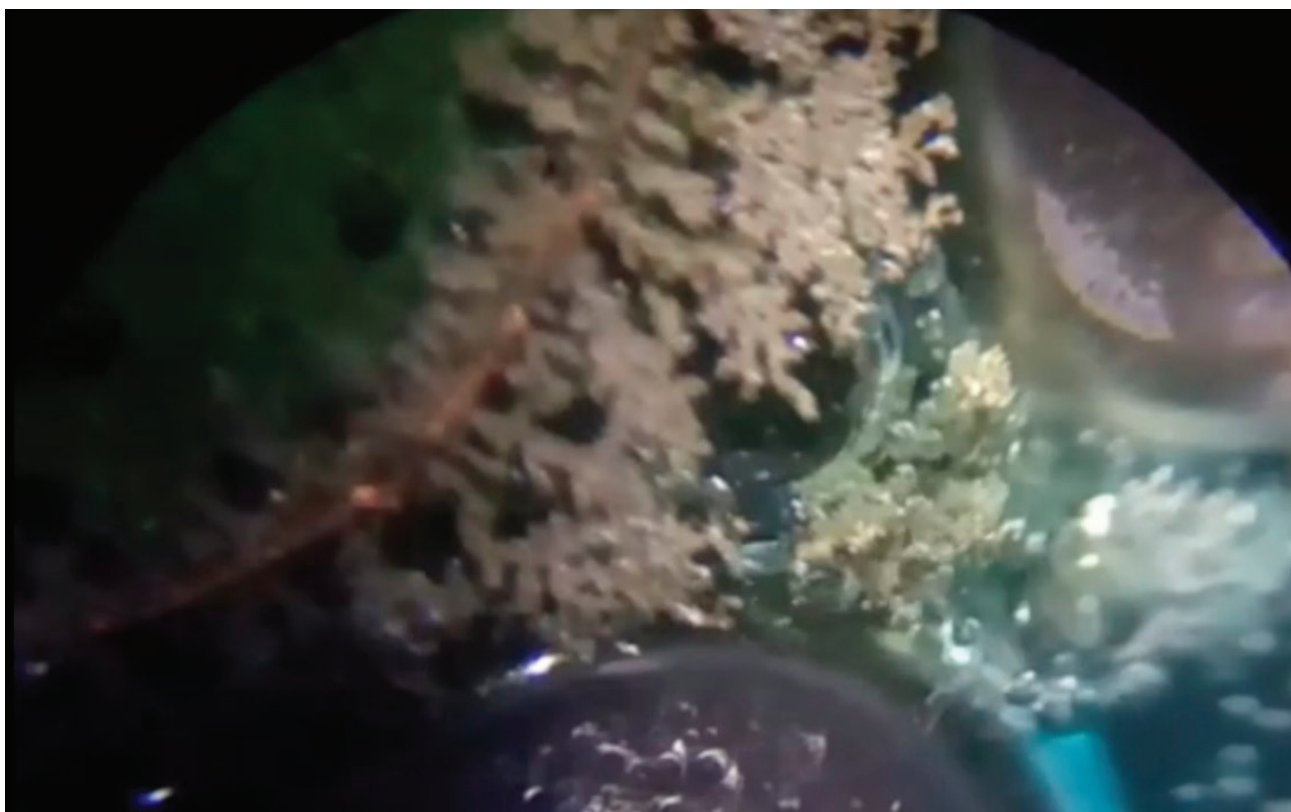


Рис. 2. Фрактальные дендриты металлического серебра на границе раздела внутреннего (вмятина, вверху слева) и внешнего (справа внизу) пространств медного корытца. Оптическая микроскопия («Микромед» Р-1, об. 4х/0.1, ок. WF16X)

При оптическом мониторинге во внешнем пространстве модели зафиксированы начальные стадии ветвления кристаллов с формированием ультрамелких игольчатых структур (Рис. 3). Появление этих микроскопических «шипикиков» детерминировано законами фрактального роста [6]: при достижении критического пересыщения плоский фронт кристаллизации теряет устойчивость (неустойчивость Маллинза–Сескерки). При этом выделившиеся микросферы газа, теряя кинетическую энергию в мениске, работали как микронные геометрические препятствия, вынуждая растущий фрактал огибать их и дополнительно модулируя сложную морфологию наноразмерных ветвей, устремленных во внешнее пространство.

