

МОЛОДОЙ УЧЁНЫЙ

ISSN 2072-0297

МЕЖДУНАРОДНЫЙ НАУЧНЫЙ ЖУРНАЛ



28 2026
ЧАСТЬ I

16+

Молодой ученый

Международный научный журнал

№ 28 (631) / 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 *Жорес Иванович Алфёров* (1930–2019), советский и российский ученый-физик, политический деятель.

Жорес Иванович Алфёров родился в Витебске. Свое необычное имя будущий нобелевский лауреат получил в честь французского социалиста Жана Жореса. Семья его часто переезжала вслед за новыми назначениями по службе отца — Ивана Карповича, который работал директором различных комбинатов целлюлозной и деревообрабатывающей промышленности. Поэтому Жорес нередко менял школы. В 1947 году юноша окончил среднюю школу № 42 в Минске с золотой медалью и поступил на энергетический факультет Белорусского политехнического института.

Окончив первый курс в Минске, Жорес Алфёров вслед за новым назначением отца переехал в Ленинград и перевелся в Ленинградский электротехнический институт (ЛЭТИ) на факультет электронной техники. Там он сходу включился не только в учебу, но и в студенческую жизнь: в составе стройотряда принимал участие в строительстве Красноборской ГЭС в Ленинградской области. Алфёров также возглавил студенческое научное общество факультета, работал в вакуумной лаборатории, где и проводил свои первые научные исследования. Темой его работ стали плёночные полупроводниковые элементы.

В 1952 году Жорес с красным дипломом окончил институт по специальности «электровакуумная техника». Как отличнику, ему дали направление в Ленинградский физико-технический институт (ЛФТИ) для подготовки кандидатской диссертации и участия в разработке первых советских транзисторов.

Алфёров устроился в лабораторию Владимира Тучкевича, которая работала над получением монокристаллов чистого германия и созданием на их основе диодов и триодов. Он очень быстро стал первоклассным специалистом по квантовой физике полупроводников. За несколько лет группа ЛФТИ, в которую входил Алфёров, создала кремниевые солнечные батареи, германиевые силовые выпрямители, транзисторы на уровне лучших мировых образцов. Весной 1958 года лаборатория В. М. Тучкевича получила важный правительственный заказ по разработке полупроводниковых устройств для первой советской атомной подводной лодки. За выполнение этого заказа в 1959 году сотрудники лаборатории, в том числе и Алфёров, были награждены орденами и медалями.

На основе проведенных исследований в 1961 году Алфёров защитил кандидатскую диссертацию, посвященную германиевым и кремниевым мощным выпрямителям.

В 1968–1969 годах на базе своего открытия гетероструктур на основе арсенида галлия и арсенида алюминия Алфёров создал первый в мире гетеролазер, разрабатывал двойные гетероструктуры, имеющие колоссальное преимущество перед обычными. Одновременное открытие оптического волокна с малыми потерями привело к бурному развитию оптоволоконных систем связи. Открытые Ал-

фёровым гетероструктуры нашли применение в качестве солнечных батарей на космических спутниках и станциях. В 1970 году ученый стал доктором наук, в 1979 году — академиком. С 1973 по 2004 годы местом работы Алфёрова была кафедра оптоэлектроники ЛЭТИ. В сфере его научных интересов оказались исследования свойств наноструктур.

В 1995 году Алфёров со своими сотрудниками успешно испытали инжекционный гетеролазер на квантовых точках, работающий в непрерывном режиме при комнатной температуре.

Также со времен перестройки ученый принимал активное участие в политической жизни страны. Он неоднократно избирался депутатом Государственной Думы.

В 2000 году Алфёров стал первым российским лауреатом Нобелевской премии после распада СССР. Фактически Нобелевская премия была присуждена Алфёрову за то, что он и сотрудники его лаборатории буквально за десять лет (1963–1973) реализовали все основные схемотехнические идеи, создавшие основу для большинства современных систем генерации, контроля и управления когерентными световыми потоками в гетероструктурах на основе GaAs–AlAs.

В 2001 году Жорес Алфёров организовал Фонд поддержки образования и науки (Алфёровский фонд).

С 2003 года Алфёров занимал пост председателя Научно-образовательного комплекса «Санкт-Петербургский физико-технический научно-образовательный центр» РАН.

В 2005 году Жоресу Алфёрову была присуждена одна из самых престижных мировых премий «Глобальная энергия».

Вклад Алфёрова в современную физику был признан мировым научным сообществом. Сверхвысокочастотные усилители сотовых телефонов, плоские экраны мониторов и телевизоров, инфракрасные компьютерные порты, лазерные декодеры особо надёжных грузовых меток, мощные светодиодные прожекторы, светофоры, автомобильные фары, проигрыватели компакт-дисков — все эти устройства используют специализированные модификации гетерогенных лазеров. Без лазерных диодных транзисторов для оптоволоконных сетей невозможна была бы работа современного интернета.

Жорес Алфёров умер 1 марта 2019 года от острой сердечно-лёгочной недостаточности. Церемония прощания прошла 5 марта в Санкт-Петербургском научном центре РАН, в тот же день ученый был похоронен на Комаровском кладбище под Петербургом. В его память была выпущена памятная почтовая марка, названы школы и гимназии в Минске, Витебске, Архангельске. 16 апреля 2019 года в Санкт-Петербурге в холле Академического университета, основанного Алфёровым, открыли памятник этому великому ученому.

*Информацию собрала ответственный редактор
Екатерина Осянина*

СОДЕРЖАНИЕ

ФИЗИКА

Bui Thi Hoan

Facile green synthesis of green emissive carbon dots from passion fruit juice: insights into surface states and optical transitions 2

Nguyen Thi Huong

Conditions for the metal-insulator transition in the three-component falicov-kimball model within the coherent potential approximation 6

Garayev E. S.

Photoluminescence and absorption coefficient spectrum of CuInSe₂ thin films 12

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Козырев П. М.

Использование генеративного искусственного интеллекта в программировании 16

Козырев П. М.

Сравнение монолитной и микросервисной архитектуры в веб-приложениях 18

Козырев П. М.

Применение чат-ботов для автоматизации поддержки пользователей 20

Козырев П. М.

Методы балансировки нагрузки в распределенных веб-системах 22

Козырев П. М.

Автоматическая генерация программного кода с использованием больших языковых моделей 25

Мишекин Д. И.

Квантовые технологии: от вычислений к сенсорике и коммуникациям 27

Назаров В. И., Петрова Е. И.

Сравнительный анализ подходов к управлению состоянием в React-приложениях Redux Toolkit, RTK Query и Zustand 29

Нгуен Суан Чыонг

Вероятность ошибки приема символа в системах символьной синхронизации 32

Пронин К. Н.

Совершенствование бизнес-процесса обработки аукционных документов на основе технологии RPA 35

Тагунков М. В.

Разработка программного средства для анализа данных о серийных преступлениях 44

Трохин Е. В.

Программный модуль автоматизированного расчёта водопотребления жилых зданий 47

Хакризоев Д. Ю.

Метод выявления неявных преобразований типов, снижающих надёжность C#-кода 51

МЕДИЦИНА

Скоросова Д. А.

Роль врача-терапевта в первичной и вторичной профилактике ишемического инсульта 53

ЭКОЛОГИЯ

Сасикова Н. С., Мартыненко С. М., Самарцева А. С.

Современное состояние биоценозов реки Чилипси и ее притоков в Туапсинском районе Краснодарского края 59

ФИЛОЛОГИЯ, ЛИНГВИСТИКА

Агапов А. К.

Русский язык в цифровую эпоху: вызовы и перспективы 65

Хуснуллин А. И.

Особенности системы персонажей в басне И. А. Крылова «Ворона и Лисица» 67

Челпанова А. С.

Способы перевода прецедентных феноменов в современной англоязычной детской литературе (на материале произведения Нила Геймана «Odd and the Frost Giants») 71

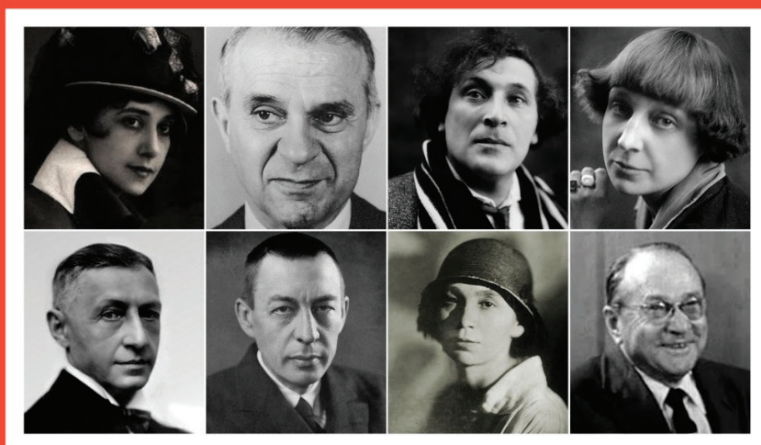
ИНФОРМАЦИОННЫЕ МАТЕРИАЛЫ

Открыт прием заявок на Конкурс для студентов, аспирантов и молодых ученых России «Культурно-историческое наследие русской эмиграции первой половины XX века». Конкурс проводит Фонд наследия русского зарубежья.

К участию приглашаются студенты бакалавриата (III–IV курсы), специалитета, магистратуры, СПО, аспирантуры, соискатели ученой степени и молодые ученые до 35 лет. Победители получают гранты на проведение исследований и создание просветительских проектов. Общий грантовый фонд составляет 2 млн рублей, размеры грантов — от 50 000 до 200 000 рублей.

Подать заявку могут студенты и специалисты любых направлений. В заявках прошедших конкурсов были представлены не только профильные направления (история, политология, культурология, философия), но и история музыки, картография, режиссура и другие области. Результатом конкурса станет создание победителями исследовательских и просветительских проектов: лекций, видеороликов, подкастов, публикаций на заявленную тему.

Положение о Конкурсе, форма заявки, шаблоны заполнения необходимых документов и прочие подробности — на сайте Фонда: <https://fnrz.ru/konkurs-dlya-molodyh-uchenyh-2026>



ГРАНТОВЫЙ КОНКУРС

для студентов, аспирантов и молодых ученых России

КУЛЬТУРНО-ИСТОРИЧЕСКОЕ НАСЛЕДИЕ РУССКОЙ ЭМИГРАЦИИ ПЕРВОЙ ПОЛОВИНЫ XX ВЕКА

Приём заявок
с 26 июня по 25 сентября 2026

ФНРЗ
ФОНД НАСЛЕДИЯ
РУССКОГО ЗАРУБЕЖЬЯ

Подробности



ФИЗИКА

Facile green synthesis of green emissive carbon dots from passion fruit juice: insights into surface states and optical transitions

Bui Thi Hoan, doctor of science, senior lecturer
Thuyloi University (Hanoi, Vietnam)

In this study, green-emitting carbon nanodots (CDs) were successfully fabricated from passion fruit juice using a simple one-step hydrothermal method. The optical properties of the CDs were systematically investigated to elucidate the material's unique electronic transition characteristics. The UV-VIS absorption spectrum showed a distinct absorption region around 280 nm, characteristic of the $n \rightarrow p^$ or $p \rightarrow p^*$ electronic transitions of the carbon core structure. However, the fluorescence excitation spectrum showed a strongly excited region within the visible light region at around 480 nm. The strongest emission region was in the green light region. The distinct separation between the intrinsic absorption peak of the carbon core (~ 280 nm) and the optimal excitation/emission wavelength pair in the visible region (~ 480 nm/ ~ 550 nm) suggests that the luminescence mechanism of this material system is primarily governed by surface energy states and luminescent molecules formed during the hydrothermal process. With its environmentally friendly synthesis process, low cost, and unique optical properties, these green-emitting carbon dots promise to be potential candidates for applications in next-generation optical sensing and bioelectronics.*

Keywords: carbon nanodots, passion fruit, hydrothermal, UV-Vis absorption spectrum, surface state, green light emission.

Introduction

Recently, the discovery of luminescent carbon nanodots (CDs) has been revolutionary in the field of materials science. Thanks to their superior properties such as high photoluminescence, good water solubility, low toxicity, and excellent biocompatibility, CDs have quickly become a superior alternative to traditional organic dyes and heavy metal-based inorganic semiconductor quantum dots. Therefore, this type of material has been widely applied in many fields such as sensors [1], [2], [3], bioimaging [2], [4], [5], [6], drug delivery [7], [8] and photocatalysis [9], [10], [11]. Despite their immense potential, developing efficient and environmentally friendly carbon dot synthesis processes remains a significant challenge. Numerous different techniques for synthesizing carbon dots have been widely published, including ultrasonic-assisted techniques [12], graphite laser etching [13], microwave-assisted techniques [14], strong acid oxidation [15], electrochemical methods [16], glycerol pyrolysis [17]. However, these methods often require complex equipment, consume significant energy, or utilize toxic chemical solvents. Therefore, the trend of utilizing readily available, inexpensive, and natural biomass sources for carbon dot fabrication in a «green chemistry» approach is attracting considerable interest from researchers.

In this study, we present a method for fabricating carbon dots (CDs) using a one-stage hydrothermal technique from

a natural precursor, passion fruit juice. To elucidate the structural properties and optical mechanisms of the material, a series of systematic spectroscopic measurements were performed. First, electron absorption spectra (UV-Vis) were analyzed to identify the background energy transition regions. Next, Fourier transform infrared (FTIR) spectra were used to identify the abundant chemical functional groups on the particle surface. The relationship between these functional groups and the electronic structure was then clarified through fluorescence excitation spectra (PLE) to find the optimal trapping energy region. Finally, fluorescence emission spectra (PL) were investigated to evaluate the actual luminescence efficiency of the material in the visible light region. This comprehensive approach provides insight into the surface state-dependent emission mechanisms, laying the foundation for practical applications of passion fruit CDs such as bioimaging, sensing, etc.

Carbon dots (CDs) were synthesized via a one-step hydrothermal carbonization method using passion fruit juice as a natural biomass precursor. Briefly, fresh passion fruits were washed thoroughly, and the seeds were separated using a sieve to obtain pure juice. The pristine passion fruit juice was then diluted with doubly distilled water at a volume ratio of 1:1. The resulting mixture was transferred into a Teflon-lined stainless steel autoclave and heated at 200 °C for 12 h in an electric oven. After the hydrothermal reaction, the autoclave was allowed to

cool naturally to room temperature. The obtained dark brown solution was filtered through a 2 μm pore-size filter paper to remove large carbonaceous aggregates and unreacted char, yielding a homogenous CD solution for further investigations.

The surface functional groups and chemical framework of the as-prepared CDs were identified by Fourier-transform infrared (FTIR) spectroscopy using a Thermo Nicolet FTIR 6700 spectrometer in the wavenumber range of 4000–400 cm^{-1} . The optical absorption properties were investigated via ultraviolet-visible (UV-Vis) absorption spectroscopy recorded on an Agilent 8453 UV-Vis spectrophotometer (Agilent Technologies, USA) within a wavelength range of 200–800 nm. The photoluminescence (PL) properties, including emission and excitation (PLE) profiles, were systematically evaluated using a NanoLog fluorescence spectrophotometer (Horiba, Edison, USA).

Results and discussions

Passion fruit juice is predominantly composed of water (~75 %) and carbohydrates (~23 %). The carbohydrate fraction typically comprises dietary fibers and simple sugars (including sucrose, fructose, and glucose), alongside a notable content of ascorbic acid. The formation mechanism of carbon dots (CDs) from this biomass precursor under hydrothermal conditions can be systematically described through three sequential stages: (i) **Hydrolysis and degradation**: In the presence of naturally occurring organic acids like ascorbic acid, the carbohydrates undergo acid-catalyzed hydrolysis, dehydration, and degradation to generate highly soluble intermediate compounds, such as furan derivatives (e.g., furfural and hydroxymethylfurfural), ketones, and volatile organic acids (acetic, levulinic, and formic acids). (ii) **Polymerization and**

aromatization: Subsequently, these intermediate species experience cross-linking, polymerization, and condensation to form soluble polymeric chains. Aromatization and the generation of aromatic clusters proceed via aldol condensation, cycloaddition, and hydroxymethyl condensation indirectly mediated through furan rings. (iii) **Nucleation and growth**: Once the concentration of these aromatic clusters surpasses the critical supersaturation threshold, carbon nucleation is triggered. The growth of these carbonaceous nuclei is driven by the diffusion of surrounding molecular species toward the particle surface. Consequently, this process yields CDs that are inherently self-passivated with abundant oxygen- and hydrogen-containing functional groups [5].

Figure 1 shows the Fourier transform infrared (FTIR) spectrum performed to identify the chemical functional groups present on the surface of the CDs material. The spectral results show a broad and strong absorption band in the region from 3200 to 3600 cm^{-1} , characteristic of the stretching vibrations of the -OH functional groups [5], [4]. A strong absorption peak at approximately 1650 cm^{-1} is attributed to the stretching vibrations of the C=O bond in the carboxyl group or the C=C bond in the amorphous carbon structure of the particle core [4]. In addition, absorption peaks in the 1350–1400 cm^{-1} region correspond to the stretching vibrations of the C-H bond and the stretching vibrations of the C-O bond [18]. The abundant presence of these oxygen and hydrogen-containing functional groups not only demonstrates the successful carbonization and surface functionalization processes from passion fruit juice precursors, but also explains the excellent dispersion of CDs in aqueous environments. Furthermore, this oxygen-rich functional group shell acts as surface energy traps, directly influencing the photoluminescence properties of the particles in subsequent measurements.

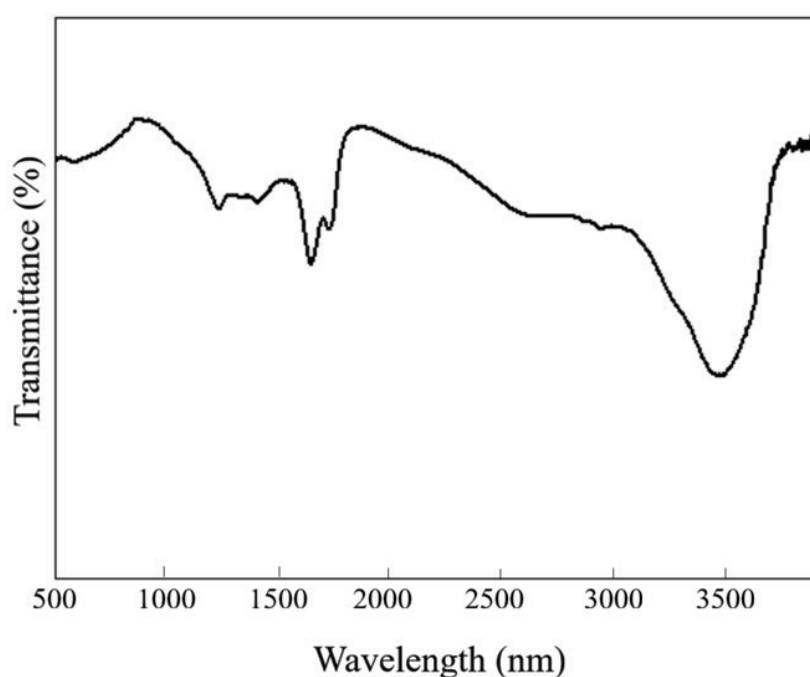


Fig. 1. Fourier transform infrared spectrum of carbon nanodots

Figure 2 shows the UV-Vis absorption spectra of CDs materials, performed to investigate the background electron energy transition states. The absorption curve displays a distinct peak at approximately 285 nm, which is thought to be related to the $p \rightarrow p^*$ energy transition of C=C double bond clusters within the amorphous carbon core network. In addition, a broad absorption shoulder peak appears in the 320–350 nm region, characteristic of the $n \rightarrow p^*$ energy

transition of oxygen-containing functional groups (such as C=O or C-O densely distributed in the surface shell of the particles), as analyzed from the FTIR spectrum in Figure 1. Notably, the UV-Vis spectrum has an absorption tail extending from the ultraviolet to the visible light region (> 400 nm). This characteristic demonstrates the existence of complex surface energy trapping states, which play a decisive role in the photoluminescence mechanism of the material [18].

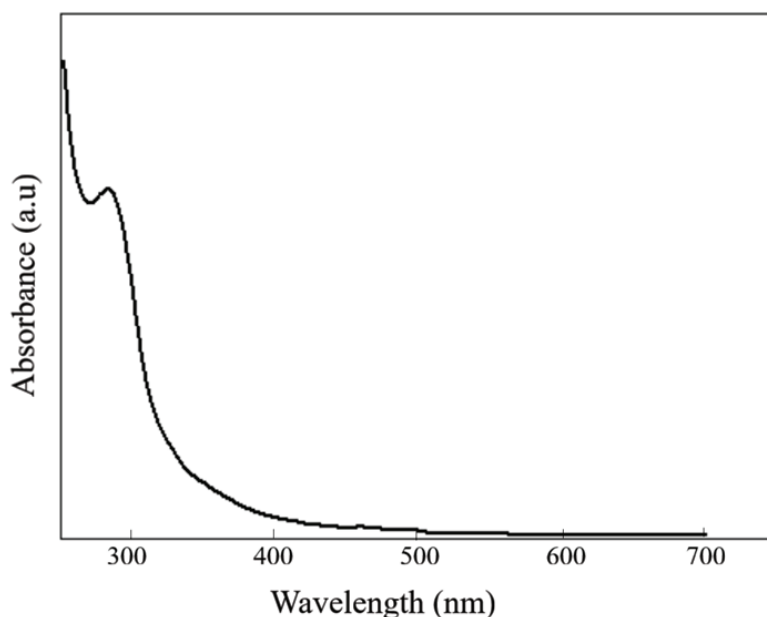


Fig. 2. UV-Vis absorption spectrum of carbon nanodots fabricated from passion fruit

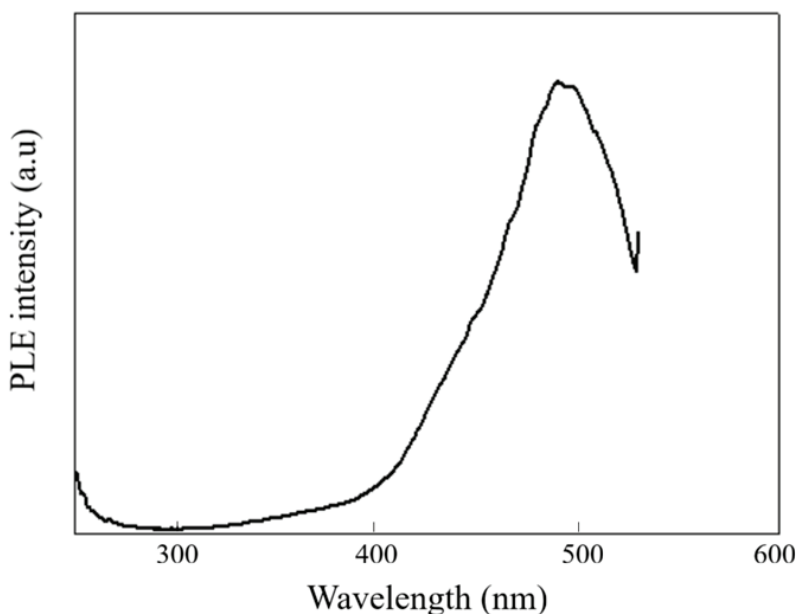


Fig. 3. Fluorescence excitation spectrum of carbon nanodots fabricated from passion fruit

Figure 3 illustrates the photoluminescence excitation (PLE) spectrum of the CDs, recorded to determine the optimal excitation wavelength and elucidate the underlying energy transitions responsible for emission. The PLE

spectrum exhibits a dominant, well-defined excitation peak at approximately 485 nm. This peak position, located in the longer wavelength region, strongly correlates with the extended absorption tail observed in the UV-Vis spectrum

(Figure 2), a typical characteristic of carbon dots with a high density of surface defect states or energy-trapping sites. This finding confirms that the photoluminescence of the passion fruit-derived CDs is primarily governed by energy transitions within the oxygen-rich functionalized surface shell, rather

than the $p \rightarrow p^*$ transitions of the amorphous carbon core [19]. Consequently, based on the PLE data, an excitation wavelength of 485 nm was selected as the optimal condition to achieve maximum fluorescence emission efficiency in subsequent optical measurements.

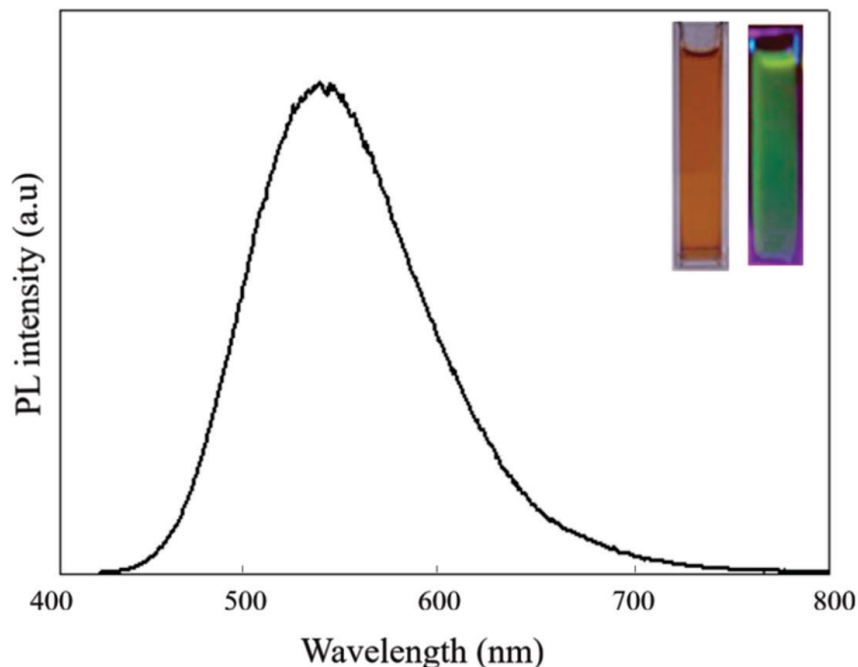


Fig. 4. Fluorescence excitation spectrum of carbon nanodots fabricated from passion fruit

Figure 4 displays the photoluminescence emission (PL) spectrum of the CDs, recorded to evaluate their actual emission performance under optimal excitation conditions. Upon excitation at 485 nm (the optimal wavelength determined from the PLE spectrum in Figure 3), the CDs exhibit a strong, symmetrical fluorescence band with a maximum emission peak at approximately 530 nm, falling within the green spectral region. The calculated Stokes shift between the excitation and emission peaks is approximately 45 nm, demonstrating an efficient radiative energy dissipation process. This relatively broad emission band is a hallmark characteristic confirming that the photoluminescence of the passion fruit-derived CDs is strongly governed by the diverse distribution of oxygen-rich surface energy-trapping states (such as C=O) and (-OH)

configurations verified by FTIR analysis), rather than quantum size effects originating from the amorphous carbon core [19].

Conclusion

Carbon dots (CDs) fabricated from passion fruit juice via a hydrothermal method possess a surface structure rich in oxygen-containing functional groups. The strong correlation among the absorption, excitation, and **green emission peaks** confirms that the photoluminescence mechanism of the material is predominantly governed by surface energy-trapping states. These findings open up promising avenues for the potential application of this material in the fields of sensing and bioimaging.

References:

1. K. Yang et al (2017). Carbon dots derived from fungus for sensing hyaluronic acid and hyaluronidase. *Sens. Actuators B Chem.*, Vol. 251, pp. 503–508, doi: 10.1016/j.snb.2017.05.086
2. Rajkumar Bandi et al, (2016). Facile and green synthesis of fluorescent carbon dots from onion waste and their potential applications as sensor and multicolour imaging agents. *RSC. Adv.*, Vol. 6, pp. 28633–28639, doi: 10.1039/c6ra01669c
3. Hui Nie et al. (2014). Carbon dots with continuously tunable full-color emission and their application in ratiometric pH sensing. *Chem. Mater.*, Vol. 26, pp. 3104–3122, doi: 10.1021/cm5003669
4. Yang Liu et al (2017). Green synthesis of fluorescent carbon dots from carrot juice for in vitro cellular imaging Original Articles Article Info. *Carbon Letters*, Vol. 21, pp. 61–67, doi:10.5714/CL.2017.21.061
5. B. B. Swagatika Sahu et al. (2012). Simple one-step synthesis of highly luminescent carbon dots from orange juice: application as excellent bio-imaging agents. *Chem Commun*, Vol. 48, pp. 8835–8837, doi: 10.1039/c2cc33796g

6. K. M. Tripathi et al. (2017). Water Soluble Fluorescent Carbon Nanodots from Biosource for Cells Imaging. J. Nanomater., Vol. 2017, doi: 10.1155/2017/7029731
7. Han Ding et al. (2015). DNA–Carbon Dots Function as Fluorescent Vehicles for Drug Delivery. ACS Appl. Mater. Interfaces, Vol. 7, No. 12, pp. 6889–6897, doi: 10.1021/acsami.5b00628
8. Hui Ding et al. (2017). Beyond a Carrier: Graphene Quantum Dots as a Probe for Programmatically Monitoring Anti-Cancer Drug Delivery, Release, and Response. ACS Appl. Mater. Interfaces, Vol. 9, No. 33, pp. 27396–27401, doi: 10.1021/acsami.7b08824
9. H. Li et al. (2012). Carbon quantum dots/Cu₂O composites with protruding nanostructures and their highly efficient (near) infrared photocatalytic behavior. J. Mater. Chem., Vol. 22, no. 34, pp. 17470–17475, doi: 10.1039/c2jm32827e
10. Benjamin C et al. (2015). Solar Hydrogen Production Using Carbon Quantum Dots and a Molecular Nickel Catalyst. J. Am. Chem. Soc., Vol. 137, no. 18, pp. 6018–6025, doi: 10.1021/jacs.5b01650
11. G. A. M. Hutton et al. (2016). Carbon Dots as Versatile Photosensitizers for Solar-Driven Catalysis with Redox Enzymes. J. Am. Chem. Soc., Vol. 138, No. 51, pp. 16722–16730, doi: 10.1021/jacs.6b10146
12. J. Xu et al. (2022). Ultrasonic-Assisted Synthesis of N-Doped, Multicolor Carbon Dots toward Fluorescent Inks, Fluorescence Sensors, and Logic Gate Operations. Nanomaterials, Vol. 12, No. 3, p. 312, doi: 10.3390/nano12030312
13. X. Li, H. Wang, Y. Shimizu, A. Pyatenko, K. Kawaguchi, and N. Koshizaki. (2011). Preparation of carbon quantum dots with tunable photoluminescence by rapid laser passivation in ordinary organic solvents. Chem. Commun., vol. 47, no. 3, pp. 932–934, doi: 10.1039/c0cc03552a
14. L. Li et al. (2012). A Facile Microwave Avenue to Electrochemiluminescent Two-Color Graphene Quantum Dots. Adv. Funct. Mater., Vol. 22, no. 14, pp. 2971–2979, doi: 10.1002/adfm.201200166
15. C. Tan, S. Zuo, Y. Zhao, and B. Shen. (2019). Preparation of multicolored carbon quantum dots using HNO₃ /HClO₄ oxidation of graphitized carbon. J. Mater. Res., Vol. 34, no. 20, pp. 3428–3438, doi: 10.1557/jmr.2019.261
16. D. Rocco, V. G. Moldoveanu, M. Feroci, M. Bortolami, and F. Vetica. (2023). Electrochemical Synthesis of Carbon Quantum Dots. ChemElectroChem, Vol. 10, No. 3, Feb. 2023, doi: 10.1002/celec.202201104
17. C.-W. Lai, Y.-H. Hsiao, Y.-K. Peng, and P.-T. Chou. (2012). Facile synthesis of highly emissive carbon dots from pyrolysis of glycerol; gram scale production of carbon dots/mSiO₂ for cell imaging and drug release. J. Mater. Chem., Vol. 22, No. 29, p. 14403, 2012, doi: 10.1039/c2jm32206d
18. L. A. A. Chunduri, A. Kurdekar, S. Patnaik, B. V. Dev, T. M. Rattan, and V. Kamiseti. (2016). Carbon Quantum Dots from Coconut Husk: Evaluation for Antioxidant and Cytotoxic Activity. Materials Focus, Vol. 5, No. 1, pp. 55–61, doi: 10.1166/mat.2016.1289
19. J. Yu et al. (2018). Luminescence Mechanism of Carbon Dots by Tailoring Functional Groups for Sensing Fe³⁺ Ions. Nanomaterials, Vol. 8, No. 4, p. 233, doi: 10.3390/nano8040233

Conditions for the metal-insulator transition in the three-component falicov-kimball model within the coherent potential approximation

Nguyen Thi Huong, doctor of science, senior lecturer
Thuyloi University (Hanoi, Vietnam)

We apply the coherent potential approximation to find the conditions for the particle fillings at which the metal-insulator transition can occur in the three-component Falicov-Kimball model, in which one-component and two-component fermions are mixed in optical lattices. While the one-component fermions are localized, the two-component ones can hop across the lattice. The conditions for the Mott transition are determined as a function of particle fillings for different values of the local Coulomb interaction.

Keywords: Three component Falicov–Kimball model, condition, metal–insulator transition, coherent potential approximation, optical lattices.

Introduction

The metal-insulator transition (MIT) is a fascinating problem in physics, which continues to attract significant attention from theoretical physicists [1]. Notably, the Mott insulating state cannot be described by the band theory of solids because of strong electron-electron interactions, which are neglected in such frameworks. A system becomes a Mott insulator when its conduction electrons become localized due to these strong repulsions [2]. In theoretical models studying the MIT, the local Coulomb interaction plays a decisive role in determining the physical properties of the system. The study of Mott transitions

has been extended following the successful loading of ultracold fermionic atoms into optical lattices, which are created by the interference of laser beams [3].

While the MIT has been extensively studied in three-component Hubbard models with equal particle masses [4], [5], the realization of optical lattices has opened a new avenue for exploring multi-component correlated systems with mass imbalances. Motivated by these experimental advances, this paper investigates the three-component Falicov-Kimball model (FKM). This system can be viewed as an asymmetric limit of the three-component Hubbard model, where a single-component fermion species is completely localized due to its heavy mass, whereas the two-component fermions remain itinerant. The MIT in this framework is driven by the competition between the hopping energy of the light particles and the on-site Coulomb interaction. Ultimately, our primary focus is to establish the precise conditions under which the MIT emerges in the three-component FKM.

Since exact solutions for multi-component strongly correlated systems are generally unattainable, various approximations have been employed to study these systems. The MIT in the three-component FKM has been investigated using methods, including dynamical mean-field theory (DMFT) [6], the slave-boson (SB) approach [7] and the coherent potential approximation (CPA) [8]. However, previous studies employing the SB [7], and CPA [8] approximations have not yet derived a generalized condition for the MIT within the FKM framework. Specifically, although the previous CPA study [8] observed the MIT, it only focused on a few specific particle fillings and interaction strengths. It did not provide a general condition for the phase transition across all particle concentrations. The behavior of the system at arbitrary fractional fillings remained unclear. By investigating the entire parameter space, our present work fills this gap. In this paper, we employ the CPA to establish the generalized condition for the MIT within the three-component FKM. This approach has demonstrated significant efficacy in investigating strongly correlated electron systems, where Coulomb interactions play a critical role in driving the MIT [8], [9], [10], [11]. Compared with DMFT and SB, the CPA offers distinct analytical simplicity. This method yields valuable analytical results and does not require much computational demand, making it well-suited for implementation on personal workstations.

Model and Formalism

We consider a three-component FKM, the Hamiltonian of the model is given by [6] [7] [8]:

$$H = -t \sum_{\langle i,j \rangle, \sigma} [c_{i\sigma}^\dagger c_{j\sigma} + \text{H.c.}] - \mu \sum_{i, \sigma} c_{i\sigma}^\dagger c_{i\sigma} + E_f \sum_{i, \sigma} f_i^\dagger f_i + U_{cc} \sum_i c_{i\uparrow}^\dagger c_{i\uparrow} c_{i\downarrow}^\dagger c_{i\downarrow} + U_{cf} \sum_i f_i^\dagger f_i c_{i\sigma}^\dagger c_{i\sigma}, \quad (1)$$

where $c_{i\sigma}^\dagger$ ($c_{i\sigma}$) is the creation (annihilation) operator for light fermion particles with spin ($\sigma \equiv \uparrow, \downarrow$) at lattice site i , and $f_i^\dagger$ (f_i) is the creation (annihilation) operator for localized fermion particles at lattice site i . t is the hopping parameter between the nearest-neighbor light fermion particles. The energy level E_f is the chemical potential of the localized fermion particles, which controls the localized fermion atom filling $n_f = \sum_i \frac{\langle f_i^\dagger f_i \rangle}{N}$. μ is the chemical potential of the system, which controls the filling in the system $n = \sum_\sigma n_{c\sigma} + n_f$, with $n_{c\sigma} = \sum_i \frac{\langle c_{i\sigma}^\dagger c_{i\sigma} \rangle}{N}$. U_{cc} is the on-site Coulomb interaction of the light fermion particles, and U_{cf} is the on-site Coulomb interaction between the light and localized fermion particles.

We apply the alloy analogy approach (AAA) to the model. By viewing the system in terms of a disordered alloy where light fermion particles with spin up can hop in the potential of light fermion particles with spin down and localized fermion particles. The many-body Hamiltonian (1) may be approximated by a one-particle Hamiltonian of the form:

$$H_{AAA} = -t \sum_{\langle i,j \rangle, \sigma} [c_{i\sigma}^\dagger c_{j\sigma} + c_{j\sigma}^\dagger c_{i\sigma}] + E_f \sum_{i, \sigma} f_i^\dagger f_i + \sum_{i, \sigma, \lambda} \varepsilon_{c\sigma}^\lambda c_{i\sigma}^\dagger c_{i\sigma}, \quad (2)$$

where energy levels $\varepsilon_{c\sigma}^\lambda$ and configuration probabilities $P_{c\sigma}^\lambda$ are given in Table 1. Here n_f is the filling of localized fermion particles, $n_{c_\downarrow}^{(0)} (n_{c_\downarrow}^{(1)})$ is the filling of light fermion particles with spin down in the lattice site that not occupied (occupied) by localized fermion particles.

