# Цифровые образовательные среды и цифровые ресурсы в образовании / Digital Educational Environments and Digital Resources in Education

Научная статья Original article
**DOI:** 10.12731/2658-4034-2026-17-1-947 **EDN:** JCOMNQ
**УДК 004.896:372.862:681.5.07**

## Технические вопросы создания методического комплекса для обучения робототехническим дисциплинам

**Д.В. Морохин, А.В. Иванов**
*Поволжский государственный технологический университет, Йошкар-Ола, Российская Федерация*

***Аннотация***

**Обоснование.** Программы обучения в области робототехники требуют современных методик, сочетающих теорию и практику. Использование симуляторов позволяет приблизить обучение к реальным задачам, но существующие образовательные ресурсы имеют ограничения: платный доступ, ориентация на широкую аудиторию, зависимость от конкретного аппаратного обеспечения.

**Цель исследования –** представить опыт разработки, внедрения и апробации методического комплекса для обучения проектированию интеллектуальных робототехнических систем с использованием открытого программного обеспечения и симуляторов для студентов старших курсов технических ВУЗов, обучающихся по направлению подготовки «Информатика и вычислительная техника».

**Материалы и методы.** Основу методологии составили анализ существующих образовательных ресурсов, практико-ориентированное моделирование и педагогический эксперимент. Технологический стек комплекса включает симуляционную среду Coppelia Sim, средства разработки (Visual Studio Code, Python), аппаратную платформу на базе микроконтроллера ESP32, а также локальную языковую модель (Gemma 3) для поддержки учебного процесса. Апробация проводилась в рамках дисциплины «Автономные транспортные системы» с участием студентов третьего курса. Общее количество студентов – 40. Для оценки эффективности применялись методы наблюдения, анкетирования и анализа результатов выполнения лабораторных работ.

**Результаты.** Разработан комплексный методический инструментарий, включающий структурированный курс из девяти лабораторных работ, методические

указания, аппаратный прототип колесной платформы и библиотеку для переноса кода между симулятором и реальным устройством. Анкетирование показало рост мотивации и уверенности в работе с инструментами. На основе анализа трудностей студентов (сложность первичного освоения интерфейсов, разный уровень подготовки) сформулированы рекомендации по улучшению курса: внедрение уровневой дифференциации заданий, оптимизация инструкций за счет добавления визуальных материалов, создание системы взаимопомощи. Также внедрение технологий симуляторов в курс выявила повышение мотивации студентов и развитие практических навыков, а также позволила сформулировать рекомендации по улучшению курса, такие как уровневая дифференциация заданий и оптимизация инструкций.

**Ключевые слова:** робототехника; симулятор; открытое программное обеспечение; Coppelia Sim; методика обучения; микроконтроллер ESP32; программно-аппаратный комплекс

**Для цитирования.** Морохин, Д. В., & Иванов, А. В. (2026). Технические вопросы создания методического комплекса для обучения робототехническим дисциплинам. *Russian Journal of Education and Psychology*, *17*(1-2), 27–49. https://doi.org/10.12731/2658-4034-2026-17-1-947

# Technical issues in developing a curriculum for teaching robotics

**D.V. Morokhin, A.V. Ivanov**
*Volga State University of Technology, Yoshkar-Ola, Russian Federation*

***Abstract***

**Background.** Robotics training programs require modern methods that combine theory and practice. The use of simulators makes it possible to bring training closer to real tasks, but existing educational resources have limitations: paid access, focus on a wide audience, or dependence on specific hardware.

**Purpose** of the article is to present the experience of developing, implementing and testing a methodological package for teaching the design of intelligent robotic systems using open source software and simulators for senior students of technical universities studying in the field of Computer Science and Computer Engineering.

**Materials and methods**. Methodology is based on existing educational resources analysis, practice-oriented modeling and pedagogical experiment. The technological stack of the complex includes the Coppelia Sim simulation environment, development tools (Visual Studio Code, Python), a hardware platform based on the ESP32 microcontroller, as well as a local language model (Gemma 3) to support the learning process. The testing was carried out within the framework of the discipline “Autonomous transport systems” with the participation of third-year students. The total number of students

is 40. To assess the effectiveness, methods of observation, questioning and analysis of laboratory work results were used.

**Results**. A comprehensive methodological toolkit has been developed, including a structured course of nine laboratory papers, guidelines, a hardware prototype of a wheeled platform, and a library for transferring code between a simulator and a real device. The survey showed an increase in motivation and confidence in working with tools. Based on the analysis of students' difficulties (the complexity of the initial development of interfaces, different levels of training), recommendations for improving the course are formulated: the introduction of level differentiation of tasks, the optimization of instructions by adding visual materials, the creation of a system of mutual assistance. Also, the introduction of simulation technologies into the course revealed an increase in students' motivation and the development of practical skills, and also allowed us to formulate recommendations for improving the course, such as level differentiation of tasks and optimization of instructions.

**Keywords**: robotics; simulator; open source software; Coppelia Sim; teaching methodology; ESP32 microcontroller; software and hardware complex

**For citation.** Morokhin, D. V., & Ivanov, A. V. (2026). Technical issues in developing a curriculum for teaching robotics. *Russian Journal of Education and Psychology*, *17*(1-2), 27–49. https://doi.org/10.12731/2658-4034-2026-17-1-947

## Введение

Программы обучения в области робототехники и автоматизированных систем в наши дни являются важной частью инновационной сферы, требующей всестороннего подхода к разработке и использованию современных методик обучения [11; 12; 15; 20; 21]. Современные стандарты инженерного образования требуют от студентов не только освоения теории, но и умения применять её на практике: программировать микроконтроллеры, разрабатывать аппаратные решения и тестировать их в виртуальных средах. Формирование таких компетенций напрямую связано с качеством подготовки будущих специалистов для сферы автоматизации и искусственного интеллекта. Эффективный учебный курс должен сочетать лабораторный практикум, теоретическую подготовку и работу с симуляторами – это позволяет приблизить обучение к реальным профессиональным задачам и отвечает запросам современного технологического рынка.

В настоящее время на образовательном рынке представлен значительный выбор учебных курсов и материалов по робототехнике [1; 4;

9; 11; 17]. Однако анализ этих ресурсов позволил выявить ряд характерных недостатков, к основным из которых относятся: ограничение доступа для пользователей из России на зарубежных платформах, платная основа многих курсов и их ориентация на широкую аудиторию, а не на специализированную подготовку в рамках конкретных дисциплин.