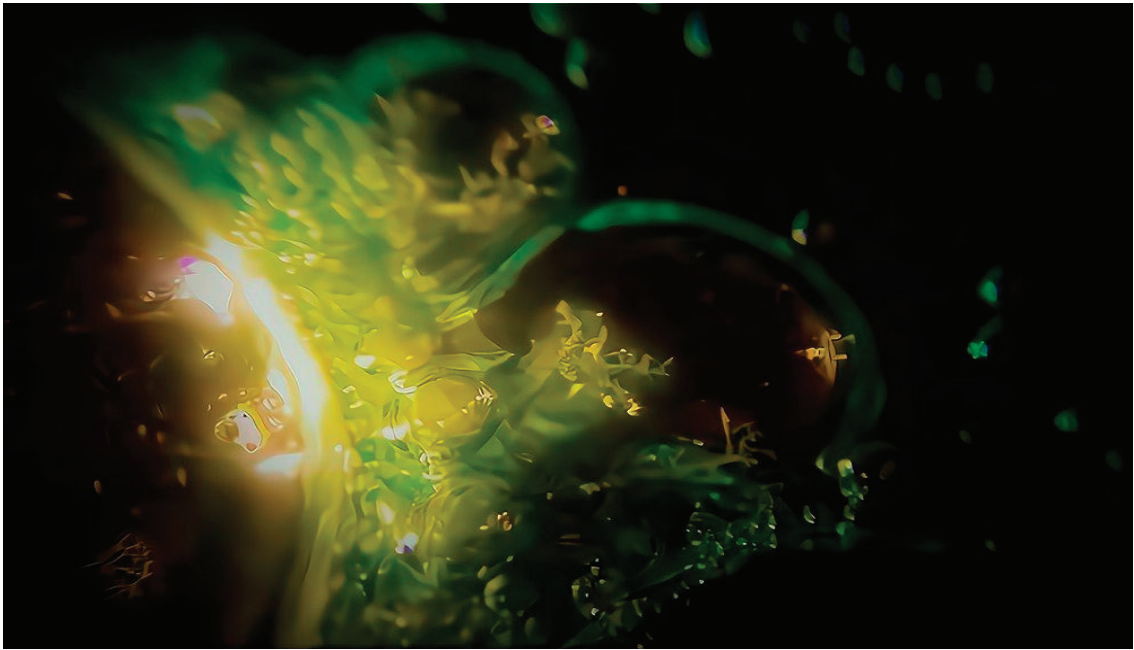


Рис. 3. Начальные стадии роста («шипики») во внешнем пространстве модели в окружении микропузырьков NO. Оптическая микроскопия («Микромед» Р-1, об. 4х/0.1, ок. WF16X)

4. Сопоставление с методом Гольджи (Метамодельный анализ)

Разработанная макромодел ь служит прямой физико-химической аналогией классического серебрения нейронов, заменяя биологические структуры эквивалентными физическими аналогами (Таблица 1).

Таблица 1. Комплементарность элементов физико-химической макромодел и биологической системы метода Гольджи

Элемент макромодел и (физико-химический)	Элемент биологической системы (метод Гольджи)
Геометрия медного кратера (бортики 0.1–0.2 мм)	Пространственная (топологическая) граница сомы нейрона
Внутренняя полость вмятины	Внутриклеточное пространство (цитоплазма сомы) [2]
Локальное занижение бортика (h=0.1 мм)	Зона аксонного холмика / основание крупного дендрита
Частицы сульфид а серебра («серный щит») на бортике	Мембранные белковые комплексы / хроматные заправки [4]
Микросферы выделившегося воздуха	Гетерогенные пространственные препятствия в межклеточном матриксе
Капиллярный градиент и диффузия в мениске	Диффузионно-осмотический градиент переноса масс через мембрану [5]