In principle, the Green's function corresponding to Hamiltonian (2) has to be averaged over all possible disorder configurations. Since this averaging cannot be performed exactly, we employ the CPA to replace the original system with an effective Hamiltonian characterized by a periodic system $\Sigma_{c\sigma}(\omega)$. It is the CPA self-energy of the system. Generally, it is complex and energy-dependent single-site and independent of vector wave $\vec{k}$. Within this CPA framework, the effective Hamiltonian is expressed as:

$$H_{CPA} = -t \sum_{\langle i,j \rangle, \sigma} [c_{i\sigma}^\dagger c_{j\sigma} + c_{j\sigma}^\dagger c_{i\sigma}] + E_f \sum_{i, \sigma} f_i^\dagger f_i + \Sigma_\sigma(\omega) \sum_{i, \sigma} c_{i\sigma}^\dagger c_{i\sigma}. \quad (3)$$

Table 1. Energy levels and configuration probabilities for light fermion particles with spin up hopping in the potential of light fermion particles with spin down and localized fermion particles [12]

Configurations λ	$c_{\uparrow}$	$c_{\downarrow}$	f	Energy levels $\varepsilon_{c\uparrow}^{\lambda}$	Probabilities $P_{c\uparrow}^{\lambda}$
1	1	0	0	$-\mu$	$(1-n_f)(1-n_{c\downarrow}^{(0)})$
2	1	1	0	$-\mu+U_{cc}$	$(1-n_f)n_{c\downarrow}^{(0)}$
3	1	0	1	$-\mu+U_{cf}$	$(1-n_f)n_{c\downarrow}^{(0)}$
4	1	1	1	$-\mu+U_{cc}+U_{cf}$	$n_f n_{c\downarrow}^{(1)}$

The lattice Green function of light fermionic particles can be expressed as:

$$F_{c\sigma}(\omega) = \int \frac{\rho_0(x)}{\omega + i\eta - x - \Sigma_{c\sigma}(\omega)} dx, \quad (4)$$

where η is an infinitesimally small positive number, ω is the Matsubara frequency, and $\rho_0(\varepsilon)$ denotes the bare density of states (DOS), which is chosen here for convenience to have a semicircular profile corresponding to the infinite-dimensional Bethe lattice [11], [13]

$$\rho_0(x) = \frac{2}{\pi W^2} \sqrt{W^2 - x^2}, \quad (5)$$

where W is the half bandwidth and we will use it as the energy unit. The average Green's function is calculated via the configurational probabilities $P_{c\sigma}^{\lambda}$ and the partial Green's functions $G_{c\sigma}^{\lambda}(\omega)$

$$G_{c\sigma}(\omega) = \sum_{\lambda=1}^4 P_{c\sigma}^{\lambda} G_{c\sigma}^{\lambda}(\omega), \quad (6)$$

where $G_{c\sigma}^{\lambda}(\omega)$ represents the partial Green's functions with configuration λ , defined as [12]:

$$G_{c\sigma}^{\lambda}(\omega) = \frac{G_{c\sigma}(\omega)}{1 + (\Sigma_{c\sigma}(\omega) - \varepsilon_{c\sigma}^{\lambda}) G_{c\sigma}(\omega)}. \quad (7)$$

Within the CPA framework, the self — consistent condition dictates that the average Green's function in equation (6) must match the local Green's function in equation (4). Consequently,

$$G_{c\sigma}(\omega) = F_{c\sigma}(\omega). \quad (8)$$

After determining the Green's function, we can calculate the hopping particle filling as follows

$$n_{c\sigma} = -\frac{1}{\pi} \int_{-\infty}^0 \text{Im}(G_{c\sigma}(\omega)) d\omega. \quad (9)$$

Results and discussions

The self-consistent equation (8) is solved numerically to determine the self-energy and the Green's function via an iterative procedure [14]. Given an initial guess for the self-energy $\Sigma_{c\sigma}(\omega)$, the lattice Green's function $F_{c\sigma}(\omega)$ and the average Green's function $G_{c\sigma}(\omega)$ are calculated from equation (4) and (6), respectively. Subsequently, an updated self-energy $\Sigma_{c\sigma}(\omega)$ is evaluated as

$$\Sigma_{c\sigma}(\omega) = \Sigma_{c\sigma}(\omega) + \frac{1}{F_{c\sigma}(\omega)} - \frac{1}{G_{c\sigma}(\omega)}. \quad (10)$$

This procedure is iterated until convergence is achieved. During the numerical calculations, the value of η in equation (4) is chosen within the range of 10^{-3} to 10^{-2} for iteration convergence to happen. Using a η value smaller than 10^{-3} significantly increases the computational cost and typically prevents the iterative process from converging.

In the following, we consider $n_f = \frac{a}{b}$, where a, b are integers ($0 < a < b$). Fig. 1 indicates the filling of hopping particles $n_{c\sigma}$ as a function of the chemical potential μ with $b = 5$ and $a = 1, 2, \dots, b - 1$ for large U_{cc} and $U_{cf} > W$. It can be seen that there are three plateaus for large U_{cc} . Therefore, insulating states exist at three different fillings $n_{c\sigma}$ at $n_{c\sigma} = \frac{1}{2} - \frac{a}{2b}$, $n_{c\sigma} = \frac{1}{2}$ and $n_{c\sigma} = 1 - \frac{a}{2b}$ which correspond to the total filling $n = 1$, $n = 1 + \frac{a}{b}$, and $n = 2$, respectively. The total filling $n = 1 + \frac{a}{b}$ can be arbitrary in the range $1 < n < 2$. Fig. 2 describes the two-component particle filling as a function of the chemical potential for different values of U_{cc} at $n_f = \frac{3}{5}$ with $U_{cf} > W$. When U_{cc} has small values, the system is insulating at $n_{c\sigma} = \frac{4}{10}$. For general filling $n_f = \frac{a}{b}$, the system is insulating at $n_{c\sigma} = 1 - \frac{a}{b}$, which is equivalent to $n_{c\sigma} + n_f = 1$. This result can be understood through its equivalence to the spinless FKM when $U_{cc} = 0$. This indicates that weak correlations of hopping particles do not affect the insulating phase at $n_{c\sigma} + n_f = 1$.

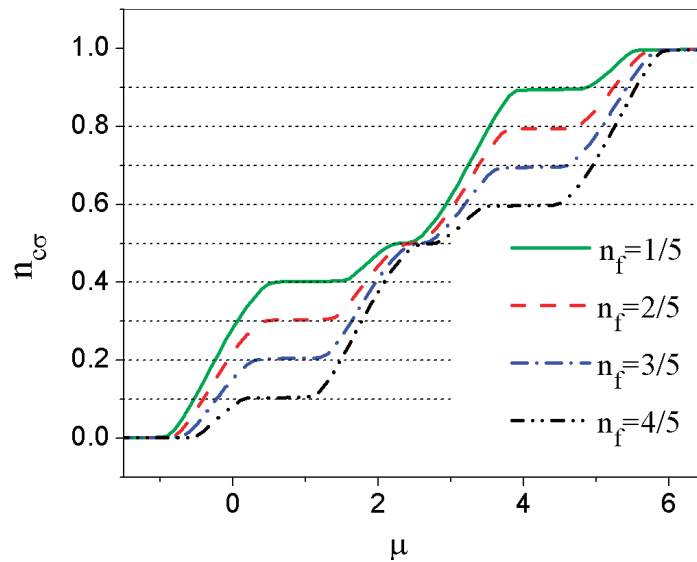


Fig. 1. (Color online) The filling of hopping particles $n_{c\sigma}$ as a function of the chemical potential μ at $n_f = \frac{a}{b}$ with $b = 5$, $0 < a < b$ for $U_{cc} = 3.0$ and $U_{cf} = 2.0$. The horizontal dotted lines show $n_{c\sigma} = \frac{i}{2b}$, where $i = 1, 2, \dots, 2b - 1$.

When U_{cc} increases, it drives the system into a metallic phase. For large values of U_{cc} , the insulating state occurs at $n_{c\sigma} = \frac{1}{2} - \frac{a}{2b}$, $n_{c\sigma} = \frac{1}{2}$ and $n_{c\sigma} = 1 - \frac{a}{2b}$. Therefore, the reentrance of the insulating phase can happen at $n_{c\sigma} = 1 - \frac{a}{b} = \frac{1}{2}$, which is equivalent to $\frac{a}{b} = \frac{1}{2}$. In general, for the case $U_{cf} > W$, in the domain of parameters (n, n_f) and for large values of U_{cc} , the system is insulating at total filling $n = 1$ and $n = 2$ with arbitrary $0 < n_f < 1$ and at arbitrary total filling $1 < n < 2$ with $n_{c\sigma} = \frac{1}{2}$. In addition, for small values of U_{cc} , the insulating phase occurs at $n_{c\sigma} + n_f = 1$, which is equivalent to $n = 2 - n_f$ with $0 < n_f < 1$. The MIT at $n_{c\sigma} + n_f = 1$ can be considered an inverse MIT.

Turning to the case $U_{cf} < W$, Fig. 3 describes the filling of two-component atoms $n_{c\sigma}$ as a function of the chemical potential μ at $n_f = \frac{a}{b}$ with $b = 5$, $0 < a < b$ for $U_{cc} = 3.0$ and $U_{cf} = 0.6$. The plot of $n_{c\sigma}$ has only one plateau at $n_{c\sigma} = \frac{1}{2}$ for large U_{cc} . This indicates that the insulating phase appears only at total filling $n = 1 + \frac{a}{b}$. Therefore, in the domain of parameters (n, n_f) for $U_{cf} < W$, the insulating phase occurs with $n = 1 + n_f$ at arbitrary $1 < n < 2$. This is also shown in Fig. 4, which plots the filling of two-

component atoms $n_{c\sigma}$ as a function of the chemical potential μ for different values of U_{cc} and fixed $U_{cf} = 0.6$ at $n_f = \frac{3}{5}$. We can see that for $U_{cf} < W$ and small values of U_{cc} , the system is in the metallic phase. When U_{cc} increases, it drives the system into an insulating phase, and the MIT can be observed at arbitrary $1 < n < 2$ with $n = 1 + n_f$, which is equivalent to $n_{c\sigma} = \frac{1}{2}$.

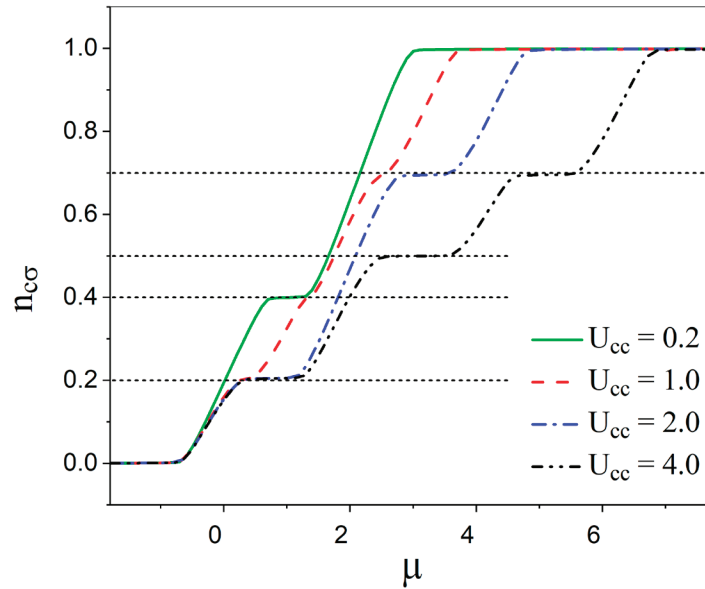


Fig. 2. (Color online) The filling of hopping particles $n_{c\sigma}$ as a function of the chemical potential μ for different values of U_{cc} and fixed $U_{cf} = 2.0$ at $n_f = \frac{3}{5}$. The horizontal dotted lines show $n_{c\sigma} = \frac{2}{10}, \frac{4}{10}, \frac{5}{10}, \frac{7}{10}$

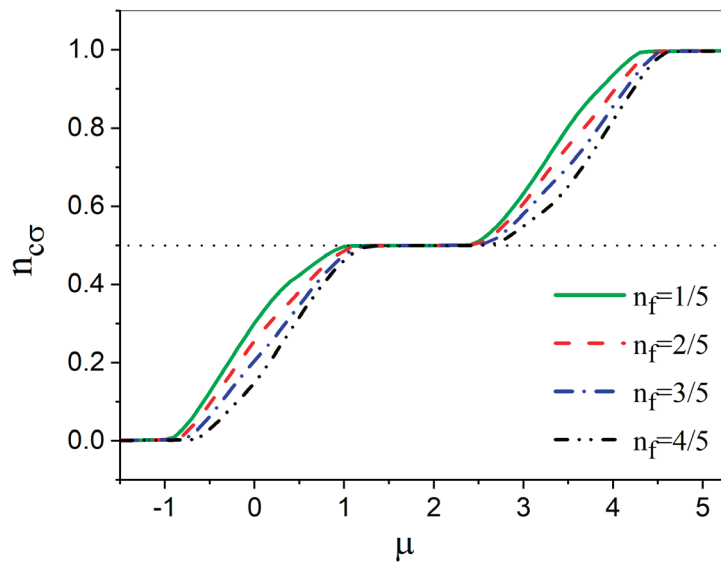


Fig. 3. (Color online) The filling of hopping particles $n_{c\sigma}$ as a function of the chemical potential μ at $n_f = \frac{a}{b}$ with $b = 5, 0 < a < b$ for $U_{cc} = 3.0$ and $U_{cf} = 0.6$. The horizontal dotted lines show $n_{c\sigma} = \frac{1}{2}$

Generally, the parameter regimes allowing for the MIT in the three-component FKM are summarized in Fig. 5. As illustrated, the transition behavior differs significantly depending on the interaction strength of U_{cf} relative to W . The black dot at $n_f = \frac{1}{2}$, $n = \frac{3}{2}$ indicates the reentrant effect.

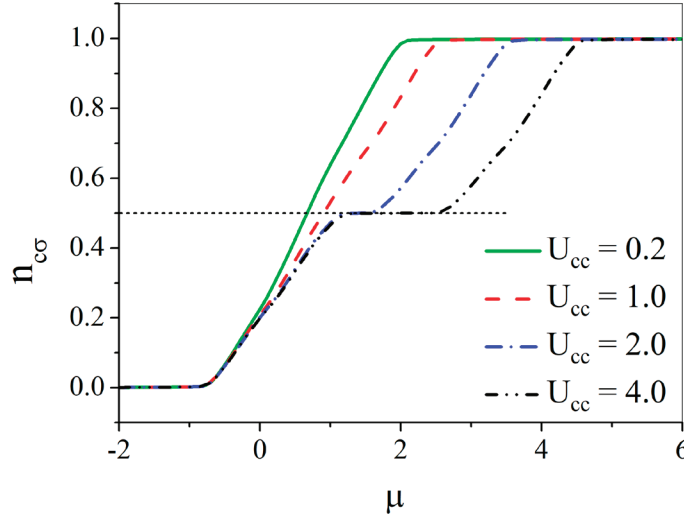


Fig. 4. (Color online) The filling of hopping particles $n_{c\sigma}$ as a function of the chemical potential μ for different values of U_{cc} and fixed $U_{cf} = 0.6$ at $n_f = \frac{3}{5}$. The horizontal dotted lines show $n_{c\sigma} = \frac{1}{2}$

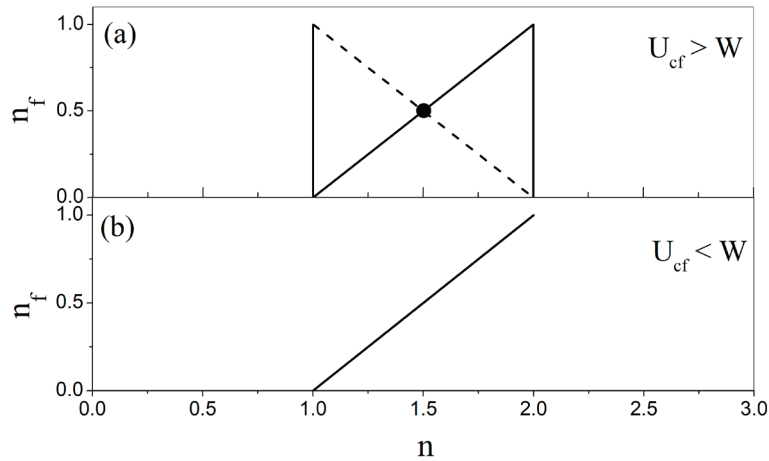


Fig. 5. Diagram showing the conditions for the MIT in the (n, n_f) plane for two regimes: (a) $U_{cf} > W$ and (b) $U_{cf} < W$. Solid lines show the conditions under which the MIT can occur, and the dashed line shows the inverse MIT. The black dot at $n_f = \frac{1}{2}, n = \frac{3}{2}$ indicates the reentrant effect

Conclusion

We have applied the CPA to study the condition under which the MIT can occur in the three-component FKM. This condition of phase transition is found based on the particle fillings. As a result, the MIT can only be observed in the region where the total filling is $1 \leq n \leq 2$; outside this region, the MIT does not occur. In particular, when $1 < n < 2$ and $n = 2n_{c\sigma} + n_f$, the MIT can be observed at $n_{c\sigma} = \frac{1}{2}$ regardless of whether $U_{cf} > W$ or $U_{cf} < W$. Therefore, under these conditions, the MIT is influenced by the local Coulomb interaction U_{cc} between the hopping particles. This means that in this case, each hopping particle having a light mass occupies a lattice site and Coulomb interaction U_{cc} prevents the double occupancy of the two-component particles.

In the case of $n = 1$ or $n = 2$, the MIT only occurs for $U_{cf} > W$ with arbitrary $0 < n_f < 1$. In this case, the roles for MIT of the two-component particles are the same as those of the single-component particles. Therefore, the same MIT can be witnessed in the three-component Hubbard model in which all components of the model have the same masses [4] [5]. In addition, the inverse MIT can be observed when $n_{c\sigma} + n_f = 1$ and $U_{cf} > W$ with small values of U_{cc} . Under this condition, each lattice site captures a single-component particle or a two-component particle, and Coulomb interaction between the single-component and two-component particles prevents them from occupying the same lattice site. Moreover, at half filling, the system can exhibit the reentrant effect. Comparing these results with the DMFT in the three-component FKM [6], there is obviously a good agreement

between the two methods although the CPA that was applied is more analytically simple. The results of this study contribute to the completion of the MIT problem in the three-component FKM within the CPA [8].

References:

1. N. F. Mott (1990). Metal-Insulator Transitions. Taylor and Francis, London.
2. U. Schneider et al. (2008). Metallic and insulating phases of repulsively interacting fermions in a 3D optical lattice. Science, Vol. 322, pp. 1520–1525.
3. Bloch, J. Dalibard, W. Zwerger (2008). Many-body physics with ultracold gases. Reviews of Modern Physics, Vol. 80, pp. 885.
4. E. V. Gorelik, N. Blumer (2009). Mott transitions in ternary flavor mixtures of ultracold fermions on optical lattices. Physical Review A, Vol. 80, 051602.
5. K. Inaba et al. (2010). Mott transitions of three-component fermionic atoms with repulsive interaction in optical lattices. Physical Review A, Vol. 82, pp. 051602.
6. D. B. Nguyen, M. T. Tran (2013). Metal-insulator transition in three-component Falicov-Kimball model. Physical Review B, Vol. 87, pp. 045125.
7. D. A. Le, M. T. Tran (2015). Mott transitions in a three-component Falicov-Kimball model: A slave boson mean-field study. Physical Review B, Vol. 91, pp. 195144.
8. T. H. Nguyen, D. A. Le, A. T. Hoang (2023). Metal-insulator transitions in three-component Falicov-Kimball model within coherent potential approximation. Communications in Physics, Vol. 33, pp. 31–40.
9. M. Plischke (1972). Coherent potential approximation calculation on the Falicov-Kimball model of the metal-insulator transition, Phys. Rev. Lett. Vol. 28, 361.
10. T. Hoang (2010). Metal-insulator transitions in the half-filled ionic Hubbard model, J. Phys.: Cond. Matt. Vol. 22, 09560.
11. D. A. Le and A. T. Hoang (2012). Mott transitions in the 2-band Hubbard model: A coherent potential approximation study. Comm. Phys. Vol. 22 (2012) 223.
12. J. van der Rest and F. Brouers (1981). Spin susceptibility of the doubly degenerate Hubbard model. Physical Review B. Vol 24, 450.
13. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg (1996). Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. Reviews of Modern Physics. Vol. 68, 13.
14. F. Ducastelle (1974). Analytic properties of the coherent potential approximation and of its molecular generalizations. Journal of Physics C: Solid State Physics. Vol 7, pp. 1795–1816.

Photoluminescence and absorption coefficient spectrum of CuInSe₂ thin films

Garayev Eldar Samed oglu, candidate of physical and mathematical sciences, associate professor
Baku State University (Azerbaijan)

Compounds of group I-III-VI₂ crystallize in structures with tetrahedral coordination of atoms, and the tetragonal and orthorhombic systems of these compounds are close to the pseudocubic structure. The band gap of I-III-VI₂ ternary semiconductors varies from 1 to 3.5 eV, which is slightly lower than that of II-VI electronic binary analogues. In contrast to compounds II-VI, which, as a rule, have a stable type of conductivity.

Keywords: triple compounds, single crystal photoluminescence, substrate absorption. optical excitation, crystal excitation, monocrystalline compound, laser radiation, absorption coefficient, spectra of thin films.

Фотолюминесценция и спектр коэффициента поглощения тонких пленок CuInSe₂

Гараев Эльдар Самед оглы, кандидат физико-математических наук, доцент
Бакинский государственный университет (Азербайджан)

Соединения группы I-III-VI₂, кристаллизуются в структурах с тетраэдрической координацией атомов, причем тетрагональная и орторомбическая сингонии этих соединений близки к псевдокубической структуре. Ширина запрещенной зоны тройных полупроводников I-III-VI₂ изменяется от 1 до 3,5 эВ, что несколько ниже, чем у электронных бинарных аналогов II-VI. В отличие от соединений II-VI, обладающих, как правило, устойчивым типом проводимости.

Ключевые слова: тройные соединения, фотолюминесценция монокристалла, поглощения подложки. оптическая возбуждения, возбуждения кристалла, монокристаллическое соединение, лазерный излучения, коэффициент поглощения, спектры тонких пленок.

It is noted in the literature [1] that ternary chalcogenides of the I-III-VI₂ type are considered to be actively luminescent semiconductors, and this assumption has been confirmed experimentally for some compounds in this class.

Despite the fact that this compound has been studied extensively compared to other compounds in the I-III-VI₂ class, no one has studied its luminescence properties. A study of radiative recombination could expand and clarify some understanding of the physical properties of *CuInSe2*.

Optical excitation of single-crystal and polycrystalline *CuInSe2* was performed using an "LTI-701" solid-state laser. The following laser pulse parameters were selected:

- wavelength $\lambda = 0.535 \mu\text{m}$;
- pulse duration $5 \mu\text{s}$;
- repetition rate 8 kHz ;
- beam diameter 1 mm ;
- average radiation power $0.8\text{--}2 \text{ W}$.

Since this wavelength of radiation falls within the crystal's intrinsic absorption region, it's clear that most of the energy is absorbed at the surface. For normal beam incidence, we calculate the beam penetration depth using the formula:

$$I = I_0 e^{-\alpha d}$$

Where I_0 — is the intensity of the incident beam, I — is the intensity of the passing daylight beam, α — is the absorption coefficient, and d — is the penetration length.

The intensity of the incident beam is equal to

$$I = \frac{W}{nh\nu}$$

where W — radiation power, $h\nu$ — photon energy, n — laser pulse frequency.

Assuming that $W = 2 \text{ W}$, $n = 8000 \text{ c-1}$, $h\nu = 3.7 \cdot 10^{-19} \text{ J}$, then $I = 6.7 \cdot 10^{14}$.

So, each fully absorbed pulse can generate $\sim 10^{14}$ free non-equilibrium carriers. In order for the intensity of the beam to decrease by four orders of magnitude as a result of absorption, it is necessary for the beam to penetrate to the depth of the crystal up to $1 \mu\text{m}$. At the same time, the absorption coefficient is taken to be equal to 105 cm-1 . Based on the obtained value of the depth of penetration, we will observe both surface and bulk recombination radiation. On the other hand, after each excitation of the crystal, a complete relaxation of non-equilibrium carriers occurs, and then the second optical excitation follows, etc.

Typical spectra of photoluminescence (fig.1) monocrystalline and polycrystalline *CuInSe2* thin films were photographed in the mode of measuring the luminescence from the reflective surface of the excitation radiation. The spectrum of radiation of a monocrystalline *CuInSe2* sample covers the $0.98 \div 1.8 \text{ eV}$ energy interval of electromagnetic radiation. Emissive recombination transitions are interband. As is known from

theory, the width of the forbidden band during interband radiative recombination is related to the maximum band of intrinsic radiation

$$E_g = h\nu \frac{kT}{2} \max.$$

Then, taking the energy of 1.031 eV as the corresponding maximum of the self-radiation band, we find $E_g = 1.023 \text{ eV}$. This value is slightly larger than the corresponding value found from absorption measurements (0.94 eV). In the interval $1.03 \div 1.39 \text{ eV}$, nine peaks associated with different radiation bands were observed. We failed to find out the mechanisms of recombination expressed by these peaks, because the intensity of the radiation is so weak that it is not possible to investigate the dependence of the intensity of the radiation on the temperature and the level of excitation. From the analysis of the absorption spectra, depressions in the spectra around 1.4 eV were found. In this same part of the photoluminescence spectrum, we detect a radiation band with a maximum peak at 1.39 eV [2,3].

Spectrum (curve 2) photoluminescence polycrystalline thin films of *CuInSe2*, grown on glass, are found in accordance with the radiation spectrum of a single crystal sample. Obviously, the measurement of photoluminescence of thin films at a high level of excitation is associated with many problems. Some of these problems have been solved by us thanks to the design of the crystal holder and the technical characteristics of the "LTI-701" laser installation. In contrast to the method of measuring photoluminescence in a large sample of *CuInSe2*, in thin films, the power of the excitation radiation did not exceed 0.8 W , which allowed to avoid heating of the thin film [4].

It can be seen from curve 2 that the long wave edge of the spectrum is shifted towards high energies compared to spectrum 1. The peak of the first band is located at 1.102 eV . In addition, peaks at 1.35 and 1.405 eV can be distinguished [5].

However, somewhat more precise energy values characterizing the band structure of the compound can be extracted from the $\alpha(h\nu)$ spectrum in thin films $\sim 40 \mu\text{m}$ thick. Figure 2 shows the spectral dependence of the absorption coefficient in a $38 \mu\text{m}$ -thick *CuInSe2* film grown on glass. The spectrum was calculated taking into account the absorption spectrum of the substrate, but ignoring the reflectance spectrum. As a rule, the latter is often taken into account by measuring the absorption spectrum of films of different thicknesses.

In the thin films we obtained, we also found that, at certain thicknesses, the absorption spectrum depends significantly on the film thickness, with the absorption coefficient in the fundamental absorption region increasing with increasing film thickness. The choice of the presented spectrum from those studied was based on the fact that, when fabricating optoelec-

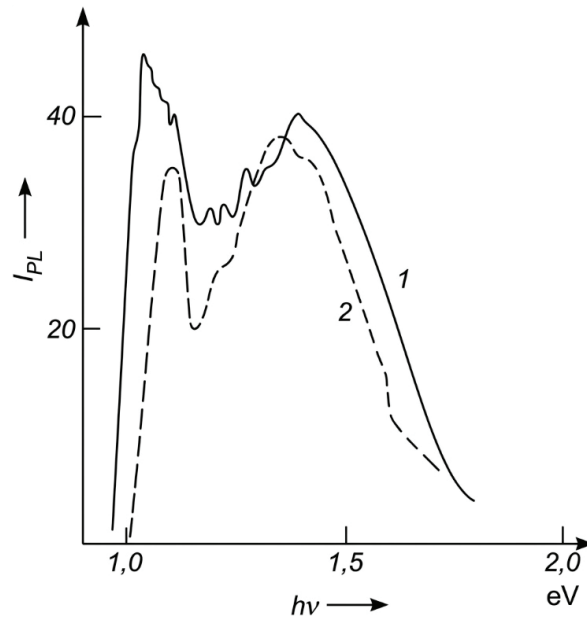


Fig. 1. Typical photoluminescence spectra of single crystals (1) and polycrystalline thin film (2) CuInSe₂

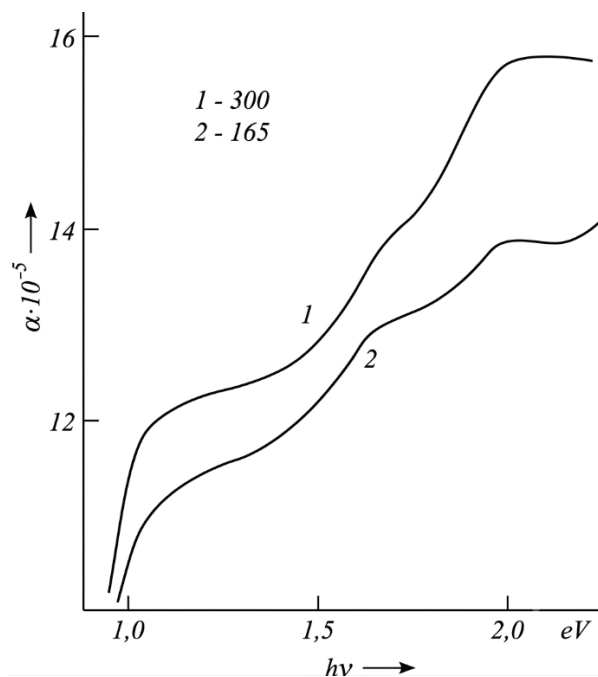


Fig. 2. Spectral dependence of the absorption coefficient in a thin film of CuInSe₂; 1–300K; 2–165K

tric devices based on thin CuInSe₂ layers, films with a thickness of 8–15 μm were most suitable for us. As can be seen from the $\alpha(h\nu)$ dependence at temperatures of 300 K and 165 K, the fundamental absorption edge is formed in the quantum energy range of 0.9–1.0 eV. However, determining the exact value of the band gap and other characteristic energy values characterizing the bands in these spectra presents difficulties.

An extreme maximum is observed in the spectrum at approximately 1.2 eV, reflecting an optical transition of electrons from the subband formed by the spin-orbit splitting of the valence band. This result is in good agreement with the results of the authors of the study. A second extreme maximum is ob-

served at approximately 1.5 eV. However, no explanation for such an optical transition in CuInSe₂ has been found in the literature. In principle, the spectra of thin films of CuInSe₂ presented by us are consistent with the spectra of this compound studied prior to our work. This result provides essential proof that the composition of the films obtained stoichiometrically corresponds to that of CuInSe₂ [6].

It was found that adding up to 1 % selenium to the starting material produces photosensitive films of Cu₃In₅Se₂. The intensities of the photoluminescence bands of these films are enhanced in comparison with the analogous luminescence bands of films of the pure compound.

References:

1. Керимова Э. М. Кристаллофизика низкоразмерных халькогенидов. Из.-во Элм, Баку, 2012, 712 с.
2. Керимова Э. М., Мустафаева С. Н., Джаббаров А. И. Температурные зависимости проводимости, термоэдс и теплоемкости TlCoS_2 , Физика низких температур. 2004, т.4, с.395.
3. Пашаев А. Н. Фотолюминесценции наночастиц, твердого раствора (Ca_2S_2) (Eu_2O_2), Журнал прикладной спектроскопии, 2011, т.78, № 2, с.288–292.
4. Керимова Э. М., Мустафаева С. Н., Температурные зависимости проводимости, термоэдс и теплоемкости TlInSe , Физика низких температур. 2002, т.2, с.212–215.
5. Мустафаева С. Н., Керимова Э. М., Перенос заряда в TlInSe , TlFeS_2 , ФТТ, 2001, т.43, № 3, с.427–430.
6. Тагиев О. Б., Абушов С. А., Люминесценция активированных ионами Eu^{2+} и Ge^{2-} кристаллов BaGa_2Se_4 // ЖТФ, 2010, т.77, с.124–126.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Использование генеративного искусственного интеллекта в программировании

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

В статье рассматривается использование генеративного искусственного интеллекта в программировании как вспомогательного инструмента для работы с кодом. Цель работы — определить, какие задачи такие системы помогают выполнять быстрее, а в каких случаях результат требует обязательной проверки разработчиком. Рассмотрены сценарии применения генеративного ИИ при написании кода, поиске ошибок, подготовке тестов и объяснении уже существующих программных фрагментов. Показано, что основной эффект таких инструментов связан не с заменой программиста, а с перераспределением рутинных операций между человеком и программным ассистентом. Сделан вывод о необходимости сочетать использование генеративного ИИ с ревью кода, статическим анализом, проверкой лицензий и контролем безопасности.

Ключевые слова: генеративный ИИ, программирование, большие языковые модели, GitHub Copilot, кодогенерация, ревью кода, безопасность программного обеспечения.

Генеративный искусственный интеллект постепенно становится частью повседневной работы программиста. Если первые инструменты автодополнения в основном предлагали отдельные идентификаторы или синтаксические конструкции, то современные ассистенты способны формировать функции, тесты, комментарии, SQL-запросы и пояснения к фрагментам кода. GitHub в своей документации определяет Copilot как AI coding assistant, доступный в средах разработки, командной строке, интерфейсе GitHub и сценариях работы с pull request. Это означает, что генеративный ИИ уже встроен не только в момент написания строки кода, но и в более широкий цикл разработки программного обеспечения.

На практике такие системы применяются в двух близких, но разных режимах. Первый режим связан с ускорением уже понятной задачи: разработчик знает, какую функцию или тест необходимо написать, а ИИ предлагает черновой вариант реализации. Второй режим связан с исследованием возможного решения: программист уточняет требования, просит объяснить неизвестный фрагмент кода, сравнивает варианты реализации или формулирует запрос иначе, чтобы получить более подходящий ответ. В исследовании Grounded Copilot эти режимы описаны как acceleration и exploration, что хорошо показывает двойственный характер инструментов кодогенерации.

Наиболее заметный практический результат генеративного ИИ состоит в сокращении времени на рутинные операции. В эксперименте GitHub Copilot с участием 95

профессиональных разработчиков группа, использовавшая Copilot, завершила задачу на 55,8 % быстрее. Такой результат не следует переносить на все виды разработки без оговорок, потому что эффект зависит от языка программирования, опыта разработчика, качества постановки задачи и типа проекта. Однако он подтверждает, что для локальных задач с ясной целью ассистент может заметно уменьшить время написания первичного варианта кода.

При этом генеративный ИИ не работает как детерминированный компилятор или формальный верификатор. Он формирует вероятностный ответ на основании запроса и контекста. Поэтому небольшое изменение формулировки может привести к другому варианту программы. В эмпирическом исследовании устойчивости GitHub Copilot семантически близкие перефразировки меняли рекомендацию примерно в 46 % случаев и могли влиять на корректность примерно в 28 % случаев. Для практики это означает, что запрос к ассистенту становится частью инженерной работы, а не нейтральной технической деталью.

Качество сгенерированного кода также остаётся неоднородным. Сравнительное исследование GitHub Copilot, Amazon CodeWhisperer и ChatGPT на наборе HumanEval показало, что Copilot генерировал корректный код в 46,3 % случаев. Этот показатель полезен не как универсальная оценка инструмента, а как напоминание о границе применимости: модель может предложить правдоподобный фрагмент, который компилируется или выглядит логично, но не выполняет требование полностью. Осо-

бенно это важно для кода, работающего с данными пользователя, сетевыми запросами, правами доступа и финансовыми операциями.

Отдельное значение имеет безопасность. Исследование уязвимостей Copilot-generated code в GitHub-проектах выявило уязвимости в 29,5 % Python-фрагментов и 24,2 % JavaScript-фрагментов из изученной выборки. GitHub в документации для Copilot cloud agent также отдельно описывает риски prompt injection, доступа к чувствительным данным и необходимость ограничений для автономных агентов. Следовательно, использование генеративного ИИ должно сопровождаться обычными инженерными мерами: статическим анализом, проверкой зависимостей, ревью человеком, тестированием граничных случаев и запретом на передачу секретов в запрос.

Вопрос авторского права и лицензий также нельзя сводить к формальному предупреждению. GitHub описывает механизм code referencing, при котором предложения Copilot могут сверяться с публичным кодом и при совпадении показываться вместе со ссылкой и сведениями о лицензии. Для образовательных и исследовательских работ это важно по двум причинам. Во-первых, автор должен понимать происхождение заимствованного фрагмента. Во-вторых, организация обязана учитывать лицензионные ограничения, если сгенерированный код переносится в коммерческий или закрытый проект.

В образовательной среде генеративный ИИ создаёт дополнительное противоречие. Для подготовленного студента он может ускорить проверку гипотез, объяснить ошибку компиляции или показать альтернативный способ решения. Для новичка тот же инструмент способен усилить иллюзию понимания: программа работает,

но студент не может объяснить, почему выбран такой алгоритм и где он сломается. Исследования по восприятию GenAI студентами показывают, что использование таких инструментов зависит от уровня курса, политики преподавателя и целей обучения. Поэтому запрет или полное разрешение без правил одинаково плохо описывают реальную ситуацию.

Рациональный подход состоит в том, чтобы рассматривать генеративный ИИ как источник черновика, а не как автора готового решения. Разработчик может поручить ассистенту подготовить каркас функции, набор тестовых случаев, документацию к API или объяснение старого кода. Но окончательное решение остаётся за человеком, который проверяет требования, анализирует ошибки, оценивает сложность алгоритма, соблюдает лицензионные условия и отвечает за безопасность результата. В этом смысле генеративный ИИ меняет структуру труда программиста: часть времени переносится с набора кода на постановку задачи, проверку и принятие инженерных решений.

Таким образом, использование генеративного ИИ в программировании оправдано там, где задача допускает быстрый черновик и последующую проверку. Наиболее полезными выглядят сценарии автодополнения, генерации тестов, объяснения кода, поиска типовых ошибок и подготовки документации. Наиболее рискованными остаются области, где ошибка проявляется не сразу: безопасность, обработка персональных данных, параллельное выполнение, финансовая логика, миграции баз данных и лицензионно чувствительный код. Поэтому развитие ИИ-ассистентов не отменяет инженерную дисциплину, а делает её более заметной.

