

DOI: 10.20913/2224-1841-2026-1-7

УДК 37.02: 004.8

Оригинальная научная статья

Использование генеративных инструментов искусственного интеллекта в обучении студентов программированию

А. А. Алетдинова

*Сибирский государственный университет инженерии и биотехнологий
Новосибирск, Российская Федерация
e-mail: kaf-aoi418@mail.ru*

Д. Д. Петров

*Новосибирский государственный технический университет
Новосибирск, Российская Федерация
e-mail: kaf_asu@corp.nstu.ru*

М. Е. Фролов

*Новосибирский государственный технический университет
Новосибирск, Российская Федерация
e-mail: kaf_asu@corp.nstu.ru*

Аннотация. *Введение.* Генеративные нейросетевые инструменты используются в настоящее время для обработки текстов, написания тезисов, решения задач и т.д. Их возможности позволяют еще и генерировать, интерпретировать, и исправлять код. Неконтролируемое использование студентами таких сервисов приводит к более низкому усвоению основ программирования, невозможности формирования умений по оптимизации кода. В условиях, когда нет возможности ограничить использование генеративных нейросетевых инструментов, необходимо менять методики преподавания, добавляя в них изучение основ промпт-инжиниринга. *Постановка задачи.* В статье рассматриваются ИИ-сервисы: ChatGPT, DeepSeek, Cursor, OpenAI Codex, Qwen, Gemini, Claude и Perplexity для обучения программированию. Цель данного исследования – в обосновании необходимости введения промпт-инжиниринга в учебные программы подготовки бакалавров ИТ-специальностей. На основе анализа научной литературы и собственного опыта преподавания выделены особенности применения генеративных нейросетевых сервисов в области программирования. Проведено статистическое исследование предпочтений и опыта студентов в их использовании. *Методика и методология исследования.* В исследовании применялись методы системного и сравнительного анализа, анкетирования, а также обобщения и статистической обработки данных. Рассмотрены наиболее распространённые генеративные нейросетевые сервисы, которые используются студентами в процессе изучения программирования. Обоснованы их достоинства и недостатки. Кроме того, проведён опрос студентов направления «Прикладная информатика» для выявления частоты и специфики использования генеративных нейросетевых ассистентов в образовательной деятельности. Эмпирические данные получены путём сводки, группировки и вычисления относительных показателей. *Результаты.* В работе выделены ключевые функции генеративных нейросетевых ассистентов, которые чаще всего применяются студентами в образовательной деятельности. К ним относятся создание программного кода с нуля, дополнение и исправление уже существующих программ, поиск и устранение ошибок, пояснение алгоритмической логики, освоивание новых технологий и библиотек, а также подготовка отчётов. Анализ показал, что наибольшей популярностью пользуются сервисы ChatGPT и DeepSeek, в то время как Cursor и OpenAI Codex, как правило, используются в качестве вспомогательных средств разработки, а Claude и Perplexity – преимущественно для поиска информации и разъяснения сложных концепций. Кроме того, выделены основные педагогические риски, связанные с использованием генеративных нейросетевых ассистентов, среди которых отмечены снижение уровня самостоятельности студентов, поверхностное выполнение учебных задач и повышенная вероятность академического плагиата. *Выводы.* Признана необходимость включения промпт-инжиниринга в учебные программы бакалавров ИТ-специальностей.

Ключевые слова: генеративный искусственный интеллект, генеративные нейросетевые ассистенты, обучение программированию, студенты, промпт-инжиниринг, генерация кода, цифровые образовательные технологии, академический плагиат

Для цитирования: Алетдинова А. А., Петров Д. Д., Фролов М. Е. Использование генеративных инструментов искусственного интеллекта в обучении студентов программированию // Профессиональное образование в современном мире. 2026. Т. 16, №1. С. 49–57. DOI: <https://doi.org/10.20913/2224-1841-2026-1-7>

DOI: 10.20913/2224-1841-2026-1-7

Full Article

Using generative AI tools in teaching students programming

Aletdinova, A. A.

*Siberian State University Engineering and Biotechnology
Novosibirsk, Russian Federation
e-mail: kaf-aoi418@mail.ru*

Petrov, D. D.

*Novosibirsk State Technical University
Novosibirsk, Russian Federation
e-mail: kaf_asu@corp.nstu.ru*

Frolov, M. E.

