



НАПРАВЛЕНИЯ ИСПОЛЬЗОВАНИЯ ГЕНЕРАТИВНОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ НАЧАЛЬНОМ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ В УНИВЕРСИТЕТАХ

Долинский М. С.¹, канд. техн. наук, доцент, ✉ dolinsky@gsu.by, 0000-0002-3057-4051

¹Гомельский государственный университет им. Ф. Скорины,
ул. Кирова, д. 119, 246019, Гомель, Республика Беларусь

Аннотация

В статье приводится обзор литературы по использованию генеративного искусственного интеллекта (ГенИИ) при начальном обучении программированию в вузах.

Приведены основные направления применения ГенИИ: специализированные узко-темные разработки, встраивание в онлайн платформы обучения и проверки решений, работа студентов с ГенИИ без ограничений, направляющие системы взаимодействия студентов с ГенИИ (без предоставления решений), помощь преподавателю, инструменты для разработки интеллектуальных обучающих систем.

Также содержится обзор работ, анализирующих достигнутые результаты и нерешённые проблемы.

Ключевые слова: генеративный искусственный интеллект, обучение программированию первокурсников, интеллектуальные обучающие системы.

Цитирование: Долинский М. С. Направления использования генеративного искусственного интеллекта при начальном обучении программированию в университетах // Компьютерные инструменты в образовании. 2024. № 2. С. 85–96 doi:10.32603/2071-2340-2024-2-85-96

1. ВВЕДЕНИЕ

Цель начального обучения программированию в университете — формирование навыков “Computational Thinking” [1], включающих следующие компоненты: абстракция, декомпозиция, распознавание паттернов, алгоритмизация, отладка.

Существенное повышение эффективности обучения приносит использование онлайн платформ для обучения программированию, таких, например, как EduCoder [2, 3], HTProgramming [4], Code4brownies [5].

Новый этап повышения эффективности начального обучения программированию связан с внедрением генеративного искусственного интеллекта. Текстовые генеративные системы имеют чат-боты в качестве фронтенда, чтобы взаимодействовать с пользователем, и основываются на LLM — генеративных моделях, которые могут производить новый контент, основываясь на данных, на которых они натренированы. Наиболее известные LLM: GPT (OpenAI), Gemini (Google), Llama (Meta), Claude3

(Anthropic). Взаимодействие пользователя с чат-ботом реализуется запросами. В работе [6] сформулированы задачи для таких чат-ботов во время процесса начального обучения программированию: объяснение базовых знаний, конструирование кода, объяснение кода, рефакторинг кода, форматирование кода, проверка стиля кодирования, комментирование кода. Стандартные практики использования ГенИИ при обучении программированию представлены в работе [7].

Проектирование и формулирование запросов для того, чтобы получить оптимальный ответ, часто называется промпт-инжинирингом. Отличная иллюстрация, как с помощью промпт-инжиниринга получать фантастические результаты, представлена в работе [8], посвящённой контекстному дообучению ChatGPT в области биоинформатики. Представленный в данной работе GeneGPT использует NCBI Web API документацию и демонстрацию в запросе для обучения контексту, а затем отвечает на тесты GeneTuring и GeneHop с помощью NCBI Web API.

Для полноты картины надо иметь в виду и мнение скептиков, считающих, что на начальном этапе обучения программированию необходим полный запрет использования студентами ГенИИ [9].

2. ИНСТРУМЕНТАЛЬНЫЕ СИСТЕМЫ, ИСПОЛЬЗУЮЩИЕ ГЕНИИ-КОДОГЕНЕРАТОРЫ ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ СТУДЕНТОВ

2.1. Первые шаги

В работе [10] представлен продукт, помогающий писать Python-программы, отформатированные по стандарту PEP-8. MMAPR [11] позволяет автоматически исправлять ошибки в программах, написанных студентами на Python. Codeboard [12] помогает студентам в диалоге находить и исправлять ошибки в предложенных JavaScript-программах. Работа [13] представляет продукт, обеспечивающий автоматический контроль соблюдения правил для Java-программ, размещённых студентами на Github. В работе [14] описывается программа, которая помогает студентам в диалоге выполнять задания на SQL-запросы к PostgreSQL. В [15, 16] представлены разработки, которые помогают студентам в диалоге выполнять задания по программированию, отвечая на их вопросы и предлагая поясняющие примеры.

2.2. Платформы, позволяющие студентам прозрачно взаимодействовать с ChatGPT

Более эффективным представляется встраивание ГенИИ в онлайн-системы обучения и тестирования программ. Этот подход предлагают следующие системы.

TutorBot+ [17, 18] интегрирован в WEB-систему обучения и онлайн-платформу тестирования программ. TutorBot+ позволяет брать условия задач, получать решения в диалоге с ГенИИ, проверять решения.

KOGI [19] — система поддержки обучения, которая интегрирует ChatGPT и среду Jupyter. KOGI помогает студенту получить совет от ChatGPT в ответ на ошибки и вопросы. KOGI внедрена в два вводных курса: Алгоритмы, Data Science. Как результат — существенное уменьшение количества студенческих ошибок и получение преподавателями информации о вопросах и ошибках.

IPSSC [20] (Intelligent programming scaffolding system using ChatGPT). Вместо того чтобы студент взаимодействовал непосредственно с ChatGPT, авторы разработали три модуля:

Solution Assessment (SA) — оценивание решений, Code Assessment (CA) — оценивание кода, Free Interaction (FI) — свободное взаимодействие. В модуле SA студенты декомпозируют сложную задачу на несколько простых и абстрагируют их, чтобы решение подходило для похожих задач (таким образом, этот модуль фокусируется на декомпозиции и абстракции). В модуле CA студенты разрабатывают алгоритмы, пишут и непрерывно отлаживают код, используя модуль для идентификации ошибок в коде, затем их исправляют. ChatGPT помогает улучшить алгоритмы и код. В модуле FI студенты могут непосредственно взаимодействовать с ChatGPT, обсуждая темы, которые не покрыты в модулях SA и CA.