Учитывая вышеизложенное, *целью исследования* является создание b аппробация методического комплекса, направленного на повышение эффективности обучения студентов проектированию и применению интеллектуальных робототехнических систем с использованием открытого программного обеспечения и симуляторов.

**Материалы и методы**

Методологической основой исследования стал комплексный подход, включающий анализ, моделирование и апробацию. Был проведен сравнительный анализ существующих отечественных и зарубежных образовательных курсов и платформ в области робототехники, таких как Coursera Robotics Specialization, курсы на основе ROS2 и др. В качестве основного инструмента для симуляции и моделирования робототехнических систем выбрана среда Coppelia Sim благодаря её открытости, функциональности и поддержке различных языков программирования (Python, Lua). Для разработки программного обеспечения использовалась среда Visual Studio Code с поддержкой удаленной разработки. Аппаратная часть исследования включала проектирование и сборку физического прототипа колесной платформы на базе микроконтроллера ESP32. Особенностью методики является интеграция локальной языковой модели Gemma 3 (через платформу Ollama) в учебное окружение для контекстной поддержки студентов при решении рутинных технических задач, что способствует снижению когнитивной нагрузки и развитию навыков самостоятельной работы.

**Результаты и обсуждение**

*Анализ проблемной области*

Для определения требований к разрабатываемому методическому комплексу был проведён сравнительный анализ современных образо-

вательных ресурсов и подходов к обучению робототехнике. Критерии сравнения были выбраны исходя из ключевых проблем: 1) доступность (финансовая и территориальная), 2) техническая и методическая ориентация (глубина погружения, широта охвата, структурированность), 3) независимость от конкретного аппаратного обеспечения (использование открытого ПО и симуляторов), 4) практическая направленность и безопасность (наличие сквозных проектов, связь с реальными устройствами, минимизация рисков). Результаты анализа систематизированы в Таблице 1.

Методология анализа, критерии сравнения и оценка ресурсов базируются на изучении открытых материалов платформ, а также на исследованиях, посвящённых методике преподавания робототехники [13; 16], анализу симуляторов [1; 9; 17] и применению микроконтроллерных платформ в образовании [3; 5; 11].

*Таблица 1.*

**Сравнительный анализ подходов и ресурсов в обучении робототехнике по ключевым методическим линиям**

**Table 1.** A Comparative Analysis of Approaches and Resources in Robotics Education by Key Methodological Approaches

| Ресурс / Критерий | 1. Доступность | 2. Методическая ориентация | 3. Зависимость от аппаратного обеспечения | 4. Практическая направленность и безопасность | Оценка применимости для целей работы |
|---|---|---|---|---|---|
| **Coursera. Robotics Specialization** | Платформенный, платный доступ; географические ограничения. | Широкий обзорный курс. Финальный проект позволяет углубиться в детали реализации. | Сильная зависимость от MATLAB (проприетарное ПО). Частично используется симулятор. | Есть трек с физическим роботом, но требует специфичного и часто дорогого оборудования. | **Ограниченная**. Не соответствует критериям 1 и 3 (доступность, открытое ПО). |
| **«Летающая робототехника» на основе «Клевер»** | Открытый доступ к материалам на русском языке. | Специализированная документация по платформе. Отсутствие структурированного курса. | Жёсткая привязка к конкретному аппаратному комплексу. | Прямая работа с реальным БПЛА, но в рамках строго заданной конфигурации. Риск поломки. | **Низкая**. Не соответствует критериям 2 и 3 (методическая глубина, независимость от «железа»). |
| **ROS2 – Народный курс** | Открытый доступ, активное русскоязычное сообщество. | Фокусируется на освоении middleware (ROS 2). Недостаточно материала по моделированию и алгоритмам управления. | Абстрагирован от «железа» через ROS, но требует погружения в экосистему ROS для начала работы. | Предполагает работу с реальными роботами, но начальный порог входа высок. Риск аппаратных ошибок на ранних этапах. | **Средняя**. Соответствует критерию 1, частично — 3 и 4. Не даёт готовой методики для начального моделирования. |

| | | | | | |
|---|---|---|---|---|---|
| **Методические пособия** | Доступны в электронных библиотеках. | Теоретическая база и отдельные примеры. Отсутствие целостного практикума, интегрированного с симулятором. | Часто привязаны к конкретным стендам и ПО учебного заведения. | Задания требуют уникальной аппаратной базы, что затрудняет воспроизведение. Риск несовместимости. | **Недостаточная**. Не соответствуют критериям 2 (целостный практикум), 3 и 4 (аппаратная независимость и воспроизводимость). |
| **Учебные стенды на основе микроконтроллеров** | Высокая (низкая стоимость базовых компонентов, распространённость). | Глубокое погружение в низкоуровневое программирование, электронику и «железо». Высокая практическая ценность. Отсутствует системная методика. | Курс определяется конкретной выбранной платформой и набором сенсоров. | Прямая работа с физическими компонентами. Высокий риск поломки, сложность отладки, отсутствие визуализации процессов для новичков. | **Частичная**, как конечный этап. Полностью соответствует критерию 4, но не соответствует критериям 3 и частично 2 (отсутствие виртуального, безопасного этапа для отработки алгоритмов). Является идеальной целью для переноса готовых решений, но не для начального обучения. |

Сравнительный анализ электронных курсов (Coursera Robotics Specialization, «Летающая робототехника», ROS2 - Народный курс) выполнен на основе открытых методических материалов, представленных на соответствующих образовательных платформах, а также с учетом обзоров и анализа, приведенных в работах [1; 9; 13; 17].

Также в сравнении использовались источники:

- По общим проблемам и методике [1; 13; 16];
- По симуляторам и Coppelia Sim/V-REP [1; 9; 17];
- По микроконтроллерам и аппаратной части [3; 5; 11];
- По Gazebo/ROS [17; 18; 19];
- По общим вопросам робототехники [2; 4; 20].