*Novosibirsk State Technical University
Novosibirsk, Russian Federation
e-mail: kaf_asu@corp.nstu.ru*

Abstract. *Introduction.* Generative neural network tools are currently used for text processing, thesis writing, problem solving, and more. Their capabilities also allow for code generation, interpretation, and correction. Uncontrolled use of such services by students leads to a lower understanding of programming fundamentals and an inability to develop code optimization skills. In a context where it is impossible to limit the use of generative neural network tools, teaching methods must be modified to incorporate the fundamentals of prompt engineering. *Purpose setting.* This article examines the following AI services for teaching programming: ChatGPT, DeepSeek, Cursor, OpenAI Codex, Qwen, Gemini, Claude, and Perplexity. The purpose of this study is to justify the need to introduce prompt engineering into undergraduate IT curricula. Based on an analysis of scientific literature and the authors' own teaching experience, the authors highlight the specific features of using generative neural network services in programming. A statistical study of students' preferences and experiences with them was conducted. *Methodology and methods of the study.* The study utilized methods of systemic and comparative analysis, questionnaires, and data aggregation and statistical processing. The most common generative neural network services used by students in their programming studies are examined, and their advantages and disadvantages are discussed. A survey of students majoring in Applied Computer Science was also conducted to determine the frequency and specifics of their use of generative neural network assistants in educational activities. Empirical data was obtained through summarizing, grouping, and calculating relative indicators. *Results.* The paper highlights the key functions of generative neural network assistants most frequently used by students in their educational activities. These include creating program code from scratch, expanding and correcting existing programs, finding and fixing errors, clarifying algorithmic logic, mastering new technologies and libraries, and preparing reports. The analysis revealed that ChatGPT and DeepSeek are the most popular services, while Cursor and OpenAI Codex are typically used as development aids, and Claude and Perplexity are primarily used for information retrieval and clarification of complex concepts. Furthermore, the paper identifies the key pedagogical risks associated with the use of generative neural network assistants, including decreased student independence, superficial completion of learning tasks, and an increased risk of academic plagiarism. *Conclusion.* The paper acknowledges the need to incorporate prompt engineering into undergraduate IT curricula.

Keywords: Generative artificial intelligence, generative neural network assistants, programming training, students, prompt engineering, code generation, digital educational technologies, academic plagiarism

Citation: Aletdinova, A. A., Petrov, D. D., Frolov, M. E. [Using generative AI tools in teaching students programming]. *Professional education in the modern world*, 2026, vol. 16, no. 1, pp. 49–57. DOI: <https://doi.org/10.20913/2224-1841-2026-1-7>

Введение. Методика преподавания программирования в университетах давно отработана. Она включает в себя гармоничное сочетание теории и практики, постепенный переход от базовых понятий, переменных и функций к их применению с последующим усложнением задач. При этом, как отмечает И. В. Баженова, у студентов 1 курса наблюдается низкий уровень системного мышления, алгоритмической грамотности и навыков программирования [1].

С появлением нейросетевых сервисов, умеющих генерировать код, в наш лексикон вошел термин «пром프트-инжиниринг». Под ним пользователи понимают умение составлять запросы к нейросетевым сервисам. В. Б. Алферьева-Термсинос отмечает, что пром프트-инжиниринг в настоящее время формирует информационную культуру обучающихся [2].

При этом мнения о необходимости его использования резко разделились в научном сообществе. Одни – пишут о новых возможностях, которые открывают генеративные нейросетевые сервисы, и подчеркивают преимущества построения индивидуальной образовательной траектории [3; 4]. Другие ученые отмечают проблемы, связанные с их использованием, наиболее частые из них следующие:

- «галлюцинации», которые приводят к ошибкам в коде [5];
- неучет бизнес-логики, условий или архитектурных ограничений проекта [6; 7];
- риск использования нелицензионного или запатентованного кода из обучающих данных [8];
- получение некорректных решений из-за плохо структурированных и некорректных промптов [9–11], отсутствие знаний предметной области для составления эффективных запросов [12; 13];
- «замыливание взгляда», чрезмерное доверие ИИ-сервисам [14].

Как отмечает группа ученых A. Hellas, J. Leinonen, S. Sarsa, C. Koutcheme, L. Kujanpää, J. Sorva [14], в рамках их исследования доказано, что участники эксперимента демонстрировали проблемы с переносом знаний между задачами, пониманием алгоритмических концепций, отладкой сгенерированного кода. В качестве решения этой проблемы они предлагают следующие педагогические приемы: использования ИИ-ассистентов с наставником, развитие метакогнитивных навыков проверки кода и поэтапное сокращение зависимости от генеративных сервисов.