Литература:

1. GitHub Docs. What is GitHub Copilot? // GitHub Docs. — 2026. — URL: <https://docs.github.com/en/copilot/get-started/what-is-github-copilot> (дата обращения: 19.06.2026).
2. GitHub Docs. GitHub Copilot code referencing // GitHub Docs. — 2026. — URL: <https://docs.github.com/en/copilot/concepts/completions/code-referencing> (дата обращения: 19.06.2026).
3. GitHub Docs. Risks and mitigations for GitHub Copilot cloud agent // GitHub Docs. — 2026. — URL: <https://docs.github.com/en/copilot/concepts/agents/cloud-agent/risks-and-mitigations> (дата обращения: 19.06.2026).
4. Barke S., James M. B., Polikarpova N. Grounded Copilot: How Programmers Interact with Code-Generating Models. — 2022. — DOI: 10.48550/arXiv.2206.15000. — URL: <https://arxiv.org/abs/2206.15000>.
5. Finnie-Ansley J., Denny P., Becker B. A., Luxton-Reilly A., Prather J. Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. — 2022. — DOI: 10.48550/arXiv.2210.15157. — URL: <https://arxiv.org/abs/2210.15157>.
6. Peng S., Kalliamvakou E., Cihon P., Demirel M. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. — 2023. — DOI: 10.48550/arXiv.2302.06590. — URL: <https://arxiv.org/abs/2302.06590>.
7. Liang J. T., Yang C., Myers B. A. A Large-Scale Survey on the Usability of AI Programming Assistants: Successes and Challenges. — 2023. — DOI: 10.48550/arXiv.2303.17125. — URL: <https://arxiv.org/abs/2303.17125>.
8. Yetistiren B., Ozsoy I., Ayerdem M., Tuzun E. Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT. — 2023. — DOI: 10.48550/arXiv.2304.10778. — URL: <https://arxiv.org/abs/2304.10778>.
9. Mastropaolo A., Pascarella L., Guglielmi E., Ciniselli M., Scalabrino S., Oliveto R., Bavota G. On the Robustness of Code Generation Techniques: An Empirical Study on GitHub Copilot. — 2023. — DOI: 10.48550/arXiv.2302.00438. — URL: <https://arxiv.org/abs/2302.00438>.

10. Fu Y., Liang P., Tahir A., Li Z., Shahin M., Yu J., Chen J. Security Weaknesses of Copilot-Generated Code in GitHub Projects: An Empirical Study. — 2023. — URL: <https://arxiv.org/abs/2310.02059>.
11. Dakhel A. M., Majdinasab V., Nikanjam A., Khomh F., Desmarais M. C., Jiang Z. M. GitHub Copilot: the perfect Code compLeeter? — 2024. — DOI: 10.48550/arXiv.2406.11326. — URL: <https://arxiv.org/abs/2406.11326>.
12. Prather J., Reeves B., Leinonen J., MacNeil S., Randrianasolo A. S., Becker B., Kimmel B., Wright J., Briggs B. The Widening Gap: The Benefits and Harms of Generative AI for Novice Programmers. — 2024. — URL: <https://arxiv.org/abs/2405.17739>.
13. Prather J., Denny P., Leinonen J., Becker B. A., Albluwi I., Craig M., Keuning H., Kiesler N., Kohn T., Luxton-Reilly A., MacNeil S., Petersen A., Pettit R., Reeves B. N., Sarsa S. Students' Perceptions and Use of Generative AI Tools for Programming Across Different Computing Courses. — 2024. — URL: <https://arxiv.org/abs/2410.06865>.
14. Song F., Agarwal A., Wen W. The Impact of Generative AI on Collaborative Open-Source Software Development: Evidence from GitHub Copilot. — 2024. — DOI: 10.48550/arXiv.2410.02091. — URL: <https://arxiv.org/abs/2410.02091>.
15. Hou X., Wang Z., Liu X., Zhao Y., Wang D., Wang H. The Effects of GitHub Copilot on Computing Students' Programming Effectiveness, Efficiency, and Processes in Brownfield Programming Tasks. — 2025. — URL: <https://arxiv.org/abs/2506.10051>.

Сравнение монолитной и микросервисной архитектуры в веб-приложениях

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Монолитная и микросервисная архитектуры по-разному распределяют сложность веб-приложения: монолит концентрирует код, данные и релизный цикл в одном приложении, а микросервисы переносят часть ответственности в сетевые контракты, отдельные сервисы и эксплуатационную инфраструктуру. Цель работы — определить, какие эксплуатационные и организационные условия делают монолит рациональным выбором, а какие требуют перехода к микросервисам. Сравнение проводится по критериям сложности разработки, развёртывания, масштабирования, сопровождения, тестирования, наблюдаемости, согласованности данных и стоимости миграции. Особое внимание уделяется не преимуществам архитектур в общем виде, а механизмам, через которые они влияют на релизы, отказоустойчивость и сопровождение пользовательских сценариев.

Ключевые слова: монолитная архитектура, микросервисная архитектура, веб-приложение, масштабирование, эксплуатационные практики, наблюдаемость, согласованность данных, миграция, сопровождение.

Веб-приложение можно построить как единый серверный модуль или как набор сервисов, взаимодействующих через сетевые протоколы и сообщения. В первом случае большая часть бизнес-логики, доступа к данным и пользовательских сценариев развёртывается как одно приложение. Во втором случае функции распределяются между несколькими сервисами, каждый из которых имеет собственный код, жизненный цикл и часто собственное хранилище данных. Сравнение этих подходов важно не как выбор «современного» или «устаревшего» стиля, а как оценка того, где именно система будет изменяться, масштабироваться и отказывать.

Монолитная архитектура обычно проще на начальном этапе разработки. Один репозиторий, единая схема данных, общий процесс сборки и локальный запуск уменьшают число инфраструктурных решений, которые команда должна принять до появления продукта. Для небольшого веб-приложения это означает более короткий путь от требования до работающей функции: разработчик меняет код, запускает тесты и развёртывает один артефакт. В такой схеме легче проследить выполнение пользо-

вательского запроса, потому что вызовы проходят внутри одного процесса или одной серверной части.

Эта простота имеет предел. По мере роста системы монолит может стать узким местом для параллельной разработки, независимых релизов и масштабирования отдельных функций. Изменение небольшой функции требует пересборки и развёртывания всего приложения, а ошибка в одном модуле может повлиять на весь серверный процесс. Если разные команды одновременно меняют независимые части системы, общий код и общий релизный цикл увеличивают число конфликтов. Поэтому монолит лучше подходит там, где доменная область ещё не стабилизировалась, команда небольшая, а требования к независимому масштабированию отдельных функций невысоки.

Микросервисная архитектура решает часть этих проблем через декомпозицию по бизнес-возможностям. Отдельный сервис может отвечать за каталог, оплату, уведомления, профиль пользователя или обработку заказов. Если граница выбрана корректно, команда меняет один сервис без пересборки всей системы и развёртывает его

независимо. Это полезно в веб-приложениях с разными профилями нагрузки: например, каталог может требовать масштабирования чтения, а сервис платежей — строгих проверок и более консервативного релизного цикла.

Главная цена микросервисов состоит в том, что локальная простота заменяется распределённой сложностью. В монолите вызов функции обычно завершается внутри одного процесса, а в микросервисной системе тот же сценарий превращается в сетевой обмен с таймаутами, повторными запросами и частичными отказами. Если сервис рекомендаций недоступен, веб-приложение должно решить, показывать ли страницу без рекомендаций, повторять запрос или возвращать ошибку. Поэтому микросервисы требуют механизмов изоляции отказов, очередей сообщений, ограничителей сбоя и заранее заданного упрощённого режима работы.

Развёртывание также меняет характер работы. Монолит развёртывается как один артефакт, поэтому команда управляет одной версией приложения и одной процедурой отката. Это проще для небольших систем, но затрудняет независимый выпуск отдельных функций. Микросервисы позволяют выпускать сервисы по отдельности, однако требуют конвейера непрерывной интеграции и поставки, управления конфигурациями, контейнерной инфраструктуры, контроля совместимости программных интерфейсов и наблюдаемости. Без этих практик независимое развёртывание превращается в набор ручных операций, где ошибка версии одного сервиса ломает сценарий пользователя.

Масштабирование в монолите чаще выполняется копированием всего приложения. Такой подход понятен и хорошо работает, пока нагрузка распределена относительно равномерно. Если же только один модуль потребляет большую часть ресурсов, приходится масштабировать весь монолит, включая части, которым дополнительные экземпляры не нужны. Микросервисы позволяют масштабировать отдельный компонент: увеличить число экземпляров сервиса поиска, рабочих процессов обработки изображений или программного интерфейса каталога. Экономический выигрыш появляется только тогда, когда инфраструктурная стоимость распределённой системы ниже стоимости избыточного масштабирования монолита.

Работа с данными — один из самых сложных пунктов сравнения. Монолит часто использует единую базу данных и транзакции, поэтому операция оформления заказа может быть выполнена как целостное изменение нескольких таблиц. В микросервисной архитектуре каждый сервис стремится владеть своими данными, чтобы не зависеть от внутренней схемы другого компонента. Это снижает связность на уровне БД, но создаёт проблему согласованности. Например, платёж может пройти успешно, а резервирование товара завершиться ошибкой из-за отсутствия остатка на складе. Тогда система должна отменить платёж, изменить статус заказа, не отправить ложное уведомление пользователю и не выполнить компенсацию

дважды при повторной доставке сообщения. Для таких сценариев нужны события, саги, идемпотентные операции и правила компенсации.

Тестирование монолита и микросервисов также отличается. В монолитном приложении проще поднять локальную среду и выполнить интеграционные тесты внутри одного процесса или одной базы данных. В микросервисной системе тест должен учитывать совместимость контрактов между, например, сервисом заказов и сервисом оплаты: один компонент публикует формат запроса и ответа, а другой проверяет, что его клиентский код не сломается при новой версии программного интерфейса. Поэтому возрастает роль контрактного тестирования, тестовых окружений, имитаторов внешних сервисов и проверок в конвейере поставки. Иначе команда получает быстрые релизы отдельных сервисов, но теряет уверенность в поведении всей системы.

Наблюдаемость в монолите обычно строится вокруг журналов приложения, метрик процесса и состояния базы данных. В микросервисах пользовательский запрос проходит через несколько компонентов, поэтому требуется связать вызовы в единую трассировку. Идентификатор запроса должен проходить через шлюз программного интерфейса, сервисы бизнес-логики, очереди и хранилища, чтобы команда могла увидеть, где возникла задержка или ошибка. Без трассировки и централизованных метрик распределённая архитектура становится труднее для сопровождения, чем монолит, потому что дефект проявляется в одном месте, а причина находится в другом сервисе.

Миграция от монолита к микросервисам не должна рассматриваться как автоматическое улучшение архитектуры. Систематическое исследование Hassan, Bahsoon и Kazman выделяет гранулярность сервисов как одну из центральных проблем перехода: слишком крупный сервис сохраняет свойства монолита, а слишком мелкий увеличивает число сетевых зависимостей. Если сервисы выделяются по текущим классам или таблицам, система получает сетевую оболочку поверх прежней связанности. Если границы строятся вокруг бизнес-возможностей и транзакционных контекстов, появляется шанс уменьшить область изменения. Поэтому миграция должна начинаться не с контейнеризации всего кода, а с анализа доменной модели, зависимостей, частоты изменений и критичных сценариев.

В некоторых случаях рациональным решением остаётся монолит или модульный монолит. Такой подход сохраняет единое развёртывание и локальную простоту, но требует внутренней модульности: явных границ пакетов, запрета на случайные зависимости, отдельных интерфейсов и дисциплины тестирования. Принцип поэтапного выделения сервисов полезен для веб-приложений, где продукт ещё ищет рынок и требования быстро меняются. Ранняя микросервисная декомпозиция в такой ситуации может зафиксировать ошибочные границы и увеличить стоимость разработки до того, как команда поймёт реальную структуру предметной области.

Таким образом, монолитная и микросервисная архитектуры решают разные инженерные задачи. Монолит выигрывает за счёт простоты разработки, единой транзакционной модели, более дешёвого локального тестирования и меньших требований к инфраструктуре. Микросервисы выигрывают там, где нужны независимые релизы, масштабирование отдельных функций, автономия команд и изоляция изменений. Но эти преимущества появляются только вместе с платой за распределён-

ность: сетевые отказы, версионирование программных интерфейсов, наблюдаемость, согласованность данных и организационная координация. Практический критерий выбора прост: если изменение функции требует участия одной команды и одного релиза, монолит часто остаётся достаточным; если разные части системы должны изменяться, масштабироваться и отказывать независимо, микросервисы становятся оправданным, но более дорогим вариантом.

Литература:

1. Microsoft Learn. Стиль архитектуры микрослужб // Azure Architecture Center. — 2025. — URL: <https://learn.microsoft.com/ru-ru/azure/architecture/guide/architecture-styles/microservices> (дата обращения: 05.07.2026).
2. Microsoft Learn. Выбор варианта вычислений Azure для микрослужб // Azure Architecture Center. — 2026. — URL: <https://learn.microsoft.com/ru-ru/azure/architecture/microservices/design/compute-options> (дата обращения: 05.07.2026).
3. Microsoft Learn. Шаблон Circuit Breaker // Azure Architecture Center. — 2025. — URL: <https://learn.microsoft.com/ru-ru/azure/architecture/patterns/circuit-breaker> (дата обращения: 05.07.2026).
4. Microsoft Learn. Асинхронное взаимодействие на основе сообщений // .NET Architecture. — 2023. — URL: <https://learn.microsoft.com/ru-ru/dotnet/architecture/microservices/architect-microservice-container-applications/asynchronous-message-based-communication> (дата обращения: 05.07.2026).
5. Microsoft Learn. Принципы проектирования надёжности // Azure Well-Architected Framework. — 2025. — URL: <https://learn.microsoft.com/ru-ru/azure/well-architected/reliability/principles> (дата обращения: 05.07.2026).
6. Fowler M., Lewis J. Microservices. — 2014. — URL: <https://martinfowler.com/articles/microservices.html>.
7. Dragoni N., Dustdar S., Larsen S. T., Mazzara M. Microservices: Migration of a Mission Critical System. — 2017. — URL: <https://arxiv.org/abs/1704.04173>.
8. Al-Debagy O., Martinek P. A Comparative Review of Microservices and Monolithic Architectures. — 2019. — URL: <https://arxiv.org/abs/1905.07997>.
9. Hassan S., Bahsoon R., Kazman R. Microservice Transition and its Granularity Problem: A Systematic Mapping Study. — 2019. — URL: <https://arxiv.org/abs/1903.11665>.
10. Taibi D., Lenarduzzi V., Pahl C. Continuous Architecting with Microservices and DevOps: A Systematic Mapping Study. — 2019. — DOI: 10.1007/978-3-030-29193-8_7. — URL: <https://arxiv.org/abs/1908.10337>.
11. Lenarduzzi V., Lomio F., Saarimäki N., Taibi D. Does Migrate a Monolithic System to Microservices Decrease the Technical Debt? — 2020. — DOI: 10.1016/j.jss.2020.110710. — URL: <https://arxiv.org/abs/1902.06282>.
12. Waseem M., Liang P., Shahin M. A Systematic Mapping Study on Microservices Architecture in DevOps. — 2020. — URL: <https://arxiv.org/abs/2008.07729>.
13. Montesi F., Peressotti M., Picotti V. Sliceable Monolith: Monolith First, Microservices Later. — 2021. — URL: <https://arxiv.org/abs/2103.09518>.
14. Ayas H. M., Leitner P., Hebig R. Facing the Giant: a Grounded Theory Study of Decision-Making in Microservices Migrations. — 2021. — DOI: 10.48550/arXiv.2104.00390. — URL: <https://arxiv.org/abs/2104.00390>.
15. Andrade B., Santos S., Silva A. R. From Monolith to Microservices: Static and Dynamic Analysis Comparison. — 2022. — URL: <https://arxiv.org/abs/2204.11844>.
16. Borges M. C., Bauer J., Werner S., Gebauer M., Tai S. Informed and Assessable Observability Design Decisions in Cloud-native Microservice Applications. — 2024. — URL: <https://arxiv.org/abs/2403.00633>.

Применение чат-ботов для автоматизации поддержки пользователей

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Чат-боты применяются в поддержке пользователей для обработки повторяющихся обращений, первичной классификации проблемы и выдачи типовых инструкций. Цель статьи — рассмотреть, в каких сценариях чат-бот действительно автоматизирует поддержку, а где он не сокращает число шагов до решения. В работе анализируются сценарные

боты и боты на основе искусственного интеллекта, распознавание намерений, база знаний, передача диалога оператору, метрики качества и риски ошибочных ответов. Сделан вывод, что чат-бот полезен там, где вопрос повторяется, ответ формализован, а граница передачи человеку описана заранее.

Ключевые слова: чат-боты, поддержка пользователей, автоматизация поддержки, распознавание намерений, база знаний, эскалация, разговорный ИИ, система учета заявок.

Поддержка пользователей часто сталкивается с большим числом повторяющихся обращений: восстановление доступа, статус заявки, настройка уведомлений, ошибка входа, поиск инструкции, изменение контактных данных. Если каждый такой вопрос обрабатывает оператор, часть рабочего времени уходит не на сложные случаи, а на повторение одинаковых ответов. Чат-бот снижает эту нагрузку, когда он принимает обращение, уточняет параметры и возвращает пользователю инструкцию или результат без участия человека.

По архитектуре чат-боты можно разделить на сценарные и основанные на искусственном интеллекте. Сценарный бот работает по заранее заданным правилам: пользователь выбирает пункт меню или вводит фразу, которая сопоставляется с правилом. Такой бот предсказуем, но плохо переносит свободные формулировки. Бот на основе искусственного интеллекта использует методы обработки естественного языка: выделяет намерение пользователя, извлекает сущности и подбирает ответ из базы знаний или генерирует его. Обзор развития чат-ботов показывает, что современные решения объединяют несколько подходов: шаблоны, машинное обучение, диалоговое состояние и внешние сервисы [4].

Ключевой элемент поддержки — распознавание намерения пользователя. Фраза «не могу войти», «забыл пароль» и «не приходит код» могут относиться к одному сценарию восстановления доступа, но требовать разных уточнений. Бот должен определить намерение, собрать недостающие данные и не задавать лишние вопросы. Если он каждый раз просит пользователя описать проблему заново, автоматизация не сокращает время обращения. Поэтому проектирование чат-бота начинается не с выбора платформы, а с анализа истории обращений и группировки типовых сценариев.

База знаний определяет качество ответов. Если статьи устарели, противоречат друг другу или написаны для операторов, бот будет выдавать неполные инструкции. В поддержке важна связка: вопрос пользователя -> проверенный фрагмент базы знаний -> понятный ответ -> ссылка на подробную инструкцию. IBM описывает чат-ботов как программные приложения, которые имитируют человеческий разговор и могут использоваться для поддержки клиентов [1]. Но сама технология диалога не гарантирует полезный ответ: его обеспечивает актуальная база знаний и процесс ее обновления.

Большое значение имеет эскалация к человеку. Пользователь должен понимать, когда бот не справился и что произойдет дальше. Передача оператору нужна при конфликте, сложной технической ошибке, финансовом во-

просе, работе с персональными данными или нескольких неудачных попытках распознать намерение. Эскалация должна передавать оператору историю диалога, уже собранные данные и классификацию проблемы. Иначе пользователь повторяет информацию, а бот не сокращает, а увеличивает путь к решению.

Чат-бот может автоматизировать ответы и отдельные операции. Например, он может проверить статус заявки, создать обращение в систему учета заявок, отправить ссылку для сброса пароля, собрать диагностические данные или записать пользователя на услугу. Для таких действий нужен безопасный контракт программного интерфейса приложения: бот должен знать, какие поля можно менять, как проверяется личность пользователя, какие операции требуют подтверждения и что делать при ошибке. Если бот получает слишком широкие права, ошибка распознавания может привести к изменению данных без достаточной проверки.

Метрики качества чат-бота должны отражать не число диалогов, а результат поддержки. Полезны доля обращений, решенных без оператора, среднее время до ответа, доля эскалаций, повторные обращения по той же теме, оценка пользователя после диалога и точность распознавания намерений. Если бот закрывает много диалогов, но пользователи после этого создают повторные заявки, автоматизация только скрывает проблему. Поэтому метрики чат-бота нужно связывать с метриками службы поддержки: соблюдением соглашения об уровне обслуживания, очередью нерешенных обращений, долей решений при первом обращении и удовлетворенностью.

Риск ошибочного ответа особенно заметен у генеративных ботов. Большая языковая модель может сформулировать уверенный текст, который не соответствует политике компании или реальному состоянию системы. Для поддержки это опасно: пользователь может получить неверную инструкцию, раскрыть лишние данные или выполнить действие, которое ухудшит проблему. Поэтому генеративный бот должен работать в ограниченном контуре: получать ответ из проверенной базы знаний, ссылаться на источник, не придумывать регламенты и передавать диалог человеку при недостатке данных.

Данные пользователей требуют отдельной защиты. В диалоге поддержки могут появляться номера договоров, телефоны, адреса, сведения о платежах и технические логи. Бот должен обрабатывать только те данные, которые нужны для сценария, а журнал диалога должен храниться по правилам организации. Если используется внешняя платформа искусственного интеллекта, нужно оценить, какие данные уходят поставщику, где они хранятся и ис-

пользуются ли для обучения. Без этого чат-бот становится новым каналом утечки, а не инструментом поддержки.

Внедрение чат-бота лучше начинать с узкой области. Например, выбрать пять самых частых обращений, где ответы стабильны и не требуют сложной диагностики. После запуска нужно анализировать неудачные диалоги, пополнять базу знаний, уточнять намерения пользователей и добавлять правила эскалации. Такой итерационный подход безопаснее, чем попытка сразу заменить первую линию поддержки. Бот должен расширяться только там, где статистика показывает сокращение времени решения и отсутствие роста повторных обращений.

Качество диалога зависит от того, как бот задает уточняющие вопросы. Если он сразу просит длинное описание проблемы, пользователь повторяет то, что уже написал в первом сообщении. Лучше задавать короткие вопросы по одному параметру: устройство, версия приложения, номер заявки, текст ошибки, время возникновения проблемы. Такой порядок помогает собрать данные для диагностики и не превращает диалог в анкету. Если часть данных уже известна из профиля пользователя, бот не должен спрашивать ее повторно.

Для службы поддержки важна прозрачность границ автоматизации. Пользователь должен видеть, что общается с ботом, какие действия бот может выполнить и как

перейти к оператору. Скрытая имитация человека снижает доверие, особенно когда бот ошибается или не понимает вопрос. Ясная маркировка канала, короткие ответы и заметная команда для передачи оператору делают автоматизацию спокойнее для пользователя и уменьшают раздражение при сложных обращениях.

Чат-боты полезны и для внутренней аналитики поддержки. Классификация обращений показывает, какие инструкции непонятны, какие ошибки чаще появляются после релизов и где пользователи не находят нужную функцию в интерфейсе. Если команда продукта регулярно анализирует темы диалогов, бот становится источником обратной связи, а не только фильтром перед оператором. Это помогает исправлять первопричины обращений: текст ошибки, форму настройки, документацию или сам бизнес-процесс.

Таким образом, чат-боты полезны для автоматизации поддержки пользователей, если задача повторяется, данные для ответа доступны, а сценарий имеет понятные границы. Они сокращают нагрузку на операторов, ускоряют первичный ответ и помогают собрать информацию до эскалации. Но чат-бот не должен становиться единственным каналом решения сложных проблем. Его место — в связке с базой знаний, системой учета заявок, программным интерфейсом приложения, оператором и метриками качества поддержки.

Литература:

1. IBM. What is a chatbot? // IBM Think. — 2026. — URL: <https://www.ibm.com/think/topics/chatbots> (дата обращения: 05.07.2026).
2. Microsoft Learn. Bot Framework documentation // Microsoft. — 2026. — URL: <https://learn.microsoft.com/en-us/azure/bot-service/> (дата обращения: 05.07.2026).
3. Google Cloud. Dialogflow documentation // Google Cloud. — 2026. — URL: <https://cloud.google.com/dialogflow/docs> (дата обращения: 05.07.2026).
4. Al-Amin Md., Ali M. S., Salam A., Khan A., Ali A., Ullah A., Alam Md. N., Chowdhury S. K. History of generative Artificial Intelligence (AI) chatbots: past, present, and future development. — 2024. — URL: <https://arxiv.org/abs/2402.05122>.
5. Behrooz M., Ngan W., Lane J., Morse G., Babcock B., Shuster K., Komeili M., Chen M., Kambadur M., Boureau Y.-L., Weston J. The HCI Aspects of Public Deployment of Research Chatbots: A User Study, Design Recommendations, and Open Challenges. — 2023. — URL: <https://arxiv.org/abs/2306.04765>.
6. Chaidrata A., Shafeeu M. I., Chew S. K., Chen Z., Cham J. S., Yong Z. L., Yap U. H., Bahrin D. I. B. K. Intent Matching based Customer Services Chatbot with Natural Language Understanding. — 2022. — URL: <https://arxiv.org/abs/2202.00480>.
7. Zendesk. Customer service chatbot guide // Zendesk. — 2026. — URL: <https://www.zendesk.com/service/messaging/chatbot/> (дата обращения: 05.07.2026).
8. Yandex Cloud Documentation. YandexGPT API and assistants // Yandex Cloud. — 2026. — URL: <https://yandex.cloud/ru/docs/foundation-models/> (дата обращения: 05.07.2026).

Методы балансировки нагрузки в распределенных веб-системах

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Балансировка нагрузки в распределенной веб-системе решает две связанные задачи: распределяет запросы между несколькими экземплярами приложения и ограничивает влияние отказа одного узла на работу всей системы. Цель статьи — сравнить основные методы балансировки нагрузки и показать, в каких условиях каждый метод помогает

управлять временем ответа, доступностью и эксплуатационной сложностью. Рассмотрены статические и динамические алгоритмы, DNS-балансировка через систему доменных имен, балансировка на транспортном и прикладном уровнях, проверки состояния, привязка сессии и связь балансировщика с автоматическим масштабированием. Сделан вывод, что балансировка нагрузки не заменяет оптимизацию приложения, но задает правило маршрутизации трафика и сценарий деградации при перегрузке или отказе.

Ключевые слова: балансировка нагрузки, распределенные веб-системы, обратный прокси, DNS-балансировка, Kubernetes, NGINX, HAProxy, Maglev, автоматическое масштабирование.

Распределенная веб-система обычно состоит из нескольких экземпляров приложения, прокси, базы данных, кэша, очередей и внешних сервисов. Когда пользовательский запрос приходит в такую систему, его нужно направить к экземпляру, который может обработать его без лишней задержки. Если все запросы попадают на один сервер, остальные ресурсы простаивают, а перегруженный узел становится точкой отказа. Балансировщик нагрузки устраняет эту концентрацию: он принимает входящий трафик и выбирает сервер приложения по заранее заданному правилу.

Самый простой класс методов — статические алгоритмы. Круговое распределение передает запросы серверам по очереди. Взвешенное круговое распределение добавляет веса: более мощный сервер получает больше запросов, чем слабый. Эти методы просты в настройке и хорошо подходят для однородных экземпляров, которые выполняют похожие операции. В документации NGINX круговое распределение указано как базовый метод балансировки группы вышестоящих серверов, а метод наименьшего числа соединений и привязка по IP-адресу рассматриваются как альтернативы для других сценариев [1]. Ограничение статического подхода состоит в том, что он слабо учитывает текущую занятость сервера. Если один запрос обрабатывается 10 мс, а другой запускает долгую операцию, простая очередь распределения перестает отражать фактическую нагрузку.

Динамические алгоритмы учитывают состояние серверных узлов. Метод наименьшего числа соединений выбирает сервер с меньшим числом активных соединений. Такой подход полезен там, где длительность запросов заметно различается: долгие соединения не должны случайно накапливаться на одном экземпляре. HAProxy также поддерживает несколько алгоритмов распределения: круговое распределение, метод наименьшего числа соединений и привязку к источнику запроса, а выбор зависит от типа трафика и требований к привязке клиента [2]. Однако динамический метод сам создает накладные расходы: балансировщик должен хранить состояние соединений, обновлять счетчики и корректно реагировать на закрытие долгих запросов.

Отдельно стоит выделить алгоритм выбора из двух случайных вариантов. Его идея проста: балансировщик случайно выбирает два сервера и отправляет запрос к менее загруженному из них. В классической работе Mitzenmacher показано, что такая схема резко уменьшает дисбаланс по сравнению с полностью случайным выбором [5].

Для веб-систем это важно потому, что алгоритм не требует полного глобального знания о состоянии всех узлов, но снижает риск отправить запрос на перегруженный экземпляр. Такой компромисс полезен в крупных кластерах, где полное измерение состояния каждого серверного узла становится дорогим.

Балансировка через систему доменных имен работает раньше, чем запрос попадет к приложению. DNS возвращает разные IP-адреса для одного доменного имени, а клиент подключается к одному из них. Этот метод прост и хорошо подходит для географического распределения, но управляется временем жизни записи и кэшированием DNS-ответов. Если адрес уже закэширован у клиента или резолвера, мгновенно вывести отказавший узел из обращения трудно. Поэтому DNS-балансировку обычно используют как внешний слой распределения между регионами или крупными точками входа, а внутри региона ставят балансировщики транспортного и прикладного уровней.

Балансировка на четвертом уровне модели OSI (модель взаимодействия открытых систем) работает с TCP (протокол управления передачей) или UDP (протокол пользовательских датаграмм). Она быстрее, потому что не анализирует HTTP-содержимое запроса. Такой слой подходит для высоконагруженных входных точек, где нужно распределить поток соединений и сохранить малую задержку. Балансировка на седьмом уровне анализирует HTTP-заголовки, адрес запроса, клиентские файлы с данными сеанса и иногда тело запроса. Это позволяет направлять `/api` к одному пулу сервисов, статические файлы — к другому, а административные запросы — к третьему. Цена прикладного подхода состоит в большем объеме обработки: балансировщик становится частью маршрута приложения и требует аккуратной настройки таймаутов, заголовков и правил безопасности.

В распределенной системе балансировка нагрузки не отделима от проверок состояния. Если балансировщик не проверяет состояние серверного узла, он может продолжать отправлять запросы на процесс, который уже не принимает соединения или отвечает с ошибкой. Проверка может быть сетевой, например открытие TCP-соединения, или прикладной, например HTTP-запрос к `/health`. Прикладная проверка точнее, но ее нужно проектировать аккуратно: если конечная точка проверки обращается к базе данных и внешним сервисам, то временная проблема зависимого сервиса может вывести из пула все экземпляры приложения. Поэтому проверка состояния должна отра-

жать способность экземпляра принять запрос, но не превращаться в тяжелый тест всей системы.

Состояние пользователя усложняет балансировку. Если веб-приложение хранит сессию в памяти процесса, повторный запрос того же клиента должен попасть на тот же экземпляр. Для этого используют привязку сессии: балансировщик привязывает клиента к серверному узлу по клиентскому файлу с данными сеанса, IP-адресу или другому признаку. В Kubernetes для службы также предусмотрены механизмы распределения трафика и политики, влияющие на то, как запрос попадает к контейнерным группам [3]. Но привязка сессии снижает свободу балансировщика: перегруженный сервер нельзя так же легко разгрузить, если к нему уже привязаны активные пользователи. Более устойчивый вариант — вынести состояние сессии во внешнее хранилище или использовать токены без состояния, когда это допустимо по требованиям безопасности.

Крупные системы используют несколько слоев балансировки. В работе Google Maglev описан программный балансировщик нагрузки, рассчитанный на высокую пропускную способность, отказоустойчивость и согласованное распределение трафика между серверными узлами [4]. Практический вывод из таких архитектур состоит в том, что балансировщик — не одиночный компонент перед приложением, а часть цепочки: глобальная система доменных имен или единый сетевой адрес направляет пользователя в регион, региональный балансировщик выбирает кластер, а внутренний прокси маршрутизирует за-

прос к сервису. Каждый слой решает свою задачу и имеет собственный сценарий отказа.

Балансировка тесно связана с автоматическим масштабированием. Если метрики показывают рост нагрузки на центральный процессор, памяти, числа запросов или задержки, оркестратор может добавить новые экземпляры приложения. Но новый экземпляр полезен только после того, как балансировщик включит его в пул и убедится, что он готов принимать трафик. Если проверка готовности настроена неверно, система начинает отправлять запросы в контейнер до завершения инициализации, и пользователь видит ошибки. Поэтому масштабирование требует согласованной настройки метрик, пределов ресурсов, проверок готовности и правил балансировки.

Выбор метода балансировки зависит от характера нагрузки. Для однородных HTTP-запросов часто достаточно кругового или взвешенного кругового распределения. Для долгих соединений и неодинаковых запросов лучше подходит метод наименьшего числа соединений или выбор из двух случайных вариантов. Для географического распределения нужен слой системы доменных имен или единого сетевого адреса. Для маршрутизации по адресу запроса, заголовкам и версиям программного интерфейса приложения нужен балансировщик прикладного уровня. Если приложение хранит состояние в процессе, возникает потребность в привязке сессии, но она увеличивает риск неравномерной нагрузки. Поэтому балансировку нагрузки следует рассматривать как инженерный механизм управления трафиком, а не как универсальное средство ускорения системы.

Литература:

1. NGINX Documentation. HTTP Load Balancing // NGINX. — 2026. — URL: https://nginx.org/en/docs/http/load_balancing.html (дата обращения: 05.07.2026).
2. HAProxy Technologies. HAProxy Configuration Manual. Load balancing algorithms // HAProxy Documentation. — 2026. — URL: <https://docs.haproxy.org/> (дата обращения: 05.07.2026).
3. Kubernetes Documentation. Services, Load Balancing, and Networking // Kubernetes. — 2026. — URL: <https://kubernetes.io/docs/concepts/services-networking/> (дата обращения: 05.07.2026).
4. Eisenbud D. E., Yi C., Contavalli C., Smith C., Kononov R., Mann-Hielscher E., Cilingiroglu A., Cheyney B., Shang W., Hosein J. D. Maglev: A Fast and Reliable Software Network Load Balancer. — 2016. — URL: <https://research.google/pubs/maglev-a-fast-and-reliable-software-network-load-balancer/>.
5. Mitzenmacher M. The Power of Two Choices in Randomized Load Balancing. — 2001. — URL: <https://www.eecs.harvard.edu/~michaelm/postscripts/mythesis.pdf>.
6. Dean J., Barroso L. A. The Tail at Scale. — 2013. — URL: <https://research.google/pubs/the-tail-at-scale/>.
7. Cloudflare Docs. Load Balancing // Cloudflare. — 2026. — URL: <https://developers.cloudflare.com/load-balancing/> (дата обращения: 05.07.2026).
8. Yandex Cloud Documentation. Network Load Balancer // Yandex Cloud. — 2026. — URL: <https://yandex.cloud/ru/docs/network-load-balancer/> (дата обращения: 05.07.2026).

Автоматическая генерация программного кода с использованием больших языковых моделей

Козырев Петр Михайлович, студент

Московский государственный технологический университет «СТАНКИН»

Автоматическая генерация программного кода с использованием больших языковых моделей переносит часть работы разработчика с ручного набора фрагментов на постановку задачи, отбор вариантов и проверку результата. Цель статьи — определить, какие задачи кодогенерации такие модели решают наиболее надёжно и какие ограничения необходимо учитывать при их применении в разработке. Рассмотрены генерация кода по естественно-языковому описанию, автодополнение, заполнение пропусков в существующем коде и многошаговое уточнение программы. Показано, что практическая ценность таких инструментов зависит от связки «запрос — сгенерированный вариант — тест — ревью», а не от самого факта получения правдоподобного фрагмента. Поэтому автоматическую генерацию кода следует рассматривать как способ подготовки кандидатов на решение, которые проходят инженерную проверку перед включением в проект.

Ключевые слова: большие языковые модели, генерация кода, программная инженерия, HumanEval, MBPP, CodeXGLUE, infilling, тестирование программ, ревью кода.

Автоматическая генерация программного кода с использованием больших языковых моделей возникла на пересечении обработки естественного языка, машинного обучения и классической задачи синтеза программ. Если традиционные средства разработки подсказывали идентификаторы, сигнатуры функций или шаблонные конструкции, то современные модели способны получать на вход описание задачи на естественном языке и формировать полноценный фрагмент программы. Меняется и интерфейс работы программиста, и сама постановка задачи: разработчик всё чаще описывает требуемое поведение, а затем проверяет, насколько предложенный моделью код соответствует этому поведению.

Технически такая генерация опирается на обучение моделей на больших корпусах исходного кода, документации, комментариев и текстовых описаний задач. В работе OpenAI Codex была показана возможность генерировать программы из docstring-описаний и оценивать результат по прохождению тестов HumanEval. Этот подход важен тем, что качество ответа определяется не внешней похожестью на правильное решение, а фактическим выполнением требований. Для программного кода это принципиально: фрагмент может выглядеть убедительно, но ошибаться на граничном случае, неверно обрабатывать исключение или использовать неподходящий алгоритм.

Существуют разные режимы автоматической генерации кода. Первый режим — генерация функции или программы по краткому описанию. Он характерен для бенчмарков MBPP, HumanEval и APPS, где модель получает условие задачи и должна выдать исполняемый код. Второй режим — автодополнение, когда модель продолжает уже начатый фрагмент с учётом локального контекста файла. Третий режим — заполнение пропусков в существующей программе. Модели InCoder и StarCoder показывают, что для реальной разработки важны продолжение кода слева направо и редактирование фрагмента с учётом строк до и после изменяемого участка.

Задача генерации кода отличается от генерации обычного текста тем, что результат можно проверять формальными средствами. Компилятор, интерпретатор, модульные тесты, статический анализатор и линтер дают разработчику обратную связь, которой нет в большинстве текстовых задач. Поэтому наиболее практичный сценарий применения больших языковых моделей состоит не в принятии первого ответа, а в построении цикла: сформулировать требование, получить несколько вариантов, выполнить тесты, отсеять неработающие решения и доработать оставшийся код вручную. В AlphaCode такой принцип был доведён до масштабной схемы генерации множества решений с последующей фильтрацией, что позволило приблизиться к задачам соревновательного программирования.

Наиболее заметные успехи кодогенерации связаны с короткими и хорошо специфицированными задачами. В таких условиях модель получает достаточно явный контракт: входные данные, ожидаемый результат и иногда примеры. Исследования Program Synthesis with Large Language Models и Evaluating Large Language Models Trained on Code показывают, что увеличение размера модели, обучение на коде и повторная выборка вариантов повышают долю решений, проходящих тесты. Однако этот результат не означает, что модель понимает задачу в инженерном смысле. Она может воспроизвести распространённый алгоритмический шаблон, но не учесть скрытое ограничение, которое в реальном проекте содержится в бизнес-логике, схеме данных или неявном API-контракте.