Проведённый анализ по выделенным методическим линиям показал, что ни один из рассмотренных подходов или ресурсов в полной мере не удовлетворяет комплексу требований для реализации поставленной цели. Электронные курсы часто ограничены в доступе или имеют общий характер, пособия привязаны к конкретной базе, а работа напрямую с микроконтроллерными стендами, при всей своей практической ценности, сопряжена с высокими начальными барьерами, рисками и отсутствием безопасной среды для экспериментов. Это подтвердило

необходимость разработки методического комплекса, который бы интегрировал преимущества разных подходов: симуляционное моделирование (для безопасного освоения алгоритмов), структурированную педагогическую последовательность на основе открытого ПО, финальный перенос на реальную микроконтроллерную платформу (как этап, следующий за виртуальным прототипированием).

*Выбор инструментов для реализации обучающей системы*

Формирование технологического стека является ключевым этапом при построении учебной среды для интеллектуальной робототехники, так как непосредственно влияет на эффективность, стабильность и производительность разрабатываемых решений. Ниже представлен обоснованный выбор инструментов, положенных в основу методического комплекса.

1. Операционная система.

Базовой операционной системой был выбран дистрибутив Linux (Arch Linux). Его главными преимуществами являются открытая архитектура, высокая степень кастомизации и актуальность пакетов в репозиториях, что критически важно для исследовательских и учебных задач, требующих тонкой настройки под специфические нужды робототехнических проектов. В качестве альтернативы, обладающей большей «коробочной» готовностью и простотой освоения, можно рассматривать Microsoft Windows, однако её проприетарная природа ограничивает возможности глубокой адаптации.

2. Средства разработки.

В качестве основной интегрированной среды разработки (IDE) используется Visual Studio Code. Его ключевое преимущество для учебного процесса – хорошо проработанная поддержка удалённой разработки через SSH, позволяющая студентам работать в единой, предварительно настроенной среде (например, в Linux-контейнере), что устраняет проблемы с совместимостью и обеспечивает идентичность условий разработки и выполнения кода. Альтернативой для опытных пользователей может служить редактор Neovim, предлагающий максимальную гибкость настройки и высокую производительность.

3. Языки программирования.

Основным языком для реализации алгоритмов управления и взаимодействия с симуляторами выбран Python. Он обладает низким

порогом входа, богатой экосистемой библиотек для научных вычислений и машинного обучения, а также хорошо интегрируется с ROS. Для задач, требующих максимальной производительности и работы в реальном времени (например, низкоуровневое управление аппаратной платформой), целесообразно применение C/C++ или, в более современном варианте - RUST, часто используемых в связке с Python в рамках ROS-систем.

4. Библиотеки для работы с изображением в ROS.

Для работы с компьютерным зрением и обработки изображений с камер роботов используется библиотека OpenCV. Она является стандартом де-факто в робототехнике благодаря широкому набору алгоритмов, высокой производительности и отличной интеграции с ROS. В качестве инструментов для специфических задач обработки видео или сложной научной визуализации могут применяться FFmpeg и VTK (Visualization Toolkit) соответственно.

5. Системы для тестирования и симуляции.

В качестве среды для симуляции используется Coppelia Sim (ранее известный как V-REP) – среда для моделирования роботов и динамических систем, разработанная компанией Coppelia Robotics. Она позволяет создавать виртуальные модели роботов, сенсоров, приводов и окружающей среды. Coppelia Sim активно используется как в образовательных целях, так и в исследовательских и промышленных проектах.

Среди аналогов можно рассмотреть Gazebo – симулятор, используемый в ROS и позволяющий тестировать и отлаживать роботов в виртуальной среде, что ускоряет процесс разработки и снижает риски при работе с реальными роботами, а также Rviz, предоставляющий удобный графический интерфейс для отображения и анализа данных, получаемых от сенсоров. Webots – инструмент для моделирования робототехнических систем, разработанный компанией Cyberbotics. Он предоставляет 3D-среду для проектирования, тестирования и отладки, а также поддерживает различные языки программирования (C++, Python, Java).

6. Микроконтроллер.

В качестве аппаратной платформы, на которой будет проводиться тестирование программного кода, разработанного в симуляторе, был

выбран ESP32 – микроконтроллер от компании Espressif, разработанный для использования в различных проектах. Он сочетает в себе поддержку Wi-Fi и Bluetooth, низкое энергопотребление, большие возможности по работе с интерфейсами (UART, I2C, SPI) и возможность работы с аналоговыми и цифровыми сигналами.

Альтернативные варианты:

Arduino – платформа с открытым исходным кодом, состоящая из плат микроконтроллеров (в основном, серии ATmega от Microchip) и удобной интегрированной среды разработки (IDE).

Raspberry Pi – универсальная и очень компактная платформа для выполнения вычислительных задач и взаимодействия с внешними модулями и периферией.

Orange Pi – альтернатива Raspberry Pi, часто предлагающая больше вычислительных ресурсов по более выгодной цене.

*Интеграция локальной языковой модели для поддержки учебного процесса*

Для усиления самостоятельной работы студентов и снижения когнитивной нагрузки при освоении сложных инструментов было принято решение интегрировать в учебное окружение (Linux с набором симуляторов) локальную языковую модель (LLM). Эта модель призвана выступать в роли контекстного ассистента, способного оперативно помогать в решении рутинных проблем, возникающих при изучении новых сред разработки и симуляторов, таких как поиск нужной функции, разбор синтаксических ошибок или понимание логики работы API.

Ключевым преимуществом локального развертывания является его автономность и безопасность: модель работает без доступа к интернету, что исключает риски утечки данных и обеспечивает стабильную доступность. При этом, чтобы сохранить образовательную ценность, модель будет работать с предустановленным системным промптом (инструкцией). Этот промпт ограничит её способность выдавать готовые решения ключевых учебных задач, мотивируя студентов самостоятельно декомпозировать проблемы, анализировать условия и глубже вникать в предметную область. Таким образом, модель становится не «решателем», а навигатором и консультантом, который помогает пре-

одолеть первоначальные технические барьеры и сосредоточиться на сути задания.

