

ДОБРОВОЛЬСКАЯ Наталья Юрьевна
кандидат педагогических наук, доцент кафедры ин-
формационных технологий
Кубанский государственный университет
(г. Краснодар, Российская Федерация)
dnu10@mail.ru

ХАРЧЕНКО Анна Владимировна
кандидат педагогических наук, доцент кафедры ин-
формационных технологий
Кубанский государственный университет
(г. Краснодар, Российская Федерация)
fz@mail.ru

РАЗВИТИЕ МЕТАКОГНИТИВНЫХ НАВЫКОВ ЧЕРЕЗ ДИАЛОГОВЫЕ ПАТТЕРНЫ ВЗАИМОДЕЙСТВИЯ С ИИ-АССИСТЕНТОМ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ*

Аннотация: широкое внедрение больших языковых моделей в обучение, в том числе и программированию, создает риск использования готовых решений студентами, что может препятствовать развитию критических метакогнитивных навыков: планирования, мониторинга, анализа. В работе проверяется гипотеза о том, что педагогически структурированное взаимодействие с ИИ-ассистентом, в отличие от директивной помощи, способствует более глубокому и устойчивому освоению этих навыков. Материалы и методы. Разработана и практически апробирована таксономия диалоговых паттернов промптов, реализующих стратегии метакогнитивного скаффолдинга. Эффективность паттернов оценивалась в ходе эксперимента с участием 90 студентов-первокурсников, изучающих C++, разделенных на три группы: экспериментальную, взаимодействующую с LLM через педагогические паттерны, контрольную с директивной помощью LLM и группу без ИИ-поддержки. Результаты исследования. Результаты показали, что группа, использовавшая паттерны, продемонстрировала статистически значимо более высокие результаты в тестах на способность применять изученные паттерны в незнакомых предметных областях, а анализ логов диалогов выявил качественно более глубокий мыслительный процесс. Заключение. Показано, что образовательный эффект LLM определяется не самой технологией, а педагогической стратегией взаимодействия, и предлагает конкретный инструментарий для ее реализации.

Ключевые слова: большие языковые модели, искусственный интеллект в образовании, обучение программированию, метакогнитивные навыки, инженерия промптов, образовательные технологии.

Дата поступления: 28.01.2026

Дата публикации: 26.06.2026

Для цитирования: Добровольская Н. Ю., Харченко А. В. Развитие метакогнитивных навыков через диалоговые паттерны взаимодействия с ИИ-ассистентом при обучении программированию // Непрерывное образование: XXI век. 2026. Т. 14. № 2. DOI: 10.15393/j5.art.2026.12246

* Источником финансового обеспечения является грант на обеспечение обучения студентов по образовательным программам высшего образования для топ-специалистов в сфере искусственного интеллекта, предоставленный Аналитическим центром при Правительстве Российской Федерации № 70-2025-000735 от 29.05.2025 ИГК 000000Ц330325P2J0002.

DOBROVOLSKAYA Natalia Yu.

PhD in Pedagogical Sciences, Associate Professor of the
Department of Information Technologies
Kuban State University
(Krasnodar, Russian Federation)

dnu10@mail.ru

KHARCHENKO Anna V.

PhD in Pedagogical Sciences, Associate Professor of the
Department of Information Technologies
Kuban State University
(Krasnodar, Russian Federation)

fz@mail.ru

DEVELOPMENT OF METACOGNITIVE SKILLS THROUGH DIALOGUE PATTERNS OF INTERACTION WITH AN AI ASSISTANT IN PROGRAMMING EDUCATION

Abstract: the widespread integration of large language models (LLMs) into education, including programming instruction, creates the risk of students using ready-made solutions, which may hinder the development of critical metacognitive skills: planning, monitoring, and analysis. This study tests the hypothesis that pedagogically structured interaction with an AI assistant, as opposed to directive assistance, fosters deeper and more sustainable acquisition of these skills. Materials and Methods. A taxonomy of dialogue prompt patterns implementing metacognitive scaffolding strategies was developed and practically tested. The effectiveness of the patterns was evaluated through an experiment involving 90 first-year students studying C++, divided into three groups: an experimental group interacting with an LLM via pedagogical patterns, a control group receiving directive LLM assistance, and a group without AI support. Research Results. The results showed that the group which used patterns demonstrated statistically significantly higher scores on tests measuring the ability to apply learned patterns in unfamiliar subject areas, and log analysis of the dialogues revealed a qualitatively deeper thinking process. Conclusion. It is shown that the educational effect of LLMs is determined not by the technology itself, but by the pedagogical strategy of interaction, and a specific set of tools for its implementation is proposed.

Keywords: large language models, artificial intelligence in education, programming instruction, metacognitive skills, prompt engineering, educational technologies.