Coding Step [11, 21] — это web-приложение, предназначенное для обучения основам программирования на Python. Это дружественная к начинающему программисту среда программирования с серией задач, вводящих новые концепции. Ученик может двигаться по этим задачам независимо, имея доступ к ГенИИ-кодогенератору от OpenAI Codex. Пользователи вводят текст описания поведения программы на естественном языке и по клику на кнопку «Сгенерировать код» получают программу, сгенерированную OpenAI Codex. Каждое обращение к API-вызову Codex кастомизируется добавлением определённых примеров, имеющегося кода в редакторе и затребованного пользователем поведения. Это обеспечивает то, что ГенИИ-кодогенератор сгенерирует код на Python для начинающего программиста, который соответствует контексту пользователя. Студенты работают со сгенерированным кодом. При таком подходе уменьшается количество ошибок и сокращается время на выполнение задания.

LearningProgrammingWithGPT [22] — среда, которая используется студентами в курсе CS1 при изучении Python. Курс интегрирует ChatGPT как средство поддержки студентов и инструкторов в учебном классе. Платформа действует как посредник между студентами и их ГенИИ-инструктором, выдаёт персонализированный обучающий материал, основанный на препромптах, встроенных в студенческие запросы. Собственно программа обучения состоит из трёх частей:

1. Введение — Переменные — Операторы — Ввод и Вывод.
2. Комментарии — Условия — Массивы (Списки) — Ошибки и отладка.
3. Циклы — Функции — Библиотека.

GPTutor [23] — это расширение для VS Code, поддерживает более 50 языков программирования, включая C, C++, Java, Python, C#. GPTutor позволяет пользователям настраивать подсказки для различных языков программирования и сценариев. Он предоставляет объяснение кода, рецензии и настраиваемые подсказки. В курс по Python (8 недель) входят задания на неделю по теме недели (можно использовать ChatGPT); домашняя работа (можно обсудить с ChatGPT сложные моменты); индивидуальный мини-проект в последнюю неделю — уникальная программа менее 1000 строк кода на любую тему от игр до чат-ботов, можно использовать ChatGPT, но включить в отчёт диалог ChatGPT. ChatGPT можно использовать не только для отладки, объяснения и оптимизации, но также для мозгового штурма идей, уточнения требований проекта, более глубокого понимания технических аспектов проекта.

Copilot [24] — это ГенИИ-помощник, который использует LLM, натренированную на миллиардах строк кода. Copilot доступен как бесплатный IDE плагин для студентов. Copilot может сгенерировать корректный и читабельный код на общих языках программирования.

В работе [25] дополнительно обеспечивается проверка на плагиат.

2.3. Помощь без представления решений

В рамках конструктивного ответа скептикам внедрения ГенИИ в начальное обучение программированию разрабатываются продукты, которые контролируют и модифицируют запросы студентов и ответы ГенИИ таким образом, чтобы предоставлять помощь без представления решений.

Разработчики CodeHelp [26, 27] утверждают, что преподаватели программирования имеют огромные проблемы с возможностью уделить достаточно времени каждому студенту. LLM-модели помогают студентам решать их проблемы, но есть опасности двух видов: решение задач вместо студентов, неправильное решение задач. CodeHelp обеспечивает студентам помощь по требованию, не предлагая непосредственных решений, являясь примером «сократического» наставника, избегающего раскрытия решений непосредственно пользователю. CodeHelp поддерживает любой язык программирования, поскольку обращается к ГенИИ. В то же время авторы подчёркивают, что CodeHelp является не заменой учителя-человека, а дополнением к нему. Студент в любой момент времени может попросить помощь и немедленно получает ответ — советы без предоставления полного решения, которые получаются запросами к ИИ на основе студенческого ввода. Кроме того, преподаватель имеет возможность отслеживать работу студентов. CodeHelp спроектирован так, чтобы отвечать на любые вопросы студентов, не генерируя кода решения вне зависимости от формулировки запроса. Это достигается серией различных запросов, которые включают инструкции не генерировать код и отдельно верифицируют, что код отсутствует в ответе от LLM. Цель — обеспечить получение студентами пошаговой помощи как смеси естественного языка и псевдокода, после чего студенты должны самостоятельно написать корректный код. CodeHelp также использует промпты, чтобы сгенерировать ответы «положительные и одобряющие», как это делает учитель-человек. Разработчики CodeHelp подчёркивают три основные характеристики своего цифрового помощника учителя: обучать, а не давать готовые решения, вести эффективный диалог, учитывать уровень подготовки студента.

CodeAid [28] также избегает прямого ответа с кодом, стараясь направлять студента к решению задачи. CodeAid позволяет студентам задать общий вопрос по программированию, задать вопрос о представленном коде, попросить объяснить представленный код, помочь исправить ошибку в представленном коде, помочь написать код. За семестр тестового использования были собраны следующие данные, которые позволили улучшить CodeAid: более 8000 взаимодействий студентов с CodeAid с их оценкой сгенерированного ответа, 10 еженедельных отчётов студентов об использовании CodeAid и других ресурсов, 22 интервью студентов о CodeAid, финальный анонимный обзор сравнения CodeAid с ChatGPT.

Разработчики NotebookGPT [29] считают, что прямое использование ChatGPT может помешать обучению студента, поэтому NotebookGPT даёт доступ к GPT, но не возвращает полных решений, обеспечивая обратную связь на стиль программирования, пояснения, как работают куски кода, помощь с отладкой кода, возможность увидеть альтернативные решения задач.