В качестве конкретной реализации была выбрана модель Gemma 3 от Google в конфигурации на 4 миллиарда параметров (4B). Данный выбор обусловлен оптимальным балансом между вычислительной эффективностью и качеством ответов. Gemma 3 с 4B параметрами представляет собой одну из самых компактных и в то же время достаточно мощных языковых моделей, способную функционировать на компьютерах с ограниченными ресурсами, включая одноплатные компьютеры (например, Raspberry Pi 4/5 с достаточным объемом ОЗУ) или смартфоны. Для её стабильной работы рекомендуется наличие не менее 4 ГБ свободной оперативной памяти. Для удобства развертывания, управления и взаимодействия с моделью используется платформа Ollama, которая предоставляет простой CLI и REST API интерфейс, совместимый с широким спектром клиентских приложений и плагинов для сред разработки, что обеспечивает беспрепятственную интеграцию в существующий рабочий процесс студентов.

*Требования к курсу*

Структура лабораторной работы

Общие требования к лабораторным работам

Цель: Развитие практических компетенций в области проектирования, программной реализации и верификации робототехнических систем с использованием симуляционной среды.

Инструментарий: Coppelia Sim, языки программирования (Python/Lua), API Coppelia Sim.

Формат выполнения: Индивидуальные задания с возможностью групповой работы.

Критерии оценки: Качество разработанной модели, точность реализуемых алгоритмов, корректность отчета, а также понимание теоретических аспектов.

*Перечень лабораторных работ*

1. Создание четырёхколёсной платформы.

Основы моделирования роботов, физические параметры (масса, трение), интерфейсы Coppelia Sim. Задание: Создать модель колёсной платформы, настроить динамику и визуализацию.

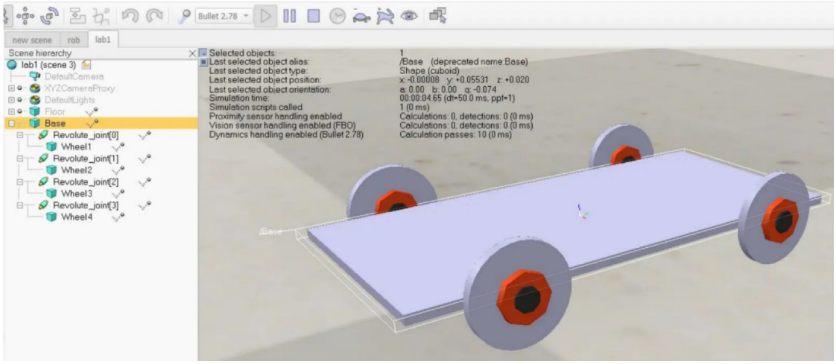

**Рис. 1.** Лабораторная работа №1. Построение колесной платформы
**Fig. 1.** Lab Assignment No. 1: Building a Wheeled Platform

В данной работе модифицируется конструкция колёсной платформы. Вместо двух обычных колёс устанавливается поворотное колесо, а шарниры настраиваются в режим двигателя (электромотора). Создается скрипт управления электромоторами.

2. Основы программирования движения колесной платформы.

Кинематика колёсных роботов, методы управления (прямая/обратная кинематика). Задание: Реализовать алгоритм перемещения по заданным координатам.

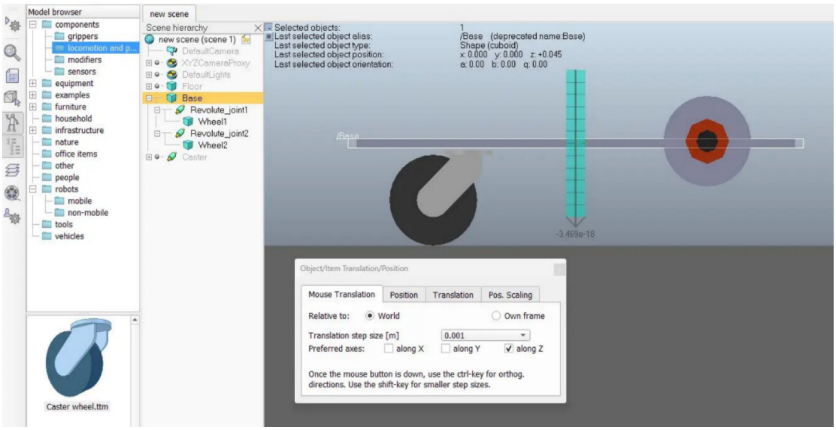

**Рис. 2.** Лабораторная работа №2. Модификация конструкции тележки
**Fig. 2.** Lab Assignment No. 2. Modification of the Cart Design

В рамках данной лабораторной работы выполняется модернизация базовой конструкции. На шасси заменяется пара стандартных колёс на одно поворотное (рис. 2). Параллельно производится перенастройка шарниров: вместо пассивного режима подшипника они переводятся в активный режим «двигателя». Это позволяет программно задавать и контролировать угловую скорость вращения каждого колеса.

Финальным этапом является разработка управляющего скрипта и подробный анализ программного кода, реализующего взаимодействие с виртуальными моторами через API симулятора Coppelia Sim.

3. Взаимодействие с объектами при помощи датчиков приближения.

Изучение типов датчиков и принципов обработки сигналов. Задание: Создать модель, оснащенную датчиками, и разработать алгоритм объезда препятствий колёсной платформы в пространстве.

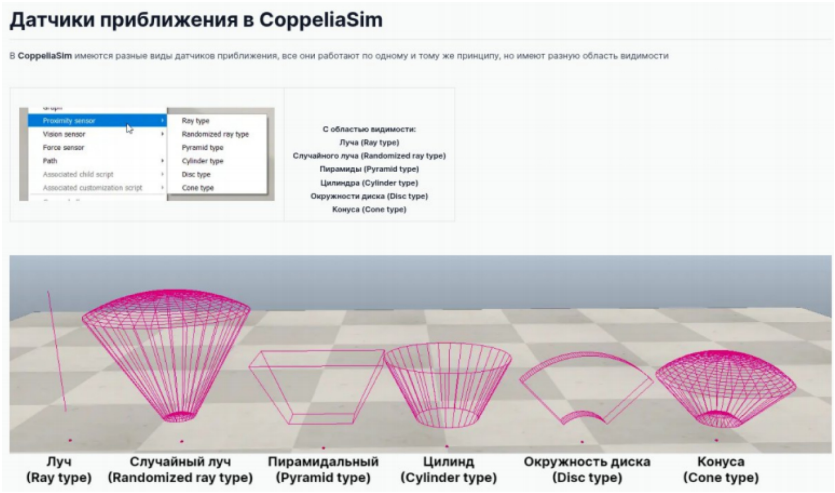


**Рис. 3.** Лабораторная работа №3. Взаимодействие с объектами.
**Fig. 3.** Lab Assignment No. 3. Interaction with objects

Приведен пример обработчика данных робота для движения по линии на основе объединения нескольких датчиков.