Received: January 28, 2026

Date of publication: June 26, 2026

For citation: Dobrovolskaya N. Yu., Kharchenko A. V. Development of metacognitive skills through dialogue patterns of interaction with an AI assistant in programming education. *Nepreryvnoe obrazovanie: XXI vek [Lifelong education: the 21st century]*. 2026. Vol. 14. No. 2. DOI: 10.15393/j5.art.2026.12246

Доступность больших языковых моделей (LLM) и их внедрение в образовательный процесс, в частности в обучение программированию, формирует новые направления для персонализированной поддержки студентов [1; 2]. С одной стороны, LLM-ассистенты демонстрируют высокую операционную эффективность, позволяя практически мгновенно преодолевать синтаксические и логические затруднения при разработке алгоритмических решений. С другой стороны, они потенциально устраняют те необходимые познавательные усилия,

которые обеспечивают приобретение новых знаний и формирование глубокого понимания базовых концепций программирования. Критически важные метакогнитивные навыки, такие как планирование, мониторинг и рефлексия, становятся уязвимыми при взаимодействии с системами, ориентированными на предоставление готовых решений [3–5]. В связи с этим возникает конфликт между мгновенным удобством использования искусственного интеллекта (ИИ) и глубиной устойчивого усвоения знаний.

Актуальность данного исследования определяется тем, что образовательное влияние LLM-ассистента обусловлено не столько техническими возможностями модели, сколько способом взаимодействия, способом формулировки и обработки запроса. Существующие подходы к инженерии промптов преимущественно сосредоточены на технической эффективности решения изолированных задач, игнорируя их педагогический контекст [6; 7]. В то же время накопленный опыт исследований интеллектуальных обучающих систем (ITS) указывает на фундаментальную важность диалоговых стратегий, включая сократический диалог и скаффолдинг (дозированное педагогическое сопровождение с последующей передачей ответственности учащемуся), используемые для активизации конструирования знаний учащимся [8]. Следовательно, возникает проблема составления структурированных и воспроизводимых шаблонов промптов в соответствии с традиционными педагогическими принципами – паттерны промптов для современных генеративных моделей.

Цель работы заключается в системном проектировании, реализации и эмпирической оценке набора конкретных паттернов промптов, обеспечивающих стратегию метакогнитивного скаффолдинга в контексте обучения программированию на языке C++. Выдвигается гипотеза о том, что обучающий LLM-ассистент, управляемый специально разработанными диалоговыми паттернами, окажет более значимое положительное влияние на развитие устойчивых метакогнитивных навыков конструирования алгоритмических решений у начинающих программистов по сравнению с LLM-ассистентом, дающим прямые ответы, или организацией самостоятельной работы без использования ИИ-технологий, только с применением статических ресурсов по обучению программированию. Для этого решаются следующие взаимосвязанные задачи: разработка теоретически обоснованной таксономии педагогических паттернов промптов, связывающей целевые метакогнитивные процессы с конкретными способами диалоговой поддержки; проведение контролируемого межгруппового эксперимента; анализ эмпирических данных, объединяющий количественную оценку эффективности, метрики процесса и качественный анализ мышления учащихся.

Исследования в области обучения программированию последовательно выделяют метакогнитивные навыки – декомпозицию задач, отладку (мониторинг) и объяснение – как ключевые для успешного обучения. Отладка, в частности, понимается как комплексный метакогнитивный процесс, включающий выдвижение гипотез, локализацию ошибки и проверку решений. В современных условиях, когда широкодоступные ИИ-инструменты способны мгновенно давать готовые решения, актуальной становится задача разработки методов, ко-

которые не упрощали бы, а целенаправленно развивали эти навыки, стимулируя активную познавательную деятельность и противодействуя риску пассивной зависимости [9–11].

Ранее задача поддержки метакогнитивного развития решалась в рамках интеллектуальных обучающих систем (ITS), где доказали свою эффективность такие диалоговые стратегии, как сократический диалог и скаффолдинг [8]. Современные большие языковые модели представляют собой новую, более универсальную, но и менее структурированную платформу для образовательного взаимодействия по сравнению со специализированными ITS. Большинство научных работ по интеграции ИИ в образование часто ограничиваются констатацией общих возможностей и барьеров, не предлагая конкретных педагогических подходов для этих новых инструментов [12; 13]. Формирующаяся дисциплина промпт-инжиниринга рассматривается в литературе прежде всего как общая техническая компетенция, а не как инструмент для целенаправленного педагогического проектирования образовательного диалога [6; 7].

Ряд эмпирических исследований рассматривает непосредственное использование языковых моделей студентами-программистами [14; 15]. Эти работы фиксируют неоднозначные эффекты: с одной стороны, отмечается рост доступности помощи и успешности в решении задач, с другой – выявляются риски формирования привычки к пассивному получению ответов и снижение усилий по самостоятельному анализу и поиску ошибок. Одновременно в области компьютерных наук развивается инженерия промптов, предлагая техники для управления выводом и поведением моделей [16; 17], анализируется стихийное, нерегламентированное использование моделей студентами [18].