Бесконтрольное использование студентами ИИ-сервисов в процессе обучения программиро-

ванию приводит к грубым ошибкам, неоптимальности кода. Наиболее частый пример дублирования кода, когда загрузка одних и тех же библиотек встречается несколько раз. Это признак того, что студент обращался к ИИ-сервису последовательно, итерация за итерацией, реализуя задачу по шагам. Преподаватель не всегда понимает причины такого качества выполненных заданий и пытается отработать так называемые «слабые места» в знаниях студентов, что дает меньше эффекта, если бы он взял под контроль еще и написание промптов.

Таким образом, представляется актуальным выделить особенности существующих наиболее популярных ИИ-сервисов для программирования; задачи, для которых студенты их используют; проблемы, которые у них возникают, чтобы на основе этого корректировать методики преподавания программирования. Пром프트-инжиниринг должен влиться в уже существующие курсы по программированию или стать отдельной дисциплиной для специальностей ИТ-направления.

Постановка задачи. Цель исследования – обоснование необходимости введения пром프트-инжиниринга в учебные программы подготовки бакалавров ИТ-специальностей.

Для ее достижения авторами решаются следующие задачи:

- выделить особенности использования генеративных нейросетевых сервисов в области программирования (на основе анализа научных публикаций и собственного опыта преподавания);
- проанализировать практический опыт использования студентами ИТ-специальностей генеративных инструментов искусственного интеллекта в программировании (провести статистическое исследование).

В исследовании рассматриваются следующие ИИ-сервисы: ChatGPT, DeepSeek, Cursor, OpenAI Codex, Qwen, Gemini, Claude и Perplexity.

Научная новизна состоит в систематизации особенностей использования генеративных нейросетевых сервисов в программировании.

Методика и методология исследования. Среди множества существующих генеративных нейросетевых сервисов наибольшую известность в обучении программированию получили ChatGPT, DeepSeek, Cursor, OpenAI Codex, Qwen, Gemini, Claude и Perplexity. Данный выбор обусловлен массовым применением студентами для генерации и исправления кода, поиска ошибок, объяснения кода, изучения новых технологий, а также для оформления отчетов (табл. 1).

Таблица 1. Особенности использования нейросетевых сервисов в области программирования
 Table 1. Features of using neural network assistants in the field of programming

Сервис	Предназначение	Ограничения, недостатки использования при программировании	Наличие ограничений на использование в России
ChatGPT	Генерация и объяснение кода	Возможны логические ошибки; недостоверные ответы; ограниченность контекста [15]	Да
DeepSeek	Генерация и объяснение кода	Нестабильность качества; ошибки в сложных задачах [15; 16]	Нет
Cursor	Генерация кода	Ошибки автодополнения; зависимость от модели	Нет
OpenAI Codex	Генерация кода	Ограниченная доступность; генерация без внешнего поиска; ошибки интерпретации [17]	Да
Qwen	Генерация и объяснение кода	Снижение точности в специализированных задачах [16]	Нет
Gemini	Генерация и объяснение кода	Ограничения доступа; вариативность качества результатов [16]	Частично
Claude	Генерация и объяснение кода	Ограничения на типы запросов; избыточная формализация [18; 19]	Да
Perplexity	Поиск и обобщение технической информации	Ограниченная применимость к генерации кода; зависимость от источников [20]	Нет

Примечание: составлено авторами на основе анализа публикаций и собственных оценок.

Этот набор сервисов покрывает основные задачи студентов: от написания и проверки кода до настройки сложных алгоритмов и работы в средах разработки. Кроме того, учет географических ограничений (наличие/отсутствие блокировок в России) позволяет оценить практическую применимость в отечественных вузах, где доступность инструментов напрямую влияет на их внедрение в учебные программы. Анализ этих сервисов выявляет как их потенциал для повышения эффективности обучения, так и типичные ограничения – от логических ошибок до контекстных ограничений, что определяет необходимость разработки адаптированных методик преподавания.