4. Движение в пространстве с помощью датчиков зрения.

В рамках данной лабораторной работы рассматриваются принципы компьютерного зрения, включая обработку изображений (с использова-

нием библиотек, подобных OpenCV) и методы отслеживания объектов (трекинг). Практическим заданием является разработка алгоритма для распознавания визуальных маркеров или заданных цветовых областей в видеопотоке, получаемом с камеры, установленной на модели робота.

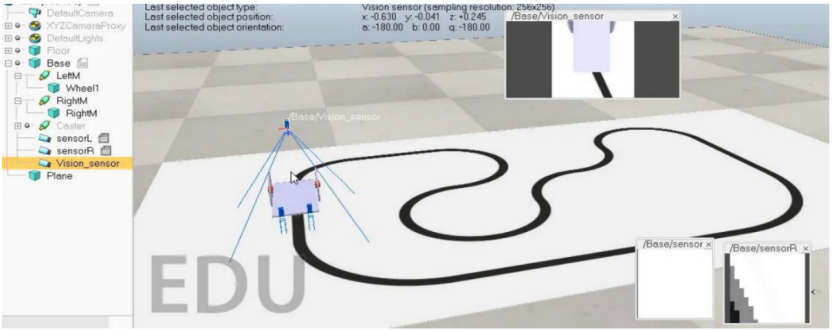


**Рис. 4.** Лабораторная работа №4. Реализация робота, перемещающегося по линии
**Fig. 4.** Lab Assignment No. 4. Implementation of a robot that follows a line

Coppelia Sim предоставляет два принципиально разных типа виртуальных сенсоров: перспективные и ортогональные камеры.

- Перспективная камера моделирует работу реального объектива, имеющего определённый угол обзора, что создаёт естественные искажения перспективы.

- Ортогональная камера является идеализированным инструментом, у которого угол обзора равен нулю. Она проецирует чётко определённую область пространства на матрицу пикселей без искажений и независимо от расстояния до объектов, что в реальных условиях практически неосуществимо.

Это уникальное свойство ортогональных камер в симуляторе было использовано для эмуляции работы другого типа сенсора – датчика линии. Такой виртуальный датчик позволяет роботу определять своё положение относительно контрастной разметки. В лабораторной работе представлен и подробно разобран практический пример программного кода, который реализует считывание данных с ортогональных камер, интерпретируя их как сигналы от датчиков линии, и использует эту информацию для управления движением робота (рис. 4).

5. Управление колёсной платформой с использованием ПИД-регулятора.

Принцип работы ПИД-контроллеров, настройка параметров (Kp, Ki, Kd) регулятора для движения по линии на основе объединения нескольких датчиков.

```python
class PID:
    def __init__(self, sim, setpoint, kp, ki, kd):
        self.setpoint = setpoint
        self.kp = kp
        self.ki = ki
        self.kd = kd
        self.dt = sim.getSimulationTimeStep()

    def pid(self, input):
        err = p = i = d = prevErr = float()
        err = self.setpoint - input
        p = err
        i = i + err * self.dt
        d = (err - prevErr) / self.dt
```

**Рис. 5.** Лабораторная работа №5. Класс ПИД регулятора в скрипте Python
**Fig. 5.** Lab Assignment No. 5. PID controller class in a Python script

Приведён пример использования ПИД регулятора для движения по линии на основе объединения нескольких датчиков.

6. Основы управления БПЛА.

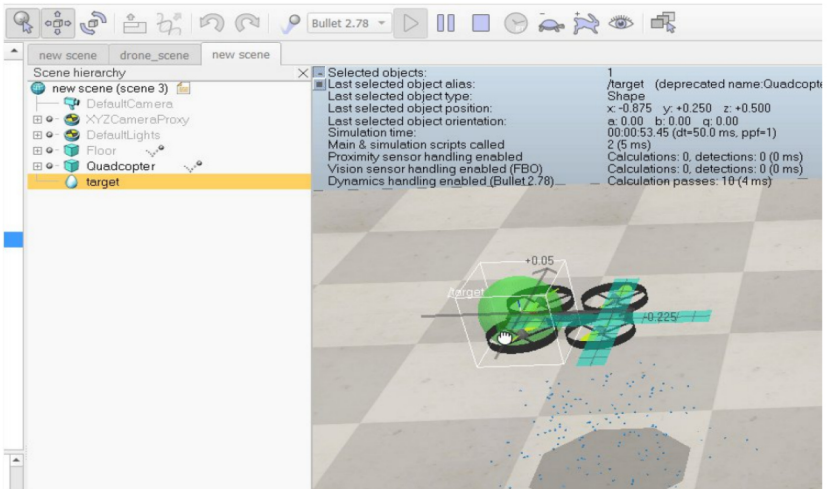


**Рис. 6.** Лабораторная работа №7. Перемещение БПЛА
**Fig. 6.** Lab Assignment No. 7. UAV Movement

Работа посвящена изучению базовых принципов динамики полёта, включая генерацию подъёмной силы, балансировку моментов и управление аэродинамическими параметрами. Практической целью является создание виртуальной модели квадрокоптера и отработка навыков её стабилизации и маневрирования в симуляционной среде Coppelia Sim (рис. 6).

7. Основы программирования БПЛА.

В рамках этой лабораторной работы студенты осваивают алгоритмы автономной навигации и планирования траекторий. Ключевой задачей становится разработка и реализация программного обеспечения, обеспечивающего самостоятельное перемещение БПЛА по заданному маршруту в виртуальном пространстве. Этап включает построение полётного задания с указанием ключевых точек (рис. 7).