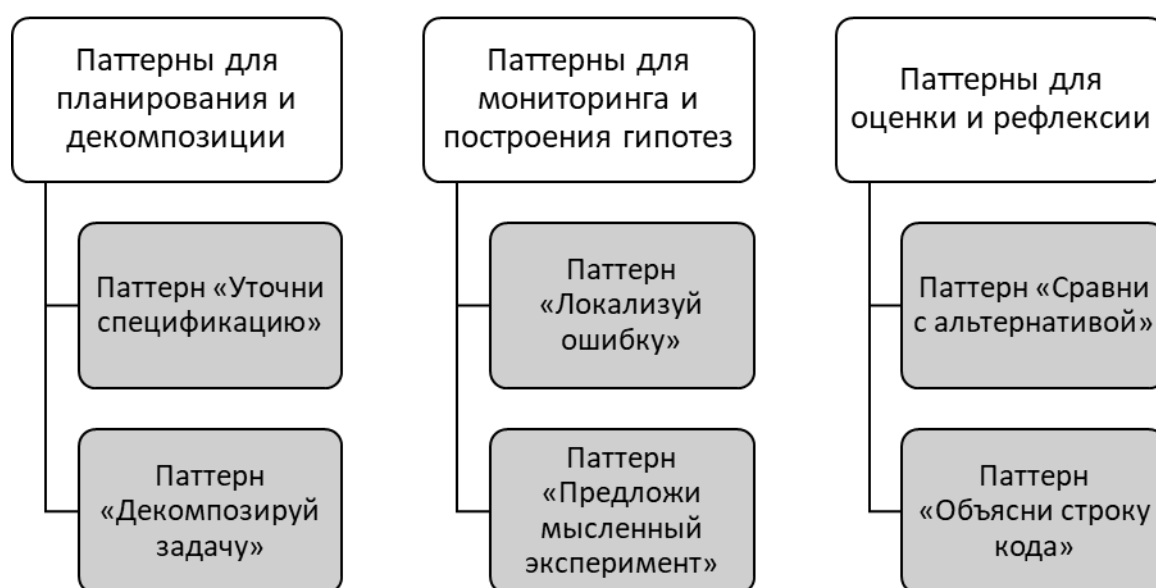
Таким образом, в современных исследованиях возникает разрыв между имеющимися педагогическими подходами к обучению и формированию метакогнитивных навыков, с одной стороны, и инновационным инструментарием управления большими языковыми моделями через промптинг – с другой. Недостаёт исследований, которые бы осуществляли системную интеграцию традиционных педагогических стратегий в конкретные, воспроизводимые и эффективные паттерны промптов для образовательных LLM-ассистентов. Настоящее исследование предлагает таксономию таких паттернов, специально предназначенных для развития метакогнитивных навыков у начинающих программистов.

Разработка педагогических паттернов промптов осуществлялась в рамках итеративного процесса, объединяющего принципы проектирования, теории диалогового обучения и современные практики инженерии промптов. Процесс проектирования состоял из трех взаимосвязанных фаз. На первой, аналитической фазе, были выявлены ключевые метакогнитивные проблемы, с которыми сталкиваются студенты – начинающие программисты – при разработке на C++: неспособность четко сформулировать ожидаемое поведение программы, склонность к бессистемному поиску ошибок, трудности в интерпретации сообщений компилятора и отсутствие привычки к рефлексии работоспособности найденного решения. На второй, конструкторской фазе, для преодоления каждого выявленного барьера был предложен соответствующий диалоговый паттерн, переводящий абстрактную педагогическую цель в конкретную инструкцию для

языковой модели. Каждый паттерн формулировался как шаблон с переменными, позволяющий адаптировать его к контексту конкретной ошибки. На третьей, валидационной фазе, предварительные формулировки паттернов тестировались в диалогах с моделью Yandex Pro 5, после чего они уточнялись для обеспечения стабильности, дидактической обоснованности и отсутствия ухода в прямые ответы. Итогом этого процесса стала таксономия паттернов, структурированная в соответствии с моделью метакогнитивного регулирования (планирование, мониторинг, анализ), представленная ниже (см. рис.).

Паттерны для планирования и декомпозиции

Данная группа паттернов нацелена на этап, предшествующий непосредственному написанию или исправлению кода, и призвана сформировать у студента привычку к анализу задачи и стратегическому планированию.



Таксономия паттернов

Pattern taxonomy

Паттерн «Уточни спецификацию». Педагогическое обоснование строится на том, что значительная часть логических ошибок происходит из неполного или неправильного понимания условий задачи. Паттерн направлен на развитие навыка формализации требований. В реализации системного промпта он задается инструкцией: «Если студент представляет код с логической ошибкой или просит помощи в реализации, прежде чем анализировать код, попроси его своими словами описать, что, по его мнению, должна делать программа в текущем фрагменте, и какие входные данные приводят к некорректному выводу». Ожидаемое поведение модели заключается в том, что она, вместо анализа кода, задает вопросы вида: «Можете ли вы своими словами описать, что, по вашему замыслу, должна делать эта программа? И что происходит, когда вы ее запускаете?». Это смещает фокус студента с синтаксиса на семантику, часто позволяя

ему самостоятельно обнаружить противоречие между идеей решения и реализацией.