Специализированные наборы данных задают общий язык сравнения моделей. CodeXGLUE включает задачи понимания и генерации кода и позволяет сопоставлять решения по нескольким направлениям, а APPS проверяет способность решать задачи программирования разной сложности с использованием скрытых тестов. Такие наборы важны для научной оценки, но их нельзя напрямую переносить на промышленную разработку. В реальном проекте код редко существует как изолиро-

ванная функция. Он связан с зависимостями, конфигурацией, правами доступа, состоянием базы данных, соглашениями команды и требованиями сопровождения.

Поэтому граница применимости больших языковых моделей проходит там, где от отдельного фрагмента кода требуется знание широкого проектного контекста. Модель может написать корректную функцию сортировки, преобразователь формата или простой SQL-запрос. Значительно сложнее получить от неё изменение, которое одновременно учитывает миграцию базы данных, обратную совместимость API, обработку ошибок клиента, ограничения безопасности и наблюдаемость. Если эти требования не переданы явно в запросе или не доступны модели через контекст репозитория, сгенерированный код будет опираться на статистически вероятное продолжение запроса, а не на фактическую архитектуру системы.

Часть этого ограничения снимается многошаговым взаимодействием. CodeGen рассматривал синтез программ через несколько запросов, при котором задача уточняется последовательно, а не решается одним ответом. Для практики это означает, что разработчик может сообщать модели об ошибках тестов, ограничениях интерфейса и требованиях к стилю проекта. Такой режим ближе к работе с ассистентом-разработчиком: первичный вариант редко является окончательным, но последующие итерации позволяют приблизить его к рабочему решению.

Тип модели влияет на то, какие ошибки она будет делать в коде. Универсальная языковая модель способна генерировать программы, но специализированные модели, обученные на больших корпусах исходного кода, обычно лучше учитывают синтаксис, идиомы языков и типовые паттерны библиотек. CodeT5 использует представление идентификаторов и объединяет задачи понимания и генерации кода. StarCoder развивает открытые модели для программирования на основе набора данных The Stack, а DeepSeek-Coder показывает влияние масштабного предварительного обучения на коде и на задачах заполнения пропусков. Эти работы подтверждают, что кодогенерация стала самостоятельным направлением, где качество за-

висит от состава обучающих данных, длины контекста, способа дообучения и процедуры оценки.

Основной риск автоматической генерации кода связан с правдоподобностью результата. Большая языковая модель может предложить программу, которая компилируется, проходит простые тесты и выглядит стилистически приемлемо, но содержит ошибку в условии, не обрабатывает пустой ввод, нарушает требования безопасности или использует неподходящую зависимость. В задачах с пользовательскими данными, доступом к файловой системе, сетевыми запросами и авторизацией такие ошибки нельзя считать косметическими. Они переходят из области качества кода в область эксплуатационного риска.

Для уменьшения этого риска генерация должна включаться в управляемый процесс разработки. Минимальный контур проверки состоит из автоматических тестов, статического анализа, проверки зависимостей и ревью человеком. В более строгом варианте можно использовать генерацию нескольких независимых решений, сравнение результатов, тестирование свойств программы на наборах случайных входных данных и выполнение кода в изолированной среде. Такой подход не устраняет ошибок полностью, но меняет роль модели: она становится поставщиком кандидатов на решение, а не источником окончательной реализации.

Автоматическая генерация программного кода с использованием больших языковых моделей наиболее оправдана в задачах с ясным локальным контрактом, доступными тестами и ограниченным числом внешних зависимостей. Она помогает быстрее получить черновой вариант функции, подобрать типовой алгоритм, заполнить повторяющийся фрагмент и объяснить ошибку начинающему разработчику. Ограничение возникает там, где решение зависит от неявного контекста проекта: схемы данных, истории миграций, правил авторизации, соглашений команды или эксплуатационных требований. Поэтому развитие таких систем не отменяет инженерные практики, а усиливает значение требований, тестирования, архитектурного контекста и ответственности разработчика за итоговый код.

Литература:

1. Lu S., Guo D., Ren S., Huang J., Svyatkovskiy A., Blanco A., Clement C., Drain D., Jiang D., Tang D., Ge L., Shou L., Zhou L., Tufano M., Gong M., Zhou M., Duan N., Sundaresan N., Deng S. K., Fu S., Liu S. CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation. — 2021. — URL: <https://arxiv.org/abs/2102.04664>.
2. Hendrycks D., Basart S., Kadavath S., Mazeika M., Arora A., Guo E., Burns C., Puranik S., He H., Song D., Steinhardt J. Measuring Coding Challenge Competence With APPS. — 2021. — URL: <https://arxiv.org/abs/2105.09938>.
3. Austin J., Odena A., Nye M., Bosma M., Michalewski H., Dohan D., Jiang E., Cai C. J., Terry M., Le Q., Sutton C. Program Synthesis with Large Language Models. — 2021. — URL: <https://arxiv.org/abs/2108.07732>.
4. Chen M., Tworek J., Jun H., Yuan Q., Pinto H. P. O., Kaplan J., Edwards H., Burda Y., Joseph N., Brockman G., Ray A., Puri R., Krueger G., Petrov G., Khlaaf H., Sastry G., Mishkin P., Chan B., Gray S., Ryder N., Pavlov M., Power A., Kaiser L., Bavarian M., Winter C., Tillet P., Such F. P., Cummings D., Plappert M., Chantzis F., Barnes E., Herbert-Voss A., Guss W. H., Nichol A., Paino A., Tezak N., Tang J., Babuschkin I., Balaji S., Jain S., Saunders W., Hesse C., Carr A. N., Leike J., Achiam J., Misra V., Morikawa E., Radford A., Knight M., Brundage M., Murati M., Mayer K., Welinder P., McGrew B., Amodei D., McCandlish S., Sutskever I., Zaremba W. Evaluating Large Language Models Trained on Code. — 2021. — URL: <https://arxiv.org/abs/2107.03374>.

5. Wang Y., Wang W., Joty S., Hoi S. C. H. CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation. — 2021. — URL: <https://arxiv.org/abs/2109.00859>.
6. Xu F. F., Alon U., Neubig G., Hellendoorn V. J. A Systematic Evaluation of Large Language Models of Code. — 2022. — URL: <https://arxiv.org/abs/2202.13169>.
7. Nijkamp E., Pang B., Hayashi H., Tu L., Wang H., Zhou Y., Savarese S., Xiong C. CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis. — 2022. — URL: <https://arxiv.org/abs/2203.13474>.
8. Fried D., Aghajanyan A., Lin J., Wang S., Wallace E., Shi F., Zhong R., Yih W.-t., Zettlemoyer L., Lewis M. InCoder: A Generative Model for Code Infilling and Synthesis. — 2022. — URL: <https://arxiv.org/abs/2204.05999>.
9. Li Y., Choi D., Chung J., Kushman N., Schrittwieser J., Leblond R., Eccles T., Keeling J., Gimeno F., Dal Lago A., Hubert T., Choy P., de Masson d'Autume C., Babuschkin I., Chen X., Huang P.-S., Welbl J., Goyal S., Cherepanov A., Molloy J., Mankowitz D. J., Robson E. S., Kohli P., de Freitas N., Kavukcuoglu K., Vinyals O. Competition-Level Code Generation with AlphaCode. — 2022. — DOI: 10.1126/science.abq1158. — URL: <https://arxiv.org/abs/2203.07814>.
10. Li R., Allal L. B., Zi Y., Muennighoff N., Kocetkov D., Mou C., Marone M., Akiki C., Li J., Chim J., Liu Q., Zheltonozhskii E., Zhu T., Wang T., Dehaene O., Davaadorj M., Lamy-Poirier J., Monteiro J., Shliazhko O., Gontier N., Meade N., Zebaze A., Yee M. H., Umapathi L. K., Zhu J., Lipkin B., Oblokulov M., Wang Z., Murthy R., Stillerman J., Patel S. S., Abulkhanov D., Zocca M., Dey M., Zhang Z., Fahmy N., Bhattacharyya U., Yu W., Singh S., Lu X., Villegas P., Kunakov M., Zhdanov F., Romero M., Lee T., Timor N., Ding J., Schlesinger C., Schoelkopf H., Ebert J., Dao T., Mishra M., Gu A., Robinson J., Anderson C. J., Dolan-Gavitt B., Contractor D., Reddy S., Fried D., Bahdanau D., Jernite Y., Ferrandis C. M., Hughes S., Wolf T., Guha A., de Vries H., von Werra L. StarCoder: may the source be with you! — 2023. — URL: <https://arxiv.org/abs/2305.06161>.
11. Guo D., Zhu Q., Yang D., Xie Z., Dong K., Zhang W., Chen G., Bi X., Wu Y., Li Y. K., Luo F., Xiong Y., Liang W. DeepSeek-Coder: When the Large Language Model Meets Programming — The Rise of Code Intelligence. — 2024. — URL: <https://arxiv.org/abs/2401.14196>.

Квантовые технологии: от вычислений к сенсорике и коммуникациям

Мишекин Даниил Игоревич, студент

Московский государственный технологический университет «СТАНКИН»

В статье рассматривается развитие квантовых технологий от фундаментальных исследований к прикладному использованию в вычислениях, сенсорике и коммуникациях. Показано, что квантовые вычисления, квантовое зондирование и квантовая связь постепенно выходят за рамки теоретических экспериментов и начинают оказывать влияние на здравоохранение, энергетику, транспорт и национальную безопасность. Особое внимание уделяется тому, почему именно эти направления считаются наиболее перспективными, какие практические эффекты они уже дают и какие ограничения пока сдерживают массовое внедрение. Делается вывод, что квантовые технологии формируют новый технологический контур, в котором научная новизна напрямую связана с экономическими и инфраструктурными задачами.

Ключевые слова: квантовые технологии, квантовые вычисления, квантовая сенсорика, квантовые коммуникации, безопасность, инновации.

Квантовые технологии долгое время воспринимались как область фундаментальной физики, далекой от практики. Однако в последние годы ситуация меняется: государственные программы, инвестиции бизнеса и развитие инженерной базы переводят квантовую повестку из лабораторий в сферу прикладных решений. Сегодня речь идет не только о квантовых компьютерах, но и о сенсорике, связи и гибридных вычислительных платформах.

Наиболее заметным стало расширение области применения. Если раньше квантовые исследования ассоциировались главным образом с задачей ускорения сложных вычислений, то теперь они рассматриваются как основа для новых инструментов измерения, защищенной передачи данных и повышения точности в критически важных

отраслях. Именно поэтому квантовые технологии начинают восприниматься как инфраструктурный ресурс, а не только как научный эксперимент [1].

Квантовые вычисления остаются самым известным направлением. Их ключевая особенность заключается в использовании кубитов, которые способны находиться в суперпозиции состояний и взаимодействовать через квантовую запутанность. Это открывает возможность решать отдельные классы задач эффективнее, чем на классических компьютерах, особенно в области моделирования сложных систем.

На практике квантовые вычисления особенно интересны для химии, материаловедения, фармацевтики и энергетики. Например, они могут помочь в мо-

делировании молекул, поиске новых материалов для аккумуляторов, катализаторов и сверхпроводников. Для здравоохранения это означает более точный поиск молекул-кандидатов при разработке лекарств и ускорение работы с биохимическими моделями [2].

При этом квантовые компьютеры пока не являются универсальной заменой обычным вычислительным системам. Их развитие ограничено шумом, ошибками, сложностью масштабирования и высокой стоимостью оборудования. Поэтому в ближайшей перспективе речь идет скорее о гибридных системах, где квантовые и классические вычисления дополняют друг друга.

Не менее важным направлением стала квантовая сенсорика. В отличие от вычислений, где эффект часто ожидается в будущем, сенсорные технологии уже сегодня выглядят ближе к практическому внедрению. Квантовые датчики позволяют измерять магнитные, гравитационные, температурные и временные параметры с очень высокой точностью.

В транспортной и промышленной сферах квантовая сенсорика тоже имеет значительный потенциал [3]. Сверхточные измерения позволяют лучше контролировать положение объектов, состояние машин и параметры сложных систем. В долгосрочной перспективе это может изменить подход к мониторингу критической инфраструктуры, где цена ошибки особенно высока.

С практической точки зрения квантовая связь рассматривается как новый уровень защиты каналов передачи данных. Она не отменяет обычную криптографию, но дополняет ее там, где требуется повышенная устойчивость к перехвату и компрометации. В этом смысле квантовые коммуникации особенно важны для национальной безопасности и управления критически важной инфраструктурой.

Однако и здесь есть ограничения. Развертывание квантовых сетей требует больших инвестиций, специализированного оборудования и развитой телекоммуникационной базы. Поэтому распространение таких решений идет постепенно и сначала затрагивает наиболее чувствительные сегменты, а уже затем — более широкий рынок.

Главная особенность квантовых технологий сегодня заключается в том, что они перестают быть чисто исследовательской темой. Их влияние заметно в трех крупных зонах. Первая — здравоохранение, где важны более точная диагностика и ускорение разработки лекарств. Вторая — энергетика, где квантовые методы могут повысить эффективность моделирования, контроля и диагностики. Третья — национальная безопасность, где критичны квантовая связь и устойчивость к киберугрозам.

Несмотря на быстрый прогресс, до массового внедрения еще далеко. Основные препятствия — высокая стоимость, сложность инженерной реализации, нехватка специалистов и недостаточная зрелость многих решений. Кроме того, не все направления одинаково готовы к коммерциализации: сенсоры и коммуникации могут развиваться быстрее, чем универсальные квантовые вычислительные платформы [4].

Тем не менее общий тренд очевиден. Квантовые технологии переходят от демонстрации принципов к созданию работающих систем, способных решать конкретные задачи. Наиболее вероятным сценарием на ближайшие годы выглядит развитие гибридных решений и узкоспециализированных приложений, где квантовый эффект дает измеримое преимущество.

Квантовые технологии уже нельзя рассматривать только как теоретическую перспективу. Квантовые вычисления, сенсорика и коммуникации постепенно становятся частью прикладной технологической повестки, влияя на медицину, энергетику, транспорт и безопасность. Их значение состоит не только в потенциальном ускорении вычислений, но и в появлении новых способов измерения и защиты информации.

В ближайшие годы именно практическая полезность будет определять темпы их развития. Если удастся снизить стоимость, улучшить инженерную надежность и подготовить специалистов, квантовые технологии смогут занять устойчивое место в экономике и инфраструктуре. Таким образом, их развитие — это не просто научный прогресс, а формирование нового технологического уклада.

Литература:

1. Как Россия стала одним из лидеров квантовой гонки // Naked Science. 2025.
2. Квант милосердия. Новая дорожная карта по квантовым вычислениям // ComNews. 2025.
3. «Кванты» определены одним из ключей к достижению технологического лидерства // Atomic Energy. 2025.
4. Государство и бизнес объединятся для развития квантовых технологий // ComNews. 2025.

Сравнительный анализ подходов к управлению состоянием в React-приложениях Redux Toolkit, RTK Query и Zustand

Назаров Владимир Игоревич, студент
Московский авиационный институт (национальный исследовательский университет)

Петрова Евгения Игоревна, инженер-тестировщик
ГК «Иннотех» (г. Москва)

В статье рассматривается проблема управления состоянием в клиентских приложениях на React. Проведён сравнительный анализ трёх инструментов — Redux Toolkit, RTK Query и Zustand, различающихся объёмом шаблонного кода, назначением и областью применения. Показано, что выбор инструмента определяется не предпочтениями разработчика, а характером задачи: разделением клиентского и серверного состояния, масштабом приложения и требованиями к кэшированию данных. Сформулированы рекомендации по выбору подхода для проектов разного масштаба.

Ключевые слова: React, управление состоянием, Redux, Redux Toolkit, RTK Query, Zustand, серверное состояние, клиентское состояние, фронтенд-разработка.

Введение

Развитие фронтенд-разработки во многом определялось поиском предсказуемых способов управления состоянием интерфейса. С ростом сложности одностраничных приложений состояние перестало быть локальной характеристикой компонента и стало отдельным слоем архитектуры. React через хуки `useState` и `useReducer` не навязывает единый способ организации глобального состояния, что привело к появлению экосистемы сторонних решений — от классического Redux [5] до современных облегчённых библиотек.

Redux Toolkit (RTK) — официальный набор инструментов поверх Redux, сокращающий шаблонный код и включающий модуль RTK Query для работы с серверными данными. Zustand — минималистичная альтернатива, построенная на хуках и не требующая провайдера контекста. Цель статьи — сравнить эти инструменты по архитектурным принципам, сценариям применения и практическим компромиссам при выборе решения для конкретного проекта.

Клиентское и серверное состояние

Прежде всего стоит разграничить два вида состояния. Клиентское состояние полностью контролируется интерфейсом и не имеет внешнего источника истины — открыт ли список, значения ещё не отправленной формы. Серверное состояние получено из внешнего источника и лишь кэшируется на клиенте; оно может устаревать независимо от действий пользователя и требует повторной синхронизации.

Долгое время оба вида состояния хранили в одном хранилище Redux, из-за чего разработчику приходилось вручную писать логику загрузки, обработки ошибок и инвалидации кэша — функциональность, не зависящую от специфики приложения. Это привело к появлению специализированных библиотек для серверного состояния, включая RTK Query, React Query и SWR.

Redux Toolkit

Redux Toolkit сохраняет базовые принципы Redux — единственный источник истины, неизменяемость состояния, чистые функции-редьюсеры [1] — но упрощает их реализацию. Основная единица RTK — функция `createSlice`, объединяющая начальное состояние, редьюсеры и автогенерируемые действия. Внутри редьюсеров допустим мутлирующий на вид синтаксис за счёт встроенной библиотеки Immer, которая под капотом строит корректные неизменяемые обновления (листинг 1).

Кроме сокращения шаблонного кода, RTK подключает Redux DevTools и рекомендуемые middleware по умолчанию, предоставляет `createAsyncThunk` для асинхронных операций и `createEntityAdapter` для нормализации коллекций. RTK остаётся решением для управления клиентским состоянием сложных приложений, где предсказуемость изменений важнее минимального объёма кода.

RTK Query

RTK Query входит в состав Redux Toolkit, но решает иную задачу — управление данными с сервера. Вместо императивной загрузки через `thunk`-функции разработчик декларативно описывает эндпоинты API, а библиотека сама генерирует хуки, управляет состояниями загрузки, ошибок и кэширования (листинг 2).

Ключевой механизм RTK Query — система тегов [2]: после успешной мутации с определённым тегом все запросы с тем же тегом помечаются устаревшими и повторно выполняются. Это избавляет от необходимости вручную отслеживать, какие части кэша нужно обновить, и снижает риск рассинхронизации интерфейса с сервером.

Zustand

Zustand проектировался как компактное решение без обязательного провайдера контекста и без разделения ло-


```
import { createSlice, configureStore } from '@reduxjs/toolkit';

const counterSlice = createSlice({
  name: 'counter',
  initialState: { value: 0 },
  reducers: {
    incremented: (state) => { state.value += 1; },
    amountAdded: (state, action) => { state.value += action.payload; },
  },
});

export const { incremented, amountAdded } = counterSlice.actions;

const store = configureStore({
  reducer: { counter: counterSlice.reducer },
});
```

Рис. 1. Листинг 1 «Создание слайса состояния с помощью createSlice»

```
import { createApi, fetchBaseQuery } from '@reduxjs/toolkit/query/react';

export const catalogApi = createApi({
  reducerPath: 'catalogApi',
  baseQuery: fetchBaseQuery({ baseUrl: '/api/' }),
  tagTypes: ['Product'],
  endpoints: (builder) => ({
    getProducts: builder.query({
      query: (categoryId) => `products/?category=${categoryId}`,
      providesTags: ['Product'],
    }),
    updateProduct: builder.mutation({
      query: ({ id, ...patch }) => ({
        url: `products/${id}/`,
        method: 'PATCH',
        body: patch,
      }),
      invalidatesTags: ['Product'],
    }),
  }),
});

export const { useGetProductsQuery, useUpdateProductMutation } = catalogApi;
```

Рис. 2. Листинг 2 «Описание API с автоматической инвалидацией кэша по тегам»

гики на действия и редьюсеры [3]. Хранилище создаётся функцией create, а обращение к нему из компонентов — через обычный хук (листинг 3).

Минимализм даёт Zustand практические преимущества: малый размер, отсутствие дополнительных провай-

деров, простую интеграцию в уже существующие приложения. Обратная сторона — меньшая формализация изменений состояния, из-за чего Zustand чаще применяется для локальных фрагментов интерфейса, а не для всей бизнес-логики целиком.

```
import { create } from 'zustand';

const useUiStore = create((set) => ({
  selectedItem: null,
  isEditing: false,
  selectItem: (item) => set({ selectedItem: item }),
  toggleEditing: () => set((state) => ({ isEditing: !state.isEditing })),
}));

function PanelHeader() {
  const isEditing = useUiStore((state) => state.isEditing);
  const toggleEditing = useUiStore((state) => state.toggleEditing);
  return <button onClick={toggleEditing}>{isEditing ? 'Сохранить' :
'Редактировать'}</button>;
}
```

Рис. 3. Листинг 3 «Хранилище Zustand и его использование в компоненте»

Сравнительный анализ

Сопоставление инструментов по ключевым критериям представлено в таблице 1.

Redux Toolkit и Zustand решают, по существу, одну задачу — управление клиентским состоянием, — но рассчитаны на разный масштаб проектов: RTK предпочтителен, где важны строгая структура и инструменты отладки, Zustand — где приоритет отдан скорости разработки. RTK Query занимает отдельную нишу и решает задачу синхронизации с сервером, поэтому на практике нередко используется совместно с Zustand: RTK Query отвечает за серверные данные, Zustand — за состояние интерфейса.

Критерии выбора

Для приложений с интенсивным взаимодействием с API оправдано применение RTK Query независимо от выбора инструмента для клиентского состояния: встроенное кэширование и инвалидация по тегам сокращают объём инфраструктурного кода.

Выбор между Redux Toolkit и Zustand целесообразно связывать с масштабом команды и жизненным циклом проекта: в крупных приложениях строгая структура RTK облегчает онбординг и отладку, в небольших и средних проектах минимализм Zustand ускоряет разработку без заметной потери читаемости кода.

Таблица 1. Сравнение Redux Toolkit, RTK Query и Zustand

Критерий	Redux Toolkit	RTK Query	Zustand
Объём шаблонного кода	Средний (slice-подход снижает объём по сравнению с классическим Redux)	Низкий (эндпоинты описываются декларативно)	Минимальный
Назначение	Клиентское состояние приложения	Серверное состояние, кэширование запросов	Клиентское состояние приложения
Кривая обучения	Средняя	Средняя (требует понимания жизненного цикла кэша)	Низкая
Встроенное кэширование запросов	Отсутствует	Есть (инвалидация по тегам)	Отсутствует
Размер бандла	Средний	Средний (входит в состав RTK)	Малый (около 1 Кб в сжатом виде)
Инструменты разработчика	Redux DevTools	Redux DevTools	Ограниченная поддержка через middleware
Типичная область применения	Крупные приложения со сложной бизнес-логикой	Приложения с интенсивным взаимодействием с API	Небольшие и средние проекты, локальные UI-состояния

Ни один из инструментов не заменяет встроенные средства React (useState, useReducer, контекст) [4] для действительно локального состояния — их применение оправдано, когда состояние нужно совместно использовать в удалённых частях дерева компонентов.

Заключение

Практика управления состоянием в React-приложениях уходит от идеи единого хранилища в сторону разде-

ления ответственности между специализированными инструментами. Redux Toolkit остаётся решением для клиентского состояния крупных приложений; RTK Query закрывает задачу синхронизации с сервером; Zustand — минималистичная альтернатива для случаев, где важны скорость разработки и простота интеграции. Осознанная комбинация этих инструментов, а не следование единственному «универсальному» решению, позволяет выстраивать архитектуру состояния под реальные требования проекта.

Литература:

1. Redux Toolkit Documentation. — Текст: электронный // redux-toolkit.js.org: [сайт]. — URL: <https://redux-toolkit.js.org/> (дата обращения: 07.07.2026).
2. RTK Query Documentation. — Текст: электронный // [redux-toolkit.js.org](https://redux-toolkit.js.org/rtk-query/overview): [сайт]. — URL: <https://redux-toolkit.js.org/rtk-query/overview> (дата обращения: 07.07.2026).
3. Zustand Documentation. — Текст: электронный // zustand.docs.pmnd.rs: [сайт]. — URL: <https://zustand.docs.pmnd.rs/> (дата обращения: 08.07.2026).
4. React Documentation: Managing State. — Текст: электронный // [react.dev](https://react.dev/learn/managing-state): [сайт]. — URL: <https://react.dev/learn/managing-state> (дата обращения: 09.07.2026).
5. Abramov D., Clark A. Redux: A Predictable State Container for JS Apps. — Текст: электронный // GitHub: [сайт]. — URL: <https://github.com/reduxjs/redux> (дата обращения: 09.07.2026).

Вероятность ошибки приема символа в системах символьной синхронизации

Нгуен Суан Чыонг, кандидат технических наук, ассистент
Школа подготовки офицеров ВВС (г. Нячанг, Вьетнам)

В статье исследуются системы символьной синхронизации как важный элемент цифровых систем связи. Рассмотрено использование фазовой автоподстройки частоты (ФАПЧ) и её инерционных свойств для сглаживания шумовых флуктуаций фронтов импульсов. Показано влияние отношения сигнал/шум и параметров дискриминатора на дисперсию временных отклонений синхрои́мпульсов. Приведены расчёты вероятности ошибки при различных видах сигналов, что позволяет определить допустимые пределы нестабильности и требования к энергетике входных сигналов.

Ключевые слова: символьная синхронизация, вероятность ошибки.

Системы символьной синхронизации (symbol synchronization systems) — это ключевой элемент цифровых систем связи, обеспечивающий правильное выделение границ символов в принимаемом сигнале. Точность символьной синхронизации напрямую влияет на устойчивость демодуляции сигналов с относительной фазовой модуляцией, а значит — на общую надёжность передачи данных [1].

Для реализации символьной синхронизации применяются как аналоговые, так и цифровые методы. Классическим решением является ФАПЧ, которая формирует опорное колебание из принятого сигнала. В современных системах связи всё чаще применяются алгоритмы цифровой обработки сигналов, основанные на корреляции, фильтрации и адаптивных методах. Эти подходы позволяют повысить точность синхронизации даже при сильных помехах. Проблема синхронизации тесно связана с задачей исправления ошибок вставки и удаления символов, и требует комплексных решений на стыке теории кодирования и цифровой обработки сигналов [2].

Использование инерциальной ФАПЧ имеет важное значение. Именно благодаря этому тактовые импульсы на выходе схемы остаются постоянными независимо от случайных импульсов, появляющихся на входе временного дискриминатора, расположенного на границах символов. Инерционность схемы ФАПЧ должна быть достаточной, чтобы обеспечить генерирование символьных синхрои́мпульсов, когда во входном сигнале будет несколько подряд следующих одинаковых символов, так что несколько тактов подряд сигнал на вход временного различителя поступать не будет. С другой стороны, инерционность не должна быть слишком большой, чтобы система ФАПЧ успевала следить за меняющейся фазой импульсов на входе временного различителя, то есть чтобы динамические ошибки были малы.

Благодаря инерционным свойствам ФАПЧ происходит эффективное сглаживание флуктуаций фронтов импульсов, вызванных воздействием шумовых компонентов. Однако формируемые синхрои импульсы сохраняют случайные отклонения относительно их среднего положения. При высоких входных отношениях сигнал/шум данные отклонения подчиняются нормальному закону распределения. Дисперсия временных флуктуаций выходных импульсов схемы символьной синхронизации определяется как величиной входного отношения сигнал/шум, так и характеристиками используемого дискриминатора.

$$B = \frac{1}{\Delta f_K \tau},$$

Где: Δf_K — полоса кольца автоподстройки ФАПЧ

τ — длительность символа (рис. 1).

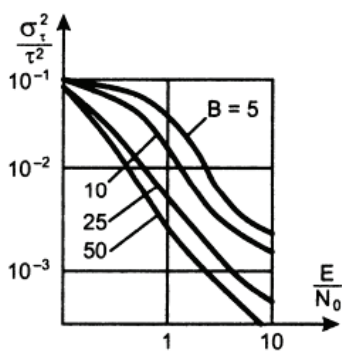


Рис. 1. Дисперсия флуктуации фронта импульса, выдаваемого схемой символьной синхронизации, содержащей ФАПЧ

При разработке системы символьной синхронизации необходимо оценить вероятность ошибки приема символа, обусловленную неточным знанием границ символа. Если смещение тактовых импульсов обозначить $\Delta\tau$, а распределение этого смещения — $P(\Delta\tau)$, то среднее значение вероятности ошибки при приеме символа p можно найти как

$$p = \int_{-\infty}^{\infty} P(\Delta\tau) p(\Delta\tau) d(\Delta\tau).$$

На рис. 2–4 приведены результаты расчетов по этой формуле для сигналов различной структуры и различных способов приема.

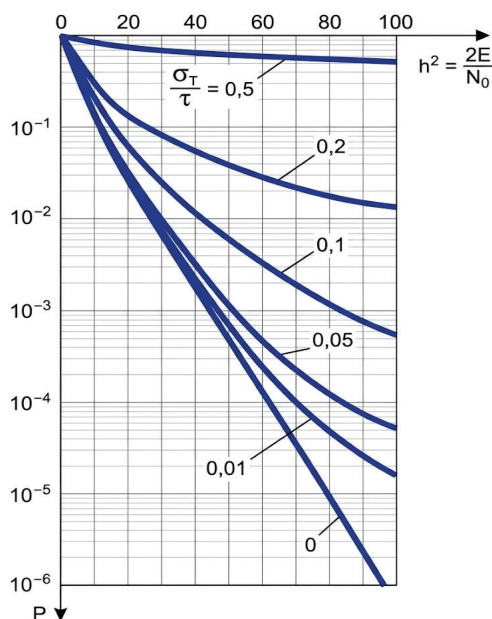


Рис. 2. Вероятность ошибки при АМн

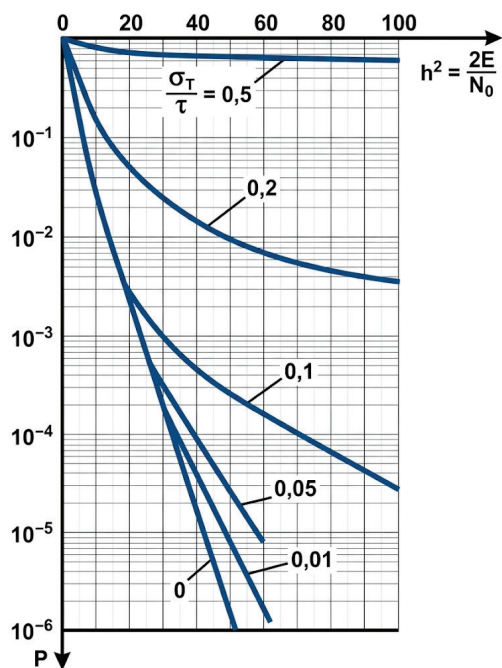


Рис. 3. Вероятность ошибки при ЧМн

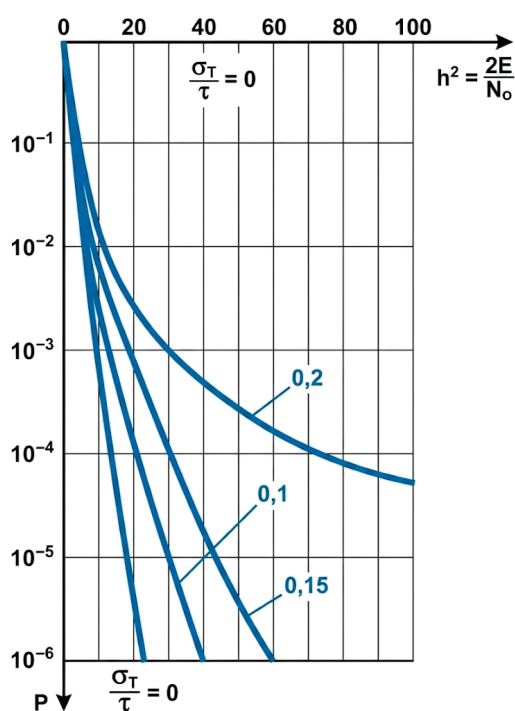


Рис. 4. Вероятность ошибки при ФМн

Рассмотрение зависимостей вероятности ошибки от входного отношения сигнал/шум позволяет сделать следующие выводы.

Для обеспечения погрешности передачи параметра порядка 1 % в системах должна достигаться вероятность ошибки приема символа порядка 10^{-4} . Если при идеальной синхронизации достигнута такая вероятность ошибки за счет создания определенного отношения сигнал-шум на входе системы, то появление нестабильности символьной частоты, характеризующееся относительным среднеквадратическим значением $\frac{\sigma_\tau}{\tau} = 0,05$, приводит к увеличению вероятности ошибки примерно до 10^{-3} для всех видов сигналов и методов приема. Такое увеличение вероятности ошибки не может быть признано допустимым. Оно может быть компенсировано соответствующим увеличением интенсивности сигнала.

Чтобы сохранить вероятность ошибки на уровне 10–4 при появлении в идеальной системе нестабильности символьных синхроимпульсов, характеризующейся $\frac{\sigma_\tau}{\tau} = 0.05$, необходимо примерно вдвое увеличить энергию входных сигналов для всех видов сигналов и методов приема. Дальнейшее увеличение нестабильности приводит к резкому увеличению необходимого отношения сигнал-шум, которое обеспечивает заданную вероятность ошибки.

Поэтому относительную нестабильность $\frac{\sigma_\tau}{\tau} = 0.05$ можно считать предельно допустимой. Как следует из рис. 2, $\frac{\sigma_\tau}{\tau} < 0.05$ может быть легко достигнута при отношении сигнал-шум на входе системы более 10.

Таким образом, рациональный выбор характеристик системы символьной синхронизации и величины отношения сигнал-шум на входе приемника позволяет обеспечить требуемую вероятность ошибки передачи символа.

Литература:

1. Филимонов В. А., Усов Н. А. Алгоритмы символьной синхронизации в системах цифровой связи. Вестник современных технологий связи, 12(3), с. 45–52, 2023.
2. Haeupler B., Shahrasbi A. Synchronization Strings and Codes for Insertions and Deletions—A Survey // IEEE Transactions on Information Theory, vol. 67, no. 6, pp. 3190–3206, June 2021.

Совершенствование бизнес-процесса обработки аукционных документов на основе технологии RPA

Пронин Кирилл Николаевич, руководитель группы разработки
ООО «Нейромед» (г. Москва)

В статье рассматриваются вопросы совершенствования бизнес-процесса обработки аукционных документов на основе технологии роботизированной автоматизации процессов (Robotic Process Automation, RPA). Исследование выполнено на примере предприятия ООО «Ново Флоре», осуществляющего закупочную деятельность на международных цветочных аукционах. Проведен анализ существующего бизнес-процесса обработки входящей аукционной документации, выявлены основные причины снижения его эффективности, связанные с высокой долей ручных операций, многократным переносом данных между корпоративными информационными системами и влиянием человеческого фактора. На основе моделирования процессов в состояниях AS-IS и TO-BE разработано проектное решение, предусматривающее применение программного робота для автоматизации ключевых операций обработки документов. Выполнен выбор платформы роботизации с использованием многокритериального подхода и реализован прототип программного робота на платформе PIX RPA. Практическая апробация разработанного решения показала сокращение продолжительности выполнения бизнес-процесса на 65 %, снижение стоимости обработки документов на 67 % и уменьшение количества ошибок до уровня менее 5 %. Полученные результаты подтверждают эффективность применения технологии RPA для совершенствования закупочных бизнес-процессов предприятий.

Ключевые слова: роботизированная автоматизация процессов, RPA, цифровая трансформация, закупочная деятельность, аукционные документы, бизнес-процесс, автоматизация.

Актуальность

В условиях цифровой трансформации экономики эффективность закупочной деятельности становится одним из ключевых факторов обеспечения устойчивого развития предприятий. Рост объемов электронных торгов, увеличение количества поставщиков и расширение информационного обмена сопровождаются усложнением процедур обработки закупочной документации, что повышает требования к скорости выполнения операций и качеству принимаемых решений.

Несмотря на широкое распространение корпоративных информационных систем класса ERP и SRM, значительная часть операций, связанных с обработкой входящей документации, продолжает выполняться вручную. К таким операциям относятся получение и регистрация документов, перенос информации между различными информационными

системами, создание карточек объектов учета, формирование сопроводительной документации и выполнение иных регламентированных действий. Высокая доля ручного труда приводит к увеличению продолжительности бизнес-процессов, росту вероятности возникновения ошибок и дополнительным трудовым затратам.

Одним из перспективных направлений повышения эффективности подобных процессов является применение технологии роботизированной автоматизации процессов (Robotic Process Automation, RPA). Использование программных роботов позволяет автоматизировать выполнение повторяющихся операций посредством взаимодействия с существующими пользовательскими интерфейсами корпоративных информационных систем без необходимости их существенной модификации. Это значительно сокращает сроки внедрения решений и снижает стоимость цифровой трансформации отдельных бизнес-процессов.

Практическая значимость применения технологии RPA особенно проявляется в закупочной деятельности предприятий, осуществляющих регулярное взаимодействие с электронными торговыми площадками и большим количеством поставщиков. В подобных условиях обработка аукционной документации представляет собой последовательность формализованных операций, обладающих высокой повторяемостью, что делает данный процесс целесообразным объектом роботизации.

Настоящее исследование выполнено на примере ООО «Ново Флоре» — предприятия, осуществляющего закупочную деятельность на международных цветочных аукционах. В качестве объекта совершенствования выбран бизнес-процесс приема и первичной обработки входящих аукционных документов, характеризующийся значительными временными затратами и высокой зависимостью от ручного выполнения операций.

Цель исследования

Целью исследования является совершенствование бизнес-процесса обработки аукционных документов посредством применения технологии роботизированной автоматизации процессов (RPA) для сокращения продолжительности выполнения операций, снижения трудозатрат и повышения эффективности закупочной деятельности предприятия.