В реальной ткани мозга межклеточные щели экстремально узки. Как только внутри «избранного» нейрона на специфических белковых центрах нуклеации (аналогах частиц сульфид а серебра на бортике вмятины) начинается автокаталитический рост осадка, клетка начинает работать как локальный сток вещества, лавинообразно поглощая ионы серебра из окружающего пространства. Межклеточный раствор в диффузионной зоне стремительно истощается, обеспечивая оптическую чистоту межклеточного фона препарата. При этом клеточная мембрана выполняет роль пространственного ограничителя для фронта кристаллизации, направляя рост фрактального дендрита строго вдоль геометрии клеточного отростка в объеме ткани [4; 5].

5. Заключение

Предложенная макромодел ь наглядно демонстрирует, что для направленного формирования сложных пространственных биоморфных дендритных структур присутствие живой клетки не является строго обязательным условием. Избирательный фрактальный рост и моделирование сложной геометрии нейрона могут быть полностью симитированы

в искусственной неорганической системе при сочетании пространственно-геометрических барьеров, локализации наноразмерных гетерогенных центров нуклеации на границе раздела фаз и перевода системы в диффузионно-ограниченный режим массопереноса [6]. Применение автогенного «серного щита» из стабильных частиц Ag_2S позволяет эффективно задавать точки старта фрактального роста дендритных ветвей, развивающихся строго во внешнее пуловое пространство.

Предложенный автокаталитический механизм локального истощения реагентов вокруг растущего дендрита дает новое фундаментальное объяснение феномена стохастической селективности классического метода К. Гольджи [1].

Кроме того, разработанная модель открывает новые подходы к прецизионному управлению процессами направленной химической самосборки сложно разветвлённых металлических микро- и наноструктур, что имеет высокую практическую ценность для создания аппаратных элементов нейроморфных процессоров и систем искусственного интеллекта.

Литература:

1. Гольджи, К. Избранные труды по гистологической анатомии и физиологии нервной системы / Пер. с ит. под ред. и с предисл. акад. А. И. Смирнова. / К. Гольджи. — М.: Медгиз, 1953. — 284 с. — Текст: непосредственный.
2. Ромейс, Б. Микроскопическая техника / Пер. с нем. под ред. проф. В. Н. Терновского. / Б. Ромейс. — М.: Иностранная литература, 1954. — 606 с. — Текст: непосредственный.
3. Коржевский, Д. Э. Микроскопическая техника / Пер. с нем. под ред. проф. В. Н. Терновского. / Д. Э. Коржевский, А. В. Гиляров. — СПб: СпецЛит, 2010. — 95 с. — Текст: непосредственный.
4. Бабминдра, В. П. Структурные основы межнейронной интеграции в коре головного мозга. / В. П. Бабминдра, Т. А. Брагина. — Л.: Наука, 1982. — 144 с. — Текст: непосредственный.
5. Дерягин, Б. В. Поверхностные силы / Б. В. Дерягин, Н. В. Чураев, В. М. Муллер; Академия наук СССР, Институт физической химии. — Москва: Наука, 1985. — 398 с. — Текст: непосредственный.
6. Федер, Е. Фракталы / Пер. с англ. Ю. А. Данилова. / Е. Федер. — М.: Мир, 1991. — 254 с. — Текст: непосредственный.

МЕДИЦИНА

Сравнительный анализ исследования ликвора у детей и подростков в Многопрофильной городской детской больнице № 3 Астаны за 2023- 2024 годы

Пучко Светлана Константиновна, заведующая медицинской лабораторией;
Койшебаева Карлыгаш Баяновна, кандидат медицинских наук, врач-бактериолог
ГКП на ПХВ «Многопрофильная городская детская больница № 3» г. Астаны (Казахстан)

Арыкбаева Мадина Кенжетаевна, ассистент
Медицинский университет Астана (Казахстан)

Елгазы Санжар Дастанулы, учащийся 11-го класса
ГКП на ПХВ «Школа-лицей № 60» акимата города Астаны (Казахстан)

Целью исследования являлась оценка эффективности рутинных анализов ликвора в дифференциальной диагностике вирусных и бактериальных менингитов и оценка динамики микроскопических, цитологических показателей СМЖ у детей и подростков ГКП на ПХВ «Многопрофильная городская детская больница № 3» для совершенствования ранней диагностики нейроинфекций и воспалительных заболеваний ЦНС.