AI TA [30] помогает студентам выполнять декомпозицию, то есть разбиение задач на подзадачи. По мнению разработчиков AI TA, такие модели как ChatGPT, LLaMA, Gemini в контексте обучения программированию показывают замечательные способности понимать, генерировать и объяснять код, делая их сильными кандидатами на Teacher Assistants (помощники учителя) в курсах обучения программированию. Они могут исправлять ошибки в коде, объяснять, почему ошибки случились, и обсуждать возможные

подходы к решению различных программистских задач. Однако есть и проблемы — они могут генерировать ошибочный код и даже различные ответы на один и тот же вопрос. Поэтому AI TA поддерживает “Subgoal learning” — эффективную стратегию обучения, помогающую студентам разбивать сложные задачи на подзадачи.

StAP-tutor [31] обеспечивает рекомендацию следующего шага решения задачи. Эксперименты показали, что в большинстве случаев StAP-tutor рекомендует правильный шаг, и он персонализирован к студенческому коду и подходу. Однако бывают и случаи ошибочных советов.

TeachYou+AlgoBo [32, 33] предлагает подход LBT (Learning By Teaching — обучение себя обучением кого-то). Здесь TeachYou — LBT-среда для обучения алгоритмам, а AlgoBo — чат-бот, обучающийся решению задач. Вместо того чтобы конструировать обучаемых агентов «с нуля», авторы предлагают методологию «сверху вниз», в которой они используют разные LLM, чтобы симулировать учеников. В своём формирующем исследовании авторы попросили 15 новичков в программировании выполнить LBT с ChatGPT, который попросили промптом выполнять роль ученика.

Авторы пришли к следующим выводам о необходимости:

1. Ограничивать уровень знаний LLM-агента, то есть программы, которая выступает в роли ученика, для того чтобы процесс был как можно ближе к реальности, чтобы его учить было не легче, чем среднестатистического студента.
2. Предписать LLM-агенту задавать вопросы «почему» и «как», как это и должен делать хороший ученик.
3. Диалоги LLM-агента и студентов должны сохраняться для последующего анализа с целью получения обратной связи на применённые методы преподавания и обучения.

Как результат, авторский подход предполагает, что в TeachYou ученик решает задачу по программированию алгоритмов (например, двоичный поиск) и затем рефлексивует эту работу обучением AlgoBo. Слово «рефлексивует» здесь означает, что студент должен снова вспомнить и гораздо лучше понять изученное, поскольку он вынужден объяснять эту тему LLM-агенту, возможно, не один раз и разными способами, чтобы в конце концов LLM-агент смог научиться сам решать подобные задачи.

В работе [9] предлагается разработка продукта, спрашивающего у ГенИИ: «Спроси у меня, что тебе нужно знать, чтобы решить задачу».

2.4. Платформы для разработки интеллектуальных обучающих систем

Дальнейшее повышение эффективности связывается с разработкой платформ создания ГенИИ-тьюторов [34].

VoLL-KI [35] — это исследовательский проект, цель которого создать набор инструментов на различных уровнях абстракции для улучшения с помощью искусственного интеллекта процесса обучения в университетах. Имеются ввиду такие инструменты как планирование обучения, рекомендательные системы для служб ответов на вопросы студентов и выбора учебных материалов, обучение с помощью виртуальной реальности, образовательные чат боты, адаптивная обучающая среда.

Практическая реализация платформы Apprentice Tutor Builder (ATB) для создания интеллектуальных обучающих систем (ИОС) описывается в работе [36]. ИОС — это образовательная технология, которая обеспечивает студентов содержанием курса, индивидуально-предлагаемыми практическими заданиями и эффективной обратной

связью. По мнению авторов АТВ, ИОС очень эффективны, однако их разработка сложна, времязатратна, требует специальных программистских и преподавательских знаний, что сдерживает их широкое распространение. Поэтому авторы и разрабатывают АТВ — платформу, которая упрощает создание и персонализацию ИОС-тьюторов. Преподаватель может, используя технологию “drag-and-drop”, создать интерфейс тьютора, а затем интерактивно обучить ГенИИ-агента решать задачи. Обучение достигается многократным интерактивным взаимодействием, включая демонстрации, обратную связь и пользовательские метки. Как результат — требуется потратить 200–300 часов на один час обучения. 14 инструкторов оценили эффективность использования АТВ, но указали на желаемые возможности для сокращения времени на разработку. В результате анализа этой тестовой работы авторы создали множество рекомендаций для использования платформы АТВ и ГенИИ-агента.

2.5. Помощь преподавателю

Отдельное направление применения генеративного ИИ — помощь преподавателям. Здесь можно отметить такие продукты как Robosourcing [11] для генерации практических заданий в масштабируемой манере, VizGroup [37], обеспечивающий контроль за деятельностью студентов в больших классах (100+ студентов), проверка решений студентов с помощью ChatGPT [38], PAIGE [39] — система обнаружения обманов, опознающая тексты, написанные генеративным ИИ, а не человеком.

2.6. Рефлексирующие работы

В работе [40] предлагаются способы использования ChatGPT для самообучения. Работы [41–43] описывают достоинства и недостатки использования ГенИИ для обучения программированию. В работе [44] излагаются результаты опроса студентов о пользе и проблемах использования ГенИИ для обучения программированию. Сбор и анализ протоколов чатов студентов во вводном курсе программирования предлагается в работе [45]. Технология тестирования способностей ChatGPT отвечать на вопросы по программированию предложена в работе [46]. Работа [47] предлагает опыт «Как учить решать задачи по примерам». Работа [48] предлагает решать проблемы прогресса искусственного интеллекта в начальном обучении программированию, используя подходы, предложенные в обучении математике после изобретения калькуляторов и компьютеров.