Анализ бизнес-процесса обработки аукционных документов и выявление проблемных зон

В качестве объекта исследования рассмотрен бизнес-процесс обработки аукционных документов ООО «Ново Флоре», осуществляющего закупочную деятельность на международных цветочных аукционах. Специфика деятельности предприятия предполагает ежедневную обработку значительного объема документов, формируемых по итогам проведения торгов, включая инвойсы, платежные документы и сопроводительную информацию, необходимую для дальнейшего выполнения логистических и учетных операций.

Обработка аукционной документации является одним из ключевых этапов закупочного процесса, поскольку именно на основании полученных документов осуществляется регистрация результатов торгов, формирование учетных записей о приобретенной продукции, подготовка данных для оплаты поставщикам, а также создание логистических правил последующей доставки товаров. Ошибки или задержки на данном этапе оказывают влияние на весь последующий цикл закупочной деятельности, включая организацию международной логистики, таможенного оформления и складского учета [4], [5].

Ознакомимся с архитектурной схемой бизнес-процесса AS IS.

Исследование существующего бизнес-процесса показало, что его выполнение осуществляется с использованием нескольких корпоративных информационных систем, включая внутреннюю информационную систему Flowers, программные продукты 1С и корпоративную электронную почту. При этом значительная часть операций выполняется сотрудниками вручную посредством последовательного переноса информации между различными программными продуктами. Далее рассмотрим BPMN модель бизнес-процесса.

В общем виде исследуемый бизнес-процесс включает следующие этапы:

- получение аукционного инвойса после завершения торгов;
- регистрацию и проверку полученных документов;
- подготовку и передачу счетов на оплату;
- создание карточек приобретенных товаров во внутренней информационной системе;
- формирование логистических правил транспортировки;
- распределение товаров по приходам;
- подготовку товаросопроводительной документации для логистического подразделения.

Проведенное обследование процесса позволило установить, что наиболее трудоемкими являются операции, связанные с обработкой информации и ее многократным переносом между различными информационными системами. При выполнении данных операций сотрудники неоднократно вводят одни и те же сведения вручную, что приводит к увеличению продолжительности обработки документов и повышению вероятности возникновения ошибок.

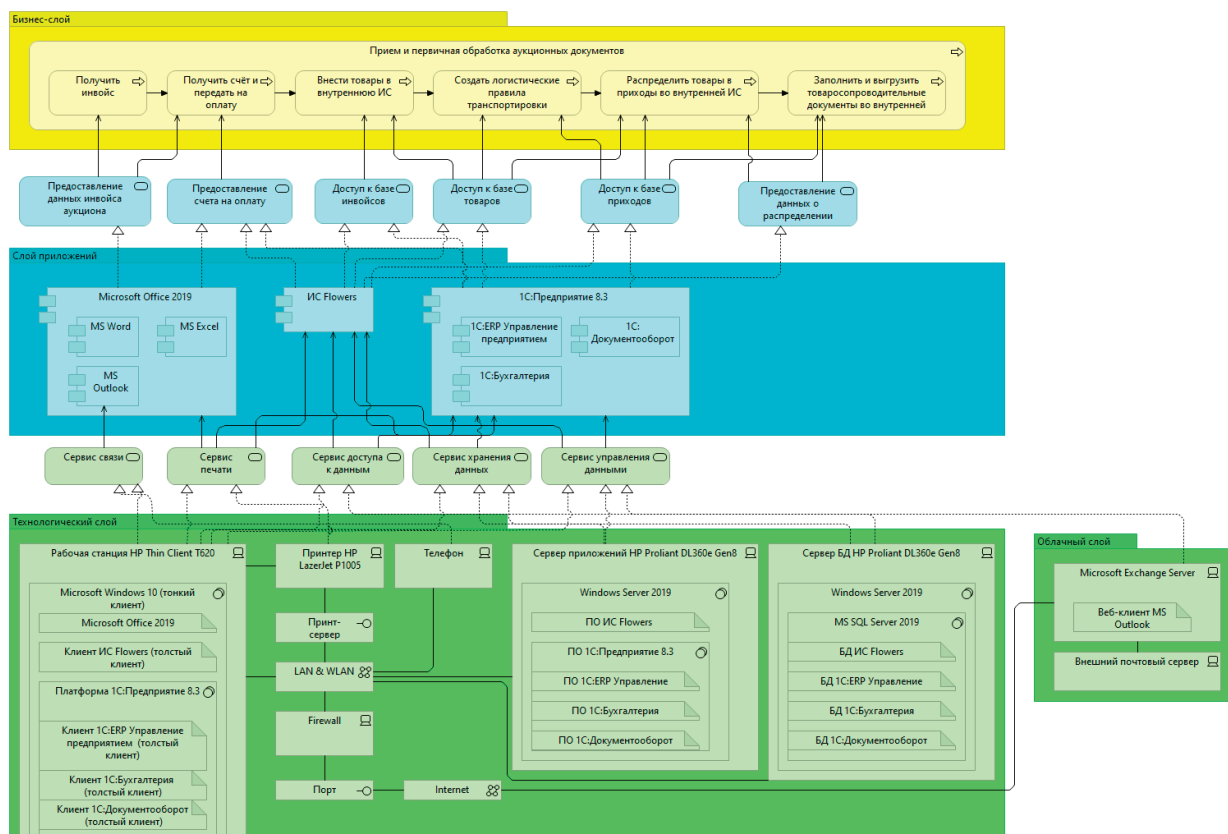


Рис. 1. Архитектурная схема бизнес-процесса AS IS

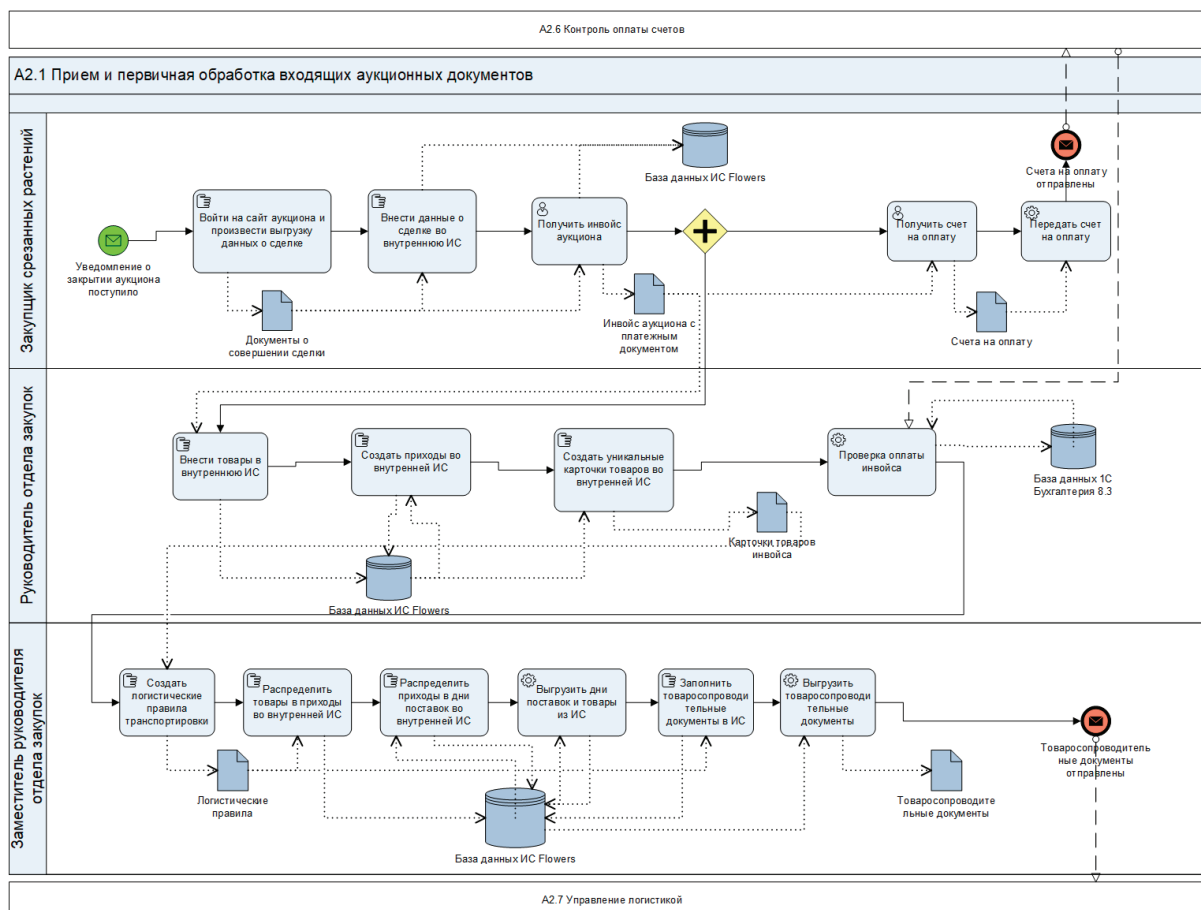


Рис. 2. BPMN бизнес-процесса AS IS

В ходе анализа были выявлены следующие основные недостатки существующего бизнес-процесса:

- значительная доля повторяющихся ручных операций при обработке аукционной документации;
- многократный перенос идентичных данных между различными информационными системами;
- отсутствие автоматизированного взаимодействия между используемыми программными продуктами;
- высокая зависимость качества выполнения процесса от квалификации исполнителя;
- увеличение продолжительности обработки документов при росте объема закупочной деятельности;
- риск возникновения ошибок при ручном вводе информации, способных привести к некорректному формированию учетных и логистических данных.

Особенностью рассматриваемого процесса является высокая степень формализованности большинства выполняемых операций. Последовательность действий определяется регламентами предприятия и практически не изменяется в зависимости от конкретного поставщика или результатов аукциона. Основная часть операций представляет собой последовательное получение документов, извлечение информации, проверку корректности данных и их внесение в корпоративные информационные системы по заранее установленным правилам [5], [6]. По результатам имитационного моделирования в среде Business Studio продолжительность бизнес-процесса для обработки одного аукциона — 44:10:18, а средняя стоимость 12 654 рубля. Далее построим таблицу выявленных проблем (узких мест).

Таблица 1. Выявленные узкие места

Выявленная проблема	Причина возникновения	Последствия
Ручной перенос данных	Отсутствие автоматизированной интеграции между системами	Увеличение трудозатрат
Повторный ввод информации	Использование нескольких информационных систем	Рост вероятности ошибок
Большое время обработки	Последовательное выполнение ручных операций	Снижение пропускной способности процесса
Зависимость от исполнителя	Высокая доля ручного труда	Нестабильность качества выполнения процесса
Ошибки при обработке документов	Человеческий фактор	Некорректные учетные и логистические данные

Таким образом, исследуемый бизнес-процесс характеризуется высокой повторяемостью операций, значительным объемом ручной обработки данных и использованием нескольких информационных систем без полной автоматизации обмена информацией между ними. Указанные особенности позволяют рассматривать процесс обработки аукционных документов как перспективный объект внедрения технологии роботизированной автоматизации процессов (RPA), применение которой способно существенно сократить продолжительность выполнения операций, уменьшить влияние человеческого фактора и повысить эффективность закупочной деятельности предприятия.

Выбор подхода к совершенствованию бизнес-процесса

Выявленные в ходе анализа недостатки исследуемого бизнес-процесса свидетельствуют о необходимости сокращения доли ручного труда, исключения повторного ввода информации и повышения скорости обработки аукционной документации. Решение указанных задач может быть достигнуто различными способами, отличающимися объемом требуемых изменений существующей информационной инфраструктуры предприятия, стоимостью внедрения и сроками реализации.

Одним из возможных подходов является разработка дополнительных модулей корпоративной информационной системы. Такой вариант позволяет реализовать необходимую функциональность непосредственно в существующем программном обеспечении и обеспечить глубокую интеграцию между используемыми информационными системами. Вместе с тем данный подход требует существенных затрат на проектирование, разработку и сопровождение программного обеспечения, а также внесения изменений в функционирующую корпоративную систему, что сопровождается дополнительными техническими и организационными рисками.

Другим вариантом является использование механизмов интеграции посредством программных интерфейсов (API). Такой подход обеспечивает автоматизированный обмен данными между информационными системами и позволяет исключить часть операций по ручному переносу информации. Однако его применение ограничивается наличием открытых интерфейсов взаимодействия у всех используемых программных продуктов. В рассматриваемом случае вну-

тренняя информационная система предприятия и отдельные внешние сервисы не предоставляют полного набора интерфейсов, необходимых для реализации требуемой автоматизации [3], [4], [5], [6].

Еще одним направлением совершенствования процесса является применение технологий интеллектуальной обработки документов, включая системы оптического распознавания символов (OCR) и алгоритмы искусственного интеллекта. Использование подобных решений позволяет автоматизировать извлечение информации из неструктурированных документов, однако не решает задачи дальнейшего выполнения последовательности регламентированных операций в корпоративных информационных системах. Вследствие этого подобные технологии целесообразно рассматривать как вспомогательные компоненты автоматизации.

Наиболее перспективным подходом в рассматриваемых условиях является применение технологии роботизированной автоматизации процессов (Robotic Process Automation, RPA). В отличие от традиционных методов автоматизации программные роботы выполняют действия, аналогичные действиям пользователя, взаимодействуя с существующими интерфейсами информационных систем без внесения изменений в их программный код. Это позволяет автоматизировать последовательности повторяющихся операций при сохранении существующей архитектуры корпоративной информационной среды.

Применительно к исследуемому бизнес-процессу использование технологии RPA обеспечивает автоматизацию получения и обработки аукционных документов, перенос информации между информационными системами, создание карточек товаров, формирование логистических данных и подготовку сопроводительной документации. При этом внедрение программного робота не требует модернизации существующих информационных систем предприятия и может быть реализовано в рамках действующей ИТ-инфраструктуры [1], [2].

Сравнительный анализ возможных подходов к совершенствованию бизнес-процесса представлен в таблице 2.

Таблица 2. Сравнительный анализ подходов совершенствования

Подход	Преимущества	Ограничения
Доработка корпоративной информационной системы	Высокая степень интеграции	Высокая стоимость разработки, необходимость изменения существующей системы
Интеграция посредством API	Автоматизированный обмен данными	Зависимость от наличия программных интерфейсов
OCR и интеллектуальная обработка документов	Автоматизация извлечения информации из документов	Не автоматизирует последующие операции бизнес-процесса
RPA	Автоматизация полного цикла пользовательских операций без изменения существующих систем	Требует стабильности пользовательских интерфейсов

Таким образом, с учетом особенностей исследуемого бизнес-процесса, необходимости минимизации изменений существующей информационной инфраструктуры и обеспечения быстрого внедрения наиболее рациональным решением является применение технологии роботизированной автоматизации процессов. На следующем этапе исследования выполняется выбор программной платформы для реализации роботизированного решения.

Реализация роботизированного решения и оценка результатов

На основании проведенного анализа существующих подходов к автоматизации бизнес-процессов в качестве инструмента совершенствования процесса обработки аукционных документов была выбрана технология роботизированной автоматизации процессов (RPA). Для реализации проектного решения использована отечественная программная платформа PIX RPA, обеспечивающая разработку программных роботов, взаимодействующих с корпоративными информационными системами на уровне пользовательского интерфейса.

Выбор платформы PIX RPA обусловлен результатами сравнительного анализа, выполненного по совокупности технических, функциональных и экономических критериев. Платформа обеспечивает интеграцию с используемыми на предприятии информационными системами, поддерживает работу с документами различных форматов, позволяет автоматизировать операции без внесения изменений в существующее программное обеспечение и соответствует требованиям импортонезависимости, предъявляемым к современным корпоративным ИТ-решениям [1], [2].

Перестроена логика исходного бизнес-процесса.

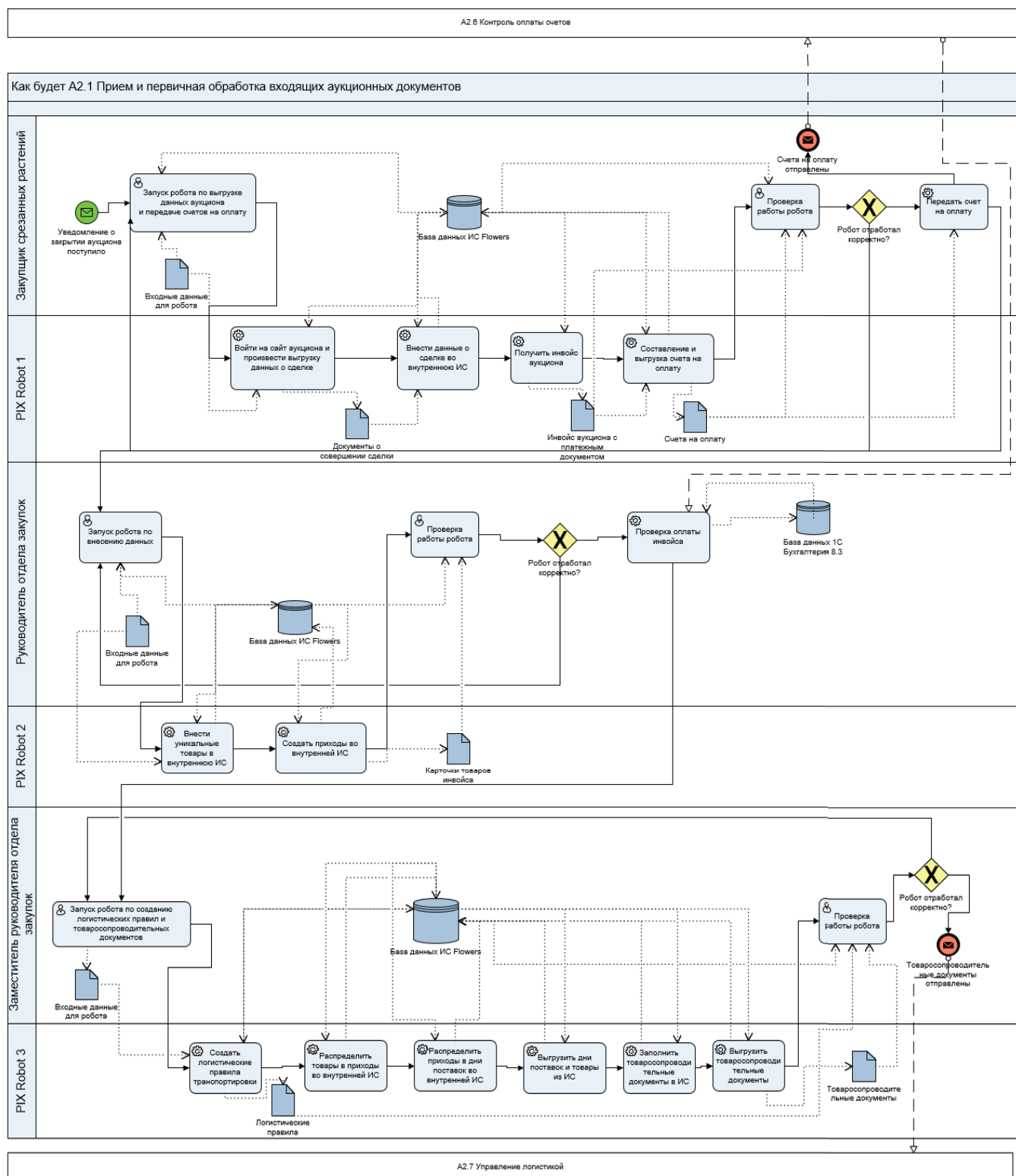


Рис. 3. BPMN бизнес-процесса TO BE

Решение на базе PIX RPA идеально встроилось в IT архитектуру предприятия.

В рамках исследования был разработан программный робот, автоматизирующий ключевые операции бизнес-процесса обработки аукционных документов.

Робот осуществляет получение входящей документации, обработку информации, перенос данных между корпоративными информационными системами, создание карточек товаров, формирование логистических данных и выполнение сопутствующих регламентированных операций, ранее выполнявшихся сотрудниками вручную.

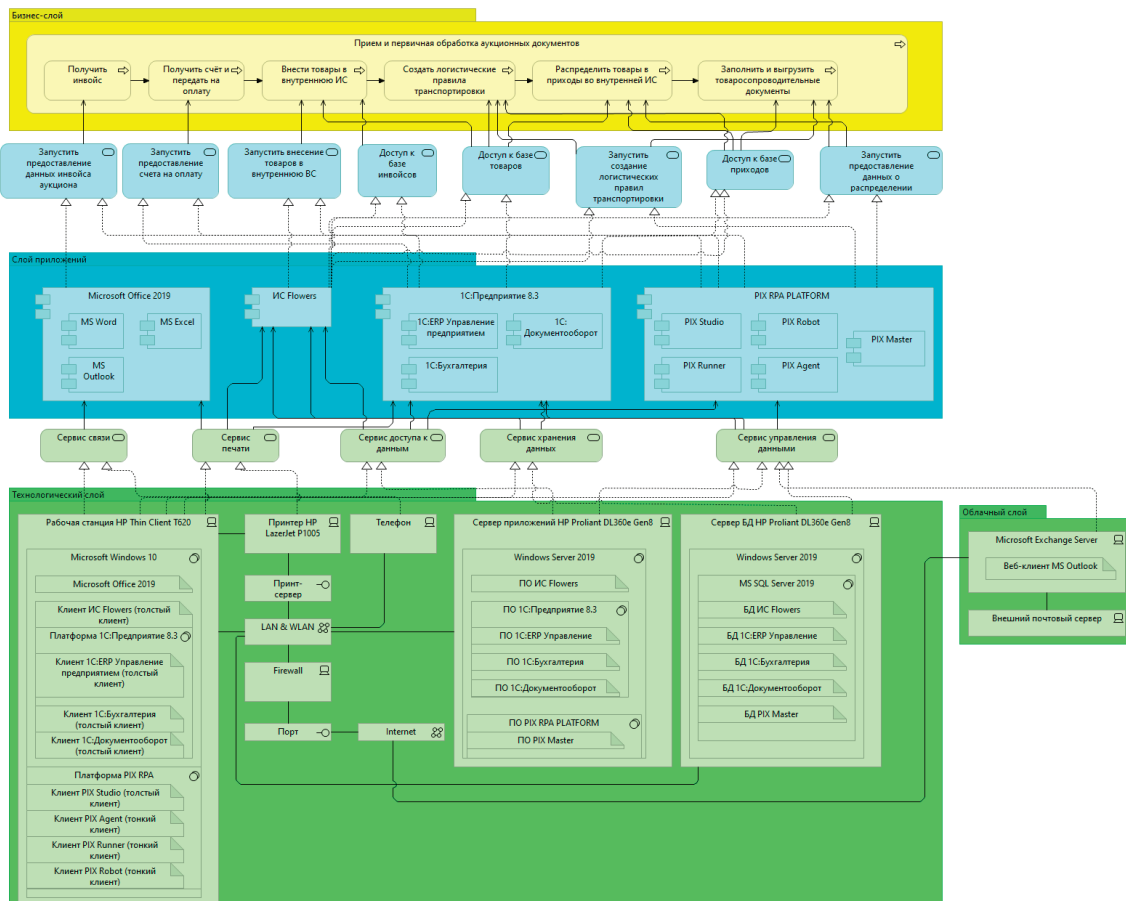


Рис. 4. Архитектурная схема бизнес-процесса TO BE

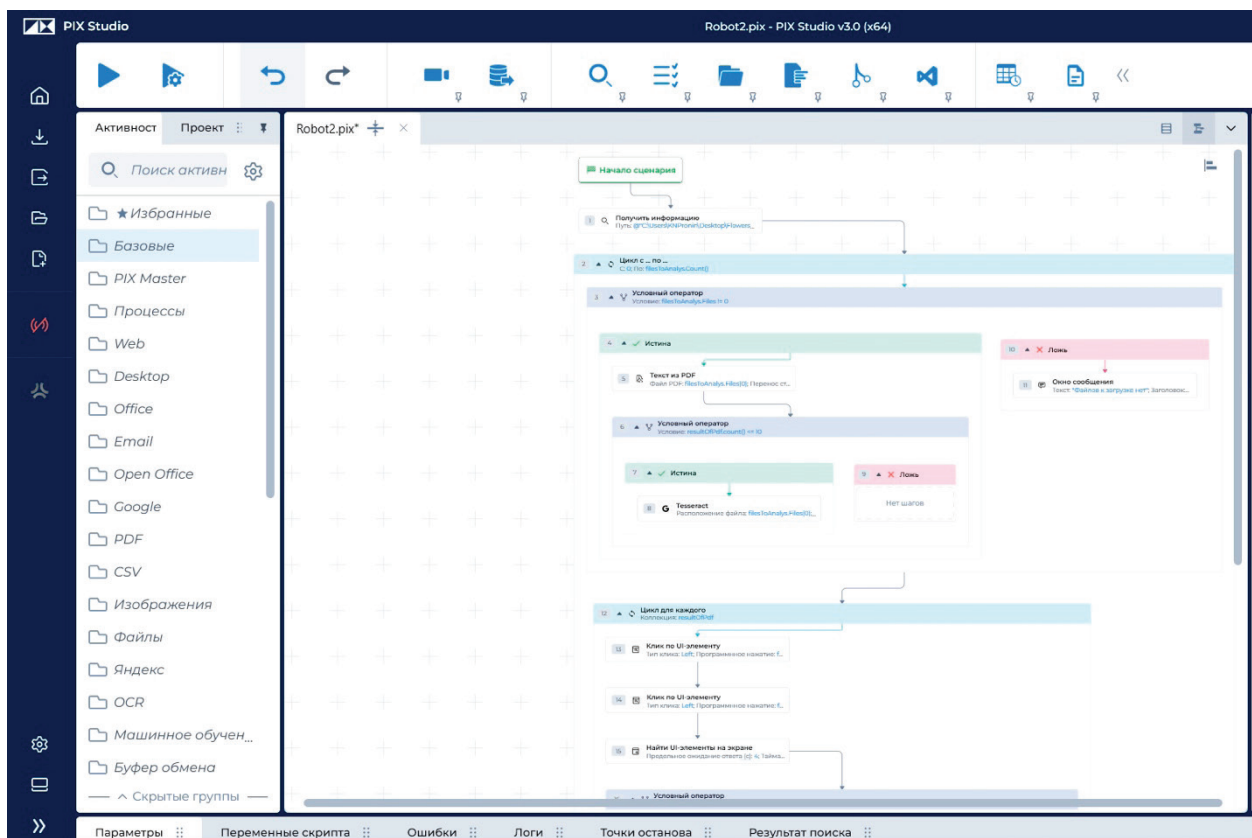


Рис. 5. Программный робот в среде PIX Studio

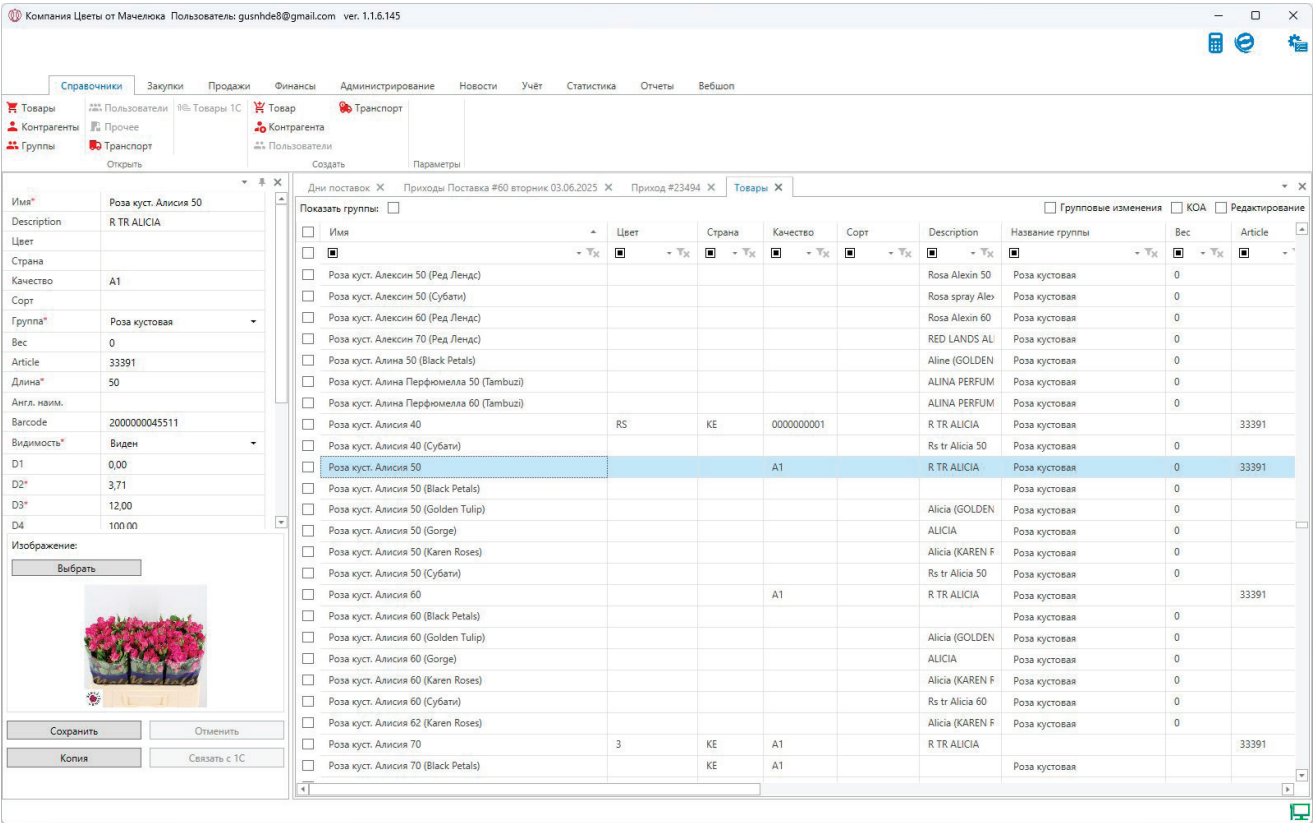


Рис. 6. Результат работы робота в внутренней ИС flowers

Внедрение роботизированного решения позволило существенно повысить эффективность исследуемого бизнес-процесса. По результатам опытной эксплуатации продолжительность обработки аукционной документации сократилась с 44,1 до 15,4 часа, что соответствует снижению времени выполнения процесса на 65 %. Стоимость выполнения процесса уменьшилась с 12 645 до 4 267 рублей, обеспечив снижение операционных затрат на 67 %. Одновременно было достигнуто существенное сокращение количества ошибок, возникающих при ручной обработке документов: уровень ошибок снизился с 20–35 % до менее 5 %.

Полученные результаты свидетельствуют о том, что применение технологии RPA позволяет не только сократить трудоемкость выполнения регламентированных операций, но и повысить качество обработки информации, снизить влияние человеческого фактора и увеличить пропускную способность закупочного процесса. Экономический эффект, достигнутый в результате внедрения роботизированного решения, подтверждает целесообразность использования технологии RPA при цифровой трансформации закупочной деятельности предприятий.

Таблица 3. Сравнительный анализ результатов трансформации бизнес-процесса

Показатель	До внедрения	После внедрения	Изменение
Продолжительность процесса	44,1 ч	15,4 ч	–65 %
Стоимость выполнения процесса	12 645 руб.	4 267 руб.	–67 %
Уровень ошибок	20–35 %	< 5 %	Существенное снижение
Степень автоматизации	Преимущественно ручная обработка	Роботизированное выполнение ключевых операций	Повышение

Направления дальнейшего развития разработанного решения

Несмотря на достигнутые результаты, разработанное роботизированное решение обладает значительным потенциалом дальнейшего развития. На текущем этапе автоматизация охватывает бизнес-процесс приема и первичной обработки аукционных документов, однако аналогичный подход может быть распространен и на другие процессы закупочной деятельности предприятия.

Одним из перспективных направлений является интеграция программных роботов с технологиями интеллектуальной обработки документов. Использование систем оптического распознавания символов (OCR) и алгоритмов искусственного интеллекта позволит автоматизировать обработку документов с различной структурой, повысить устойчивость роботизированного решения к изменениям форматов входящей документации и минимизировать необходимость предварительной стандартизации документов.

Другим направлением развития является формирование единого роботизированного контура закупочной деятельности. В этом случае программные роботы смогут автоматизировать не только обработку аукционных документов, но и выполнение последующих операций, связанных с формированием заказов поставщикам, контролем оплаты счетов, подготовкой логистической документации, мониторингом исполнения поставок и актуализацией информации в корпоративных информационных системах предприятия.

Дополнительным направлением совершенствования является внедрение средств интеллектуального анализа данных для прогнозирования объемов закупок, оценки надежности поставщиков и поддержки принятия управленческих решений. Совместное применение технологий RPA, машинного обучения и аналитических инструментов позволит перейти от автоматизации отдельных операций к комплексной цифровой трансформации закупочной деятельности предприятия.

Таким образом, разработанное решение может рассматриваться не как самостоятельный программный продукт, а как первый этап формирования интеллектуальной системы автоматизации закупочных бизнес-процессов.

Заключение

В статье рассмотрены вопросы совершенствования бизнес-процесса обработки аукционных документов на основе технологии роботизированной автоматизации процессов на примере ООО «Ново Флоре».

В результате анализа существующего бизнес-процесса установлено, что его выполнение сопровождается значительным количеством повторяющихся ручных операций, многократным переносом информации между корпоративными информационными системами и высокой зависимостью результатов обработки документов от человеческого фактора. Выявленные недостатки приводят к увеличению продолжительности процесса, росту трудозатрат и повышению вероятности возникновения ошибок.

Для устранения выявленных проблем проведен анализ современных подходов к автоматизации бизнес-процессов, по результатам которого в качестве наиболее рационального решения выбрана технология роботизированной автоматизации процессов (RPA). Реализация разработанного решения выполнена на отечественной платформе **PIX RPA**, обеспечивающей автоматизацию взаимодействия с корпоративными информационными системами без изменения их архитектуры и программного кода.

Практическая апробация разработанного программного робота подтвердила эффективность предложенного подхода. В результате внедрения роботизации продолжительность обработки аукционной документации сократилась на **65 %**, стоимость выполнения бизнес-процесса уменьшилась на **67 %**, а количество ошибок, обусловленных ручной обработкой данных, снизилось до уровня **менее 5 %**. Полученные результаты свидетельствуют о существенном повышении производительности закупочного процесса и снижении влияния человеческого фактора.

Предложенный подход обладает высокой практической значимостью и может быть использован при роботизации аналогичных бизнес-процессов предприятий, деятельность которых связана с обработкой значительных объемов регламентированной документации. Дальнейшее развитие исследования целесообразно связать с интеграцией технологий RPA, интеллектуальной обработки документов и методов искусственного интеллекта, что позволит перейти от автоматизации отдельных операций к созданию интеллектуальных систем управления закупочной деятельностью.

Литература:

1. Пронин К. Н. RPA (Robotic process automatisation) PIX Studio в педагогике основ программирования // Гуманитарные научные исследования. 2023. № 5 (141). 14–20 с.
2. Пронин К. Н. Методология проектного обучения на основе принципов разработки программного обеспечения // Гуманитарные научные исследования. 2023. № 7 (143).
3. Цуканов А. В., Русина Н. А. Анализ продуктовой модели архитектуры предприятия на основе технологии RPA & AI // Экономика и предпринимательство. — 2023. — № 2 (151). — С. 732–735.
4. Шикунова В. А., Агафонова А. Н. Актуальные подходы к управлению взаимоотношениями с поставщиками // Скиф. Вопросы студенческой науки. — 2019. — № 11 (39). — С. 19–22.
5. Зоидов К. Х., Пономарева С. В., Трегубова А. А. Автоматизация и роботизация бизнес-процессов промышленного предприятия в условиях цифровизации // Сегодня и завтра российской экономики. — 2018. — № 89–90. — С. 39–50.
6. Антониади К. С., Коваль О. И., Соломко Д. С. Роботизация как способ повышения эффективности бизнес-процессов // Colloquium-journal. — 2020. — № 4–2 (56). — С. 68–69.

Разработка программного средства для анализа данных о серийных преступлениях

Тагунков Михаил Владиславович, студент

Национальный исследовательский университет «Московский институт электронной техники» (г. Зеленоград)

Представлено настольное программное средство для извлечения и анализа хронологических данных из судебных и архивных документов. Система принимает TXT, DOCX и изображения, выполняет OCR, нормализует текст, выделяет даты преступлений и дату рождения исследуемого лица, поддерживает точечные даты и интервалы. События сопоставляются с фазами трех условных биоритмических циклов и лунного цикла, отображаются на одном периоде синусоиды, а фазовая концентрация оценивается тестом Рэлея. Программа реализована на C# с использованием Windows Forms, Open XML SDK, Tesseract, OpenCvSharp и ScottPlot. Модульные, интеграционные и интерфейсные испытания подтвердили корректность основных сценариев; покрытие кода тестами составило 73,5 %. Получаемые результаты предназначены для исследовательской поддержки и не доказывают причинную связь между циклами и преступным поведением.

Ключевые слова: анализ данных, юридические документы, OCR, извлечение дат, циклические данные, круговая статистика, C#, Windows Forms.

1. Введение

Материалы о серийных преступлениях содержат значительный объем неструктурированных сведений. Даты эпизодов распределены по тексту приговоров, экспертиз и архивных документов, записаны в разных форматах и могут задаваться приблизительно: месяцем, диапазоном месяцев или словесным периодом. Ручной поиск занимает много времени и повышает риск пропусков, дублирования и ошибочного включения служебных дат.

Дополнительная задача состоит в сопоставлении календарных событий с повторяющимися процессами. Визуализация временных данных переводит разрозненную хронологию в форму, удобную для поиска концентраций и повторяемости [5, с. 11–31]. При этом статистическое совпадение не следует трактовать как причинность: программа должна помогать проверять гипотезы и сохранять возможность экспертной проверки исходных фрагментов.

Цель работы — создать автономное средство, которое загружает документы, извлекает и позволяет исправлять даты, строит циклические графики и формирует осторожный статистический вывод. Требуются поддержка TXT, DOCX, JPG, PNG и BMP, предобработка изображений, контекстное отделение дат преступлений от дат приговоров и экспертиз, работа с точками и интервалами, а также быстрое обновление результатов после ручной корректировки.

2. Аналоги и требования

В качестве аналогов рассмотрены SAS Visual Analytics, Microsoft Power BI, Tableau Desktop и KNIME Analytics Platform [1–4]. Эти решения обладают развитыми средствами отчетности и анализа временных рядов, однако обычно предполагают, что даты уже размещены в таблице или базе данных. Извлечение событий из юридического текста и распознавание сканов отсутствуют либо требуют отдельного конвейера и настройки внешних компонентов.