Анализ данных таблицы 1 показывает, что генеративные ИИ-сервисы имеют разное назначение в обучении программированию: такие инструменты, как ChatGPT, DeepSeek, Qwen и Gemini, в основном используются для генерации и анализа кода, Cursor и OpenAI Codex помогают в процессе разработки и автоматизируют часть задач, Claude чаще дает более формализованные ответы, а Perplexity используется как инструмент для поиска информации; при этом у всех сервисов есть общие недостатки – они могут допускать ошибки в коде, хуже справляются со сложными задачами и сильно зависят от формулировки запроса, а часть из них (например, ChatGPT, OpenAI Codex, Gemini и Claude) имеет ограничения на использование в России, поэтому их применение требует проверки полученных результатов и осознанного подхода.

Для проведения статистических обследований использовались данные одного опроса. Объем выборки составил 84 студента, обучающихся по направлению «Прикладная информатика» на 1–4 курсах. Распределение студентов по курсам выглядело следующим образом: 1 курс – 10 человек, 2 курс – 15 человек, 3 курс – 18 человек, 4 курс – 39 человек. Опрошенные студенты учатся на бакалавриате 1–4 курсов по специальности «Прикладная информатика» в следующих учебных заведениях: Новосибирский государственный технический университет и Сибирский государственный университет инженерии и биотехнологий.

Авторами использовались следующие основные методы: монографический, системного и статистического анализа, включая анкетирование, экспертные оценки, сводку, группировку, расчет относительных величин.

Анкета содержала 9 вопросов и распространялась в электронном виде через Яндекс Формы в январе 2026 года. Вопросы охватывали такие аспекты, как курс обучения, используемые ИИ-инструменты, частоту применения ИИ для генерации кода, исправление ошибок, объяснение кода, изучение новых технологий, оформление отчетов, проверку ответов ИИ студентами, изменение промпта при некорректных ответах и необходимость обучения промпт-инжинирингу. Полученные данные позволили выявить периодичность и характер использования генеративных ИИ-сервисов в учебной деятельности

студентов и оценить их влияние на образовательный процесс.

Результаты. Результаты нашего исследования указывают на то, что генеративные нейросетевые сервисы прочно вошли в учебный процесс студентов направления «Прикладная информатика». Значительная часть обучающихся применяет эти инструменты регулярно, преимущественно для создания программного кода, устранения ошибок, разъяснения отдельных участков программ и освоения новых технологий. Одновременно степень использования подобных сервисов варьируется в зависимости от учебного курса и сложности поставленных задач.

Для обозначения в исследовании генеративных нейросетевых сервисов далее будет использоваться аббревиатура ГНС.

В данном исследовании была проанализирована специфика взаимодействия студентов с различными ГНС, включая ChatGPT, DeepSeek, Cursor, OpenAI Codex, Qwen, Gemini, Claude и Perplexity. Рассматривались основные направления их применения: создание программного кода с нуля, дополнение и корректировка существующих программных решений, выявление ошибок, объяснение логики работы программ, освоение новых библиотек и технологий, а также оформление отчетной документации (табл. 2).

Таблица 2. Распределение ответов студентов по регулярному использованию ИИ-сервисов, %
 Table 2. Distribution of student responses by regular use of AI services, %

ИИ-сервисы	Задачи					
	Генерация кода с нуля	Дополнение или исправление кода	Поиск и исправление ошибок	Объяснение кода	Изучение новых технологий и библиотек	Оформление отчетов
ChatGPT	23,2	22,0	22,0	12,2	15,9	14,6
DeepSeek	15,9	18,3	20,7	13,4	18,3	9,8
Cursor	12,2	13,4	12,2	7,3	11,0	4,9
OpenAI Codex	9,8	11,0	8,5	7,3	9,8	9,8
Qwen	7,3	8,5	7,3	6,1	11,0	4,9
Gemini	2,4	1,2	1,2	1,2	0,0	1,2
Claude	1,2	0,0	2,4	3,7	1,2	2,4
Perplexity	2,4	0,0	2,4	3,7	1,2	2,4

Примечание: составлено авторами.

При анализе результатов анкетирования следует учитывать, что студентам предлагалось несколько вариантов ответа на каждую задачу: «регулярно», «часто», «редко» и «не использую». Это позволяло не только выявить наиболее востребованные генеративные нейросетевые ассистенты, но и оценить степень вовлеченности студентов в их применение при решении различных учебных задач.