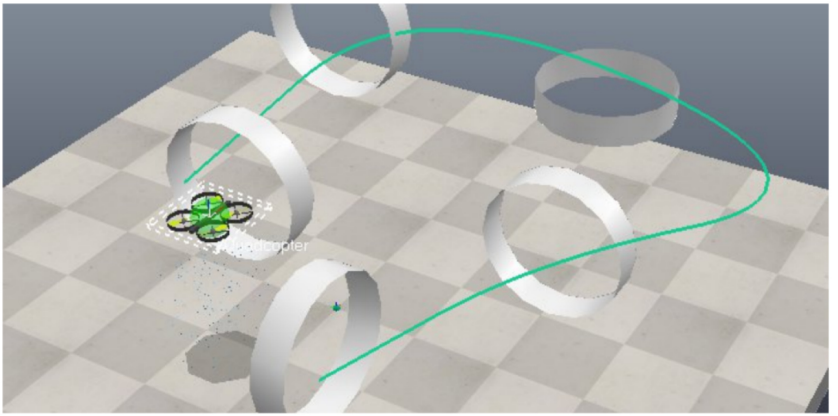

**Рис. 7.** Лабораторная работа №7. Построение полетного задания
**Fig. 7.** Lab Assignment No. 7. Creating a flight plan

8. Основы компьютерного зрения.

Лабораторная работа знакомит с фундаментальными методами обработки изображений: фильтрацией, сегментацией и выделением признаков. Практическая часть включает настройку виртуальной камеры в Coppelia Sim и разработку алгоритма для автоматического распознавания объектов на основе их цветовых характеристик или геометрической формы.

Навигация БПЛА с использованием датчика расстояния.

Методы навигации в 3D, обработка данных, полученных с лидара или дальномера. Задание: Реализация автономного полета с использованием датчиков для обхода препятствий.

9. Проектирование физической колёсной платформы

Основная цель лабораторной работы заключается в переносе навыков, полученных студентами на базе виртуального симулятора, на реальное устройство. Это позволяет обучающимся оценить влияние свойств физических элементов на работоспособность собранной из них системы.

Заданием в данной работе является реализация одного из изученных ранее алгоритмов на физическом уровне.

**Заключение**

Авторами разработан и апробирован полный методический комплекс для изучения основ робототехники. Его (ядро) составляет структурированный курс из девяти лабораторных работ для среды Coppelia Sim, охватывающих темы от создания базовой колесной платформы до навигации БПЛА с использованием компьютерного зрения. Для каждой работы разработаны подробные методические указания, примеры кода и критерии оценки.

Успешная апробация виртуального прототипа колёсной платформы в симуляторе послужила основой для проектирования и сборки её физического аналога. С учётом практического опыта эксплуатации подобных учебных стендов, а также для повышения эффективности обучения, были сформулированы ключевые требования к аппаратной реализации:

- Безопасная система питания.
- Наличие встроенных беспроводных интерфейсов.
- Наличие обратной связи для достижения качественного управления.
- Модульный подход.
- Адаптивная среда разработки с готовыми библиотеками.

На основе данных требований была спроектирована и разработана колёсная платформа на микроконтроллере ESP32 с питанием от аккуму-

ляторной батареи с контроллером заряда и разряда. Было разработано базовое программное обеспечение для взаимодействия с периферией и настроена интегрированная среда разработки на базе дистрибутива Linux в виртуальной машине, за счёт этого система является стабильной и максимально зеркальной для использования на разных компьютерах.

Разработана принципиальная электрическая схема и программное обеспечение. ПО базируется на платформе Arduino Uno с использованием официального SDK от разработчиков микроконтроллера Espressif. Также была разработана общая библиотека ESP32Car для управления колёсной платформой в которую входит настроенный ПИД регулятор для решения проблем со стабильностью движения робота на низкой скорости, данная библиотека решает проблему налипания моторов и имеет задокументированное API подключения периферии колёсной платформы.

Принципы управления платформой соответствуют изученным в процессе работы на симуляторе. Поэтому, использование в процессе обучения данной платформы позволяет студентам осуществить плавный переход от виртуальных подходов к реализации робототехнических систем к реальному устройству. Также для помощи студентам разработана библиотека для управления элементами платформы.

Комплекс был апробирован в осеннем семестре 2024/2025 учебного года в рамках дисциплины «Автономные транспортные системы» на 2-м курсе направления подготовки «Информатика и вычислительная техника». В эксперименте участвовали 32 студента. Апробация включала последовательное выполнение 9 лабораторных работ в течение семестра.

Для оценки эффективности применялись:

1. Метод наблюдения за работой студентов и фиксация типичных трудностей.

2. Анкетирование (проведено по итогам курса «Автономные транспортные системы) для оценки изменения мотивации, интереса и субъективной оценки сложности.

3. Сравнительный анализ результатов выполнения итоговых практических заданий с результатами контрольной группы предыдущего года (36 студентов), обучавшейся по традиционной методике без углубленного использования симулятора и единого методического комплекса.

Апробация курса показала ряд положительных результатов:

1. Повышение успеваемости. Средний балл за выполнение итогового комплексного задания в экспериментальной группе составил 4.3 (по 5-балльной шкале), что на 18% выше, чем в контрольной группе предыдущего года (средний балл 3.6). Качество выполненных работ (доля работ с оценкой «хорошо» и «отлично») возросло с 65% до 87%. Отметим, что для полноты результатов требуется многолетняя статистика здесь и далее.

2. Динамика мотивации. Итоговое анкетирование показало, что 85% студентов отметили повышение интереса к дисциплине благодаря наглядности симуляций и возможности экспериментировать без риска повреждения оборудования. 78% респондентов указали на рост уверенности в своих силах при работе с реальными микроконтроллерами после этапа симуляции.

3. Развитие навыков. Наблюдение выявило активное развитие навыков алгоритмического мышления и программирования. Студенты научились декомпозировать задачи, переносить логику из виртуальной среды в аппаратную и проводить отладку на разных уровнях.

4. Работа с ИИ-ассистентом. Интегрированная локальная модель Gemma 3 эффективно использовалась для решения рутинных вопросов (синтаксис, поиск функций API), что позволило сократить время на преодоление технических барьеров и сосредоточиться на сути заданий. Важным условием была настройка системного промпта, ограничивающего выдачу готовых решений ключевых алгоритмов.

Были выявлены и трудности: сложности с первичным освоением интерфейса Coppelia Sim (отмечали 40% студентов), значительный разброс в уровне начальной подготовки по программированию.