SAS и Tableau ориентированы преимущественно на корпоративную инфраструктуру; Power BI доступнее, но контекстная классификация дат и циклические модели требуют программирования. KNIME позволяет собрать такой процесс из узлов, однако рабочий поток остается сложным для непрофильного пользователя. Поэтому специализированное приложение строится как единый сценарий: загрузка документа, проверка дат, расчет и просмотр графиков.

На обычном пользовательском компьютере текстовый документ объемом до 200 страниц должен обрабатываться не более 30 секунд, изображение с OCR — не более двух минут. Пакет из десяти файлов не должен нарушать устойчивость приложения. Расчет фаз и построение четырех графиков для набора до 1000 событий должны занимать не более пяти секунд, а ручные изменения — отображаться без заметной задержки.

3. Методы обработки данных

Алгоритм состоит из независимых этапов. Система проверяет расширение файла и выбирает обработчик: TXT читается напрямую, содержимое DOCX извлекается средствами Open XML SDK [8], а изображения передаются в OCR-подсистему. Тексты объединяются, затем парсер формирует единый список событий, удаляет дубли и сохраняет связь каждого результата с исходным фрагментом.

3.1. OCR и нормализация

Перед распознаванием выполняются масштабирование, выравнивание наклона, шумоподавление, повышение локального контраста и адаптивная бинаризация. Используются Tesseract и OpenCvSharp [9; 10]. Несколько вариантов предобработки могут распознаваться параллельно; лучший результат выбирается по доле допустимых символов, количеству словарных токенов и наличию фрагментов, похожих на даты.

Нормализация удаляет невидимые символы и повторные пробелы, объединяет разорванные элементы даты и исправляет типичные OCR-ошибки. Например, кириллическая «О» в числовом окружении заменяется на ноль. Исходный текст не изменяется: нормализованная копия используется только для анализа, чтобы спорный результат можно было проверить по документу.

3.2. Контекстное извлечение дат

Парсер сочетает регулярные выражения и эвристики. Он находит числовые и словесные даты, неполные годы, конструкции «в начале месяца», «с марта по май» и другие интервалы. Затем оценивается контекстное окно. Маркеры «совершил», «напал», «эпизод», «потерпевший» повышают вероятность включения кандидата, а слова «приговор», «экспертиза», «постановление», «судимость» и «регистрация» понижают ее.

Дата рождения извлекается отдельным правилом. Неопределенное время сохраняется как объект периода с начальной и конечной датой; обратный диапазон автоматически нормализуется. Пользователь видит не только итоговую дату, но и фрагмент, на основании которого она распознана, и может добавить, изменить либо удалить событие.

3.3. Циклические координаты и статистика

Для каждого события вычисляется число суток от даты рождения. Затем календарная координата преобразуется в фазу цикла. Используются условные периоды 23, 28 и 33 дня и средняя продолжительность лунного цикла 29,530588853 дня. Для события i и периода T фаза определяется выражением:

$$\varphi_i = 2\pi \cdot ((d_i - d_0) \bmod T) / T,$$

где d_i — дата события, d_0 — дата рождения, T — период. Амплитуда равна $\sin \varphi_i$. Приведение к одному периоду позволяет совместно рассматривать события разных лет. Точечная дата отображается маркером, небольшой интервал — двумя маркерами и отрезком; слишком длинный диапазон помечается как непригодный для оценки локальной концентрации.

Равномерность фаз проверяется тестом Рэля, применяемым в круговой статистике [6, с. 94–101; 7, с. 33–49]. При числе событий менее пяти программа сообщает о недостаточности данных. В остальных случаях рассчитывается p -значение и формируется формулировка о наличии либо отсутствии статистически заметной концентрации. Дополнительно оценивается плотность событий по скользящим окнам. Результат остается исследовательским индикатором, поскольку на него влияют полнота выборки, ошибки OCR и множественная проверка циклов.

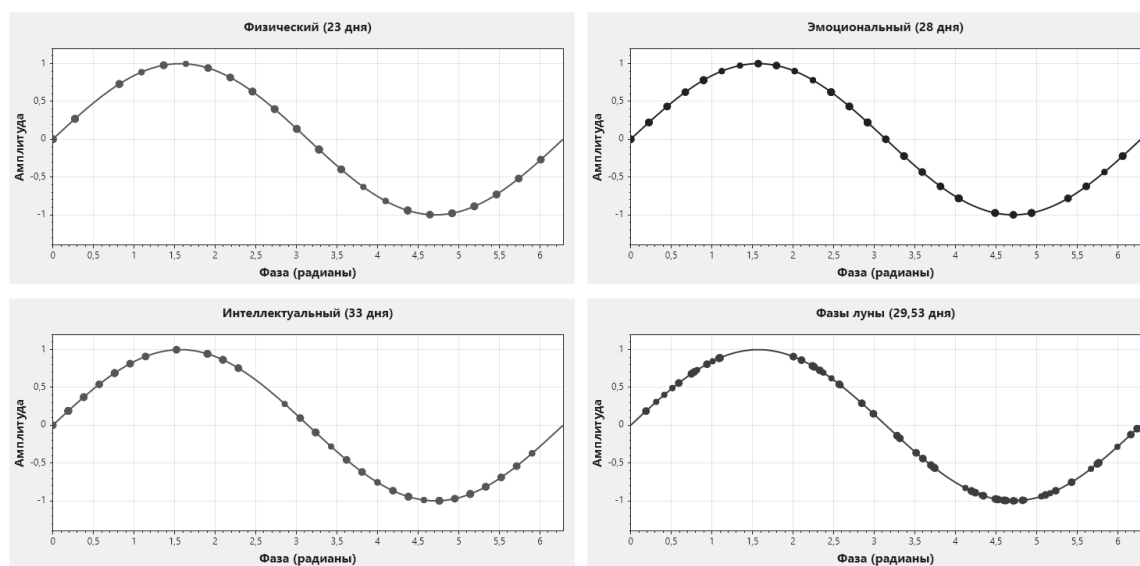


Рис. 1. Четыре циклических графика и результаты анализа

4. Реализация и интерфейс

Программа разработана на C# и .NET [8]. Windows Forms обеспечивает автономный интерфейс, Open XML SDK используется для DOCX, Tesseract и OpenCvSharp — для OCR и обработки изображений, ScottPlot — для графиков [9–11]. Архитектура разделена на сервис документов, OCR-модуль, парсер, анализатор циклов и слой интерфейса. Такое разделение упрощает модульное тестирование и добавление новых форматов.

Основная форма объединяет работу в одном окне: справа размещены загрузка документов, список дат, команды добавления, удаления и очистки и поле даты рождения; в центре — четыре графические панели и краткие выводы. Интерфейс рассчитан на черно-белую печать: смысл передается подписями, формой линий и маркеров, а не только цветом.

5. Тестирование и результаты

Испытания проведены на модульном, интеграционном и интерфейсном уровнях средствами MSTest [12]. Модульные тесты проверяли разбор дат и диапазонов, фильтрацию служебных дат, преобразование фаз и тест Рэля. Интеграционные сценарии охватывали выбор обработчика, передачу TXT и DOCX в парсер, взаимодействие OCR с предобработкой, объединение результатов и продолжение пакетной обработки при ошибке одного файла. Интерфейсные испытания проверяли загрузку, ручное редактирование, отмену выбора и блокировку расчета при отсутствии входных данных.

Таблица 1. Сводные результаты испытаний

Уровень	Проверяемые функции	Результат
Модульный	Парсер, даты и интервалы, фазы, статистика	Сценарии пройдены
Интеграционный	Файлы, OCR, объединение результатов	Сценарии пройдены
Интерфейсный	Загрузка, редактирование, запуск анализа	Сценарии пройдены
Итог	Совместное покрытие исходного кода	73,5 %

Все предусмотренные сценарии завершились успешно, совокупное покрытие кода составило 73,5 %. Испытания выявили потерю части дат в коротких списках, неверную замену кириллической «О» и попадание сносок в анализируемый текст. Ошибки исправлены и закреплены повторяемыми тестами. Для практического применения необходимы дополнительные испытания на обезличенном корпусе документов, расчет precision и recall для извлечения дат, проверка OCR на разных типах сканов и оценка статистических ошибок при множественном сравнении.

6. Заключение

Разработано автономное программное средство, объединяющее извлечение текста, контекстное распознавание дат, ручную корректировку, циклическую визуализацию и оценку фазовой концентрации. В отличие от универсальных BI-платформ, система принимает неструктурированные документы и изображения и не требует предварительного формирования таблиц.

Испытания подтвердили работоспособность модулей обработки документов, OCR, парсера, расчетной части и интерфейса. Дальнейшее развитие связано с расширением правил для юридических текстов, поддержкой дополнительных форматов, вероятностной оценкой кандидатов, автоматическим отчетом и более строгой коррекцией статистической значимости. Окончательная интерпретация результатов должна оставаться за специалистом.

Литература:

1. SAS Visual Analytics [Электронный ресурс] // SAS. URL: https://www.sas.com/en_us/software/visual-analytics.html (дата обращения: 15.12.2025).
2. Microsoft Power BI [Электронный ресурс] // Microsoft. URL: <https://www.microsoft.com/ru-ru/power-platform/products/power-bi> (дата обращения: 15.12.2025).
3. Tableau Desktop [Электронный ресурс] // Tableau. URL: <https://www.tableau.com/products/desktop> (дата обращения: 15.12.2025).
4. KNIME Analytics Platform [Электронный ресурс] // KNIME. URL: <https://www.knime.com/knime-analytics-platform> (дата обращения: 15.12.2025).
5. Aigner W., Miksch S., Schumann H., Tominski C. Visualization of Time-Oriented Data. 2nd ed. Cham: Springer Nature, 2023.

6. Mardia K. V., Jupp P. E. Directional Statistics. 2nd ed. Chichester: John Wiley & Sons, 2000. 429 p.
7. Jammalamadaka S. R., Sengupta A. Topics in Circular Statistics. Singapore: World Scientific, 2001. 322 p.
8. C# and Open XML SDK documentation [Электронный ресурс] // Microsoft Learn. URL: <https://learn.microsoft.com/> (дата обращения: 05.05.2026).
9. Tesseract OCR Documentation [Электронный ресурс]. URL: <https://tesseract-ocr.github.io/> (дата обращения: 05.05.2026).
10. OpenCvSharp Documentation [Электронный ресурс]. URL: <https://shimat.github.io/opencvsharp/> (дата обращения: 05.05.2026).
11. ScottPlot. Plotting Library for .NET [Электронный ресурс]. URL: <https://scottplot.net/> (дата обращения: 05.05.2026).
12. MSTest overview [Электронный ресурс] // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-mstest-intro> (дата обращения: 05.05.2026).

Программный модуль автоматизированного расчёта водопотребления жилых зданий

Трохин Евгений Витальевич, студент

Национальный исследовательский университет «Московский институт электронной техники» (г. Зеленоград)

В статье рассматривается разработка программного модуля расчёта водопотребления жилых зданий, ориентированного на автоматизацию инженерных вычислений по данным BIM-модели Autodesk Revit. Актуальность работы обусловлена тем, что при проектировании систем внутреннего водоснабжения расчёт водопотребления часто выполняется с использованием электронных таблиц и ручного анализа помещений и санитарно-технических приборов, что приводит к значительным трудозатратам, затрудняет повторяемость результата и повышает вероятность ошибок. В качестве основы исследования использованы тезисы, посвящённые автоматизации данной задачи, а материалы выпускной квалификационной работы применены для расширенного раскрытия архитектуры программного решения, алгоритмов обработки данных и результатов апробации. Предлагаемый модуль реализует извлечение данных о помещениях и сантехнических приборах из BIM-модели, классификацию помещений по типам водопотребителей на основе формализованных правил, загрузку и валидацию нормативных данных из XML-файла, а также формирование результирующей таблицы водопотребления. Архитектура программного модуля построена с использованием языка C#, WPF и шаблона MVVM, что обеспечивает разделение интерфейсной и вычислительной логики, удобство сопровождения и возможность развития решения. В работе показано, что интеграция расчётного инструмента непосредственно в BIM-среду позволяет сократить объём ручных операций, повысить воспроизводимость вычислений и обеспечить более удобную подготовку отчётных материалов. Дополнительная практическая ценность модуля заключается в поддержке экспорта результатов и хранении истории расчётов, что важно при итерационном развитии проектной модели.

Ключевые слова: BIM, Autodesk Revit, водопотребление, жилые здания, WPF, MVVM, XML, инженерные расчёты.

Software Module for Automated Calculating Water Consumption of Residential Buildings

Trokhin Yevgeny Vitalyevich, student

National Research University “Moscow Institute of Electronic Technology” (Zelenograd)

The article discusses the development of a software module for calculating water consumption in residential buildings, aimed at automating engineering calculations based on Autodesk Revit BIM models. The relevance of the study is determined by the fact that water consumption calculations in building water supply design are often performed using spreadsheets and manual analysis of rooms and plumbing fixtures, which leads to high labor intensity, reduces reproducibility, and increases the probability of errors. The study is based on thesis materials devoted to this automation task, while the graduation project materials are used to expand the description of the software architecture, data processing algorithms, and approbation results. The proposed module extracts room and plumbing fixture data from the BIM model, classifies rooms by water consumer categories according to formalized rules, loads and validates normative data from an XML file, and generates the resulting water consumption table. The module is implemented in C# using WPF and the MVVM pattern, which provides separation of interface and computational logic, maintainability, and extensibility. The paper

shows that embedding the calculation tool directly into the BIM environment reduces manual operations, improves reproducibility of calculations, and simplifies the preparation of reporting materials. Additional practical value is provided by export support and storage of calculation history, which is important for iterative project development.

Keywords: BIM, Autodesk Revit, water consumption, residential buildings, WPF, MVVM, XML, engineering calculations.

Введение

В последние годы технологии информационного моделирования зданий стали одной из ключевых основ цифровой трансформации строительной отрасли [1]. BIM-модель рассматривается не только как графическое представление объекта, но и как структурированный источник данных, пригодный для инженерных расчётов, координации проектных решений и последующего сопровождения объекта [1], [2]. Однако при решении ряда прикладных задач, в том числе связанных с проектированием систем внутреннего водоснабжения, потенциальные возможности BIM используются не в полной мере. Значительная часть расчётной работы по-прежнему выполняется вручную или с применением внешних электронных таблиц, что создаёт разрыв между цифровой моделью здания и инженерными вычислениями.

Одной из таких задач является определение водопотребления жилых зданий. При её выполнении инженер должен учитывать состав помещений, набор санитарно-технических приборов, наличие систем горячего водоснабжения и нормативные показатели расхода воды [2]. На практике это приводит к необходимости последовательно просматривать BIM-модель, вручную интерпретировать параметры элементов, сопоставлять полученные сведения с нормативными категориями и затем оформлять итоговую таблицу. Подобный подход трудоёмок, плохо масштабируется при корректировке проекта и повышает вероятность логических и операционных ошибок.

В связи с этим актуальной становится задача создания программного модуля, который связывает BIM-модель здания с нормативной базой и автоматизирует расчёт водопотребления непосредственно в проектной среде. Цель статьи состоит в описании программного модуля расчёта водопотребления жилых зданий, разработанного для среды Autodesk Revit, а также в раскрытии его архитектурных решений, алгоритмической основы и практической значимости. В основе статьи лежат тезисы по теме работы, а содержание выпускной квалификационной работы использовано для более полного раскрытия методики реализации и анализа полученных результатов.

1. Постановка задачи и требования к модулю

Расчёт водопотребления жилого здания относится к числу инженерных задач, в которых важно не только получить численный результат, но и обеспечить его нормативную корректность, воспроизводимость и удобство повторного применения. Для этого программный мо-

дуль должен обрабатывать несколько классов данных. К первому классу относятся сведения BIM-модели: помещения, сантехнические приборы и их параметры. Ко второму классу относятся внешние нормативные данные, определяющие категории водопотребителей и расчётные показатели расхода воды. К третьему классу относятся пользовательские параметры, задающие особенности интерпретации кодов элементов и систем в конкретной проектной практике.

Исходя из этого, к модулю были сформулированы следующие требования. Во-первых, он должен быть встроен в среду Autodesk Revit и использовать активную BIM-модель как первичный источник информации. Во-вторых, модуль должен обеспечивать автоматизированное извлечение данных о помещениях и санитарно-технических приборах без необходимости ручного переноса сведений во внешние таблицы. В-третьих, нормативная база должна подключаться из внешнего XML-файла, при этом корректность его структуры необходимо проверять до начала расчёта. В-четвёртых, алгоритм классификации помещений должен быть формализован таким образом, чтобы при одинаковых исходных данных результат оставался детерминированным. В-пятых, пользователю должны быть доступны просмотр итоговой таблицы, экспорт результатов и возможность повторного обращения к ранее выполненным расчётам.

Постановка задачи показывает, что речь идёт не о частной программной утилите, а о прикладном инженерном модуле, который должен быть встроен в повседневный проектный процесс. Следовательно, особое значение приобретают не только вычислительная корректность, но и архитектурная устойчивость, понятность интерфейса и возможность дальнейшего расширения решения.

2. Архитектура программного решения

Разработанный программный модуль реализован на языке C# в виде add-in для Autodesk Revit. Такой выбор обусловлен тем, что Revit API предоставляет необходимые средства доступа к помещениям, экземплярам семейств и пользовательским параметрам модели [3], [4]. Пользовательский интерфейс построен на WPF, а логическая структура приложения организована по шаблону MVVM, который позволяет разделить представление, состояние интерфейса и вычислительную логику [5], [6]. Для инженерного программного обеспечения такой подход особенно важен, поскольку он повышает сопровождаемость кода и упрощает добавление новых функций без переработки всего приложения.

Архитектурно программный модуль включает несколько основных подсистем. Подсистема построения модели отвечает за извлечение помещений и сантехнических приборов из документа Revit и формирование внутреннего представления объекта. Подсистема алгоритмической обработки реализует правила классификации помещений по категориям водопотребителей и вычисление итоговых показателей. Подсистема работы с нормативными данными загружает XML-файл и проверяет его структуру с использованием XSD-схемы, что повышает надёжность расчёта [7], [8]. Подсистема экспорта позволяет выгружать результаты в форматы XLSX, CSV и XML, а подсистема хранения истории использует локальную SQLite-базу данных, не требующую серверной инфраструктуры [9].

Важным преимуществом выбранной архитектуры является разделение контуров ответственности. Извлечение данных из Revit не смешивается с логикой классификации помещений, а обработка нормативных XML-данных не встроена непосредственно в код интерфейса. Благодаря этому модуль оказывается более прозрачным с точки зрения сопровождения и легче адаптируется к изменениям требований. Например, корректировка пользовательских кодов или расширение состава экспортных форматов не требует переработки алгоритмического ядра.

3. Алгоритм расчёта водопотребления

Работа программного модуля начинается со сбора данных из BIM-модели. На этом этапе формируется список помещений и связанных с ними сантехнических приборов. Для каждого прибора извлекаются параметры, необходимые для классификации: код элемента, код системы и, в ряде случаев, длина ванны. Далее выполняется нормализация признаков, включающая удаление лишних пробелов, перевод строковых значений в единый регистр и устранение неоднозначностей в символах. Этот шаг принципиально важен, поскольку в реальной проектной модели параметры могут заполняться разными специализациями и содержать неоднородное текстовое оформление.

После построения внутренней модели здания запускается алгоритм классификации помещений. Он основан на приоритетной последовательности правил, которая исключает неоднозначность при определении категории водопотребителя. Сначала анализируются признаки, связанные с наличием ванны и её длиной, затем рассматриваются сочетания приборов, характерные для помещений с определённым набором санитарно-технического оборудования, после чего оценивается наличие нагревателя и признаков централизованного горячего водоснабжения. Если специальные условия не выполняются, помещению присваивается базовая категория. Такой подход обеспечивает детерминированность алгоритма: для одного и того же набора входных данных результат классификации остаётся неизменным.

Следующим этапом является сопоставление результатов классификации с нормативной базой. Перед ис-

пользованием XML-файл проходит обязательную проверку по XSD-схеме, что исключает выполнение расчёта по структурно некорректным данным [7]. После успешной валидации каждая категория сопоставляется с нормативной записью, содержащей суточный расход воды и другие расчётные параметры. Итоговая таблица формируется только для тех категорий, которые фактически присутствуют в модели. При вычислении результирующих показателей используются число помещений данной категории и коэффициент расхода, что позволяет получить суммарное значение водопотребления для текущей BIM-модели.

С точки зрения инженерной практики значимым свойством алгоритма является устойчивость к неполным данным. Если отдельный элемент модели содержит недостаточный набор параметров, модуль не прекращает весь расчёт, а исключает проблемный элемент из обработки. Это соответствует условиям реального проектирования, когда BIM-модель может находиться в промежуточном состоянии, а часть информации ещё не доведена до окончательного вида. При этом модуль сохраняет общую корректность вычислений для той части модели, которая заполнена надлежащим образом.

4. Пользовательский сценарий и эксплуатационные особенности

Пользовательский сценарий разработан таким образом, чтобы встроить расчёт в привычную последовательность работы инженера-проектировщика. После запуска команды в Revit пользователь сначала выбирает XML-файл нормативов и профиль расчёта. Затем открывается основное окно модуля, где доступны параметры алгоритма, запуск расчёта, просмотр результирующей таблицы, экспорт и дополнительные вкладки для сравнения и истории. Такая организация интерфейса делает рабочий процесс поэтапным и понятным: подготовка входных данных отделяется от анализа результата и вспомогательных операций.

Важной частью эксплуатационной логики является поддержка истории расчётов. В условиях реального проектирования одна и та же модель многократно уточняется, а значит, инженер регулярно возвращается к пересчёту водопотребления. Сохранение истории выгрузок в SQLite позволяет фиксировать дату и время расчёта, формат экспортированного файла, параметры запуска и итоговые значения. Благодаря этому модуль становится инструментом сопровождения проектных итераций, а не только средством получения единичного расчёта.

Ещё одной особенностью решения является дружественная деградация второстепенных функций. Если, например, временно недоступна локальная база истории, основной расчётный сценарий не блокируется. Аналогично ошибка при загрузке файла сравнения не отменяет уже полученного результата расчёта. Для инженерного программного обеспечения это существенное качество, по-

скольку вспомогательные сервисы не должны разрушать базовый рабочий процесс.

5. Результаты и обсуждение

Разработанный программный модуль показывает, что интеграция расчёта водопотребления непосредственно в BIM-среду позволяет существенно изменить характер работы инженера. Вместо последовательного ручного анализа помещений и переноса данных в электронные таблицы пользователь получает инструмент, который автоматически извлекает сведения из модели, интерпретирует их по формализованным правилам и формирует итоговую таблицу в едином программном контуре. Это сокращает объём рутинных действий и делает расчёт менее зависимым от человеческого фактора.

Практическая значимость решения проявляется прежде всего в повышении воспроизводимости. Поскольку классификация помещений выполняется по явным правилам, а нормативная база проходит предварительную валидацию, инженер получает более устойчивый и прозрачный результат. Дополнительный эффект даёт история расчётов: она позволяет сравнивать версии отчётов, связывать их с изменениями BIM-модели и тем самым поддерживать итерационный характер проектирования.

В ходе разработки и апробации модуля была сформирована развитая тестовая база, включающая 70 автоматизированных проверок, направленных на контроль алгоритма классификации, работы с XML и XSD, формирования отчёта, экспортных сценариев, истории выгрузок и ряда сквозных пользовательских операций. Хотя такие проверки не заменяют эксплуатационную оценку в конкретной проектной организации, они подтверждают, что вычислительная логика системы формализована и поддаётся систематическому контролю. Это позволяет рассматривать разработанное решение не как концептуальный прототип, а как работоспособный программный модуль прикладного назначения.

Вместе с тем следует учитывать и ограничения применимости. Качество результата напрямую зависит от дисциплины ведения BIM-модели и согласованности системы кодирования элементов. Если параметры помещений и приборов заполнены непоследовательно, а нормативные файлы формируются без единого подхода, часть преимуществ автоматизации снижается. Следовательно, максимальный эффект от внедрения программного модуля достигается в той проектной среде, где уже существует хотя бы минимальный стандарт работы с BIM-данными.

Заключение

В статье рассмотрен программный модуль расчёта водопотребления жилых зданий, разработанный для автоматизации инженерных расчётов на основе BIM-модели Autodesk Revit и внешних нормативных XML-данных. Показано, что ручной подход к выполнению таких расчётов приводит к повышенной трудоёмкости, затрудняет повторное использование результата и увеличивает риск ошибок. Предложенное решение устраняет эти недостатки за счёт интеграции расчётного контура непосредственно в BIM-среду, формализации алгоритма классификации помещений и автоматизированной работы с нормативной базой.

Архитектура модуля, основанная на сочетании Revit API, C#, WPF, MVVM, XML и SQLite, обеспечивает разделение ответственности между основными подсистемами, удобство пользовательского сценария и возможность дальнейшего расширения функциональности. Практическая ценность работы состоит в том, что разработанный модуль может использоваться как инструмент инженерной подготовки расчётной документации и как средство повышения прозрачности проектных итераций. Перспективы дальнейшего развития связаны с расширением набора профилей расчёта, адаптацией к другим типам объектов и углублением аналитики истории вычислений.

Литература:

1. Sacks R., Eastman C., Lee G., Teicholz P. BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers. 3rd ed. Hoboken: Wiley, 2018.
2. СП 30.13330.2020. Внутренний водопровод и канализация зданий. М.: Минстрой России, 2021.
3. Autodesk. Revit API Developers Guide: Introduction [Электронный ресурс]. URL: https://help.autodesk.com/view/RVT/2024/ENU/?guid=Revit_API_Revit_API_Developers_Guide_Introduction_html (дата обращения: 14.04.2026).
4. Autodesk. Revit API Developers Guide: Deployment Options [Электронный ресурс]. URL: https://help.autodesk.com/view/RVT/2024/ENU/?guid=Revit_API_Revit_API_Developers_Guide_Introduction_Add_In_Integration_Deployment_Options_html (дата обращения: 14.04.2026).
5. Microsoft. WPF overview [Электронный ресурс]. URL: <https://learn.microsoft.com/dotnet/desktop/wpf/overview/> (дата обращения: 14.04.2026).
6. Microsoft. MVVM Toolkit [Электронный ресурс]. URL: <https://learn.microsoft.com/dotnet/communitytoolkit/mvvm/> (дата обращения: 14.04.2026).
7. Microsoft. XmlSerializer Class [Электронный ресурс]. URL: <https://learn.microsoft.com/dotnet/api/system.xml.serialization.xmlserializer> (дата обращения: 14.04.2026).

8. W3C. XML Schema Definition Language (XSD) 1.1 Part 1: Structures [Электронный ресурс]. URL: <https://www.w3.org/TR/xmlschema11-1/> (дата обращения: 14.04.2026).
9. SQLite Documentation [Электронный ресурс]. URL: <https://www.sqlite.org/docs.html> (дата обращения: 14.04.2026).
10. ГОСТ 19.701–90. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. М.: Издательство стандартов, 1991.

Метод выявления неявных преобразований типов, снижающих надёжность C#-кода

Хакризов Дмитрий Юрьевич, студент
Московский государственный технологический университет «СТАНКИН»

В статье рассматривается проблема неявных преобразований типов в C#-коде и их влияние на надёжность программного обеспечения. Несмотря на строгую типизацию языка C#, в прикладных программах возможны ситуации, когда значение одного типа автоматически приводится к другому без явного указания разработчика. Такие преобразования могут приводить к потере точности, скрытому boxing, ошибкам при работе с nullable-типами, некорректному выбору перегруженных методов и снижению предсказуемости поведения программы. Предлагается метод выявления потенциально опасных неявных преобразований на основе статического анализа исходного кода, классификации преобразований по уровню риска и формирования рекомендаций по их устранению.

Ключевые слова: C#, типизация, неявные преобразования, статический анализ, надёжность кода, nullable-типы, boxing, dynamic.

Введение

Язык C# относится к языкам со строгой статической типизацией. Это позволяет обнаруживать значительную часть ошибок ещё на этапе компиляции: несовместимые присваивания, некорректную передачу аргументов, ошибочные операции с объектами и другие нарушения типовой модели. Однако строгая типизация не исключает всех рисков, связанных с преобразованием данных.

В C# существуют неявные преобразования типов, при которых компилятор самостоятельно изменяет тип значения без явного оператора приведения. В простых случаях это удобно и делает код короче. Например, значение типа `int` может быть преобразовано к `long` или `double`. Но в более сложной бизнес-логике такие преобразования могут остаться незамеченными и привести к ошибкам, которые сложно обнаружить при тестировании.

Проблема особенно актуальна для серверных приложений, систем обработки данных, финансовых расчётов и проектов с большим количеством моделей, DTO, обобщённых коллекций и внешних интерфейсов. В таких системах изменение типа может повлиять на результат вычислений, сериализацию, работу с базой данных или выбор перегруженного метода. Поэтому необходимо выявлять участки кода, где неявные преобразования снижают надёжность программы.

Понятие неявного преобразования типов

Неявное преобразование типов — это автоматическое приведение значения одного типа к другому без явного

указания разработчика. Оно выполняется компилятором в случаях, разрешённых правилами языка. Например, значение производного класса может быть присвоено переменной базового типа, а значение `int` — переменной `long`.

Такие преобразования не всегда являются ошибкой. Однако их опасность зависит от контекста. Если преобразование явно видно из одной строки, риск невелик. Если же оно скрыто внутри вызова метода, LINQ-выражения, обобщённого типа или цепочки вычислений, разработчику сложнее понять фактическое поведение программы.

Неявные преобразования могут возникать при присваивании, выполнении арифметических операций, передаче параметров, сравнении значений, работе с `object`, `dynamic`, `nullable`-типами и обобщёнными коллекциями. Поэтому ручной контроль таких случаев в крупном проекте становится затруднительным.

Основные риски неявных преобразований

Первый риск связан с потерей точности или изменением результата вычислений. При смешивании числовых типов компилятор может выбрать общий тип выражения, который не всегда очевиден. Это особенно важно в расчётах, связанных с деньгами, процентами, коэффициентами и статистикой.

Второй риск связан с преобразованием значимых типов к `object`. Такое преобразование вызывает `boxing`, то есть упаковку значения в объект. Это увеличивает нагрузку на память и может скрывать исходный тип. При обратном преобразовании возникает `unboxing`, который способен привести к ошибке выполнения при несовпадении ожидаемого типа.

Третий риск возникает при работе с nullable-типами. Такие типы могут хранить как значение, так и null. Если преобразование между обычным и nullable-типом происходит неявно, разработчик может ошибочно предположить, что значение всегда существует. Это повышает вероятность ошибок при обращении к данным.

Четвёртый риск связан с использованием dynamic. В этом случае часть проверок переносится с этапа компиляции на этап выполнения. При сочетании dynamic и неявных преобразований код становится менее предсказуемым, так как ошибка может проявиться только при конкретных входных данных.

Отдельную проблему создают перегруженные методы. Если существует несколько методов с разными типами параметров, компилятор выбирает подходящий вариант на основе правил разрешения перегрузки. Из-за неявного преобразования может быть выбран не тот метод, который ожидал разработчик.

Метод выявления преобразований

Предлагаемый метод включает четыре этапа.

На первом этапе выполняется статический анализ исходного кода. Проверяются выражения присваивания, вызовы методов, арифметические операции, условия, возвращаемые значения, элементы коллекций и операции сравнения. Для каждого участка определяется фактический тип выражения и ожидаемый тип контекста.

На втором этапе найденные преобразования классифицируются. Выделяются безопасные расширяющие преобразования, потенциально опасные числовые преобразования, boxing и unboxing, преобразования с nullable-типами, приведение к object, использование dynamic и преобразования, влияющие на выбор перегруженного метода.

На третьем этапе каждому преобразованию присваивается уровень риска. Низкий риск характерен для очевидных и безопасных преобразований. Средний риск относится к случаям, ухудшающим читаемость или производительность. Высокий риск присваивается преобразованиям, которые могут привести к потере данных, ошибке выполнения, некорректной обработке null или изменению логики вызова метода.

На четвёртом этапе формируется диагностическое сообщение. Оно должно содержать место в коде, исходный тип, ожидаемый тип, вид преобразования и возможное последствие. Также целесообразно указывать рекомендацию: использовать явное приведение, изменить тип переменной,

заменить object на обобщённый тип, добавить проверку null или отказаться от dynamic в пользу конкретной модели.

Практическое применение метода

Метод может применяться на разных этапах разработки. В среде программирования он может быть реализован как анализатор кода, который подсвечивает потенциально опасные преобразования сразу при написании программы. На этапе code review он может использоваться как набор правил проверки. В процессе непрерывной интеграции такие проверки могут быть встроены в pipeline сборки.

Для снижения риска рекомендуется явно указывать преобразование в тех местах, где оно является осознанным решением. Это делает код понятнее для других разработчиков. Также следует ограничивать использование object и dynamic, если вместо них можно применить конкретный тип, интерфейс или обобщённую модель. При работе с nullable-типами важно явно проверять наличие значения перед его использованием.

Особое внимание следует уделять числовым типам. Для денежных расчётов предпочтительно использовать decimal, а не double. Если значение может превысить диапазон int, следует заранее выбирать long. Такой подход позволяет избежать скрытых ошибок, связанных с переполнением, потерей точности или неправильной интерпретацией результата.

Заключение

Неявные преобразования типов являются допустимой частью языка C# и во многих случаях делают код удобнее. Однако в крупных приложениях они могут снижать надёжность, ухудшать читаемость и приводить к ошибкам, которые трудно обнаружить при обычном тестировании.

Предложенный метод основан на статическом анализе кода, классификации найденных преобразований и оценке их уровня риска. Он позволяет заранее выявлять потенциально опасные участки программы и формировать рекомендации по их исправлению.

Наиболее важными объектами анализа являются числовые преобразования, nullable-типы, boxing и unboxing, приведение к object, использование dynamic и ситуации, влияющие на выбор перегруженных методов. Контроль этих случаев повышает предсказуемость поведения программы, упрощает сопровождение и способствует повышению качества C#-кода.

Литература:

1. Microsoft Learn. C# type system [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/types/> (дата обращения: 09.07.2026).
2. Microsoft Learn. Casting and type conversions [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/programming-guide/types/casting-and-type-conversions> (дата обращения: 09.07.2026).
3. Microsoft Learn. Nullable value types [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/language-reference/builtin-types/nullable-value-types> (дата обращения: 09.07.2026).

МЕДИЦИНА

Роль врача-терапевта в первичной и вторичной профилактике ишемического инсульта

Скоросова Дарья Алексеевна, врач-терапевт
ООО «Центр восточной медицины» (г. Белгород)

Ишемический инсульт остаётся одной из ведущих причин смертности, инвалидизации и социально-экономического бремени в Российской Федерации. Большинство пациентов с хроническими сердечно-сосудистыми заболеваниями находятся под наблюдением врача-терапевта участкового, что определяет ключевую роль терапевтического звена в раннем выявлении факторов риска, профилактике первого инсульта и долгосрочном ведении пациентов после перенесённого острого нарушения мозгового кровообращения (ОНМК). В обзоре проанализированы современные эпидемиологические данные, модифицируемые факторы риска, алгоритмы скрининга и ведения пациентов на амбулаторном этапе в соответствии с действующими клиническими рекомендациями Минздрава России. Особое внимание уделено практическим аспектам работы терапевта: оценке сердечно-сосудистого риска, коррекции артериального давления и липидного профиля, анти тромботической терапии, мотивации пациентов к изменению образа жизни и своевременной маршрутизации при подозрении на транзиторную ишемическую атаку (ТИА).

Ключевые слова: ишемический инсульт, транзиторная ишемическая атака, первичная профилактика, вторичная профилактика, врач-терапевт, факторы риска, артериальная гипертензия, шкала FAST.

Введение

Острые нарушения мозгового кровообращения занимают одно из первых мест в структуре заболеваемости и смертности населения России. По данным мониторинга, ежегодно в стране регистрируется от 430 до 470 тысяч случаев инсульта; около 80 % из них приходится на ишемический тип поражения [1, 2]. Болезни системы кровообращения остаются ведущей причиной смерти: доля цереброваскулярных заболеваний в структуре сердечно-сосудистой смертности по-прежнему превышает 30 % [3]. Несмотря на снижение региональных показателей смертности от ОНМК в ряде субъектов Федерации, вариабельность между регионами сохраняется, что отражает неоднородность организации помощи и доступности профилактических программ [4].

Врач-терапевт участковый является первым и наиболее частым контактом пациента с системой здравоохранения. Именно на уровне поликлиники формируется диспансерное наблюдение за лицами с артериальной гипертензией (АГ), сахарным диабетом (СД), ишемической болезнью сердца, хронической сердечной недостаточностью и фибрилляцией предсердий (ФП) — состояниями, непосредственно определяющими риск ишемического инсульта. Согласно клиническим рекомендациям, комплексная профилактика инсульта включает как немедикаментозные мероприятия, так и фармакотерапию, направленную на контроль модифицируемых факторов риска [3, 5].

Цель настоящего обзора — систематизировать современные представления о роли врача-терапевта в первичной и вторичной профилактике ишемического инсульта и обосновать практические алгоритмы амбулаторного ведения пациентов группы высокого и очень высокого сердечно-сосудистого риска.

Эпидемиология и социальная значимость проблемы

Инсульт характеризуется высокой остротой начала, значительной летальностью в остром периоде и выраженными долгосрочными последствиями для качества жизни. По данным обзора распространённости инсульта в Российской Федерации по результатам популяционных опросов 2003–2023 гг., частота самоотчётного инсульта в популяции демонстрирует устойчивую тенденцию к росту с возрастом и существенную географическую неоднородность [1]. Смертность в течение первого года после инсульта по отдельным исследованиям достигает 29,7 %, что подчёркивает необходимость не только экстренной, но и последующей профилактической работы [1, 2].

Ишемический инсульт в подавляющем большинстве случаев развивается на фоне атеросклеротического поражения сосудов, кардиоэмболии (прежде всего при ФП), лакунарного механизма или редких причин (диссекция, васкулиты, гиперкоагуляция). При этом до 95 % ишемических инсультов и ТИА связаны с атеротромботическими осложнениями, локализующимися в экстракраниальных отделах артериальной системы [2]. Важный клинический факт: лишь у 15 % пациентов, перенёвших инсульт, в анамнезе имелись чёткие указания на ТИА, что свидетельствует о недостаточной распознаваемости предынсультных состояний на амбулаторном этапе [2].