Ключевые слова: менингит, дети, клиническая картина, ликвор, СМЖ.

Актуальность

Важность анализа спинномозговой жидкости (СМЖ) при инфекционных заболеваниях в первую очередь определяется высокой диагностической значимостью инфекций центральной нервной системы (ЦНС), включая бактериальный и вирусный менингит, которые остаются основными проблемами во всем мире [1, 4].

По данным ВОЗ, ежегодно в мире регистрируется 500 тысяч случаев заболеваемости генерализованными формами менингококковой инфекции, из которых 50 тысяч заканчиваются летально, а у 20 % переболевших детей формируются неврологические дефекты [2, 5]. По данным Всемирной организации здравоохранения, менингококковый менингит распространен во всем мире [2, 6]. Заболевания регистрируют у людей всех возрастов, преимущественно у детей дошкольного и школьного возраста [2].

Исследование СМЖ считается золотым стандартом диагностики менингита, позволяя напрямую оценить воспалительные изменения в ЦНС и дифференцировать этиологию [3].

Цель: Провести сравнительный анализ заболеваемости менингитом у детей разных возрастов, госпитализированных в Многопрофильную городскую детскую больницу № 3 г. Астаны в зависимости от сезона, возраста и пола за 2023 и 2024 годы.

Материалы и методы: Материалом исследования служили образцы ликвора детей разного возраста с диагнозом ОРВИ, менингиты, острые кишечные инфекции с клиническими проявлениями менингизма, находившихся на стационарном лечении в отделениях кишечных инфекций, пульмонологии «Многопрофильной городской детской больницы № 3» г. Астаны. Образцы ликвора собирали в одноразовые стерильные пластиковые пробирки в объеме 1–2 мл при поступлении пациентов в отделения стационара. Исследование выполнялось с помощью тест-полосок CYBOW 11M и устройства для считывания данных тест-полосок CYBOW Reader 300. Определяли физико-химические свойства ликвора и проводили микроскопическое исследование в камере Фукса-Розенталя (определение цитоза). Биохимические исследования проводились на анализаторе COBAS 400. Исследовали заболеваемость у пациентов по сезонам года (зима, весна, лето, осень), объединили детей в группы по возрасту: 1 группа от 0–2 лет, 2 группа от 2–6 лет, 3 группа от 6–17 лет и по гендерному признаку (Таблица 2).

Результаты: В течении 2023–2024 гг. исследовано 1190 образцов цереброспинальной жидкости детей, в 836 случаях пробы оказались положительными. За 2023 год изучено 585 образцов, а за 2024 год изучено 605 образца ликвора, из них процент положительных результатов составил 57,7 % (483) и 42,3 % (353) соответственно (таблица 1)

Таблица 1. Частота заболеваемости менингитом у детей и подростков

Годы	Общее количество исследований		Из них положительные	
	Абсолютное количество	%	Абсолютное количество	%
2023	585	49,2	483	82,6
2024	605	50,8	353	58,4
Всего	1190	100	836	70,2