3. ЗАКЛЮЧЕНИЕ

Резюмируя описанные в статье возможности ГенИИ, можно разделить функционально на две группы:

1. Как можно использовать ГенИИ студентам: автоматическая коррекция синтаксиса, выполнение заданий, получение нескольких решений одной задачи, получение пояснений к решениям и сообщения об ошибках, в перспективе получить виртуального учителя.
2. Как можно использовать ГенИИ преподавателям: генерация заданий по программированию, генерация новых упражнений по имеющимся, контекстуализация условий задач, оценка студенческих работ.

Варианты будущего обучения программированию с ГенИИ-кодогенерацией:

В идеале — изменить задания так, чтобы можно было пользоваться кодогенерацией и при этом получать собственные навыки решения задач.

Радикалы предлагают сделать сдвиг фокуса в программистских курсах. ГенИИ-кодогенераторы могут решать низкоуровневые задачи реализации, а студенты будут разрабатывать высокоуровневые алгоритмы, поскольку в будущем разработчики программного обеспечения будут более нацелены на проектно-ориентированные роли, поэтому фокус курсов должен измениться на разработку запросов к AI-кодогенераторам, оценку кода и отладку.

Рационалисты предлагают начинающих учить традиционными подходами, а ИИ-кодогенерации учить на последующих курсах.

Список литературы

1. *Kaleem M., Hassa M. A.n, Khurshid S. K.* A Machine Learning-Based Adaptive Feedback System to Enhance Programming Skill Using Computational Thinking // *IEEE Access*. 2024. Vol. 12. P. 59431–59440. doi:10.1109/access.2024.3391873
2. *Zhang S., Yang J., Sang X.* Exploring the Applications of EduCoder Platform in Blended Teaching for Computer Major // *Journal of Education and Educational Research*. 2023. Vol. 4, № 2. P. 100–103. doi:10.54097/jjeer.v4i2.10819
3. *Li Z., Zhang S., Sang X.* Exploration of Machine Learning Teaching based on the EduCoder Platform // *Journal of Education and Educational Research*. 2023. Vol. 4, № 3. P. 130–133. doi:10.54097/jjeer.v4i3.11387
4. *Figueiredo J., Garcia-Penalvo F.* Teaching and Learning Tools for Introductory Programming in University Courses // 2021 International Symposium on Computers in Education (SIIE). 2021. doi:10.1109/siie53363.2021.9583623
5. *Phan V., Hicks E.* Code4Brownies: an active learning solution for teaching programming and problem solving in the classroom // *Proc. of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. 2018. P. 153–158. doi:10.1145/3197091.3197128
6. *Herden O.* Integration of Chatbots for Generating Code Into Introductory Programming Courses // *Proc. of International Conference “Future of education*. 2024. URL: <https://conference.pixel-online.net/files/foe/ed0014/FP/9091-ICT6673-FP-FOE14.pdf> (date: 26.06.2024).
7. *Köppe Ch., Keuning H., Lykourantzou I., Chacon I. A., Sosnovsky S.* Practices for the Application of Generative AI in Programming Education. Research project // <https://www.uu.nl>. 2024. URL: <https://www.uu.nl/sites/default/files/GenAI%20in%20ProgEd%20practices.pdf> (date: 26.06.2024).
8. *Jin Q., Yang Y., Chen Q., Lu Z.* GeneGPT: augmenting large language models with domain tools for improved access to biomedical information // *Bioinformatics*. 2024. Vol. 40, № 2. P. btac075. doi:10.1093/bioinformatics/btac075
9. *Lau S., Guo P.* From ‘Ban It Till We Understand It’ to ‘Resistance is Futile’: How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot // *Proc. of the 2023 ACM Conf. on Int. Computing Education Research*. 2023. Vol. 1. P. 106–121. doi:10.1145/3568813.3600138
10. *Farah J. C., Spaenlehauer B., Ingram S., Purohit A. K., Holzer A., Gillet D.* Harnessing Rule-Based Chatbots to Support Teaching Python Programming Best Practices // *Lecture Notes in Networks and Systems*. 2024. P. 455–466. doi:10.1007/978-3-031-51979-6_47
11. *Cambaz D.* Use of AI-driven code generation models in teaching and learning programming: a systematic literature review. Bachelor thesis, TU Delft, Faculty Electrical Engineering, Mathematics and Computer Science, Delft, The Netherlands, 2023.
12. *Roca M. D. L., Chan M. M., Garcia-Cabot A., Garcia-Lopez E., Amado-Salvatierra H.* The impact of a chatbot working as an assistant in a course for supporting student learning and engagement // *Computer Applications in Engineering Education*. 2024. P. e22750. doi:10.1002/cae.22750