Социально-экономическое бремя инсульта в России остаётся значительным: высокая госпитальная летальность (17,6–20,7 % в разные годы), длительная реабилитация и потребность в уходе формируют существенные затраты для системы ОМС и семей пациентов [2]. Снижение этого бремени возможно преимущественно за счёт профилактики — как первичной (предотвращение первого инсульта), так и вторичной (предупреждение рецидива).

Факторы риска ишемического инсульта

Модифицируемые и немодифицируемые факторы

К факторам, которые нельзя изменить, относятся возраст, мужской пол (для молодого возраста), наследственная предрасположенность и некоторые неизменяемые характеристики сосудистого русла. Однако клинически наиболее значимыми являются модифицируемые факторы, контроль которых доказанно снижает риск ОНМК. К ним относятся АГ, дислипидемия, СД, курение, ожирение, низкая физическая активность, злоупотребление алкоголем, ФП, бессимптомный стеноз сонных артерий и обструктивное апноэ сна [3, 5].

Совокупность факторов риска определяет индивидуальный сердечно-сосудистый риск. Для систематизации оценки риска у взрослых в Российской Федерации рекомендуется использовать шкалу SCORE-2 или модифицированные отечественные алгоритмы с учётом национальных особенностей [5]. Врач-терапевт на ежегодных профилактических осмотрах и при диспансерном наблюдении обязан идентифицировать пациентов группы высокого риска и инициировать своевременную коррекцию выявленных нарушений.

На рисунке 1 представлена условная сравнительная значимость основных модифицируемых факторов риска ишемического инсульта, подлежащих приоритетному контролю в терапевтической практике.



Рис. 1. Модифицируемые факторы риска ишемического инсульта и их относительная клиническая значимость

Артериальная гипертензия как ведущий фактор риска

Артериальная гипертензия является наиболее распространённым и клинически значимым модифицируемым фактором риска инсульта. Снижение систолического артериального давления (АД) на 10 мм рт. ст. ассоциировано с уменьшением риска инсульта примерно на 30 % [5]. Согласно клиническим рекомендациям по АГ, целевые значения АД для большинства взрослых составляют менее 140/90 мм рт. ст., а для отдельных категорий (СД, ХБП, высокий сердечно-со-

судистый риск) — менее 130/80 мм рт. ст. [6]. При вторичной профилактике ишемического инсульта/ТИА целевое АД также рекомендуется ниже 130/80 мм рт. ст. [3].

Врач-терапевт осуществляет скрининг АД у всех пациентов старше 18 лет, диагностирует АГ, подбирает антигипертензивную терапию и контролирует приверженность лечению. При резистентной АГ или подозрении на вторичную гипертензию показана консультация кардиолога или врача-нефролога, однако базовое ведение и мониторинг остаются в компетенции терапевта.

В таблице 1 систематизированы основные модифицируемые факторы риска ишемического инсульта с указанием целевых параметров контроля согласно клиническим рекомендациям Минздрава России.

Таблица 1. Модифицируемые факторы риска ишемического инсульта и целевые параметры профилактики

Фактор риска	Целевой показатель / вмешательство	Уровень доказательности
Артериальное давление	< 130/80 мм рт. ст. (ИИ/ТИА, СД, ХБП); антигипертензивная терапия	Высокий [3, 6]
Дислипидемия	ХС ЛНП < 1,4 ммоль/л или снижение $\geq 50\%$; статины	Высокий [3]
Сахарный диабет	HbA1c индивидуально; контроль гликемии, СМП риска	Высокий [3, 5]
Фибрилляция предсердий	Пероральные антикоагулянты при CHA ₂ DS ₂ -VASc ≥ 1 (муж.) / ≥ 2 (жен.)	Высокий [3]
Курение	Полный отказ; никотинзаместительная терапия	Высокий [3, 5]
Физическая активность	≥ 150 мин умеренной аэробной нагрузки в неделю	Средний [3, 5]

Первичная профилактика: задачи врача-терапевта

Скрининг и оценка сердечно-сосудистого риска

Первичная профилактика ишемического инсульта направлена на лиц, не переносивших ОНМК, но имеющих факторы риска. Врач-терапевт проводит скрининг АД, определение липидного спектра, глюкозы крови, оценку индекса массы тела, анамнеза курения и семейного анамнеза сердечно-сосудистых заболеваний. Пациентам с ФП или подозрением на нарушения ритма сердца показана регистрация электрокардиограммы (ЭКГ); при клинических показаниях — холтеровское мониторирование [3, 5, 6].

На основании совокупности факторов терапевт определяет группу сердечно-сосудистого риска и выбирает интенсивность профилактических мероприятий. Для пациентов очень высокого риска (например, при сочетании СД и АГ с поражением органов-мишеней) показана немедленная гиполипидемическая терапия статинами независимо от исходного уровня холестерина липопротеинов низкой плотности (ХС ЛНП) [5, 6].

Немедикаментозная профилактика

Модификация образа жизни является основой первичной профилактики и должна рекомендоваться всем пациентам независимо от фармакотерапии. К ключевым мероприятиям относятся отказ от курения, ограничение потребления соли (< 5 г/сут), снижение массы тела при ожирении, средиземноморский или DASH-тип питания, регулярная физическая активность и ограничение алкоголя [3, 5]. Врач-терапевт играет центральную роль в мотивационном консультировании: именно доверительные отношения с участковым врачом повышают приверженность пациентов к длительной профилактической терапии.

Медикаментозная профилактика

Антигипертензивная терапия рекомендуется пациентам с АД $\geq 140/90$ мм рт. ст. и лицам с АД $\geq 130/80$ мм рт. ст. при высоком или очень высоком сердечно-сосудистом риске [6]. Гиполипидемическая терапия назначается в соответствии с категорией риска: для первичной профилактики у лиц очень высокого риска целевой уровень ХС ЛНП — менее 1,4 ммоль/л [5]. Антитромботическая профилактика в первичной профилактике инсульта применяется преимущественно у пациентов с ФП: выбор перорального антикоагулянта определяется шкалами CHA₂DS₂-VASc и HAS-BLED [3]. Рутинное назначение антиагрегантов здоровым лицам низкого риска не рекомендуется ввиду риска геморрагических осложнений.

Сахарный диабет и фибрилляция предсердий

Сахарный диабет увеличивает риск ишемического инсульта в 1,5–3 раза и ассоциирован с более тяжелым течением ОНМК и худшими функциональными исходами [3, 5]. Пациенты с СД относятся к группе высокого или очень высокого

сердечно-сосудистого риска; им показан строгий контроль АД (< 130/80 мм рт. ст.), гликемии (индивидуальные целевые значения HbA1c) и гиполипидемическая терапия [3, 6]. Врач-терапевт при диспансерном наблюдении больных СД ежеквартально оценивает факторы риска инсульта, проводит осмотр сосудов нижних конечностей, контроль альбуминурии и при необходимости корректирует сахароснижающую терапию совместно с эндокринологом.

Фибрилляция предсердий — причина до 20 % ишемических инсультов. Частота ФП возрастает с годами, и значительная доля случаев выявляется впервые при госпитализации по поводу инсульта. Скрининг ФП путём пальпации пульса и регистрации ЭКГ рекомендуется лицам старше 65 лет, а при наличии факторов риска — с 45–50 лет [3, 5]. При выявлении ФП терапевт оценивает показания к антикоагулянтной терапии по шкале CHA₂DS₂-VASc (заболевания сердца, АГ, возраст, СД, перенесённый инсульт/ТИА, сосудистые заболевания, женский пол) и риск кровотечений по шкале HAS-BLED. Антикоагулянты при соответствующих показаниях снижают риск инсульта при ФП на 60–70 % [3].

Распознавание транзиторной ишемической атаки и неотложная маршрутизация

Транзиторная ишемическая атака — клинический синдром, обусловленный очаговым ишемическим повреждением головного мозга без формирования инфаркта и с полным регрессом симптоматики в течение 24 часов. ТИА является неотложным состоянием: риск развития инсульта в ближайшие дни достигает максимума, особенно в первые 48–72 часа [3]. Клинические рекомендации предписывают госпитализацию или организацию экстренного обследования (включая нейровизуализацию) в максимально сжатые сроки для всех пациентов с подозрением на ТИА.

Для быстрой идентификации инсульта и ТИА в первичном звене широко применяется мнемонический алгоритм FAST: Face (асимметрия лица при улыбке), Arms (слабость или парез в руке), Speech (нарушение речи), Time (необходимость немедленного вызова скорой помощи). Врач-терапевт при приёме на дому или в поликлинике обязан провести неврологический скрининг при любых жалобах на внезапную слабость, онемение, нарушение речи, зрения или координации. При подозрении на ОНМК или ТИА показана экстренная госпитализация в стационар, оснащённый для оказания помощи при инсульте, либо вызов бригады скорой медицинской помощи [3].

Важно помнить, что у большинства пациентов симптомы ТИА к моменту визита к врачу могут полностью регрессировать. Отсутствие объективной неврологической симптоматики не исключает диагноза и не должно служить основанием для отсрочки обследования. Отсрочка диагностики ассоциирована с четырёхкратным увеличением риска инсульта в последующие 90 дней.

При обследовании пациента с подозрением на ТИА врач-терапевт проводит оценку уровня сознания по шкале Глазго, исследует черепные нервы (особенно VII и XII пары), силу и тонус в конечностях, чувствительность, координацию (пробы пальценосовой и пяточноколенной), речь и когнитивные функции. Любое внезапное очаговое неврологическое дефицитное состояние, даже кратковременное, требует исключения ОНМК. Дифференциальный диагноз включает эпилептический припадок Тодда, мигрень с аурой, гипогликемию, синкопальные состояния и поражения периферических нервов; ключевое значение имеют внезапность начала, очаговый характер симптомов и наличие факторов риска атеросклероза или эмболии [3].

Вторичная профилактика ишемического инсульта

Принципы ведения пациентов после инсульта и ТИА

У пациентов, перенёвших ишемический инсульт или ТИА, риск повторного ОНМК повышен почти в 10 раз и составляет около 25–30 % [3]. Наиболее высок риск рецидива в первые недели после первичного события, поэтому вторичная профилактика должна быть начата как можно раньше — не позднее 48 часов от развития ишемического инсульта [3]. После выписки из стационара ведущую роль в долгосрочном наблюдении, коррекции терапии и контроле факторов риска выполняет врач-терапевт.

Вторичная профилактика включает три взаимодополняющих направления: немедикаментозные методы (модификация образа жизни), лекарственную терапию (антигипертензивные, антитромботические и гиполипидемические средства) и хирургические вмешательства (каротидная эндартерэктомия или стентирование при гемодинамически значимом стенозе внутренней сонной артерии) [3].

Антитромботическая и гиполипидемическая терапия

Пациентам с некардиоэмболическим ишемическим инсультом/ТИА рекомендуется длительная антитромботическая терапия (ацетилсалициловая кислота, клопидогрел или их комбинации в соответствии с клинической ситуацией) [3]. При кардиоэмболическом механизме (ФП, тромбы в полостях сердца) показаны пероральные антикоагулянты — прямые или непрямые [3]. Выбор режима определяется неврологом/кардиологом в стационаре, однако

терапевт контролирует приверженность, мониторинг клинико-лабораторных показателей и своевременное выявление осложнений на амбулаторном этапе.

Большинству взрослых пациентов с ишемическим инсультом/ТИА рекомендуется высокоинтенсивная терапия статинами в максимально переносимых дозах с целевым уровнем ХС ЛНП менее 1,4 ммоль/л или снижением на $\geq 50\%$ от исходного [3]. При недостижении целевых значений в течение 4–12 недель показано добавление эзетимиба; при дальнейшей неэффективности — ингибиторов PCSK9 [3]. Врач-терапевт осуществляет контроль липидного профиля не реже 1 раза в 3–6 месяцев на этапе подбора терапии и далее — в соответствии с клинической ситуацией.

Диспансерное наблюдение и междисциплинарное взаимодействие

После перенесённого инсульта пациент нуждается в регулярном наблюдении: контроль АД, оценка неврологического статуса, коррекция сопутствующих заболеваний, реабилитационных мероприятий и психоэмоционального состояния. Врач-терапевт координирует взаимодействие с врачом-неврологом, кардиологом, медицинским психологом и специалистами по медицинской реабилитации. Показания для плановой консультации невролога включают рецидивирующие ТИА, прогрессирование неврологического дефицита, эпилептические припадки после инсульта, когнитивные нарушения и необходимость коррекции вторичной профилактики при непереносимости стандартной терапии [3].

Особую группу составляют пациенты с бессимптомным стенозом внутренней сонной артерии: при стенозе $\geq 60\%$ и высоком периоперационном риске показано хирургическое лечение после соответствующего обследования [3]. Выявление шума над сонной артерией при аускультации на приёме у терапевта должно служить основанием для направления на ультразвуковое исследование брахиоцефальных артерий.

Практические рекомендации для врача-терапевта

На основании проанализированных данных и действующих клинических рекомендаций можно сформулировать следующие практические ориентиры для работы врача-терапевта участкового:

1. проводить измерение АД у каждого пациента старше 18 лет не реже 1 раза в год, а у лиц с АГ — при каждом визите;
2. оценивать сердечно-сосудистый риск с использованием валидированных шкал и определять индивидуальные целевые показатели;
3. назначать и титровать антигипертензивную, гиполипидемическую и антитромботическую терапию в соответствии с клиническими рекомендациями;
4. мотивировать пациентов к отказу от курения, нормализации массы тела и регулярной физической активности;
5. при подозрении на ТИА или инсульт немедленно организовывать экстренную госпитализацию, не дожидаясь полного регресса симптомов;
6. обеспечивать диспансерное наблюдение за всеми перенёсшими ишемический инсульт/ТИА с контролем АД, липидов и приверженности терапии;
7. своевременно направлять пациентов к врачу-неврологу при рецидивирующих неврологических симптомах, когнитивном снижении или непереносимости профилактической терапии.

Обсуждение

Профилактика ишемического инсульта — междисциплинарная задача, однако именно амбулаторное звено, представленное врачом-терапевтом, обеспечивает непрерывность наблюдения за наиболее многочисленной группой пациентов с хроническими сердечно-сосудистыми заболеваниями. Эффективность профилактики определяется не только наличием клинических рекомендаций, но и их практической реализацией: регулярным скринингом, достижением целевых уровней АД и ХС ЛНП, а также распознаванием прединсультных состояний.

Сохраняющаяся региональная вариабельность смертности от ОНМК [4] указывает на необходимость дальнейшего совершенствования организации помощи, в том числе развития сети инсультных центров и повышения компетенций медицинского персонала первичного звена. Для врача-терапевта, планирующего специализацию в области неврологии, углублённое владение вопросами цереброваскулярной патологии является профессиональным приоритетом: значительная часть неврологической симптоматики впервые выявляется именно на приёме у терапевта.

Следует отметить и проблему приверженности лечению. По данным отечественных исследований, не более 40–50 % пациентов с АГ достигают целевых значений АД, а отмена статинов по причине субъективной непереносимости остаётся частой причиной недостаточной гиполипидемической профилактики. Врач-терапевт, обладающий длительным контактом с пациентом и его семьёй, находится в наилучшей позиции для разъяснения необходимости постоянного приёма препаратов, раннего выявления побочных эффектов и коррекции терапии без необоснованной отмены.

Перспективным направлением является внедрение цифровых инструментов: электронных диспансерных карт, телемониторинга АД и напоминаний о приёме лекарств. Однако технологические решения не заменяют клинического мышления и неврологического скрининга на приёме. Для специалиста, переходящего из терапии в неврологию, практический опыт ведения пациентов с цереброваскулярными факторами риска формирует прочную основу для дальнейшей работы с острыми и хроническими неврологическими заболеваниями.

Ограничением настоящего обзора является акцент на амбулаторном этапе и наиболее распространённых механизмах ишемического инсульта. Вопросы острой реперфузионной терапии, нейровизуализации и специализированной реабилитации выходят за рамки компетенции терапевта, но требуют понимания для обеспечения преемственности помощи.

Заключение

Ишемический инсульт сохраняет высокую социальную и клиническую значимость в Российской Федерации. Врач-терапевт играет ключевую роль в первичной профилактике (скрининг факторов риска, коррекция АГ, дислипидемии, СД, ФП и модификация образа жизни), в раннем распознавании ТИА и в долгосрочной вторичной профилактике после перенесённого ОНМК. Реализация положений клинических рекомендаций Минздрава России [3, 5, 6] на уровне поликлиники способна существенно снизить заболеваемость, рецидивирование и инвалидизацию. Систематическое повышение квалификации терапевтов в области цереброваскулярной профилактики является важным элементом национальной программы борьбы с инсультом и представляет собой логичный профессиональный переход к специализации в неврологии.

Литература:

1. Игнатьева В. И., Стародубов В. И., Подколзин А. А., Михайлова О. П. Распространённость инсульта в Российской Федерации (по данным опросов населения 2003–2023 гг.): обзор // Клиническая медицина. — 2024. — URL: <https://cyberleninka.ru/article/n/rasprostranennost-insulta-v-rossiyskoy-federatsii-po-dannym-oprosov-naseleniya-2003-2023-gg-obzor> (дата обращения: 30.06.2025).
2. Статистика инсульта // Северо-Западный государственный медицинский университет им. И. И. Мечникова. — URL: <https://szgmu.ru/rus/m/457/> (дата обращения: 30.06.2025).
3. Клинические рекомендации «Ишемический инсульт и транзиторная ишемическая атака» (взрослые). — Министерство здравоохранения Российской Федерации, 2024. — URL: https://www.consultant.ru/document/cons_doc_LAW_491652/ (дата обращения: 30.06.2025).
4. Самородская И. В., Суханова И. Н., Малявина У. С., Стародубов В. И. Вариабельность и динамика региональных показателей смертности от острых нарушений мозгового кровообращения // Врач. — 2023. — URL: <https://vrach-journal.ru/ru/25877305-2023-10-02> (дата обращения: 30.06.2025).
5. Клинические рекомендации «Болезни сердечно-сосудистой системы. Профилактика сердечно-сосудистых заболеваний» (взрослые). — Министерство здравоохранения Российской Федерации. — URL: <https://cr.minzdrav.gov.ru/> (дата обращения: 30.06.2025).
6. Клинические рекомендации «Артериальная гипертензия у взрослых». — Министерство здравоохранения Российской Федерации, 2024. — URL: <https://cr.minzdrav.gov.ru/> (дата обращения: 30.06.2025).

ЭКОЛОГИЯ

Современное состояние биоценозов реки Чилипси и ее притоков в Туапсинском районе Краснодарского края

Сасикова Наталья Сергеевна, преподаватель-исследователь;
Мартыненко Сергей Максимович, аспирант;
Самарцева Александра Сергеевна, аспирант
Кубанский государственный аграрный университет имени И. Т. Трубилина (г. Краснодар)

В работе представлены результаты комплексных исследований качественных и количественных показателей биоценозов реки Чилипси и шести её притоков (рек Ореховка, Маслова, Индюшка, Сухая, Скакуха, Букепка), расположенных в Туапсинском районе Краснодарского края. На основе анализа географического положения, морфометрии и гидролого-гидрохимического режима водных объектов дана их краткая физико-географическая характеристика.

На базе ихтиологических и гидробиологических проб, отобранных и обработанных по общепринятым методикам, составлено структурированное описание основных компонентов экосистемы: фитопланктона, зоопланктона, зообентоса и ихтиофауны. Установлено, что средняя биомасса фитопланктона в изученных водотоках варьирует от 0,018 г/м³ (р. Сухая) до 0,071 г/м³ (р. Индюшка). Максимальная биомасса зообентоса зафиксирована в р. Чилипси и составляет 3,82 г/м². Определён современный видовой состав местной ихтиофауны.

Полученные результаты имеют важное прикладное значение и необходимы для объективной оценки состояния водных биоценозов, а также для проведения мониторинга воздействия антропогенных факторов на биоресурсы и среду их обитания.

Ключевые слова: реки, фитопланктон, зоопланктон, зообентос, концентрация, биомасса, ихтиофауна.

Введение

Гидрографическая сеть Туапсинского района, в пределах которого проводились исследования, относится к бассейну Чёрного моря и характеризуется обилием малых порожистых водотоков преимущественно горного типа [3]. На отдельных участках их русла загромождены обломками скальных пород и аллювиально-делювиальным материалом (поваленными деревьями). Дно преимущественно каменистое, в нижнем течении — песчано-галечниковое. Берега обрывистые, высотой от 2 до 7 м. Поймы водотоков обычно узкие, прерывистые, и лишь в низовьях местами достигают ширины 200–300 м.

Наиболее значимой в регионе является река Туапсе, имеющая важное водохозяйственное значение для одноименного города. Данный водоток образуется при слиянии рек Чилипси и Пшенахо [1].

В качестве основного объекта исследований была выбрана река Чилипси — правый составляющий исток р. Туапсе. Специфика её географического положения, гидрологического режима и слабая антропогенная нарушенность в верхнем течении делают этот водоток эталонным для изучения естественной динамики местных гидробиоценозов. В связи с этим детальное рассмотрение структуры её сообществ проводилось по всему продольному профилю русла — от истоков до зоны слияния.

Отсутствие в современной литературе сведений о гидробиоценозах реки Чилипси определяет научную и практическую значимость их комплексного изучения. Целью настоящей работы являлась оценка современного состояния биоценозов реки и её притоков на основе структурно-функциональных показателей развития планктонных, бентосных и ихтиологических сообществ.

Река Чилипси берет начало на южном склоне Главного Кавказского хребта в 2 км к северо-востоку от горы Лысая (976 м н. у.м.), на высоте 350 м над уровнем моря. Течёт в юго-восточном направлении и впадает, у платформы Индюк, с правого берега в реку Туапсе. Длина р. Чилипси — 16 км. После впадения, река Туапсе, носит второе название — Большая Чилипси.

Пойма реки двусторонняя: левобережная — шириной 0,2 км, частично залесена, используется под сельхозугодья; правобережная — шириной 0,2 км, луговая, залесенная.

Русло реки сравнительно прямолинейное, хорошо проработанное, ящикообразное, шириной до 19 м, сложено галькой и валунами.

Берега высотой до 1,5 м, местами обрывистые. Оба берега подвержены размывам, сложены суглинками, задернованы, поросли кустарником и лиственными деревьями.

Гидрографическая сеть реки Чилипси включает 10 малых притоков (длиной менее 10 км). В качестве объектов настоящего исследования были выбраны 6 наиболее значимых водотоков (таблица 1), репрезентативно отражающих особенности формирования биоценозов бассейна.

Таблица 1

Название	Принадлежность притока	Исток	Устье	Длина, км
р. Чилипси	пр. приток р. Туапсе	в 2 км к с-в от горы Лысая	у платформы Индюк	16
р. Маслова	лев. приток р. Чилипси	горн. массив в р-не г. Семашко и г. Круглая	в створе ж/д станции Кривенковская	около 7
р. Скакуха	лев. приток р. Чилипси	с-з склон г. Круглой	в р-не с. Индюк	около 6
р. Сухая	пр. приток р. Чилипси	водораздел между г. Агой и г. Лысой	в р-не с. Индюк	около 6
р. Букепка	пр. приток р. Чилипси	восточный склон г. Букепка	в р-не с. Индюк и платформы № 1744	около 8
р. Индюшка	лев. приток р. Чилипси	примерно 44.2596° с. ш., 39.2747° в. д.	около 44.2396° с. ш., 39.242° в. д.	около 5
р. Ореховка	пр. приток р. Чилипси	склоны г. Лысой	в р-не с. Индюк	около 3

Водный режим притоков реки Чилипси характеризуется прохождением паводков в течение осенне-зимнего периода (ноябрь–март) и довольно устойчивой меженью (с мая по октябрь). Низкий сток в летний период нарушается дождевыми паводками.

Внутригодовое распределение стока крайне неравномерное. В средний по водности год, в период с декабря по апрель, проходит 77 % годового стока. Остальные 23 % — в период с мая по ноябрь.

Годовой ход уровня характеризуется частыми и резкими подъёмами в холодную часть года от дождевых паводков, формирующих, обычно в декабре–январе, годовой максимум уровня. Иногда наивысшие в году паводки, пик которых может превысить пик зимних паводков, случаются в летнее время года в результате выпадения интенсивных ливневых осадков, вызывающих катастрофические паводки.

Паводки отличаются большой интенсивностью подъёма и короткой продолжительностью стояния высоких уровней (на протяжении суток). Количество паводков может достигать 4–5 в месяц. Время прохождения паводка исчисляется часами и не превышает суток. Максимальный уровень воды держится не более одного часа.

Реки, большую часть продольного профиля русла, мелководны, в период летне-осенней межени (май — октябрь) на отдельных участках пересыхают, разбиваясь на отдельные плёсы глубиной 10–15 см с участками сухого русла между ними. Период пересыхания приходится на август–сентябрь.

Материалы и методы

Материалами во время исследования послужили гидробиологические и ихтиологические данные. Отбор и обработка гидробиологических проб, анализ ихтиологического материала проводились по общепринятым методикам. Сбор материала проводили в верхнем, среднем и нижнем течении указанных рек, по сезонам в течении 2024–2025 гг.

Пробы фитопланктона и зоопланктона отбирались и обрабатывались по общепринятой методике [10]. Сбор проб зообентоса на каждой станции проводили в двукратной повторности. Обработка материала осуществлена стандартными методами по методикам, приведенным в списке литературы. Определение таксонов зообентоса проводили по методикам, приведенным в списке литературы.

Состав ихтиофауны приведен согласно опросам рыболовов-любителей и литературных источников.

Комплексную оценку состояния биоценозов изучаемых рек провели на основе анализа численности, биомассы и количества таксонов, обитающих в них.

Результаты и обсуждение

Фитопланктон является первым трофическим уровнем в экосистеме и основным продуцентом органического вещества во многих водоёмах и водотоках.

Экосистемы анализируемых рек относятся к экосистемам лотического типа [9]. Из-за достаточно высоких скоростей течения показатели численности и биомассы планктонных организмов водотоков сравнительно невысоки, но значительного развития достигают организмы зообентоса.

В ходе исследований установлено, что фитопланктонные сообщества реки Чилипси и ее притоков представлены сине-зелёными (*Cyanophyta*), зелёными (*Chlorophyta*), золотистыми (*Chrysophyta*) и диатомовыми (*Bacillariophyta*) водорослями. Наиболее широко были распространены диатомовые и зелёные водоросли, отмеченные во всех пробах.

Анализ данных по численности и биомассе фитопланктона рек (табл. 2) показывает, что он был развит слабо, что характерно для лотических экосистем [9].

Таблица 2. Количественные показатели фитопланктона рассматриваемых рек

Реки	Численность, тыс. кл. /м ³	Биомасса, г/м ³
Чилипси	36,0	0,029
Маслова	13,2	0,045
Скакуха	11,8	0,042
Сухая	7,4	0,018
Букепка	12,7	0,037
Индюшка	17,7	0,071
Ореховка	10,8	0,069

Сезонная динамика показателей развития фитопланктона в реках проявляется в возрастании численности и биомассы клеток от весенне-летнего периода к осеннему. Данный характер сезонной динамики развития фитопланктона связан с прогревом воды и накоплением к сентябрю-октябрю в воде большого количества биогенных веществ из-за уменьшения водности и проточности.

Биомасса фитопланктона рек варьирует от 0,018 до 0,071 г/м³. Наиболее высокие показатели средних значений биомассы отмечены в реке Индюшка.

Редкие исчезающие виды, внесённые в Красные книги России [8] и Краснодарского края [6], в составе фитопланктона р. Чилипси, в месте проведения намечаемой хозяйственной деятельности — не отмечены.

Зоопланктонные сообщества исследованных рек насчитывают несколько значимых групп. Облигатно-планктические формы («истинный» зоопланктон): *Rotatoria* (коловратки), *Cladocera* (ветвистоусые ракообразные), *Copepoda* (веслоногие ракообразные). Факультативные формы — ведущие планктонный образ жизни на определенной стадии развития. Состав группы «*Varia*» включает несколько групп беспозвоночных разного происхождения: поднимающихся в толщу воды на предимагинальных стадиях водных личинок амфибиотических насекомых, червей (*Oligochaeta*), пелагических личинок комаров из семейства из семейства хирономид (*Chironomidae*), ракушковых раков (*Ostracoda*), (табл. 3).

В анализируемых реках наиболее существенна роль временного компонента планктонных сообществ.

По количеству в зоопланктоне рек преобладают *Rotatoria* (коловратки). По биомассе доминируют представители группы «прочие» (*Varia*). Этот факт связан с тем, что в данную группу входят достаточно крупные организмы с высокой индивидуальной массой (преимущественно членистоногие).

Сезонная динамика численности зоопланктона рек заключается в её увеличении от весенне-летнего периода к осеннему. Биомасса, наоборот, незначительно снижается.

Таблица 3. Количественные показатели зоопланктона рассматриваемых рек

Реки	Численность, тыс. кл./м ³	Биомасса, г/м ³
Чилипси	542	0,18
Маслова	0,430	0,052
Скакуха	0,393	0,049
Сухая	0,115	0,031
Букепка	0,410	0,051
Индюшка	0,198	0,067
Ореховка	0,372	0,038

Биомасса зоопланктона рек (таблица 3) варьирует от 0,031 до 0,18 г/м³. Наиболее высокие показатели средних значений биомассы отмечены в реке Чилипси.

Редкие исчезающие виды, внесённые в Красную книгу Краснодарского края [5] и Красную книгу России [7], в составе зоопланктона р. Чилипси, в месте проведения намечаемой хозяйственной деятельности, не отмечены.

Донные сообщества исследуемых рек развиваются в условиях выраженного паводочного режима, приводящего к пересыханию части русла во время летне-осенней межени, а также при высоких скоростях течения и оптимальном кислородном режиме. Обилие разнородных стадий (перекаты, плёсы, галечные и песчаные грунты, древесные остатки) определяет высокое видовое и экологическое разнообразие макрозообентоса.

В составе зообентоса р. Чилипси и её притоков идентифицированы плоские, круглые и кольчатые черви, брюхоногие моллюски (устойчивые к пересыханию водотоков благодаря способности к анабиозу), а также личиночные стадии амфибиотических насекомых, представленные отрядами подёнок, ручейников и двукрылых.

Наибольшее разнообразие достигают амфибиотические насекомые, представленные поденками (*Ephemeroptera*), веснянками (*Plecoptera*), ручейниками (*Trichoptera*), жестоккрылыми (*Coleoptera*) и двукрылыми (*Diptera*).

Для донных беспозвоночных рассматриваемых водотоков характерно численное доминирование реофильных личинок двукрылых, по биомассе доминирование организмов, характеризующиеся крупными индивидуальными размерами — личинки подёнок.

Биомасса кормового бентоса рек (таблица 4) варьирует от 0,96 до 3,82 г/м². Наиболее высокие показатели средних значений биомассы отмечены в реке Чилипси.

Таблица 4. Количественные показатели кормового зообентоса рассматриваемых рек

Реки	Численность, экз./м ²	Биомасса, г/м ²
Чилипси	142,6	3,82
Маслова	36,0	1,71
Скакуха	34,0	1,62
Сухая	21,0	0,96
Букепка	35,0	1,68
Индюшка	51,0	2,78
Ореховка	37,0	1,17

Редкие исчезающие виды, внесённые в Красную книгу Краснодарского края [5] и Красную книгу России [7], в составе зообентоса р. Чилипси, в месте проведения намечаемой хозяйственной деятельности, не отмечены.

Данные исследований видового состава ихтиофауны рассматриваемых рек, представлены в таблице 5.

Ихтиофауна изученных рек оказалась однотипной и представлена от 9 до 5 видов (табл. 5).

Таблица 5. Состав ихтиофауны реки Чилипси и ее притоков

№ п/п	Вид рыб /река	Чилипси	Маслова	Скакуха	Сухая	Букепка	Индюшка	Ореховка
1.	Ручьевая форель (<i>Salmo trutta labrax</i>)	+	+	+	–	+	+	+
2.	Южная быстрянка (<i>Alburnoides bipunctatus fasciatus</i>)	+	+	+	+	+	+	+
3.	Колхидский голянь (<i>Phoxinus phoxinus colchicus</i>)	+	+	+	+	+	+	+
4.	Кавказский пескарь (<i>Gobio caucasicus</i>)	+	+	+	+	+	+	+
5.	Колхидский усач (<i>Barbus tauricus escherichii</i>)	+	+	+	–	+	–	–
6.	Кавказский голавль (<i>Leuciscus cephalus orientalis</i>)	+	+	+	+	+	+	+
7.	Азово-черноморская шемая (<i>Alburnus mento</i>)	+	+	+	–	+	+	–
8.	Колхидский подуст (<i>Chondrostoma colchicum</i>)	+	+	+	–	+	–	–
9.	Речной бычок Родиона (<i>Neogobius rhodioni</i>)	+	+	+	+	+	+	+
Количество видов		9	9	9	5	9	7	6

Самое высокое видовое разнообразие (9 видов) отмечено в четырех реках. Наибольшим разнообразием характеризовалось семейство карповые (*Cyprinidae*).

Минимальное количество видов отмечено в ихтиофауне реки Сухой. Это обусловлено спецификой гидрологического режима водотока: его малой протяжённостью, низкой водностью, а также длительной летне-осенней меженью, что в совокупности формирует неблагоприятные условия для обитания и воспроизводства рыб.

Все виды рыб, обитающие в исследуемых водотоках, являются туводными (жилыми); проходные формы не зарегистрированы. Среди редких видов, подлежащих охране (табл. 5), отмечены азово-черноморская шемая и ручьевая форель (кумжа).

По отношению к динамике водной среды всё рыбное население речной системы относится к реофильному комплексу. По характеру нерестового субстрата большинство видов представляют литофильную экологическую группу. При этом ручьевая форель откладывает икру в искусственные гнёзда, устраиваемые в галечном грунте, в то время как кавказский пескарь выступает типичным псаммофилом, выметывающим икру на песчаный субстрат.

В исследуемых реках полностью отсутствуют фитофильные виды рыб. Это обусловлено дефицитом субстрата для их естественного воспроизводства — высшей водной и воздушно-водной растительности, а также паводочным режимом водотоков в течение всего года, исключающим длительное затопление поймы.

Рыбопродуктивность русловых нерестилищ исследуемых рек, в среднем, оценивается на уровне 0,005 т/га.

Размножение большинства видов рыб, обитающих в реке Чилипси и её притоках, приурочено к весеннему периоду с пиком интенсивности в апреле — мае. В отличие от них, массовый нерест черноморской кумжи (ручьевой форели) носит осенний характер и продолжается с середины сентября до середины ноября.

Заключение

В результате проведенных исследований установлено, что наиболее высокие показатели средней биомассы фитопланктона (до 0,071 г/м³) характерны для реки Индюшка. По количественным характеристикам зоопланктона все изученные водотоки могут быть отнесены к олиготрофному типу. Максимальная биомасса зообентоса зафиксирована в реке Чилипси, где её значение достигало 3,82 г/м².

Ихтиофауна исследуемых рек представлена преимущественно туводными видами. Наибольшее видовое разнообразие рыб отмечено в реках Чилипси, Маслова, Скакуха и Букепка.

Комплексный анализ реки Чилипси и её притоков позволил дать объективную оценку состоянию кормовой базы и определить современный состав ихтиоценозов. Полученные данные могут послужить научной основой для организации рационального природопользования, разработки природоохранных мероприятий и сохранения биологического разнообразия региона в условиях возрастающего антропогенного воздействия.

Литература:

1. Аракелов М. С. Геоэкологическое районирование приморских территорий Туапсинского района на основе индикаторного подхода // Ученые записки № 18. Научно-теоретический журнал. СПб.: Изд-во РГГМУ, 2011. С. 77–82.
2. Атлас пресноводных рыб России: в 2 т. / Под ред. Ю. С. Решетникова. — М.: Наука, 2003. Т. 1–2. — 632 с.
3. Белюченко И. С. Экологические проблемы Кубани // Туапсинский район. Краснодар: Изд-во КГАУ, 2002. 342 с.
4. Емтыль М. Х., Иваненко А. М. Рыбы Юго-запада России. — Краснодар: Изд-во КубГУ, 2002. — 340 с.
5. Красная книга Краснодарского края. Животные. III издание / Отв. ред. А. С. Замотайлов, Ю. В. Лохман, Б. И. Вольфов. — Краснодар: Адм. Краснодар. края, 2017. — 720 с.
6. Красная книга Краснодарского края. Растения и грибы / Адм. Краснодар. края, отв. ред. С. А. Литвинская и др. — 3-е изд. — Краснодар: б.и., 2017. — 850 с.
7. Красная книга Российской Федерации, том «Животные». — 2-е издание. — М.: ФГБУ «ВНИИ Экология», 2021. — 1128 с.
8. Красная книга Российской Федерации. Растения и грибы. — 2-е офиц. изд. — М.: ВНИИ «Экология», 2024. — 944 с.
9. Круглова А. Н. Значение озерного зоопланктона в формировании кормовой базы озерно-речной системы р. Лижма (бассейн Онежского озера) // Гидробиологический журнал — 1981. — Т. 17. — Вып. 1. — С. 28–33
10. Лурье, П. М., Панов В. Д. Характеристика источников питания рек Черноморского побережья Кавказа // Известия Кубанского государственного университета. Естественные науки. — 2012. — № 1. — С. 88–92.
11. Методика изучения биогеоценозов внутренних водоемов / Отв. ред. Ф. Д. Мордухай-Болтовской. М.: Наука, 1975. 240 с.
12. Методические рекомендации по сбору и обработке материалов при гидробиологических исследованиях на пресноводных водоемах. Фитопланктон и его продукция / Ред. Г. Г. Винберг, Г. М. Лаврентьева. — Л.: ГосНИОРХ, ЗИН. — 1981. — 32 с.

13. Мордухай-Болтовского Ф. Д. Экология водных беспозвоночных / Ин-т биологии внутр. вод им. И. Д. Папанина РАН, Борок, 30 октября — 2 ноября 2010 г. — Ярославль: Принтхаус, 2010—376 с.
14. Определитель пресноводных водорослей СССР / Под ред. М. М. Голлербах. — М.-Л.: Изд-во АН СССР, 1951—1986. Т. 1—14.
15. Определитель пресноводных беспозвоночных Европейской части СССР (планктон и бентос) / Отв. ред. Л. А. Кутикова, Я. И. Старобогатов. — Л.: Гидрометеиздат. — 1977. — 511с.
16. Правдин И. Ф. Руководство по изучению рыб (преимущественно пресноводных) / И. Ф. Правдин. — М. Пищевая промышленность. — 1966. — 376 с.
17. Руководство по методам гидробиологического анализа поверхностных вод и донных отложений / под ред. В. А. Абакумова. — Л.: Гидрометеиздат. — 1983. — 239 с.
18. Швердяев И. В. Особенности формирования и прохождения опасных паводков на реках Северо-Западного Кавказа: диссертация на соискание учёной степени кандидата географических наук. — Санкт-Петербург, 2017. — 152 с.