По доле опрошенных студентов, выбравших вариант ответа «регулярно», ChatGPT занимает лидирующую позицию по многим направлениям использования: генерация кода с нуля – 23,2% участников исследования, дополнение и исправление кода – 22,0%, поиск и исправление ошибок – 22,0%. DeepSeek демонстрирует сопоставимые показатели по доле ответов «регулярно», особенно в задачах, связанных с поиском и исправлением ошибок (20,7%), а также изучением новых технологий и библиотек (18,3%). Cursor применяется достаточно часто, однако преимущественно вы-

полняет функции вспомогательного инструмента при разработке и редактировании кода.

Отметим, что в ответах «часто» ChatGPT и DeepSeek показывают практически идентичные значения показателя использования. Так, оба сервиса применяют по 25,6% студентов для генерации кода с нуля и по 26,8% – для дополнения и исправления кода. Это свидетельствует о том, что DeepSeek постепенно выступает как альтернатива ChatGPT в образовательном контексте, учитывая при этом отсутствие ограничений на его использование в России.

В то же время некоторые сервисы используются студентами редко.

Плохие результаты показывают Gemini, Claude и Perplexity при ответах респондентов «регулярно» и «часто». Почти никогда не используются студентами для анализа новой информации Gemini, а Claude и Perplexity вовсе им нужен исключительно для анализа кода и отчетности. Мо-

жем предположить, что это связано с отсутствием осведомленности об их возможностях или ограничениях в решении задач программирования.

Необходимо отметить, что все же ChatGPT и DeepSeek не популярны у всех студентов. Некоторые опрошенные используют их редко.

Исходя из результатов опроса, часто студенты изучают ответы ГНС и уточняют запросы повторно. К примеру, 70,2% респондентов ответили, что постоянно проверяют результаты, которые выводит нейросеть, 42,7% – часто меняют промпт при неправильном ответе, а еще 39,0% опрошенных делают это регулярно. Это может говорить о том, что изменения промпта для студентов – это способ определения качества: в случае ошибок они меняют запрос, меняют структуру. Даже несмотря на то, что 29,8% респондентов не проверяют ответы ГНС вообще, это всё равно свидетельствует об уверенности в правильном ответе нейросети при таком решении.

Следующим рассмотрим анкетный вопрос: следует ли обучать студентов промпт-инжинирингу. Как показало исследование, 82,9% респондентов указали, что стоит, а 17,1% ответили отрицательно. Из этого следует, что студенты осознают, результат нейросетей зависит от того, как сформулировать запрос.

В результате, введение генеративных нейросетевых сервисов в образовательный процесс может положительно повлиять на ускорение работы и увеличить доступность информации, но и также принести некоторые педагогические риски. К таким относятся: снижение самостоятельности студента, зависимость от генерации кода и повышение вероятности плагиата. Эта ситуация обостряется, когда студенты копируют готовые фрагменты кода без анализа. В этом случае особенно актуальными становятся методы организации и контроля самостоятельной работы студентов и развития их мышления.

Рекомендации, основанные на исследовании, предполагают использование различных ГНС в зависимости от характера учебной деятельности. Для генерации исходного кода, отладки и поиска решений при выполнении лабораторных работ желательно применять ChatGPT и DeepSeek, поскольку именно эти сервисы получили широкое распространение и продемонстрировали высокую активность использования среди студентов. ChatGPT рекомендуется для написания и интерпретацию кода, в то время как DeepSeek особенно полезен для поиска ошибок и получения новых знаний.

В настоящее время российские нейросетевые сервисы остаются менее востребованными среди студентов, но с развитием технологий искусственного интеллекта в России эта тенденция, вероятно, изменится.

Сервис Cursor стоит использовать в первую очередь как вспомогательный инструмент внутри

среды разработки, особенно при редактировании, дополнении и рефакторинге существующего кода. На наш взгляд, он может быть полезен в контексте проектной деятельности, работы с крупными программными проектами и курсовыми работами. OpenAI Codex позволяет генерировать отдельные программные решения и прототипы, однако его применение требует дополнительной верификации результатов ввиду ограничений по сложности задач и вероятности синтаксических ошибок.

Сервис Qwen можно рассматривать как дополнительный ресурс для генерации и объяснения кода, а также для знакомства с новыми библиотеками и технологиями. Claude и Perplexity целесообразнее применять не для создания кода, а для поиска информации, разбора сложных тем, подготовки теории и оформления отчетов. Gemini студенты почти не используют, так что мы его не включили в рекомендуемые инструменты.