На основе анализа процесса апробации и обратной связи от студентов сформулированы рекомендации по улучшению курса, описанного в данной статье:

1. Внедрение уровневой дифференциации заданий внутри каждой лабораторной работы (базовый, стандартный, продвинутый уровень) для адаптации к разному уровню подготовки.

2. Оптимизация методических материалов за счет добавления коротких видео-туториалов по ключевым действиям в интерфейсе и скриншотов с пояснениями.

3. Создание системы взаимопомощи, например, в формате чата или wiki-базы знаний, где студенты могли бы делиться найденными решениями типовых проблем.

4. Формирование централизованной базы готовых моделей и шаблонов кода для самых распространенных задач, чтобы ускорить начальный этап.

Эти рекомендации носят практический характер и направлены на снижение выявленных в ходе эксперимента барьеров в обучении. Планируется их внедрение в следующем учебном году.

1. Проведен анализ существующих отечественных и зарубежных образовательных курсов и ресурсов в области робототехники, что позволило выявить их ограничения и сформулировать требования к новому методическому комплексу.
2. Исследованы и применены типовые архитектурные и программно-аппаратные решения для систем управления интеллектуальными робототехническими комплексами, в частности, связка Coppelia Sim + Python + ESP32.
3. Разработан комплексный методический инструментарий, включающий структурированный курс лабораторных работ, методические пособия, аппаратный прототип и библиотеку для кроссплатформенного переноса кода.
4. Предложен и реализован подход по интеграции локальной языковой модели в учебный процесс для технической поддержки студентов, способствующий развитию самостоятельности.
5. Апробация разработанного комплекса подтвердила его эффективность в повышении мотивации и формировании практических компетенций, а также позволила выявить направления для дальнейшего совершенствования методики.

***Список литературы***

1. Бжихатлов, И. А. (2018). *Моделирование робототехнических систем в программе V-REP: учебно-методическое пособие*. Санкт-Петербург: НИУ ИТМО. https://e.lanbook.com/book/136492
2. Бурьков, Д. В., & Волощенко, Ю. П. (2020). *Математическое и имитационное моделирование электротехнических и робототехнических систем: учебное по-*

*собие*. Ростов-на-Дону; Таганрог: Издательство Южного федерального университета. ISBN: 978-5-9275-3625-2. https://znanium.com/catalog/product/1308357. EDN: https://elibrary.ru/EGHJRJ

3. Вельтищев, В. В., & Минеев, А. Б. (2018). *Методы имитационного моделирования систем управления движением подводного аппарата: учебное пособие*. Москва: МГТУ им. Н. Э. Баумана. https://e.lanbook.com/book/205493
4. Ревич, Ю. В. (2023). *ESP32-C3. Беспроводное приключение. Полное руководство по IoT*. Москва: ДМК Пресс. 442 с.
5. Закураев, Р. А. (2023). Робототехнические конструкторы для выполнения исследовательских проектов по физике. *Вестник науки*, (6), 112–120. EDN: https://elibrary.ru/YBRJSI
6. Иванов, А. В. (2024). Библиотека построения маршрута БПЛА. *Труды Поволжского государственного технологического университета. Серия: Технологическая*, (12), 30–34. EDN: https://elibrary.ru/CWIOUD
7. Иванов, А. В. (2024). Применение среды моделирования CoppeliaSim для построения маршрута беспилотного летающего аппарата. *Инженерные кадры – будущее инновационной экономики России*, (1), 499–502. EDN: https://elibrary.ru/DKLSOF
8. Ковалев, И. М., & Петрова, Е. Д. (2021). Сравнительный анализ симуляторов Gazebo и CoppeliaSim для образовательных задач. *Информатика и образование*, (5), 32–40.
9. Лаврентьев, Е. Б., Изюмов, А. И., Марченко, Э. В., & Попов, С. И. (2023). *Основы моделирования роботов в среде CoppeliaSim: учебное пособие*. Ростов-на-Дону: ДГТУ. ISBN: 978-5-7890-2097-5. EDN: https://elibrary.ru/NFEWUS
10. Локтионов, А. С., & Евдокимова, С. А. (2024). Выбор программы-симулятора для проектирования робототехнических систем. В *Материалы Всероссийской научно-практической конференции студентов и молодых учёных* (с. 212–219). Воронеж: ВГЛТУ. https://doi.org/10.58168/ROBOTICS2024_212-219. EDN: https://elibrary.ru/ADXPIV
11. Морохин, Д. В., Мясников, В. И., & Иванов, А. В. (2023). *Микропроцессорные системы на основе микроконтроллеров STM32: лабораторный практикум*. Йошкар-Ола: Поволжский государственный технологический университет.
12. Пикалёв, Я. С. (2017). Анализ существующих симуляторов робототехнических систем. *Проблемы искусственного интеллекта*, (1), 45–52. EDN: https://elibrary.ru/YOJGTN
13. Титенок, А. В. (2022). *Основы робототехники: учебное пособие*. Москва; Вологда: Инфра-Инженерия. 236 с.
14. Фокина, А. А., Каменев, Р. В., Чикова, О. А., & Витюнин, М. А. (2024). Удовлетворённость обучением робототехнике в условиях дополнительного образо-

вания. *Педагогическое образование в России*, (4), 362–376. EDN: https://elibrary.ru/METGYZ

15. Эшниязов, А. И. (2023). Преподавание основ образовательной робототехники в дистанционном формате обучения. *Информатика. Экономика. Управление / Informatics. Economics. Management*, (2), 88–95. https://doi.org/10.47813/2782-5280-2023-2-2-0301-0310. EDN: https://elibrary.ru/MEAEDS
16. Agha, R. A. A., Mahdi, Zh. H., Sefer, M. N., & Hamarash, I. (2021). A ROS-Gazebo interface for the Katana robotic arm manipulation. *UKH Journal of Science and Engineering*, *5*(1), 26–37. https://doi.org/10.25079/ukhjse.v5n1y2021.pp26-37. EDN: https://elibrary.ru/ORWDMB
17. Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control* (3rd ed.). Pearson Prentice Hall.
18. Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. B *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2149–2154, Vol. 3). Sendai, Japan: IEEE. https://doi.org/10.1109/IROS.2004.1389727
19. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*, *3*(3.2), 5.
20. Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer Handbook of Robotics* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-32552-1
21. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). *Robot Modeling and Control* (2nd ed.). Wiley.