Паттерн «Декомпозируй задачу». Основан на принципе «разделяй и властвуй», одном из базовых для программирования. Его цель – научить студента разбивать сложную ошибку на серию проверяемых подзадач. В промпте LLM-ассистента это фиксируется так: «При работе со сложной, многосоставной ошибкой предложи студенту составить поэтапный план проверки: какие подфункции или блоки кода можно проверить независимо, и в какой последовательности это сделать». Модель, следуя этому паттерну, может предложить: «План поэтапной проверки и локализации ошибки. Давайте разберемся системно. Этап 1: Уточнение задачи. Прежде чем исправлять код, нужно четко понять ТЗ. Ответьте на вопросы: Что должна делать эта программа в целом? Какой результат вы ожидаете получить для конкретного примера ввода?». Таким образом, LLM-ассистент выступает организатором плана декомпозиции ошибки, а не его исполнителем.

Паттерны для мониторинга и построения гипотез

Эти паттерны активируются на этапе активного поиска ошибки и предназначены для развития навыков выдвижения и проверки диагностических гипотез, а также систематического наблюдения за поведением программы.

Паттерн «Локализуй ошибку». Направлен на исключение бессистемного просмотра кода. Паттерн формирует навык перехода от наблюдения проблемы к формированию гипотезы о месте ошибки. Инструкция для модели формулируется так: «Когда студент сообщает об ошибке, попроси его указать минимальный фрагмент кода, который, по его предположению, содержит ошибку, и обосновать это предположение. предложи добавить промежуточные выводы для сужения области поиска». В диалоге это проявляется в репликах LLM-ассистента: «Выделите минимальный фрагмент кода, который, по вашему мнению, содержит ошибку. Обоснуйте свое предположение. Почему вы думаете, что ошибка именно там?».

Паттерн «Предложи мысленный эксперимент». Является цифровой адаптацией сократического метода. Он заставляет студента моделировать выполнение кода, что развивает понимание отдельных шагов алгоритма и их последовательности. Паттерн включается в промпт: «Вместо того чтобы сразу указывать на ошибку, предложи студенту мысленно выполнить трассировку выполнения кода на конкретном наборе входных данных. Задавай наводящие вопросы о состоянии конкретных переменных в ключевых точках». Пример выполнения: «Трассировка функции kolvo(0). Вопрос 1: Сначала давайте поймем, что должна делать kolvo по вашему замыслу. Своими словами: что она возвращает для массива [5, -2, 3]?». Этот паттерн развивает способность к предсказанию поведения программы.

Паттерны для оценки и рефлексии

Эта группа паттернов ориентирована на анализ решения и выполненных исправлений ошибок, призвана углубить понимание через объяснение и критическую оценку решения.

Паттерн «Объясни строку кода». Призван предотвратить механическое копирование работающих строк кода и использование шаблонов без понимания их сути. Системная инструкция следующая: «После того как студент предлагает исправление или получает рабочий код, попроси его объяснить смысл ключевой, самой сложной строки в этом исправлении, используя аналогии или описывая операции шаг за шагом». Реализация: «Пожалуйста, объясните своими словами, шаг за шагом: что происходит, когда выполняется эта строка? Опишите максимально подробно, как будто объясняете человеку, который только начал изучать программирование». Это превращает пассивное принятие кода в активный этап конструирования знания.

Паттерн «Сравни с альтернативой». Нацелен на развитие гибкости мышления и понимания того, что большинство проблем имеет более одного решения. Паттерн способствует поиску различных подходов к задаче. В промпте это определяется так: «Когда ошибка исправлена, предложи студенту придумать альтернативный способ решения той же подзадачи и обсудить плюсы и минусы обоих подходов (например, читабельность, производительность, использование памяти)». Модель, следуя паттерну, может сказать: «Придумайте и опишите как минимум два альтернативных способа решить ту же подзадачу. Для каждого подхода ответьте на вопросы: Как будет выглядеть код? (примерный псевдокод или описание логики). Какие у него преимущества перед текущим рекурсивным подходом? Какие недостатки или ограничения? В какой ситуации этот подход был бы предпочтительнее?». Этот паттерн предлагает студенту задания на рассуждение, формирует критический подход к решению задачи.

Предложенная таксономия представляет собой не жесткий алгоритм, а набор инструментов, которыми LLM-ассистент управляется динамически, в зависимости от контекста ошибки и наблюдаемого поведения студента. Все паттерны объединены общим запретительным правилом, внедренный в системный промпт: «Никогда не предоставляй исправленный полный код функции или блок целиком в качестве первого ответа. Действуй как преподаватель, направляющий мыслительный процесс, а не как решатель задач». Это обеспечивает соблюдение ключевого педагогического принципа – сохранения когнитивной ответственности за решение за студентом.