13. Bobadilla S., Glassey R., Bergel A., Monperrus M. SOBO: A Feedback Bot to Nudge Code Quality in Programming Courses // arxiv.org. 2023. URL: <https://arxiv.org/pdf/2303.07187> (date: 26.06.2024).
14. Sengewald J., Wilz M., Lackes R. AI-Assisted Learning Feedback: Should Gen-AI Feedback Be Restricted to Improve Learning Success? A Pilot Study in a SQL Lecture // Proc. of ECIS 2024. 2024. P. 12.
15. Pahi K., Hawlader S., Hicks E., Zaman A., Phan V. Enhancing active learning through collaboration between human teachers and generative AI // Computers and Education Open. 2024. Vol. 6. P. 100183. doi:10.1016/j.caeo.2024.100183
16. Roldán-Álvarez D., Mesa F. J. Intelligent Deep-Learning Tutoring System to Assist Instructors in Programming Courses // IEEE Transactions on Education. 2024. Vol. 67, № 1. P. 153–161. doi:10.1109/te.2023.3331055
17. Martínez-Araneda C., Gutiérrez M., Maldonado D., Gómez P., Segura A., Vidal-Castro C. Designing a Chabot to support problem-solving in a programming course // Proc. of 18th International Technology, Education and Development Conference. 2024. P. 966–975. doi:10.21125/inted.2024.0317
18. Martínez-Araneda C., Gutiérrez Valenzuela M., Gómez Meneses P., Maldonado Montiel D., Segura Navarrete A., Vidal-Castro C. How Useful TutorBot+ is for Teaching and Learning in Programming Courses: a Preliminary Study // Proc. of 42nd IEEE International Conference of the Chilean Computer Science Society (SCCC). 2023. P. 1–7. doi:10.1109/sccc59417.2023.10315697
19. Kuramitsu K., Obara Y., Sato M., Obara M. KOGI: A Seamless Integration of ChatGPT into Jupyter Environments for Programming Education // Proc. of the 2023 ACM SIGPLAN Int. Symposium on SPLASH-E. 2023. P. 50–59. doi:10.1145/3622780.3623648
20. Liao J., Zhong L., Zhe L., Xu H., Liu M., Xie T. Scaffolding Computational Thinking With ChatGPT // IEEE Transactions on Learning Technologies. 2024. Vol. 17. P. 1668–1682. doi:10.1109/tlt.2024.3392896
21. Sarshartehrani F., Mohammadrezaei E., Behravan M., Gracanin D. Enhancing E-Learning Experience Through Embodied AI Tutors in Immersive Virtual Environments: A Multifaceted Approach for Personalized Educational Adaptation // Lecture Notes in Computer Science. 2024. P. 272–287. doi:10.1007/978-3-031-60609-0_20
22. Abolnejadian M., Alipour S., Taeb K. Leveraging ChatGPT for Adaptive Learning through Personalized Prompt-based Instruction: A CS1 Education Case Study // Extended Abstracts of the CHI Conference on Human Factors in Computing Systems. 2024. P. 1–8. doi:10.1145/3613905.3637148
23. Ma B., Li C., Shin'ichi K. Enhancing programming education with ChatGPT: a case study on student perceptions and interactions in a Python course // arxiv.org. 2024. URL: <https://arxiv.org/pdf/2403.15472v3> (date: 26.06.2024).
24. Menon P. Exploring GitHub Copilot assistance for working with classes in a programming course // Issues in Information Systems. Vol. 24, № 4p. 66, 2023; doi:10.48009/4_iis_2023_106
25. Hellas A., Leinonen J., Leppänen L. Experiences from Integrating Large Language Model Chatbots into the Classroom .. arxiv.org. 2023. URL: <https://arxiv.org/pdf/2406.04817> (date: 26.06.2024).
26. Liffiton M., Sheese B., Savelka J., Denny P. CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes // arxiv.org. 2023. URL: <https://arxiv.org/pdf/2308.06921> (date: 26.06.2024).
27. Denny P., MacNeil S., Savelka J., Porter L., Luxton-Reilly A. Desirable Characteristics for AI Teaching Assistants in Programming Education // arxiv.org. 2024. URL: <https://arxiv.org/pdf/2405.14178v1> (date: 26.06.2024).
28. Kazemitabaar M. et al. CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs // arxiv.org. 2024. URL: <https://arxiv.org/pdf/2401.11314> (date: 26.06.2024).
29. George S. D., Dewan P. NotebookGPT — Facilitating and Monitoring Explicit Lightweight Student GPT Help Requests During Programming Exercises // Companion Proceedings of the 29th International Conference on Intelligent User Interfaces. 2024. P. 62–65. doi:10.1145/3640544.3645234
30. Lee Ch., Myung J., Han J., Jin J. Alice Oh Learning from Teaching Assistants to Program with Subgoals: Exploring the Potential for AI Teaching Assistants // arxiv.org. 2023. URL: <https://arxiv.org/pdf/2309.10419> (date: 26.06.2024).
31. Roest L., Keuning H., Jeuring J. Next-Step Hint Generation for Introductory Programming Using Large Language Models // Proc. of the 26th Australasian Computing Education Conference. 2024. P. 144–153. doi:10.1145/3636243.3636259