### *References*

1. Bzhikhatlov, I. A. (2018). *Modeling robotic systems in the V-REP program: Educational and methodological guide*. Saint Petersburg: ITMO University. Retrieved from https://e.lanbook.com/book/136492
2. Burkov, D. V., & Voloshchenko, Yu. P. (2020). *Mathematical and simulation modeling of electrical and robotic systems: Textbook*. Rostov-on-Don; Taganrog: Southern Federal University Publishing House. ISBN: 978-5-9275-3625-2. Retrieved from https://znanium.com/catalog/product/1308357. EDN: https://elibrary.ru/EGHJRJ
3. Veltishchev, V. V., & Mineev, A. B. (2018). *Methods of simulation modeling of control systems for underwater vehicle motion: Textbook*. Moscow: Bauman Moscow State Technical University. Retrieved from https://e.lanbook.com/book/205493
4. Revich, Yu. V. (2023). *ESP32-C3. Wireless adventure. A complete guide to IoT*. Moscow: DMK Press. 442 p.
5. Zakuraev, R. A. (2023). Robotic construction kits for research projects in physics. *Bulletin of Science*, (6), 112–120. EDN: https://elibrary.ru/YBRJSI

6. Ivanov, A. V. (2024). Library for UAV route planning. *Proceedings of the Volga State Technological University. Series: Technological*, (12), 30–34. EDN: https://elibrary.ru/CWIOUD
7. Ivanov, A. V. (2024). Application of the CoppeliaSim simulation environment for planning the route of an unmanned aerial vehicle. *Engineering Personnel — the Future of Russia's Innovative Economy*, (1), 499–502. EDN: https://elibrary.ru/DKLSOF
8. Kovalev, I. M., & Petrova, E. D. (2021). Comparative analysis of Gazebo and CoppeliaSim simulators for educational tasks. *Informatics and Education*, (5), 32–40.
9. Lavrentiev, E. B., Izyumov, A. I., Marchenko, E. V., & Popov, S. I. (2023). *Fundamentals of robot modeling in the CoppeliaSim environment: Textbook*. Rostov-on-Don: Don State Technical University. ISBN: 978-5-7890-2097-5. EDN: https://elibrary.ru/NFEWUS
10. Loktionov, A. S., & Evdokimova, S. A. (2024). Selecting a simulator program for designing robotic systems. In *Proceedings of the All-Russian Scientific and Practical Conference of Students and Young Scientists* (pp. 212–219). Voronezh: Voronezh State University of Forestry and Technologies. https://doi.org/10.58168/ROBOTICS2024_212-219. EDN: https://elibrary.ru/ADXPIV
11. Morokhin, D. V., Myasnikov, V. I., & Ivanov, A. V. (2023). *Microprocessor systems based on STM32 microcontrollers: Laboratory workshop*. Yoshkar-Ola: Volga State Technological University.
12. Pikalev, Ya. S. (2017). Analysis of existing simulators for robotic systems. *Problems of Artificial Intelligence*, (1), 45–52. EDN: https://elibrary.ru/YOJGTN
13. Titenok, A. V. (2022). *Fundamentals of robotics: Textbook*. Moscow; Vologda: Infra-Engineering. 236 p.
14. Fokina, A. A., Kamenev, R. V., Chikova, O. A., & Vityunin, M. A. (2024). Satisfaction with robotics education in extracurricular settings. *Pedagogical Education in Russia*, (4), 362–376. EDN: https://elibrary.ru/METGYZ
15. Eshniyazov, A. I. (2023). Teaching the basics of educational robotics in a distance learning format. *Informatics. Economics. Management*, (2), 88–95. https://doi.org/10.47813/2782-5280-2023-2-2-0301-0310. EDN: https://elibrary.ru/MEAEDS
16. Agha, R. A. A., Mahdi, Zh. H., Sefer, M. N., & Hamarash, I. (2021). A ROS-Gazebo interface for the Katana robotic arm manipulation. *UKH Journal of Science and Engineering*, 5(1), 26–37. https://doi.org/10.25079/ukhjse.v5n1y2021.pp26-37. EDN: https://elibrary.ru/ORWDMB
17. Craig, J. J. (2005). *Introduction to robotics: Mechanics and control* (3rd ed.). Pearson Prentice Hall.
18. Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vol. 3, pp. 2149–2154). Sendai, Japan: IEEE. https://doi.org/10.1109/IROS.2004.1389727

19. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: An open-source Robot Operating System. *ICRA Workshop on Open Source Software*, 3(3.2), 5.
20. Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer handbook of robotics* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-32552-1
21. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). *Robot modeling and control* (2nd ed.). Wiley.

**ДАННЫЕ ОБ АВТОРАХ**

**Морохин Дмитрий Витальевич**, кандидат технических наук, доцент, заведующий кафедрой информационно-вычислительных систем
*Поволжский государственный технологический университет*
*пл. Ленина, 3, г. Йошкар-Ола, 424000, Российская Федерация*
*dvmitry@mail.ru*

**Иванов Артем Владимирович**, ассистент кафедры информационно-вычислительных систем
*Поволжский государственный технологический университет*
*пл. Ленина, 3, г. Йошкар-Ола, 424000, Российская Федерация*
*artemivanovv@bk.ru*

**DATA ABOUT THE AUTHORS**

**Dmitry V. Morokhin**, PhD in Technical Sciences, Associate Professor, Head of the Department of Information and Computing Systems
*Volga State University of Technology*
*3, Lenin Sq., Yoshkar-Ola, 424000, Russian Federation*
*dvmitry@mail.ru*
*SPIN-code: 6749-2111*
*ORCID: https://orcid.org/0000-0002-0005-3976*
*ResearcherID: PEX-2442-2025*
*Google scholar ID: Bc4f8NcAAAAJ*

**Artem V. Ivanov**, Assistant of the Department of Information and Computing Systems
*Volga State University of Technology*
*3, Lenin Sq., Yoshkar-Ola, 424000, Russian Federation*
*artemivanovv@bk.ru*
*SPIN-code: 8166-3023*
*ORCID: https://orcid.org/0009-0005-2710-0890*

Поступила 15.01.2026
После рецензирования 07.02.2026
Принята 11.02.2026

Received 15.01.2026
Revised 07.02.2026
Accepted 11.02.2026