Проведенный эксперимент позволил оценить влияние разных типов взаимодействия с ИИ на развитие навыков отладки у 90 студентов 1-го курса факультета компьютерных технологий и прикладной математики Кубанского государственного университета, изучавших C++. Участники были разделены на три сопоставимые группы.

Группа А (Паттерны, N = 30): LLM-ассистент работал под управлением системного промпта, интегрировавшего разработанную таксономию педагогических паттернов («Уточни спецификацию», «Локализуй ошибку» и др.). Его поведение было регламентировано инструкцией действовать как педагог, задающий наводящие вопросы и никогда не предоставляющий полный исправленный код в первом ответе.

Группа Б (Директивная, N = 30): LLM-ассистент получал промпт, предписывающий давать прямые, конкретные ответы. В ответ на описание проблемы или код с ошибкой модель должна была сразу предоставить исправленную версию фрагмента или всей функции с кратким пояснением, что было неверно. Стратегия соответствовала типичному запросу студента «исправь мой код».

Группа В (Контрольная, N = 30): Участники этой группы не имели доступа к диалоговому LLM-ассистенту. Вместо этого для них была доступна официальная справочная информация по C++ и учебные конспекты по пройденным темам. Эта группа служила базовым уровнем для оценки эффекта любого ИИ-взаимодействия. Контроль за отсутствием использования ИИ осуществлялся путем выполнения лабораторных работ в аудитории без доступа к сети Интернет, а также устной защитой решений при выявлении подозрительных фрагментов кода.

Для всех трех групп было проведено предварительное и итоговое тестирование, а также отложенный тест через неделю. Эффективность оценивалась по логам диалогов (для групп А и Б), успешности решения учебных задач, а также по опросникам когнитивной нагрузки и самооффективности. Через неделю был проведен дополнительный тест для оценки переноса навыков и анализа метакогнитивных стратегий. Для статистического анализа использовался непараметрический U-критерий Манна – Уитни, выбранный в силу порядкового характера данных и отсутствия нормального распределения. Такая организация эксперимента позволила комплексно проверить связь между типом поддержки ИИ и развитием метанавыков.

Результаты эксперимента выявили статистически значимое влияние условий на развитие навыков программирования. В тесте по окончании эксперимента группа А (педагогические паттерны) значимо превзошла как группу Б (директивная помощь, U-критерий Манна – Уитни, $p = 0,012$), так и контрольную группу В ($p < 0,001$). В тесте, проведенном через неделю после завершения экспериментальной работы преимущество группы А усилилось (U-критерий Манна – Уитни, $p = 0,003$ против группы Б и $p < 0,001$ против группы В). Это свидетельствует о более глубоком усвоении и устойчивом переносе метакогнитивных навыков при диалоговом стиле взаимодействия с ИИ.

Качественное различие между группами проявилось в характеристиках процесса взаимодействия с LLM. Диалоги в группе А были значительно длиннее и содержали преимущественно встречные вопросы (78 % реплик), направляющие студента к самостоятельному поиску решения. В группе Б помощь была прямой, с редкими уточнениями (22 % реплик). Успешность решения учебных задач была достаточно высока и сопоставима в группах А и Б (97–100 %). Однако по результатам дополнительного теста на концептуальное понимание (открытые вопросы и задачи на модификацию кода) участники группы А продемонстрировали статистически значимо более высокие результаты (82 % против 67 % в группе Б, $p = 0,021$), что свидетельствует о более глубоком понимании учебного материала.

Эффективность работы LLM-ассистента под управлением педагогических паттернов была неабсолютной: в 11 % диалогов возникали сбои, связанные с

концептуальной неготовностью студентов или сложностью ошибки. Группа Б сообщила о меньшей когнитивной нагрузке и большей помощи в предоставлении мгновенных конкретных ответов, тогда как группа А оценила процесс как более сложный. Однако именно этот длительный диалог позволил сформировать необходимые навыки и дать лучшие результаты в долгосрочном поиске решений учебных задач.

Исследование подтверждает, что педагогически структурированные промпты, реализующие стратегии скаффолдинга, способствуют развитию метакогнитивных навыков эффективнее, чем директивная помощь. Основными ограничениями исследования являются специфика языка C++ и объем выборки. Дальнейшая работа должна проверить устойчивость эффекта в образовательной среде с другими языковыми моделями.

ИИ-ассистент, управляемый педагогическими паттернами, реализующими стратегии метакогнитивного скаффолдинга, статистически значимо эффективнее способствует развитию навыков отладки у начинающих программистов, чем директивная помощь или самостоятельная работа с учебными материалами. Определены основные различия в педагогических стратегиях. Директивный ассистент обеспечивает более высокую оперативную эффективность и субъективную удовлетворенность пользователя, но при этом может подменять необходимые усилия для приобретения навыков. Подход с использованием паттернов, напротив, создает условия самостоятельного поиска решений, и, несмотря на более высокую сложность, позволяет формировать навыки анализа и критического мышления.