32. Jin H., Lee S., Shin H., Kim J. Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education // arxiv.org. 2023. URL: <https://arxiv.org/pdf/2309.14534> (date: 26.06.2024).
33. Jin H., Lee S., Shin H., Kim J. Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education // arxiv.org. 2023. URL: <https://arxiv.org/pdf/2309.14534v2> (date: 26.06.2024).
34. Keshtkar F., Rastogi N., Chalarca S., S. A. C. Bukhari. AI Tutor: Student's Perceptions and Expectations of AI-Driven Tutoring Systems: A Survey-Based Investigation // Proc. of FLAIRS-37. 2024. URL: <urlhttps://journals.flvc.org/FLAIRS/article/download/135314/139973> (date: 26.06.2024).
35. Kohlhase M. et al. Project volL-KI // KI — Künstliche Intelligenz. 2024. doi:10.1007/s13218-024-00846-9
36. Smith G., Gupta A., MacLellan Ch. Apprentice Tutor Builder: A Platform For Users to Create and Personalize Intelligent Tutors // arxiv.org. 2024. URL: <https://arxiv.org/pdf/2404.07883> (date: 26.06.2024).
37. Tang X. et al. VizGroup: An AI-Assisted Event-Driven System for Real-Time Collaborative Programming Learning Analytics // arxiv.org. 2024. URL: <https://arxiv.org/pdf/2404.08743v1> (date: 26.06.2024).
38. Estévez-Ayres I., Callejo P., Hombrados-Herrera M. Á., Alario-Hoyos C., Delgado Kloos C. Evaluation of LLM Tools for Feedback Generation in a Course on Concurrent Programming // Int. Journal of Artificial Intelligence in Education. 2024. doi:10.1007/s40593-024-00406-0
39. Shanto S. S., Ahmed Z., Jony A. I. PAIGE: A generative AI-based framework for promoting assignment integrity in higher education // STEM Education. 2023. Vol. 3, № 4. P. 288–305. doi:10.3934/steme.2023018
40. Silva R. Using ChatGPT as a Virtual Tutor for Teaching Algorithms. [Technical Report], Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora and Governador Valadares, Brazil, 2024. doi:10.13140/RG.2.2.33407.01440
41. Perera K. G. D. K., Wijayanayake J., Prasadika J. Factors Affecting the Effectiveness of Generative Artificial Intelligence Apps on University Students' Programming Language Learning in Sri Lanka: A Systematic Literature Review // 2024 4th Int. Conf. on Advanced Research in Computing (ICARC). 2024. P. 276–281. doi:10.1109/icarc61713.2024.10499744
42. Liu S., Grey B., Watkins R. Assessing Risks, Challenges and Opportunities of Generative AI in Computer Programming Education // Lightning Talk Journal of Computing Sciences in Colleges. 2024. Vol. 39, № 8. P. 210–211.
43. Lin C.-C., Huang A. Y. Q., Lu O. H. T. Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review // Smart Learning Environments. 2023. Vol. 10, № 1. doi:10.1186/s40561-023-00260-y
44. Le T. M. T. Utilizing AI Tools in Learning Introductory Programming: An Analysis of Tampere University Students' Strategies and Outcomes. Bachelor Thesis. Tampere University Science and Engineering, Tampere, Finland, 2024.
45. Scholla, Schiffner D., Kiesler N. Analyzing Chat Protocols of Novice Programmers Solving Introductory Programming Tasks with ChatGPT // arxiv.org. 2024. URL: <https://arxiv.org/pdf/2405.19132> (date: 26.06.2024).
46. Leidner J. L., Reiche M. Language-Model Assisted Learning How to Program? // Artificial Intelligence. ECAI 2023 International Workshops. 2024. P. 425–438. doi:10.1007/978-3-031-50485-3_41
47. Hocquetta C., Langerb J., Croppera A., Schmidb U. Can Humans Teach Machines to Code? // arxiv.org. 2024. URL: <https://arxiv.org/pdf/2404.19397> (date: 26.06.2024).
48. Jacques L. Teaching CS-101 at the Dawn of ChatGPT // ACM Inroads. 2023. Vol. 14, № 2. P. 40–46. doi:10.1145/3595634

Поступила в редакцию 05.06.2024, окончательный вариант — 26.06.2024.

Долинский Михаил Семёнович, канд. техн. наук, доцент, доцент кафедры математических проблем управления и информатики, факультет математики и технологий программирования. Гомельский государственный университет им. Ф. Скорины, Гомель, Республика Беларусь, ✉ dolinsky@gsu.by

Computer tools in education, 2024

№ 2: 85–96

<http://cte.eltech.ru>

[doi:10.32603/2071-2340-2024-2-85-96](https://doi.org/10.32603/2071-2340-2024-2-85-96)

Directions for Using Generative Artificial Intelligence in Initial Programming Training at Universities

Dolinsky M. S.¹, Cand. Sc., Associate Professor, ✉ dolinsky@gsu.by, 0000-0002-3057-4051

¹Francisk Skorina Gomel State University, 119 Kirova str., 246019, Gomel, Republic of Belarus

Abstract

This paper provides a review of the literature on the use of generative artificial intelligence (GenAI) in initial programming education at universities. The main areas of application of generative artificial intelligence are given: specialized narrow-topic developments, integration into online learning platforms and testing solutions, students working with GenAI without restrictions, guiding systems for interaction of students with GenAI (without providing solutions), teacher assistance, tools for developing intelligent teaching systems. It also contains a review of works that analyze the results achieved and unresolved problems.

Keywords: *generative artificial intelligence, teaching programming to first-year students, intelligent teaching systems.*

Citation: Dolinsky M. S., "Directions for Using Generative Artificial Intelligence in Initial Programming Training at Universities," *Computer tools in education*, no. 2, pp. 85–96, 2024 (in Russian); [doi:10.32603/2071-2340-2024-2-85-96](https://doi.org/10.32603/2071-2340-2024-2-85-96)

References

1. M. Kaleem, M. A. Hassan, and S. K. Khurshid, "A Machine Learning-Based Adaptive Feedback System to Enhance Programming Skill Using Computational Thinking," *IEEE Access*, vol. 12, pp. 59431–59440, 2024; [doi:10.1109/access.2024.3391873](https://doi.org/10.1109/access.2024.3391873)
2. S. Zhang, J. Yang, and X. Sang, "Exploring the Applications of EduCoder Platform in Blended Teaching for Computer Major," *Journal of Education and Educational Research*, vol. 4, no. 2, pp. 100–103, 2023; [doi:10.54097/jeer.v4i2.10819](https://doi.org/10.54097/jeer.v4i2.10819)
3. Z. Li, S. Zhang, and X. Sang, "Exploration of Machine Learning Teaching based on the EduCoder Platform," *Journal of Education and Educational Research*, vol. 4, no. 3, pp. 130–133, 2023; [doi:10.54097/jeer.v4i3.11387](https://doi.org/10.54097/jeer.v4i3.11387)
4. J. Figueiredo and F. Garcia-Penalvo, "Teaching and Learning Tools for Introductory Programming in University Courses," *2021 International Symposium on Computers in Education (SIIE)*, 2021; [doi:10.1109/siie53363.2021.9583623](https://doi.org/10.1109/siie53363.2021.9583623)
5. V. Phan and E. Hicks, "Code4Brownies: an active learning solution for teaching programming and problem solving in the classroom," in *Proc. of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pp. 153–158, 2018; [doi:10.1145/3197091.3197128](https://doi.org/10.1145/3197091.3197128)
6. O. Herden, "Integration of Chatbots for Generating Code Into Introductory Programming Courses," in *Proc. of International Conference "Future of education"*, 2024. [Online]. Available: <https://conference.pixel-online.net/files/foe/ed0014/FP/9091-ICT6673-FP-FOE14.pdf>
7. Ch. Köppe, H. Keuning, I. Lykourantzou, I. A. Chacon, and S. Sosnovsky, "Practices for the Application of Generative AI in Programming Education. Research project," in <https://www.uu.nl>, 2024. [Online]. Available: <https://www.uu.nl/sites/default/files/GenAI%20in%20ProgEd%20practices.pdf>
8. Q. Jin, Y. Yang, Q. Chen, and Z. Lu, "GeneGPT: augmenting large language models with domain tools for improved access to biomedical information," *Bioinformatics*, vol. 40, no. 2, p. btae075, 2024; [doi:10.1093/bioinformatics/btae075](https://doi.org/10.1093/bioinformatics/btae075)