Исходя из полученных результатов, в программы бакалавриата по направлениям «Прикладная информатика» и смежным, например, «Информатика и вычислительная техника», рекомендуется включать основы промпт-инжиниринга. Хотя многие студенты меняют запросы для достижения корректных ответов, делают они это преимущественно интуитивно и без глубокого понимания принципов построения эффективных промптов. Отсутствие этих навыков может приводить к генерации некорректного кода, логическим ошибкам и снижению качества самостоятельной работы.

Целесообразно вводить элементы промпт-инжиниринга в рамках дисциплин, связанных с программированием, анализом данных, разработкой программного обеспечения и цифровыми технологиями. В рамках обучения следует развивать навыки формулирования точных запросов, разделения сложных задач, прояснения контекста, а также проверки и критического анализа ответов нейронной сети, включая аспекты академической этики при работе с ГНС.

Как показывает опрос, 82,9% студентов поддерживают идею учить промпт-инжиниринг. Он позволит обновить программы по прикладной информатике и другим ИТ-направлениям, и соответствовать запросам, появившимся в цифровой среде. Это поможет эффективнее использовать нейросети, а главное – уменьшит риски плагиата, слепого доверия к их ответам и зависимости от готовых решений.

Выводы. Генеративные нейросетевые сервисы играют важную роль в обучении студентов, в образовании студентов направлению «Прикладная информатика». Они способны помочь в генерации кода, обнаружению и исправлению ошибок, разъяснению основных принципов программирования, а также помогают осваивать новые технологии. В основном студенты используют ChatGPT и DeepSeek,

но для решения некоторых задач используются и другие сервисы. Проведённые исследования показывают, что использование этих инструментов действительно может ускорить выполнение учебных задач и улучшить понимание программирования.

Распространение подобных технологий связано с рисками. Существует риск излишней уверенности в готовых ответах. Это может привести к снижению способности самостоятельной работы, а также к копированию кода без доработки и понимания. Следующая серьёзная проблема – академический плагиат. Особенно в отчетах, где материалы, сгенерированные нейронными сервисами, используются без тщательной проверки. Для того, чтобы снизить описанные выше риски, в вузах необходимо ввести новые методы обучения, такие, в которых искусственный интеллект выступает в качестве помощника, а не заменяет усилия студентов.

Выполненные исследования демонстрируют, что успех во внедрении генеративных нейросете-

вых сервисов напрямую зависит от уровня цифровой грамотности. Значительная часть опрошенных студентов уже на регулярном уровне использует элементы промпт-инжиниринга, при этом уточняя и видоизменяя вопросы при получении искажённых ответов. Параллельно большинство студентов отмечает необходимость включения основ промпт-инжиниринга в учебные планы. Это свидетельствует о перспективности интеграции обучения работе с нейросетевыми сервисами в процесс обновления бакалаврской подготовки по направлению «Прикладная информатика».

Можно сделать вывод, что представленные результаты исследования подтверждают потенциал генеративных нейросетевых сервисов в повышении качества и эффективности обучения программированию. Вместе с этим их применение необходимо сочетать с развитием умений критически оценивать, проверять и контролировать результаты.