Таксономию педагогических промптов можно рассматривать как инструмент, использующий традиционные педагогические подходы (скаффолдинг, сократический диалог), трансформированные в конкретные, воспроизводимые инструкции для больших языковых моделей. Наиболее универсальными и эффективными паттернами оказались «Локализуй ошибку» и «Предложи мысленный эксперимент», которые активизировали у студентов навыки самостоятельного анализа и планирования действий.

Исследование подтверждает, что дидактические особенности LLM существенно зависят от стратегии взаимодействия. Результаты показывают, что не технология сама по себе, а ее педагогически обоснованная конфигурация определяет, будет ли ассистент формировать зависимость обучаемого от готовых ответов или развивать навыки самостоятельного аналитического мышления.

Предложенная таксономия паттернов может быть использована при разработке адаптивных систем, динамически подстраивающихся под уровень студента. В дальнейшем следует расширить таксономию на другие сложные темы программирования, например на объектно-ориентированное программирование, выполнить оценку долгосрочного влияния применяемых стратегий на академические результаты и мотивацию.

Список литературы

1. Долинский М. С. Направления использования генеративного искусственного интеллекта при начальном обучении программированию в университетах [Электронный ре-

сурс] // Компьютерные инструменты в образовании. 2024. № 2. С. 85–96. Электрон. дан. URL: <https://cyberleninka.ru/article/n/napravleniya-ispolzovaniya-generativnogo-iskusstvennogo-intellekta-pri-nachalnom-obuchenii-programmirovaniyu-v-universitetah> (дата обращения 18.12.2025).

2. Яшина И. А. Искусственный интеллект в обучении программированию студентов педагогического вуза [Электронный ресурс] // Открытое образование. 2024. № 4. С. 23–32. Электрон. дан. URL: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-obuchenii-programmirovaniyu-studentov-pedagogicheskogo-vuza> (дата обращения 18.12.2025).

3. Беленкова Ю. С. Обучение метакогнитивным навыкам и методы оценки их сформированности [Электронный ресурс] // Гуманитарные, социально-экономические и общественные науки. 2015. № 3-2. С. 20–22. Электрон. дан. URL: <https://cyberleninka.ru/article/n/obuchenie-metakognitivnym-navykam-i-metody-otsenki-ih-sformirovannosti> (дата обращения 18.12.2025).

4. Абдулвахабова Б. Б.-А., Магомедова П. К., Батчаева З. Б. Педагогические условия как инструмент развития когнитивных способностей выпускника вуза [Электронный ресурс] // Проблемы современного педагогического образования. 2024. № 85-4. С. 7–10. Электрон. дан. URL: <https://cyberleninka.ru/article/n/pedagogicheskie-usloviya-kak-instrument-razvitiya-kognitivnyh-sposobnostey-vypusknika-vuza> (дата обращения 18.12.2025).

5. Непесова С. Развитие когнитивных навыков студентов через интеграцию цифровых технологий в образовательный процесс [Электронный ресурс] // Международный журнал гуманитарных и естественных наук. 2025. № 3-2 (102). С. 95–100. Электрон. дан. URL: <https://cyberleninka.ru/article/n/razvitie-kognitivnyh-navykov-studentov-cherez-integratsiyu-tsifrovyyh-tehnologiy-v-obrazovatelnyy-protsess> (дата обращения 18.12.2025).

6. Алферьева-Термсинос В. Б. Промт-инжиниринг как стратегия формирования информационной культуры обучающихся [Электронный ресурс] // Международный журнал гуманитарных и естественных наук. 2024. № 9-1 (96). С. 10–15. Электрон. дан. URL: <https://cyberleninka.ru/article/n/promt-inzhiniring-kak-strategiya-formirovaniya-informatsionnoy-kultury-obuchayuschih-sya> (дата обращения 18.12.2025).

7. Лукинский И. С., Горшенева И. А. Промт-инжиниринг в образовательном процессе и научной деятельности или к вопросу о необходимости обучения работе с искусственным интеллектом [Электронный ресурс] // Психология и педагогика служебной деятельности. 2024. № 4. С. 148–154. Электрон. дан. URL: <https://cyberleninka.ru/article/n/promt-inzhiniring-v-obrazovatelnom-protsesse-i-nauchnoy-deyatelnosti-ili-k-voprosu-o-neobhodimosti-obucheniya-rabote-s> (дата обращения 18.12.2025).

8. Расулова Н. Ю. Мультиагентный подход в создании адаптивных интеллектуальных обучающих систем [Электронный ресурс] // Экономика и социум. 2021. № 2-2 (81). С. 524–531. Электрон. дан. URL: <https://cyberleninka.ru/article/n/multiagentnyy-podhod-v-sozdanii-adaptivnyh-intellektualnyh-obuchayuschih-sistem> (дата обращения 04.12.2025).