ФИЛОЛОГИЯ, ЛИНГВИСТИКА

Русский язык в цифровую эпоху: вызовы и перспективы

Агапов Артём Камоевич, учитель русского языка и литературы
МОБУ основная общеобразовательная школа № 44 города Сочи имени Веры Ивановны Пруидзе (Краснодарский край)

В статье автор рассматривает трансформацию русского языка под влиянием цифровых технологий. Анализируются основные тенденции языковых изменений в интернет-коммуникации, влияние социальных сетей и мессенджеров на речевую культуру, а также роль образовательных и государственных институтов в сохранении языковой нормы. Особое внимание уделено балансу между инновациями в языке и поддержанием его литературной основы. Результаты исследования могут быть полезны педагогам, лингвистам, специалистам в области медиакоммуникаций и всем, кто интересуется вопросами современного языкознания.

Ключевые слова: русский язык, цифровая эпоха, интернет-коммуникация, языковая норма, речевая культура, цифровизация образования.

Введение

Цифровая эпоха радикально изменила способы коммуникации, а вместе с ними — и функционирование языка. Русский язык, будучи одним из крупнейших мировых языков, не остаётся в стороне от этих процессов. С одной стороны, цифровые технологии расширяют возможности распространения языка и делают его более доступным для изучения и использования. С другой — порождают новые формы коммуникации, которые нередко идут вразрез с традиционными нормами литературного языка.

Актуальность темы обусловлена необходимостью осмысления происходящих изменений и выработки стратегий, которые позволят сохранить богатство и выразительность русского языка, одновременно адаптируя его к новым реалиям. Цель данной статьи — выявить ключевые тенденции трансформации русского языка в цифровой среде и определить возможные пути поддержания языковой культуры в условиях технологического прогресса.

Основные тенденции трансформации языка в цифровой среде

Современные цифровые платформы формируют особый тип речевой практики, для которого характерны следующие черты:

— **Упрощение синтаксиса и сокращение длины высказываний.** В мессенджерах и социальных сетях пользователи стремятся передать мысль максимально кратко.

Это приводит к распространению эллиптических конструкций, опущению союзов и вспомогательных слов. Например, вместо «Я сейчас не могу ответить, потому что занят» часто пишут просто «Сейчас не могу».

— **Активное заимствование англицизмов и использование компьютерного сленга.** Цифровая среда изначально развивалась на базе англоязычных технологий, что предопределило широкое проникновение английских слов и выражений в повседневную речь: «лайк», «чат», «пост», «флешмоб» и т. д.

— **Визуализация коммуникации.** Эмодзи, стикеры, гифки и мемы становятся полноценными элементами общения, дополняя или даже заменяя вербальную составляющую. Такой способ передачи смысла требует новых подходов к анализу коммуникации и расширяет понятие «текст».

— **Снижение требований к орфографической и пунктуационной норме.** В неформальной переписке пользователи часто пренебрегают правилами правописания, считая их избыточными. Это может приводить к размыванию представлений о норме, особенно у молодого поколения.

— **Демократизация языка и рост вариативности.** Интернет-пространство даёт возможность высказаться каждому, что способствует распространению разговорных, просторечных и жаргонных форм. В результате границы между литературным языком и его нелитературными вариантами становятся менее чёткими.

Эти тенденции не являются исключительно негативными: они отражают естественную адаптацию языка к новым условиям. Однако их масштаб и скорость распро-

странения требуют внимательного отношения со стороны лингвистического сообщества и системы образования.

Влияние цифровой среды на речевую культуру

Одним из наиболее заметных последствий цифровизации коммуникации является изменение представлений о речевой культуре. Если ранее владение нормами литературного языка считалось показателем образованности, то в цифровой среде на первый план нередко выходит скорость и удобство общения. Это создаёт определённый конфликт между традиционной языковой нормой и новыми коммуникативными практиками.

Особую тревогу вызывает влияние цифровой среды на языковую компетенцию детей и подростков. Постоянное пребывание в пространстве неформальной коммуникации может формировать у них устойчивые привычки к использованию сниженной лексики, нарушению грамматических и пунктуационных норм. В результате при переходе к формальным видам речевой деятельности (написанию сочинений, изложений, научных текстов) учащиеся испытывают серьёзные трудности.

Вместе с тем цифровая среда предоставляет и значительные возможности для развития речевой культуры. Образовательные платформы, онлайн-курсы, интерактивные тренажёры и цифровые словари позволяют сделать изучение языка более доступным и увлекательным. Важно научиться использовать эти инструменты для формирования осознанного отношения к языку и развития навыков грамотного письма.

Роль образования в сохранении языковой нормы

Система образования играет ключевую роль в поддержании языковой культуры в цифровую эпоху. Перед педагогами стоит сложная задача: с одной стороны, не отрицать реальность новых форм коммуникации, а с другой — формировать у учащихся понимание ценности литературного языка и необходимости соблюдения его норм в соответствующих ситуациях.

Эффективными могут быть следующие подходы:

— **Интеграция анализа цифровой коммуникации в учебный процесс.** На уроках русского языка можно разбирать примеры сообщений из социальных сетей, выявлять типичные ошибки и обсуждать, как можно выразить ту же мысль в соответствии с нормами литературного языка. Такой подход помогает учащимся осознать разницу между разными регистрами речи и научиться выбирать подходящий стиль в зависимости от ситуации.

— **Развитие критического отношения к информации и языку.** Важно учить школьников и студентов анализировать тексты, различать достоверные и недостоверные источники, понимать, как языковые средства могут использоваться для манипуляции. Эти навыки особенно актуальны в эпоху информационных технологий.

— **Использование цифровых инструментов для обучения.** Онлайн-тренажёры, интерактивные диктанты, образовательные игры и другие цифровые ресурсы могут сделать процесс изучения языка более мотивирующим и эффективным. При этом важно, чтобы эти инструменты были ориентированы на формирование устойчивых навыков грамотного письма и речи.

— **Формирование медиаграмотности.** Понимание особенностей цифровой коммуникации, умение распознавать языковые манипуляции и критически оценивать контент — важные компетенции современного человека. Их развитие должно стать неотъемлемой частью образовательного процесса.

Государственные и общественные инициативы по поддержке русского языка

Помимо системы образования, важную роль в сохранении языковой культуры играют государственные и общественные институты. В России реализуется ряд программ и проектов, направленных на поддержку русского языка:

— **Законодательные меры.** Федеральный закон «О государственном языке Российской Федерации» закрепляет статус русского языка как государственного и устанавливает требования к его использованию в официальной сфере.

— **Просветительские проекты.** Проводятся всероссийские акции, такие как «Тотальный диктант», которые способствуют популяризации грамотности и привлечению внимания к проблемам языковой культуры.

— **Поддержка научных исследований.** Лингвистические исследования, посвящённые изучению современного состояния русского языка и прогнозированию его развития, получают государственную поддержку и способствуют формированию научно обоснованных рекомендаций для системы образования и СМИ.

— **Развитие цифровых ресурсов.** Создаются электронные библиотеки, словари и справочные системы, которые делают знания о русском языке доступными для широкой аудитории.

Эти инициативы создают необходимую инфраструктуру для поддержания языковой нормы и формирования уважительного отношения к русскому языку в обществе.

Заключение

Русский язык в цифровую эпоху переживает период активной трансформации. Новые формы коммуникации, порождённые развитием технологий, вносят существенные изменения в речевую практику и представления о языковой норме. Эти изменения нельзя однозначно оценивать как негативные: они отражают естественный процесс адаптации языка к меняющимся условиям.

Однако стремительное распространение неформальных языковых практик создаёт риски для сохранения

языковой культуры, особенно среди молодого поколения. В этих условиях особенно важна роль системы образования, которая должна помочь учащимся осознать ценность литературного языка и научиться грамотно использовать его в разных ситуациях.

Государственные и общественные инициативы также вносят значительный вклад в поддержку русского языка. Сочетание научных исследований, просветительских про-

ектов и современных образовательных технологий позволяет создать условия для гармоничного развития языка в цифровую эпоху.

Таким образом, будущее русского языка зависит от способности общества найти баланс между инновациями и сохранением традиций, между свободой самовыражения и ответственностью за качество речевой коммуникации.

Литература:

1. Федеральный закон от 01.06.2005 № 53-ФЗ «О государственном языке Российской Федерации».
2. Кронгауз М. А. Русский язык на грани нервного срыва. — М.: АСТ, 2017.
3. Караулов Ю. Н. Русский язык и языковая личность. — М.: Наука, 1987.
4. Материалы проекта «Тотальный диктант» (ежегодные отчёты и аналитические обзоры).

Особенности системы персонажей в басне И. А. Крылова «Ворона и Лисица»

Хуснуллин Азат Ильнарович, студент
Казанский (Приволжский) федеральный университет

В статье автор рассматривает инварианты басенного жанра и основные направления новаторства, обусловленные национальным аспектом, культурным кодом и индивидуальным стилем писателя. Особое внимание уделяется анализу персонажей басни И. А. Крылова «Ворона и Лисица», выделяются основные черты образов Вороны и Лисицы в русской культуре, обосновывается их выбор И. А. Крыловым для аллегорического изображения обличаемых характеристик.

Ключевые слова: басня, жанровые инварианты, моральная сентенция, образная система, связь с фольклором и культурным кодом.

Басня — это малый эпический жанр, в котором имеется иносказательный смысл. В поучительном басенном сюжете действующими лицами чаще всего являются условные звери, отражающие в своем образе определенную черту или порок человечества.

Важно отметить, что сам по себе жанр басни очень древний. Согласно «Словарю литературоведческих терминов» Тимофеева Л. И., Тураева С. В его история начинается с произведений легендарного греческого баснописца Эзопа. Также отмечается, что «позднее басни Эзопа были переложены стихами на латинский язык Федром (1 в. н. э.). В дальнейшем они стали достоянием всех европейских литератур и послужили основным источником басенных сюжетов самых разных баснописцев, от Лафонтена до Крылова» [3].

Басня имеет ряд инвариантных элементов, структурных и содержательных компонентов, четко определяющих её жанровую природу. В первую очередь, стоит отметить, что любой басенный текст можно разделить на два композиционных элемента: занимательный рассказ и моральную сентенцию. Так, в басне совмещаются два начала художественное и дидактическое, первое способствует раскрытию второго, используется для создания иллюстрации к морали. При этом «в лучших баснях Лафонтена, Крылова и других баснописцев поэзия, образное выражение мысли преобладает и подчиняет себе дидактический элемент» [3].

К тому же постоянной особенностью жанра является использование аллегии, то есть воплощение какой-либо идеи в конкретных художественных образах и формах. Иносказательность проявляется в изображении тех или иных пороков и качеств в образах вымышленных персонажей, чаще всего зверей. При этом часто за конкретным животным закрепляется определенная черта и в дальнейшем она сохраняется в баснях различных авторов на протяжении тысяч лет, так, например, образ Лисицы отражает хитрость и у Эзопа, и у Крылова (хотя авторов разделяют 2 тысячелетия), образ Льва постоянно соотносится с деспотизмом и властью.

Басня настолько тесно переплетается с традицией и опирается на предыдущий литературный опыт, что для него характерно даже такое явление, как кочующие или бродящие сюжеты, устойчивые повествовательные схемы, фабульные комплексы, которые переходят из одной культуры в другую и трансформируются в зависимости от среды бытования.

Если учесть огромное число общих элементов (персонажи-животные, олицетворяющие определенные характеристики, кочующие сюжеты), может сложиться ощущение, что многие басни различных авторов одинаковы, фактически являются переводами одних и тех же античных текстов. Однако это неверное предположение, так

как при общности с точки зрения использования жанровых инвариантов, каждый автор привносит в свои басни новые компоненты и яркие черты. В большинстве случаев это связано с осознаваемым или неосознаваемым влиянием национальной среды, культурного кода. Так, любое произведение включает в себя особенности исторической эпохи и культуры, сложившейся вокруг автора художественного текста. В статье В. А. Тихомировой «Отражение этнокультурных различий восприятия текста в баснях» отмечается: «Стереотипные басенные тексты, сюжеты которых приписывают легендарному Эзопу, никогда не являлись застывшими и подвергались постоянному варьированию, приспособляясь к изменяющимся ситуациям исполнения. Однако даже в случае существования единого письменного греческого инварианта басенных текстов, каждый раз при переводе на другие языки когнитивная база переводчика, который воспринимал готовые сюжеты, пропускал их через себя и переводил на свой родной язык, непременно проявляла бы себя, и данные тексты подвергались бы своеобразной «национально-культурной» обработке и трансформации» [4].

В чем же проявляется это новаторство? Обычно новое выражалось через 2 направления изменений. Первое направление модификаций можно связать со стилем и формальной стороной произведения: выборе прозы или стиха, а в стихотворных текстах — в выборе того или иного размера. Например, А. П. Сумароков стал открывателем вольного ямба для басен, а И. А. Крылов закрепил его в своих вершинных произведениях и фактически возвел его в статус эталонного для этого жанра. Новаторство может быть связано с выбором разного рода лексических средств (например, ориентацией на язык простого населения, использование просторечных и разговорных оборотов, использование диминутивов) или особым соотношением дидактического и повествовательного компонентов.

Другое направление изменений связано с переосмыслением традиций в бродячих басенных сюжетах, чаще всего это выражается в выборе тех или иных образов, которые станут заменой для закреплённых в первичном тексте античного автора. Именно подобного рода изменения в басенном тексте «Вороны и Лисицы» станут предметом нашего дальнейшего рассмотрения.

Начнем анализ образной системы басни И. А. Крылова с выбора тех текстов предшественников, с которыми мы будем сравнивать произведения «самого народного поэта, превзошедшего всех баснописцев» (так высказывался об И. А. Крылове А. С. Пушкин). За основу для сравнения нами были взяты тексты Эзопа как основателя сюжета, связанного с обманом птицы лисой для получения еды, Федра как первого человека, переложившего этот сюжет стихами на латинский язык, византийского писателя Игнатия Диякона (VIII–IX вв.), русских поэтов XVIII века — В. К. Тредиаковского и А. П. Сумарокова.

Сюжет всех этих басен построен на противопоставлении двух образов – Вороны (в ранних произведениях

Ворона), имеющей сыр (у Эзопа — мясо), и Лисицы, пытающейся лестию и хитростью его присвоить.

С самого начала нужно обратить внимание на вариации наименования птицы в различных версиях одного и того же сюжета. Вплоть до Сумарокова основным действующим лицом произведения являлся Ворон, птица, символизирующая мудрость. Выбор этой птицы античными авторами был обоснован и моральной сентенцией, отличной от основного дидактического послания Крылова. У Эзопа основным пороком, обличаемым автором, была не лесть, а глупость. Так, в тексте легендарного баснописца в переводе М. Гаспарова Лисица говорит: «Эх, ворон, кабы у тебя еще и ум был в голове, — ничего бы тебе больше не требовалось, чтоб царствовать». Далее в античном тексте следует замечание о том, что басня уместна против человека неразумного.

Краткая обработка Игнатия Диякона данного сюжета выглядит следующим образом:

«Ел ворон сыр; лиса хитрить пустилася:
«Будь голос у тебя — ты стал великим бы!»
Закаркал глупый, сыр из клюва выронив;
А та: «Есть голос у тебя, да мозгу нет».

Следует обратить внимание на мораль, вложенную в слова Лисицы. Она вновь акцентирует внимание на отсутствии мозга, следовательно, и разума у птицы, которую автор описывает с помощью прилагательного «глупый».

В этих примерах ворон используется для создания контраста, мудрый ворон оказывается в глупом положении, контраст усиливает дидактизм произведения.

В басне «Ворон и Лисица» Лафонтена к пороку глупости добавляется и лесть. Мораль также отражается в финальной речи Лиса:

«Подхватил его лис и молвил:
«Сударь, запомните: всякий льстец
Кормится от тех, кто его слушает, -
Вот урок вам, а урок стоит сыра».
И поклялся смущённый ворон (но поздно!),
Что другого ему урока не понадобится».

На первый план выходит именно лесть, ведь в речи героя отсутствует упоминание недостаточной умственной развитости птицы, вместо этого в цитате подчеркивается урок — «льстец кормится от тех, кто его слушает». При этом сама басня не лишена и привычного для сюжета обличения глупости, однако в данном произведении это выражается менее явно, не в дидактической, а в повествовательной части. Так, басня Лафонтена начинается со строк:

«Maitre Corbeau, sur un arbre perche,
Tenait en son bec un fromage.
Maitre Renard, par l'odeur alleche,
Lui tint à peu pres ce langage...»

Согласно Французскому словарю *Reverso* слово «Maitre» может переводиться как существительное — «учитель, хозяин, властелин, господин», или как прилагательное — «главный, ведущий, мастерский». М. Гаспаров использует в качестве его замены русский эквивалент «дядюшка». Одна эта характеристика для Лиса и Ворона от-

сылает нас к моральной сентенции античных и средневековых предшественников. Важно отметить, что в тексте не только герой-трикстер оказывается в роли «учителя», но и обманутый Ворон при этом сохраняет свою позицию мудрой, ведающей птицы.

В произведениях русских авторов образ Ворона был заменен на образ глупой Вороны-разини, ведь основной целью Сумарокова и Крылова стало освещение лести как чего-то гнусного и вредного. Так, в произведении А. П. Сумарокова мы можем обнаружить откровенную лесть, которая выражается через сравнение с яркими, необычными птицами:

«И попугай ничто перед тобой, душа,
Прекраснее стократ твои павлиньих перья!»

Завершается эта хвала следующим замечанием автора, помещенным в скобки: «Нелестны похвалы приятно нам терпеть».

В итоге дидактический компонент в басне Крылова окончательно перешел из речи Лисицы в речь автора и отразился в отделенных от основного текста строках:

«Уж сколько раз твердили миру,
Что лесть гнусна, вредна; но только всё не впрок,
И в сердце льстец всегда отыщет уголок».

Отсутствие необходимости в создании контраста мудрой птицы и глупого поступка поспособствовало изменению самого образа героя из Ворона в Ворону. Отдельно стоит рассмотреть и то, какую роль эти птицы имели в русской культуре, в первую очередь, в фольклоре. Для этого обратимся к описанию животных в книге Ирины Мудровой «Священные птицы в легендах, преданиях, приметах».

Задержим внимание на замечании, предшествующем основному описанию Ворона в книге: «Не следует путать ворона с вороной, такая путаница происходит только в русском языке из-за практически одинакового названия двух разных видов. В других языках эти птицы имеют четко различающиеся названия — например, в английском ворон — raven, ворона — crow. От серой вороны ворон отличается значительно более крупными размерами, чисто черной окраской, более массивным клювом и клиновидным хвостом» [2]. Можно сделать вывод, что разграничение этих птиц в культуре многих народов отражено довольно ярко, поэтому можно предположить, что замена одной птицы на другую является важнейшим элементом в истории развития данного бродячего сюжета.

«Ворон — мудрый, вещий, зловещий. Вестник войны, смерти и печали. Но в то же время — даритель живой и мёртвой воды, летающий за море, чтобы её достать, и хранитель ключей от небесных врат, провожающий души умерших в Ирий (Рай)» [2]. Яркая связь с потусторонним миром, миром мертвых делает Ворона вещим. Во многих верованиях карканье ворона связано с предсказанием войн и смертей. Нахождение на границе миров объясняет разумность птицы в русской культуре. Идею особой мудрости подкрепляют и мифы о воронах-долгожителях. Ирина Мудрова отмечает: «Миф о вороне-долгожителе родился благодаря верности неразлучной семьи воронов

месту гнездования. Когда, много лет спустя, другая пара сменяет «стариков», люди просто не замечают этого» [2]. В человеческом сознании мудрость крепко связана с возрастом и опытом существа, поэтому мифы о долгожительстве также подкрепляют взаимосвязь воронов и мудрости. Развитие умственных способностей этого вида птиц отмечают и ученые-биологи, Альфред Блэм и вовсе писал: «Ворон часто показывает понятливость, почти равную человеческой, и кто не признаёт у животных ума, пусть подольше понаблюдает ворона».

Отдельно хотелось бы выделить представления о том, что вороны хранят спрятанные в земле клады, это богатство часто связывают с чем-то несправедливым, поэтому за образом птицы закрепился и мотив кражи. Этот же мотив ярко проявляется и в образе Вороны, выражение «Над ним ворона каркает» отражает представления о том, что ворона может раскрыть вора, издавая звуки рядом с ним. В баснях русских поэтов также сохраняется мотив кражи. У Тредиаковского и Сумарокова он выражен прямо: «Однажды ворон своровал с окошка сыр» и «Ворона сыру кус когда-то унесла». У Крылова мотив кражи сыра Вороной не явен, но при этом он подразумевается в строке: «Вороне где-то бог послал кусочек сыру». Именно неопределенное местоимение «где-то», с одной стороны, конструирует свойственный басням абстрактный хронотоп, создающий эффект возможности этого события везде: и в реальном, и в вымышленном мире, способствующий раскрытию универсального и вневременного характера данной ситуации; с другой стороны, используется для создания ироничного объяснения причины появления сыра у Вороны, если бы действительно Бог решил благословить птицу сыром, то он, вероятнее всего, послал бы пищу птице напрямую, а не «где-то». В басне присутствует явный намек на первоначальную кражу, осуществленную Вороной, взявшей у кого-то «куда-то посланный ей» кусочек сыра.

В отличие от мудрого ворона, Ворона во всех сказках, присказках, баснях и приметах — это хвастуны и раззява, не слишком умная и дальновидная. Например, в русской народной сказке «Ворона и рак» описывается история о том, как рак, используя лесть, смог заставить ворону выпустить его из клюва. Эта птица часто и совершенно безосновательно хвастается, а в итоге попадает в смешную ситуацию, выставляя себя в самом невыгодном свете, или остается «у разбитого корыта».

В то же время, стоит отметить, что близость слов «ворон» и «ворона» в русской культуре сближало этих двух птиц. Согласно этимологическому словарю М. Фасмера слово ворона произошло «от праслав. *vorna <...>. Исконнородственно лит. varna «ворона», др.-прусс. warne — то же. Праслав. *vorna, лит. vārna < *vornā (от *vornos «ворон»): ворона обозначается как связанная с вороном» [5]. Отметим, что после появления басни «Ворона и Лисица» в культуре закрепилось осознание вороны, противопоставленное ворону, связанное с хвастовством, глупостью и тщеславием и выраженное в таких выражениях как «проворонить». В словаре В. И. Даля был зафиксирован

бесприставочный глагол «воронить» в значении «зевать, разинув рот», а «проворонить» там поясняется словом «зевать». То есть уже в языке была закреплена связь: ворона = разиня, тот, кто упускает что-то из-за невнимательности. Басня Крылова упрочила эту связь и возвела образ вороны до аллегорического представителя-хранителя невнимательности, глупости и недалёковидности.

Можно сделать первый вывод: образ Ворона был заменен в русской традиции на образ Вороны в связи с изменением самого дидактического посыла басни. Если в предшествующих произведениях основная задача автора была обличить глупость через контраст мудрого, но обманутого Ворона, то у Сумарокова и Крылова цель была изменена на насмешку над льстецами и тщеславными людьми, принимающими лесть за чистую монету, поэтому и был использован образ глупой Вороны, которая усиливала идею, что лесть может поставить всякого в глупое положение, соответственно, те, кто готовы принять ее за правду, на самом деле столь же тщеславны и неразумны, как и птица-разиня.

Противостоит глупой Вороне умная и хитрая Лисица, образ которой также часто встречается в фольклоре (особенно в сказках) и вбирает в себя в русском национальном сознании особые характерные черты: склонность плести интриги и создавать коварные планы, умение играть на мелких слабостях других персонажей, хитрость и изворотливость. Отличный научный анализ образа Лисицы в русских сказках из сборника А. Н. Афанасьева был представлен в статье М. С. Киселевой, А. В. Козачёк «Принципы создания образа лисы в русских сказках о животных, записанных в Сибири и на Дальнем Востоке». Так, в исследовании отмечается связь лисицы с женским началом: «В текстах XIX в. образ лисы создаётся с помощью следующих приёмов: лиса наделяется женской символикой, именно через женский образ в культуру транслируется возможность заменить силу умом, хитростью и изворотливостью; лису можно соотносить с женскими образами, превосходящими <...> мужчин своей премудростью» [1]. Отдельно отмечается и связь рыжего цвета с огненной символикой: «Примечательно, что лисица является единственным диким животным, не боящимся огня».

Образ Лисицы также близок к идее красноречия, используемого для достижения своих целей, лести. Эта же идея отражается в строчке: «И говорит так *сладко*, чуть дыша».... Часто сладкоречие проявляется в использовании диминутивов. Так, в сказках из сборника А. Н. Афанасьева можно заметить обращения «Петенька» («Лиса-исповедница»), «Кочетунюшка» («Кот, петух и лиса»). С помощью этих слов героиня способна расположить к себе других персонажей. Диминутивы часто сопровождают и речь Лисицы в басне Крылова: «**Голубушка**, как хороша! / Ну что за **шейка**, что за **глазки**! / Рассказывать, так, право, сказки! / Какие **перушки**! какой **носок**! / И верно ангельский быть должен **голосок**! / Спой, **светик**, не стыдись! / Что ежели, **сестрица**, / При красоте такой, и петь ты мастерица, / Ведь ты б у нас была царь-птица!». Слова с уменьшительно-ласкательными суффиксами встречаются и в тексте басни

Сумарокова: «Какие **ноженьки**, какой **носок**»..., «Что паче всех ты мер, мой **светик**, хороша!».

Стоит отметить, что способность попадать в близкие отношения к другим представителям леса отчетливо отображается и в наименованиях животного «лисица-сестричка», «кума», отражающих родственные связи лисицы и других животных.

Помимо хитрости и умения наладить отношения с другими героями, яркой чертой Лисицы является жадность, обжорство. Часто в фольклоре с образом этого животного связаны сюжеты с убийством, попыткой получить добычу обманом. Так, Лисица тесно сплетается с мотивами смерти, убийства, начинает олицетворять природный хаос: «восточные славяне воспринимали дикий (животный) мир как что-то чужое, хаотичное, непонятное, а домашние животные для людей становились частью «прирученного» хаоса — мира порядка. Именно в «неприрученном» мире хаоса лиса чувствует себя на своём месте, но в то же время её принадлежность к диким животным не мешает убивать и съедать их, или же участвовать в гибели косвенно» [1].

Образ Лисицы можно соотносить и с типичным для многих культур образом трикстера. К основным характеристикам трикстера можно отнести ловкость и изобретательность в трудных положениях, способность к трансформации ситуации в неожиданную сторону. Трикстер — герой, ломающий систему, и генератор перемен. Он разрушает догмы и предвзятые мнения, подталкивает к экспериментам и поиску новых путей. Лисица также находит самые необычные пути достичь цели (прельщает Ворону, хотя, возможно, легче было бы просто испугать птицу), она же нарушает правила, совершив кражу сыра, продолжая грех, запущенный Вороной.

Связь Лисицы с образом огня, смерти, хаоса дает основания сближать ее образ с демоническим началом: «Демоническая сущность лисы проявляется в постоянных поисках жертвы, во многих сюжетах лисица или напрямую связана с гибелью животных, или косвенно — со смертью человека. Поскольку в традиционной культуре смерть воспринималась как переход в потусторонний мир, здесь мы видим реализацию древнего представления о лисе как о проводнике между миром живых и мёртвых. Упоминания красного цвета шерсти, который является цветом жизни, солнца, плодородия, здоровья, но в то же время — губительного пламени, позволяет нам сделать заключение о сохранности народных представлений о лисе как животном, наделённом огненной символикой» [1]. Так, образ Лисицы приобретает черты отрицательного персонажа, который далее проявляется в обмане Вороны, игре на ее слабостях — тщеславии, приумножении зла через кражу.

Подводя итоги, следует отметить, что образы Вороны и Лисицы на протяжении многих веков были тесно вплетены в культуру русского народа, имели яркое отражение в фольклорных и литературных текстах. Изначальная наполненность смыслами данных образов обусловила выбор И. А. Крылова этих героев в качестве басенных персонажей, аллегорически передающих тщеславие и глуп-

пость (Ворона), хитрость и коварство (Лисица). Подобный выбор автора был во многом обусловлен и моральной тенденцией, он поспособствовал достижению цели яркого обличения лести и тщеславия в басне.

Литература:

1. Киселева, М. С. Принципы создания образа лисы в русских сказках о животных, записанных в Сибири и на Дальнем Востоке / М. С. Киселева, А. В. Козачёк. — Текст: непосредственный // Эпосоведение. — 2024. — № 1. — С. 84–92.
2. Мудрова, И. А. Священные птицы в легендах, преданиях, приметах / И. А. Мудрова. — М.: Центрполиграф, 2023. — 224 с. — Текст: непосредственный.
3. Тимофеев Л. И., Тураев С. В. (сост.) Краткий словарь литературоведческих терминов — 2-е изд., дораб. — М.: Просвещение, 1985. — 312 с.
4. Тихомирова, В. А. Отражение этнокультурных различий восприятия текста в баснях / В. А. Тихомирова. — Текст: непосредственный // Ярославский педагогический вестник. — 2011. — № 4 — Том I (Гуманитарные науки).

Способы перевода прецедентных феноменов в современной англоязычной детской литературе (на материале произведения Нила Геймана «Odd and the Frost Giants»)

Челпанова Анна Сергеевна, студент
Санкт-Петербургский государственный экономический университет

В статье исследуются способы перевода прецедентных феноменов в современной англоязычной детской литературе на материале произведения Нила Геймана «Odd and the Frost Giants». В результате анализа установлено, что наиболее распространенным способом перевода является подбор эквивалента, тогда как транскрибирование, калькирование и грамматическая замена используются значительно реже. Полученные результаты могут быть использованы в исследованиях художественного перевода и при подготовке переводчиков.

Ключевые слова: прецедентные феномены, детская художественная литература, способы перевода, особенности передачи.

В связи с расширением страноведческих реалий, вызванным экономическими, культурными и политическими связями, число культурных лакун непрерывно растет, что приводит к появлению новых прецедентных феноменов. Изучение прецедентных феноменов важно для понимания исходного текста, а также культурных особенностей стран.

Цель научного исследования — определить способы перевода прецедентных феноменов на русский язык на материале современной англоязычной детской литературы.

Прецедентный феномен — текст, значимый для той или иной личности в познавательном и эмоциональном отношении, имеющий сверхличностный характер, обращение к которому возобновляется неоднократно в дискурсе данной языковой личности [9].

Прецедентные феномены выполняют в художественном тексте различные функции: способствуют характеристике персонажей, создают комический эффект, а также служат средством передачи национально-культурного своеобразия произведения. При переводе они могут сохраняться, адаптироваться к культуре языка перевода или утрачиваться, если их адекватная передача оказывается затруднительной. Особую сложность представляет перевод прецедентных феноменов в детской литературе, поскольку переводчику необходимо найти баланс между сохранением культурной специфики оригинала и обеспечением понятности текста для юного читателя. При этом важную роль играют возрастные особенности целевой аудитории и степень ее знакомства с реалиями англоязычной культуры.

Прецедентные феномены вызывают интерес лингвистов, так как они имеют высокую значимость для глубокого осмысления основ межкультурной коммуникации, стоит отметить таких специалистов, как Ю. Н. Караулов [9], Д. В. Гудков [6,7,8], Д. В. Багаева, И. В. Захаренко, В. В. Красных [3,4], И. В. Арнольд [2], В. Н. Комиссаров [10], чьи работы внесли весомый вклад в изучение прецедентных феноменов и вошли в аналитический обзор в данном исследовании.

Классификация прецедентных феноменов, предложенная такими исследователями, как В. В. Красных, Д. В. Гудков, И. В. Захаренко и Д. В. Багаева, легла в основу эмпирической части исследования. Согласно данной классификации, все феномены делятся на прецедентные тексты, прецедентные ситуации, прецедентные высказывания и прецедентные имена [3].

В качестве материала исследования выступает произведение «Odd and the Frost Giants» английского писателя-фантаста Нила Геймана [1] (2009 г., «Одд и Ледяные Великаны», переводчик — Т. Ю. Покидаева [5]). В книге рассказывается

история на основе скандинавской мифологии, где мальчик, живший во времена викингов, встречается на своем пути могущественных богов.

Все примеры прецедентных феноменов, встретившиеся в произведении, являются прецедентными именами. Сферой-источником прецедентных имен в произведении является скандинавская мифология, в книге встречаются такие имена, как *Thor, Odin, Loki, Mjollnir*. Стоит отметить, что прецедентные высказывания и прецедентные тексты используются в детской литературе гораздо реже в сравнении с прецедентными именами. Данная тенденция может быть обусловлена сложностью их органичного включения в текст произведения. Также прецедентные высказывания и тексты могут редко встречаться в детской литературе по причине сложности понимания для детской аудитории в силу возраста, так как формирование кругозора ребенка находится на начальном этапе.

Рассмотрим несколько примеров перевода подробнее.

Таблица 1

Оригинал	Перевод
The fox said, «In the great hall of Odin sat all the Gods, drinking mead, eating and telling stories».	«В большом зале Одина , — продолжил лис, — пировали все боги. Пили хмельной мед, вкушали яства и рассказывали истории».

Один — верховный бог германо-скандинавской мифологии, а зал Одина (Вальхалла) — место пребывания павших воинов. Использование общепринятого русского эквивалента позволяет сохранить культурную узнаваемость прецедентного имени.

Таблица 2

Оригинал	Перевод
« Mjollnir », says the maiden. «The Hammer of Thor».	« Мьёльнир , — молвила дева. — Молот Тора».

В данном примере прецедентное имя передано с помощью транскрибирования. Мьёльнир — молот бога Тора в скандинавской мифологии. Использование этого способа перевода позволяет сохранить оригинальное звучание имени и его культурную узнаваемость.

В ходе анализа было рассмотрено 48 примеров перевода прецедентных феноменов, а также были определены основные способы перевода, используемые специалистами при работе с детской литературой. Самым распространенным способом перевода является подбор эквивалента в русском языке, используется в 81 % всех проанализированных случаев. Частота использования данного приема перевода обусловлена высокой степенью узнаваемости прецедентных имен в культуре перевода. Вторым по частоте использования способом перевода стало транскрибирование (15 %), данный прием помог сохранить фонетическую схожесть с оригиналом. Также в тексте произведения встретились примеры с использованием калькирования (2 %) и грамматической замены (2 %).



Рис. 1. Приемы, используемые для перевода прецедентных феноменов

Таким образом, для успешной передачи прецедентных феноменов переводчику нужно не только разбираться в различных способах перевода, но и обладать знаниями о культурных реалиях обоих языков. Результаты исследования могут быть полезны как для дальнейшего изучения особенностей перевода прецедентных феноменов, так и для практической деятельности переводчиков художественной литературы.

Заключение

В результате проведенного исследования были выявлены особенности перевода прецедентных феноменов в современной англоязычной детской литературе на материале произведения Нила Геймана «Odd and the Frost Giants». Анализ показал, что в исследуемом произведении прецедентные феномены представлены преимущественно прецедентными именами, источником которых выступает скандинавская мифология. Наиболее распространенным способом их передачи на русский язык является подбор эквивалента, что позволяет сохранить культурную узнаваемость и смысловое содержание оригинала. В ряде случаев переводчик также прибегает к транскрибированию, калькированию и грамматической замене, выбор которых определяется особенностями конкретного прецедентного феномена и его функцией в тексте.

Полученные результаты подтверждают, что перевод прецедентных феноменов в детской литературе требует комплексного подхода, сочетающего знание переводческих стратегий, понимание культурных реалий и учет возрастных особенностей целевой аудитории. Практическая значимость исследования заключается в возможности использования его результатов при анализе художественных переводов, в подготовке будущих переводчиков, а также при дальнейшем изучении способов передачи прецедентных феноменов в произведениях современной англоязычной литературы.

Литература:

1. Gaiman N. *Odd and the Frost Giants* / N. Gaiman // HarperCollins, 2009. — 128 p.
2. Арнольд И. В. Проблемы диалогизма, интертекстуальности и герменевтики (в интерпретации художественного текста) // Вестник Русской христианской гуманитарной академии. Филологические науки, 2025. Т. 6 № 2. с. 99–122. [Электронный ресурс]. URL: <https://elibrary.ru/item.asp?id=83045792&ysclid=mrhk4r5s35132593542> (Дата обращения: 01.07.2026)
3. Багаева Д. В., Гудков Д. Б., Захаренко И. В., Красных В. В. Когнитивная база и прецедентные феномены в системе других единиц и в коммуникации // Вестник Московского университета. Сер. 9: Филология, 1997.
4. Багаева Д. В., Гудков Д. Б., Захаренко И. В., Красных В. В. Прецедентное имя и прецедентное высказывание как символы прецедентных феноменов, 1997.
5. Гейман Н. *Одд и ледяные великаны* / Нил Гейман; пер. с англ. — Покидаева Т. — Москва: Издательство АСТ, 2018. — 128 с.
6. Гудков Д. Б. Прецедентное имя в когнитивной базе современного русского (результаты эксперимента) // Язык, сознание, коммуникация: Сб. статей / Ред. В. В. Красных, А. И. Изотов. — М.: «Филология», 1998. — Вып. 4. С. 128.
7. Гудков Д. Б. Прецедентное имя и проблемы прецедентности, М., 1999.
8. Гудков Д. Б. Теория и практика межкультурной коммуникации // Гнозис, 2003.
9. Караулов Ю. Н. *Русский язык и языковая личность*. Изд. 7-е. — М.: Издательство ЛКИ, 2010. С. 216.
10. Комиссаров, В. Н. Теория перевода (лингвистические аспекты): учеб. для студентов ин-тов и фак. иностр. яз.: доп. Гос. ком. СССР по народ. образованию / В. Н. Комиссаров. — Москва: Высш. шк., 1990.

Молодой ученый

Международный научный журнал

№ 28 (631) / 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 22.07.2026. Дата выхода в свет: 29.07.2026.

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.