СПИСОК ЛИТЕРАТУРЫ

1. Баженова И. В. Особенности методики обучения программированию на основе проективно-рекурсивной стратегии и когнитивных технологий // Педагогическое образование в России. 2015. №3. С. 52–57.
2. Алферьева-Термсинос В. Б. Промпт-инжиниринг как стратегия формирования информационной культуры обучающихся // Международный журнал гуманитарных и естественных наук. 2024. №9–1 (96). С. 10–15.
3. Vukojicic M., Krstic J. ChatGPT in programming education: ChatGPT as a programming assistant // InspirED Teachers' Voice. 2023. Т. 2023. №1. Р. 7–13.
4. Chen E. et al. GPTutor: a ChatGPT-powered programming tool for code explanation // International conference on artificial intelligence in education. Cham: Springer Nature Switzerland. 2023. P. 321–327.
5. Allal L. B., Li R., Kocetkov D., Mou C., Akiki C., Ferrandis C. M., Muennighoff N., Mishra M., Gu A., Dey M. et al. Don't Reach for the Stars // Proceedings of the 17th International Conference on Software Engineering for Adaptive and Self-Managing Systems. 2023. P. 1–12.
6. Barke S., James M. B., Polikarpova N. Grounded copilot: How programmers interact with code-generating models // Proceedings of the ACM on Programming Languages. 2023. Т. 7. №. OOPSLA1. P. 85–111.
7. Ziegler, A., Kalliamvakou, E., Li, X., Rice, A., Rifkin, D., Simister, S., Sittampalam, G., Nenadic, A. Productivity, Efficiency, and Safety in AI-Assisted Software Development: An Empirical Study of GitHub Copilot // arXiv preprint arXiv:2206.15075. 2022. P. 1–32.
8. Dakhel A. M. et al. Github copilot ai pair programmer: Asset or liability? // Journal of Systems and Software. 2023. Т. 203. P. 111–734.
9. Ye X. W. et al. Copyright Protection and Risk Mitigation for AI-Generated Content (AIGC) Creations // Journal of Computers. 2025. Т. 36. №. 5. P. 167–181.
10. Shi Y. Study on security risks and legal regulations of generative artificial intelligence // Science of law journal. 2023. Т. 2. №. 11. P. 17–23.
11. Atkinson D., Morrison J. A legal risk taxonomy for generative artificial intelligence // arXiv preprint arXiv:2404.09479. 2024. P. 1–25.
12. George A. S. Preparing students for an AI-driven world: Rethinking curriculum and pedagogy in the age of artificial intelligence // Partners Universal Innovative Research Publication. 2023. Т. 1. №. 2. P. 112–136.
13. Baidoo-Anu D., Ansah L. O. Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning // Journal of AI. 2023. Т. 7. №. 1. P. 52–62.
14. Hellas A. et al. Exploring the responses of large language models to beginner programmers' help requests // Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1. 2023. P. 93–105.
15. Uppalapati V. K., Nag D. S. A comparative analysis of AI models in complex medical decision-making scenarios: evaluating ChatGPT, Claude AI, Bard, and Perplexity // Cureus. 2024. Т. 16. №. 1.
16. Aydin O. et al. Generative ai in academic writing: A comparison of deepseek, qwen, chatgpt, gemini, llama, mistral, and gemma // arXiv preprint arXiv:2503.04765. 2025. P. 1–21.

17. Singh A. et al. Openai gpt-5 system card // arXiv preprint arXiv:2601.03267. 2025. P.1–19.
18. Martins S. Artificial intelligence-assisted classification of library resources: The case of Claude AI // Artificial Intelligence. 2024. Т. 2. P. 27.
19. Baninemeh E. et al. A security risk assessment method for Distributed Ledger Technology (DLT) based applications: three industry case studies // arXiv preprint arXiv:2401.12358. 2024. P. 1–39.
20. Xu Z., Sheng V.S. Detecting AI-generated code assignments using perplexity of large language models // Proceedings of the aaai conference on artificial intelligence. 2024. Т. 38. №. 21. P. 23155–23162.