9. Губашева Х. А., Магомедова Д. М., Магазиева З. А. Инновационные методы обучения программированию и ИТ в российских вузах [Электронный ресурс] // Международный научно-исследовательский журнал. 2022. № 5-3 (119). С. 69–71. Электрон. дан. URL: <https://cyberleninka.ru/article/n/innovatsionnye-metody-obucheniya-programmirovaniyu-i-it-v-rossiyskih-vuzah> (дата обращения 04.12.2025).

10. Скворчевский К. А., Дятлова О. В. Современные адаптивные и интеллектуальные цифровые системы обучения: механизмы и потенциал [Электронный ресурс] // Вопросы образования. 2024. № 3 (2). С. 237–299. Электрон. дан. URL: <https://cyberleninka.ru/article/n/sovremennye-adaptivnye-i-intellektualnye-tsifrovye-sistemy-obucheniya-mehanizmy-i-potentsial> (дата обращения 04.12.2025).

11. Попова Ю. Б. От LMS к адаптивным обучающим системам [Электронный ресурс] // Системный анализ и прикладная информатика. 2019. № 2. С. 58–64. Электрон. дан. URL: <https://cyberleninka.ru/article/n/ot-lms-k-adaptivnym-obuchayuschim-sistemam> (дата обращения 04.12.2025).

12. Платов А. В., Гаврилина Ю. И. Искусственный интеллект в образовании: эволюция и барьеры // Научный результат. Педагогика и психология образования. 2024. № 1. С. 26–43.
13. Ярцева Е. Я. Интеграция искусственного интеллекта в образование // Проблемы современного педагогического образования. 2024. № 85-2. С. 398–401.
14. Finnie-Ansley J., Denny P., Becker B. A. [et al.]. The robots are coming: Exploring the implications of OpenAI Codex on introductory programming // Proceedings of the 24th Australasian Computing Education Conference. 2022. P. 10–19.
15. Kazemitabaar M., Chow J., Ma C. K. T. [et al.]. Studying the effect of AI code generators on supporting novice learners in introductory programming // Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. 2023. P. 1–23.
16. Arora S., Narayan A., Chen M. F. [et al.]. Ask me anything: A simple strategy for prompting language models // arXiv preprint arXiv:2210.02441. 2022.
17. Kiesler N., Schiffner D. Large language models in introductory programming education: ChatGPT's performance and implications for assessments // arXiv:2308.08572. 2023.
18. Pearce H., Tan B., Krishnamurthy P. [et al.]. Pop quiz! Can a large language model help with reverse engineering? // arXiv:2202.01142. 2022.

References

1. Dolinsky M. S. Directions for Using Generative Artificial Intelligence in Initial Programming Training at Universities [Electronic resource]. *Komp'yuternye instrumenty v obrazovanii [Computer Tools in Education]*. 2024. No. 2. P. 85–96. Electron. dan. URL: <https://cyberleninka.ru/article/n/napravleniya-ispolzovaniya-generativnogo-iskusstvennogo-intellekta-pri-nachalnom-obuchenii-programmirovaniyu-v-universitetah> (date of access 18.12.2025). (In Russ.)
2. Yashina I. A. Artificial Intelligence in Teaching Programming to Pedagogical University Students [Electronic resource]. *Otkrytoe obrazovanie [Open Education]*. 2024. No. 4. P. 23–32. Electron. dan. URL: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-obuchenii-programmirovaniyu-studentov-pedagogicheskogo-vuza> (date of access 18.12.2025). (In Russ.)
3. Belenkova Yu. S. Teaching Metacognitive Skills and Methods for Assessing Their Development [Electronic resource]. *Gumanitarnye, sotsial'no-ekonomicheskie i obshchestvennye nauki [Humanities, Socio-Economic and Social Sciences]*. 2015. No. 3-2. P. 20–22. Electron. dan. URL: <https://cyberleninka.ru/article/n/obuchenie-metakognitivnym-navykam-i-metody-otsenki-ih-sformirovannosti> (date of access 18.12.2025). (In Russ.)
4. Abdulvahabova B. B.-A., Magomedova P. K., Batchaeva Z. B. Pedagogical Conditions as a Tool for Developing the Cognitive Abilities of a University Graduate [Electronic resource]. *Problemy sovremennogo pedagogicheskogo obrazovaniya [Problems of Modern Pedagogical Education]*. 2024. No. 85-4. P. 7–10. Electron. dan. URL: <https://cyberleninka.ru/article/n/pedagogicheskie-usloviya-kak-instrument-razvitiya-kognitivnyh-sposobnostey-vypusknika-vuza> (date of access 18.12.2025). (In Russ.)
5. Nepesova S. Development of Students' Cognitive Skills through the Integration of Digital Technologies into the Educational Process [Electronic resource]. *Mezhdunarodnyy zhurnal gumanitarnykh i estestvennykh nauk [International Journal of Humanities and Natural Sciences]*. 2025. No. 3-2 (102). P. 95–100. Electron. dan. URL: <https://cyberleninka.ru/article/n/razvitiye-kognitivnyh-navykov-studentov-cherez-integratsiyu-tsifrovyyh-tehnologiy-v-obrazovatelnyy-protsess> (date of access 18.12.2025). (In Russ.)
6. Alfer'eva-Termisikova V. B. Prompt Engineering as a Strategy for Forming the Information Culture of Learners [Electronic resource]. *Mezhdunarodnyy zhurnal gumanitarnykh i estestvennykh nauk [International Journal of Humanities and Natural Sciences]*. 2024. No. 9-1 (96). P. 10–15. Electron. dan. URL: <https://cyberleninka.ru/article/n/prompt-inzhiniring-kak-strategiya-formirovaniya-informatsionnoy-kultury-obuchayuschihsya> (date of access 18.12.2025). (In Russ.)
7. Lukinsky I. S., Gorsheneva I. A. Prompt Engineering in the Educational Process and Scientific Activity or on the Issue of the Need for Training in Working with Artificial Intelligence.