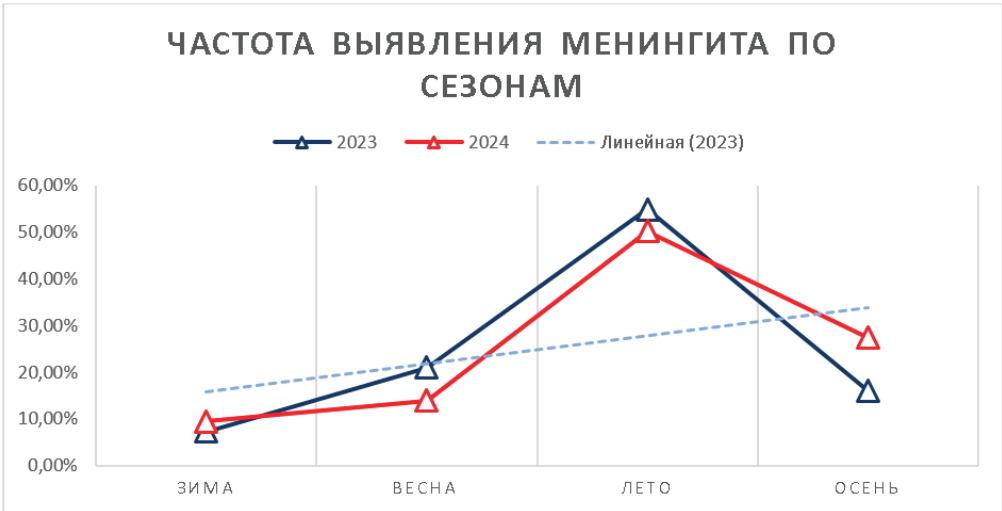


Рис. 1

Согласно полученным данным, заболеваемость за 2023 год зимой составила 8,5 %, летом 48,1 %, а в весенне-осенний период 20,3–23,1 %, а по половому признаку составило мужчин 60 % и женщин 40 %. Положительные результаты исследования ликвора на менингит составили зимой — 7,3 %, весной — 21,9 %, летом — 54,9 %, осенью — 15,9 %. Из 585 анализов на цитоз в 2023 году заболеваемость менингитом зимой составила 8,5 %, весной — 20,3 %, летом — 48,1 %, осенью — 23,1 %. Количество мальчиков составило 57,6 %, а девочек — 42,4 %. Количество положительных результатов в зимний период 2024 года составило 9,4 %, летом — 50,2 %, весенне-осенний — 13–27,4 %. Таким образом, заболеваемость в летний период 2023 и 2024 годов в два-три раза выше, чем в другие сезоны года, а из них выявление положительных результатов составило от 7,3 % до 54,9 % в 2023 году и от 9,4 % до 50,2 % в 2024 году.

Состав ликвора характеризовался цитозом от 13 до 970 клеток, нейтрофилы составляли 50 %-98 %, белок у большинства был 0,33г/л-0,96г/л, лишь в 3 случаях составил 1,39г/л.

Таблица 2. Анализ данных исследования ликвора на менингит у пациентов МГДБ № 3 за 2023–2024 годы

Общее кол-во исследований				Положительный результат		0–2		2–6		6–17		муж		жен	
		абс	%	абс	%	абс	%	абс	%	абс	%	абс	%	абс	%
Зима	2023	50	8,5	35	7,3	6	17,1	8	22,8	21	60,1	21	60,0	14	40
	2024	82	13,6	33	9,4	5	15,1	9	27,3	19	57,5	24	72,7	9	27,3
Весна	2023	119	20,3	106	21,9	10	9,4	37	35	59	55,8	78	73,6	28	26,4
	2024	86	14,2	46	13,0	11	23,9	19	41,3	16	34,8	25	54,4	21	45,6
Лето	2023	281	48,1	265	54,9	15	5,7	26	9,8	224	84,5	182	68,7	83	31,3
	2024	265	43,8	177	50,2	23	13	37	20,9	117	66,1	108	61,0	69	38,9
Осень	2023	135	23,1	77	15,9	4	5,2	12	15,6	61	79,2	50	65	27	35
	2024	172	28,4	97	27,4	6	6,1	14	14,4	77	79,4	60	61,8	37	38,2
Общее количество	2023	585	100	483	82,5	35	7,2	83	17,2	365	75,6	331	68,5	152	31,5
	2024	605	100	353	58,3	45	12,7	79	22,4	229	64,9	217	61,5	136	38,5

Результаты исследования положительно выявленных по менингиту детей по возрастным группам независимо от сезона показали, что у детей от 0–2 лет составляет от 5,2 % до 17,1 % в 2023 году и от 6,1 % до 23,9 % в 2024 году, в воз-

растной категории от 2 до 6 лет составляет от 9,8 % до 35,0 % в 2023 году и от 14,4 % до 41,3 % в 2024 году, в возрастной категории от 6 до 17 лет составляет от 45,7 % до 66,8 % в 2023 году и от 34,8 % до 79,4 % в 2024 году.