REFERENCES

1. Bazhenova I. V. Features of the methodology of teaching programming based on projective-recursive strategy and cognitive technologies. *Pedagogicheskoye obrazovaniye v Rossii*, 2015, no. 3, pp. 52–57. (In Russ.).
2. Alferyeva-Termisikov V. B. Prompt engineering as a strategy for developing students' information culture. *Mezhdunarodnyy zhurnal gumanitarnykh i yestestvennykh nauk*, 2024, no. 9–1 (96), pp. 10–15. (In Russ.).
3. Vukojcic M., Krstic J. ChatGPT in programming education: ChatGPT as a programming assistant. *InspirED Teachers Voice*, 2023, vol. 2023. no. 1. pp. 7–13.
4. Chen E. et al. GPTutor: a ChatGPT-powered programming tool for code explanation. *International conference on artificial intelligence in education*, Cham: Springer Nature Switzerland, 2023, pp. 321–327.
5. Allal L. B., Li R., Kocetkov D., Mou C., Akiki C., Ferrandis C. M., Muennighoff N., Mishra M., Gu A., Dey M. et al. SantaCoder: Don't Reach for the Stars. *Proceedings of the 17th International Conference on Software Engineering for Adaptive and Self-Managing Systems*, 2023, pp.1–12.
6. Barke S., James M. B., Polikarpova N. Grounded copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages*, 2023, vol. 7, no. OOPSLA1, pp. 85–111.
7. Ziegler, A., Kalliamvakou, E., Li, X., Rice, A., Rifkin, D., Simister, S., Sittampalam, G., Nenadic, A. Productivity, Efficiency, and Safety in AI-Assisted Software Development: An Empirical Study of GitHub Copilot. *arXiv preprint arXiv:2206.15075*, 2022, pp. 1–32.
8. Dakhel A. M. et al. Github copilot ai pair programmer: Asset or liability. *Journal of Systems and Software*, 2023, vol. 203, pp. 111–734.
9. Ye X. W. et al. Copyright Protection and Risk Mitigation for AI-Generated Content (AIGC) Creations. *Journal of Computers*, 2025, vol. 36, no. 5, pp. 167–181.
10. Shi Y. Study on security risks and legal regulations of generative artificial intelligence. *Science of law journal*, 2023, vol. 2, no. 11, pp. 17–23.
11. Atkinson D., Morrison J. A legal risk taxonomy for generative artificial intelligence. *arXiv preprint arXiv:2404.09479*, 2024, pp. 1–25.
12. George A. S. Preparing students for an AI-driven world: Rethinking curriculum and pedagogy in the age of artificial intelligence. *Partners Universal Innovative Research Publication*, 2023, vol. 1, no. 2, pp. 112–136.
13. Baidoo-Anu D., Ansah L. O. Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *Journal of AI*, 2023, vol. 7, no. 1, pp. 52–62.
14. Hellas A. et al. Exploring the responses of large language models to beginner programmers' help requests. *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*, 2023, pp. 93–105.
15. Uppalapati V. K., Nag D. S. A comparative analysis of AI models in complex medical decision-making scenarios: evaluating ChatGPT, Claude AI, Bard, and Perplexity. *Cureus*, 2024, vol. 16, no. 1.
16. Aydin O. et al. Generative ai in academic writing: A comparison of deepseek, qwen, chatgpt, gemini, llama, mistral, and gemma. *arXiv preprint arXiv:2503.04765*, 2025, pp. 1–21.
17. Singh A. et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025, pp.1–19.
18. Martins S. Artificial intelligence-assisted classification of library resources: The case of Claude AI. *Artificial Intelligence*, 2024, vol. 2, pp. 27.
19. Baninemeh E. et al. A security risk assessment method for Distributed Ledger Technology (DLT) based applications: three industry case studies. *arXiv preprint arXiv:2401.12358*, 2024, pp. 1–39.
20. Xu Z., Sheng V. S. Detecting AI-generated code assignments using perplexity of large language models. *Proceedings of the aaai conference on artificial intelligence*, 2024, vol. 38, no. 21, pp. 23155–23162.

Информация об авторах

Алетдинова Анна Александровна – доктор экономических наук, профессор кафедры информационных технологий и моделирования, Сибирский государственный университет инженерии и биотехнологий (Российская Федерация, 630039, г. Новосибирск, ул. Никитина, 155, e-mail: kaf-aoi418@mail.ru);

Петров Даниил Дмитриевич – студент кафедры автоматизированных систем управления, Новосибирский государственный технический университет (Российская Федерация, 630073, г. Новосибирск, просп. Карла Маркса, 20, e-mail: почта кафедры kaf_asu@corp.nstu.ru);

Фролов Марк Евгеньевич – студент кафедры автоматизированных систем управления, Новосибирский государственный технический университет (Российская Федерация, 630073, г. Новосибирск, просп. Карла Маркса, 20, e-mail: kaf_asu@corp.nstu.ru).

Статья поступила в редакцию 02.04.2026

После доработки 16.04.2026

Принята к публикации 17.04.2026

Information about the authors

Anna A. Aletdinova – doctor of economical sciences, professor at the department of information technology and modeling, Siberian State University Engineering and Biotechnology (155 Nikitina Str., Novosibirsk, 630039, Russian Federation, e-mail: kaf-aoi418@mail.ru);

Daniil D. Petrov – student of the department of automated control systems, Novosibirsk State Technical University (20 Karl Marks ave., Novosibirsk, 630073, Russian Federation, e-mail: kaf_asu@corp.nstu.ru);

Mark E. Frolov – student of the department of automated control systems, Novosibirsk State Technical University (20 Karl Marks ave., Novosibirsk, 630073, Russian Federation, e-mail: kaf_asu@corp.nstu.ru).

The paper was submitted 02.04.2026

Received after reworking 16.04.2026

Accepted for publication 17.04.2026