[Electronic resource]. *Psikhologiya i pedagogika sluzhebnoy deyatel'nosti [Psychology and Pedagogy of Official Activities]*. 2024. No. 4. P. 148–154. Electron. dan. URL: <https://cyberleninka.ru/article/n/promt-inzhiniring-v-obrazovatelnom-protseesse-i-nauchnoy-deyatelnosti-ili-k-voprosu-o-neobhodimosti-obucheniya-rabote-s> (date of access 18.12.2025). (In Russ.)

8. Rasulova N. Yu. Multi-Agent Approach in Creating Adaptive Intelligent Tutoring Systems [Electronic resource]. *Ekonomika i sotsium [Economics and Society]*. 2021. No. 2-2 (81). P. 524–531. Electron. dan. URL: <https://cyberleninka.ru/article/n/multiagentnyy-podhod-v-sozdanii-adaptivnyh-intellektualnyh-obuchayuschiy-sistem> (date of access 04.12.2025). (In Russ.)

9. Gubasheva Kh. A., Magamedova D. M., Magazieva Z. A. Innovative Methods of Teaching Programming and IT in Russian Universities [Electronic resource]. *Mezhdunarodnyy nauchno-issledovatel'skij zhurnal [International research journal]*. 2022. No. 5-3 (119). P. 69–71. Electron. dan. URL: <https://cyberleninka.ru/article/n/innovatsionnye-metody-obucheniya-programmirovaniyu-i-it-v-rossijskikh-vuzah> (date of access 04.12.2025). (In Russ.)

10. Skvortshevsky K. A., Dytlova O. V. Modern Adaptive and Intelligent Digital Learning Systems: Mechanisms and Potential [Electronic resource]. *Voprosy obrazovaniya [Educational Studies]*. 2024. No. 3 (2). P. 237–299. Electron. dan. URL: <https://cyberleninka.ru/article/n/sovremennye-adaptivnye-i-intellektualnye-tsifrovye-sistemy-obucheniya-mehanizmy-i-potentsial> (date of access 04.12.2025). (In Russ.)

11. Popova Yu. B. From LMS to Adaptive Learning Systems. [Electronic resource]. *Sistemnyy analiz i prikladnaya informatika [System Analysis and Applied Informatics]*. 2019. No. 2. P. 58–64. Electron. dan. URL: <https://cyberleninka.ru/article/n/ot-lms-k-adaptivnym-obuchayuschim-sistemam> (date of access 04.12.2025). (In Russ.)

12. Platov A. V., Gavrilina Yu. I. Artificial Intelligence in Education: Evolution and Barriers. *Nauchnyy rezul'tat. Pedagogika i psikhologiya obrazovaniya [Research Result. Pedagogy and Psychology of Education]*. 2024. No. 1. P. 26–43. (In Russ.)

13. Yartseva E. Ya. Integration of Artificial Intelligence into Education. *Problemy sovremennogo pedagogicheskogo obrazovaniya [Problems of Modern Pedagogical Education]*. 2024. No. 85-2. P. 398–401. (In Russ.)

14. Finnie-Ansley J., Denny P., Becker B. A. [et al.]. The robots are coming: Exploring the implications of OpenAI Codex on introductory programming. *Proceedings of the 24th Australasian Computing Education Conference*. 2022. P. 10–19.

15. Kazemitabaar M., Chow J., Ma C. K. T. [et al.]. Studying the effect of AI code generators on supporting novice learners in introductory programming. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023. P. 1–23.

16. Arora S., Narayan A., Chen M. F. [et al.]. Ask me anything: A simple strategy for prompting language models. *arXiv preprint arXiv:2210.02441*. 2022.

17. Kiesler N., Schiffner D. Large language models in introductory programming education: ChatGPT's performance and implications for assessments. *arXiv:2308.08572*. 2023.

18. Pearce H., Tan B., Krishnamurthy P. [et al.]. Pop quiz! Can a large language model help with reverse engineering? *arXiv:2202.01142*. 2022.