Следовательно, дети от 6–17 лет, положительно выявленные по менингиту, составляют 2/3, дети от 2–6 лет составляют 1/3, а дети от 0–2 лет незначительное количество.

Заключение

Выявление положительных результатов в весенне-летний период выше, чем в осенне-зимний период в 2023 и 2024 годах. Выявляемость положительных результатов у детей в возрасте от 6 до 17 лет составляет 2/3 от всех групп, что свидетельствует о большей распространенности по сравнению с другими возрастными категориями. По гендерной принадлежности процент выявленных положительных образцов составил у мужского пола около 58 %, а женского пола около 42 %.

Литература:

1. Singh N. Inside the Brain: Cerebrospinal Fluid Insights in Meningitis. Cureus. 2024. PMID: 39280394
2. Сатаева А. М. Анализ эпидемиологической ситуации по менингококковой инфекции в Республике Казахстан / А. М. Сатаева, Ж. А. Куланчиева // Фармация Казахстана. — 2019
3. Николаева И. В. Диагностика менингитов у детей. //Педиатрия. Издание для врачей Consilium Medicum 2014// 1.2014
4. Abeer Qasim, Minu C Abraham, Nismat Javed, Patrik Schmidt, Joshua Davidson. Unveiling the Truth: Diagnosing Bacterial Meningitis Through Repeat Lumbar Punctures. 2023 Jun 22;15(6):e40811 doi: 10.7759/cureus.40811.
5. Таточенко В. К. Вакцинопрофилактика в XXI веке. Ремедиум. 2006;2:16
6. Т. А. Рогачёва [и др.] Клинико-лабораторное исследование цереброспинальной жидкости: учебно-методическое пособие / — Минск: БГМУ, 2018. — 23 с.
7. WHO. WHO|Meningitis. WHO.2018. [Electronic resource]: <http://www.who.int/emergencies/diseases/meningitis/en/>. [Opening date]: 27 Jul 2018.
8. Анализ заболеваемости менингококковой инфекцией в Республике Казахстан по итогам 2018 года филиала НПЦСЭИМ. [Электронный ресурс]: <http://npcses.kz/ru/sanepidsituatsiya/obzory.html>.

Молодой ученый

Международный научный журнал

№ 25 (628) / 2026

Выпускающий редактор Г. А. Письменная
Ответственные редакторы Е. И. Осянина, О. А. Шульга, З. А. Огурцова
Художник Е. А. Шишков
Подготовка оригинал-макета П. Я. Бурьянов, М. В. Голубцов, О. В. Майер

За достоверность сведений, изложенных в статьях, ответственность несут авторы.
Мнение редакции может не совпадать с мнением авторов материалов.
При перепечатке ссылка на журнал обязательна.
Материалы публикуются в авторской редакции.

Журнал размещается и индексируется на портале eLIBRARY.RU, на момент выхода номера в свет журнал не входит в РИНЦ.

Свидетельство о регистрации СМИ ПИ № ФС77-38059 от 11 ноября 2009 г., выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор).

ISSN-L 2072-0297

ISSN 2077-8295 (Online)

Учредитель и издатель: ООО «Издательство Молодой ученый». 420029, Республика Татарстан, г. Казань, ул. Академика Кирпичникова, д. 25, пом. 1, 3, 4, 5, 6.

Номер подписан в печать 1.07.2026. Дата выхода в свет: 8.07.2026.

Формат 60×90/8. Тираж 500 экз. Цена свободная.

Почтовый адрес редакции: 420140, Республика Татарстан, г. Казань, ул. Юлиуса Фучика, д. 94А, а/я 121.

Фактический адрес редакции: 420029, Республика Татарстан, г. Казань, ул. Академика Кирпичникова, д. 25, пом. 1, 3, 4, 5, 6.

E-mail: info@moluch.ru; <https://moluch.ru/>

Отпечатано в типографии издательства «Молодой ученый», 420029, Республика Татарстан, г. Казань, ул. Академика Кирпичникова, д. 25, пом. 1, 3, 4, 5, 6.