9. S. Lau and P. Guo, "From 'Ban It Till We Understand It' to 'Resistance is Futile': How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot," in *Proc. of the 2023 ACM Conf. on Int. Computing Education Research*, vol. 1, pp. 106–121, 2023; doi:10.1145/3568813.3600138
10. J. C. Farah, B. Spaenlehauer, S. Ingram, A. K. Purohit, A. Holzer, and D. Gillet, "Harnessing Rule-Based Chatbots to Support Teaching Python Programming Best Practices," *Lecture Notes in Networks and Systems*, pp. 455–466, 2024; doi:10.1007/978-3-031-51979-6_47
11. D. Cambaz, *Use of AI-driven code generation models in teaching and learning programming: a systematic literature review*, Bachelor thesis, TU Delft, Faculty Electrical Engineering, Mathematics and Computer Science, Delft, The Netherlands, 2023.
12. M. D. L. Roca, M. M. Chan, A. Garcia-Cabot, E. Garcia-Lopez, and H. Amado-Salvatierra, "The impact of a chatbot working as an assistant in a course for supporting student learning and engagement," *Computer Applications in Engineering Education*, p. e22750, 2024; doi:10.1002/cae.22750
13. S. Bobadilla, R. Glassey, A. Bergel, and M. Monperrus, "SOBO: A Feedback Bot to Nudge Code Quality in Programming Courses," in *arxiv.org*, 2023. [Online]. Available: <https://arxiv.org/pdf/2303.07187>
14. J. Sengewald, M. Wilz, and R. Lackes, "AI-Assisted Learning Feedback: Should Gen-AI Feedback Be Restricted to Improve Learning Success? A Pilot Study in a SQL Lecture," in *Proc. of ECIS 2024*, p. 12, 2024.
15. K. Pahi, S. Hawlader, E. Hicks, A. Zaman, and V. Phan, "Enhancing active learning through collaboration between human teachers and generative AI," *Computers and Education Open*, vol. 6, p. 100183, 2024; doi:10.1016/j.caeo.2024.100183
16. D. Roldán-Álvarez and F. J. Mesa, "Intelligent Deep-Learning Tutoring System to Assist Instructors in Programming Courses," *IEEE Transactions on Education*, vol. 67, no. 1, pp. 153–161, 2024; doi:10.1109/te.2023.3331055
17. C. Martínez-Araneda, M. Gutiérrez, D. Maldonado, P. Gómez, A. Segura, and C. Vidal-Castro, "Designing a Chatbot to support problem-solving in a programming course," in *Proc. of 18th International Technology, Education and Development Conference*, pp. 966–975, 2024; doi:10.21125/inted.2024.0317
18. C. Martínez-Araneda, M. Gutiérrez Valenzuela, P. Gómez Meneses, D. Maldonado Montiel, A. Segura Navarrete, and C. Vidal-Castro, "How Useful TutorBot+ is for Teaching and Learning in Programming Courses: a Preliminary Study," in *Proc. of 42nd IEEE International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–7, 2023; doi:10.1109/sccc59417.2023.10315697
19. K. Kuramitsu, Y. Obara, M. Sato, and M. Obara, "KOGI: A Seamless Integration of ChatGPT into Jupyter Environments for Programming Education," in *Proc. of the 2023 ACM SIGPLAN Int. Symposium on SPLASH-E*, pp. 50–59, 2023; doi:10.1145/3622780.3623648
20. J. Liao, L. Zhong, L. Zhe, H. Xu, M. Liu, and T. Xie, "Scaffolding Computational Thinking With ChatGPT," *IEEE Transactions on Learning Technologies*, vol. 17, pp. 1668–1682, 2024; doi:10.1109/tlt.2024.3392896
21. F. Sarshartehrani, E. Mohamadrezaei, M. Behravan, and D. Gracanin, "Enhancing E-Learning Experience Through Embodied AI Tutors in Immersive Virtual Environments: A Multifaceted Approach for Personalized Educational Adaptation," *Lecture Notes in Computer Science*, pp. 272–287, 2024; doi:10.1007/978-3-031-60609-0_20
22. M. Abolnejadian, S. Alipour, and K. Taeb, "Leveraging ChatGPT for Adaptive Learning through Personalized Prompt-based Instruction: A CS1 Education Case Study," *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–8, 2024; doi:10.1145/3613905.3637148
23. B. Ma, C. Li, and K. Shin'ichi, "Enhancing programming education with ChatGPT: a case study on student perceptions and interactions in a Python course," in *arxiv.org*, 2024. [Online]. Available: <https://arxiv.org/pdf/2403.15472v3>
24. P. Menon, "Exploring GitHub Copilot assistance for working with classes in a programming course," *Issues in Information Systems*, vol. 24, no. 4, p. 66, 2023; doi:10.48009/4_iis_2023_106
25. A. Hellas, J. Leinonen, and L. Leppänen, "Experiences from Integrating Large Language Model Chatbots into the Classroom," in *arxiv.org*, 2023. [Online]. Available: <https://arxiv.org/pdf/2406.04817>
26. M. Liffiton, B. Sheese, J. Savelka, P. Denny, "CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes," in *arxiv.org*, 2023. [Online]. Available: <https://arxiv.org/pdf/2308.06921>
27. P. Denny, S. MacNeil, J. Savelka, L. Porter, and A. Luxton-Reilly, "Desirable Characteristics for AI Teaching Assistants in Programming Education," in *arxiv.org*, 2024. [Online]. Available: <https://arxiv.org/pdf/2405.14178v1>
28. M. Kazemitabaar et al., "CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs," in *arxiv.org*, 2024. [Online]. Available: <https://arxiv.org/pdf/2401.11314>
29. S. D. George and P. Dewan, "NotebookGPT – Facilitating and Monitoring Explicit Lightweight Student GPT Help Requests During Programming Exercises," in *Companion Proceedings of the 29th International Conference on Intelligent User Interfaces*, pp. 62–65, 2024; doi:10.1145/3640544.3645234
30. Ch. Lee, J. Myung, J. Han, and J. Jin, "Alice Oh Learning from Teaching Assistants to Program with Subgoals:

- Exploring the Potential for AI Teaching Assistants,” in *arxiv.org*, 2023. [Online]. Available: <https://arxiv.org/pdf/2309.10419>
31. L. Roest, H. Keuning, and J. Jeuring, “Next-Step Hint Generation for Introductory Programming Using Large Language Models,” in *Proc. of the 26th Australasian Computing Education Conference*, pp. 144–153, 2024; doi:10.1145/3636243.3636259
 32. H. Jin, S. Lee, H. Shin, and J. Kim, “Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education,” in *arxiv.org*, 2023. [Online]. Available: <https://arxiv.org/pdf/2309.14534>
 33. H. Jin, S. Lee, H. Shin, and J. Kim, “Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education,” in *arxiv.org*, 2023. [Online]. Available: <https://arxiv.org/pdf/2309.14534v2>
 34. F. Keshtkar, N. Rastogi, S. Chalarca, and S. A. C. Bukhari, “AI Tutor: Student’s Perceptions and Expectations of AI-Driven Tutoring Systems: A Survey-Based Investigation,” in *Proc. of FLAIRS-37*, 2024. [Online]. Available: [urlhttps://journals.flvc.org/FLAIRS/article/download/135314/139973](https://journals.flvc.org/FLAIRS/article/download/135314/139973)
 35. M. Kohlhase et al., “Project voll-KI,” *KI - Künstliche Intelligenz*, 2024; doi:10.1007/s13218-024-00846-9
 36. G. Smith, A. Gupta, Ch. MacLellan, “Apprentice Tutor Builder: A Platform For Users to Create and Personalize Intelligent Tutors,” in *arxiv.org*, 2024. [Online]. Available: <https://arxiv.org/pdf/2404.07883>
 37. X. Tang et al., “VizGroup: An AI-Assisted Event-Driven System for Real-Time Collaborative Programming Learning Analytics,” in *arxiv.org*, 2024. [Online]. Available: <https://arxiv.org/pdf/2404.08743v1>
 38. I. Estévez-Ayres, P. Callejo, M. Á. Hombrados-Herrera, C. Alario-Hoyos, and C. Delgado Kloos, “Evaluation of LLM Tools for Feedback Generation in a Course on Concurrent Programming,” *Int. Journal of Artificial Intelligence in Education*, 2024; doi:10.1007/s40593-024-00406-0
 39. S. S. Shanto, Z. Ahmed, and A. I. Jony, “PAIGE: A generative AI-based framework for promoting assignment integrity in higher education,” *STEM Education*, vol. 3, no. 4, pp. 288–305, 2023; doi:10.3934/steme.2023018
 40. R. Silva, *Using ChatGPT as a Virtual Tutor for Teaching Algorithms*, [Technical Report], Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora and Governador Valadares, Brazil, 2024; doi:10.13140/RG.2.2.33407.01440
 41. K. G. D. K. Perera, J. Wijayanayake, and J. Prasadika, “Factors Affecting the Effectiveness of Generative Artificial Intelligence Apps on University Students’ Programming Language Learning in Sri Lanka: A Systematic Literature Review,” in *Proc. 2024 4th Int. Conf. on Advanced Research in Computing (ICARC)*, pp. 276–281, 2024; doi:10.1109/icarc61713.2024.10499744
 42. S. Liu, B. Grey, and R. Watkins, “Assessing Risks, Challenges and Opportunities of Generative AI in Computer Programming Education,” *Lightning Talk Journal of Computing Sciences in Colleges*, vol. 39, no. 8, pp. 210–211, 2024.
 43. C.-C. Lin, A. Y. Q. Huang, and O. H. T. Lu, “Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review,” *Smart Learning Environments*, vol. 10, no. 1, 2023; doi:10.1186/s40561-023-00260-y
 44. T. M. T. Le, “Utilizing AI Tools in Learning Introductory Programming: An Analysis of Tampere University Students’ Strategies and Outcomes,” Bachelor Thesis, Tampere University Science and Engineering, Tampere, Finland, 2024.
 45. A. Scholl, D. Schiffner, and N. Kiesler, “Analyzing Chat Protocols of Novice Programmers Solving Introductory Programming Tasks with ChatGPT,” in *arxiv.org*, 2024. [Online]. Available: <https://arxiv.org/pdf/2405.19132>
 46. J. L. Leidner and M. Reiche, “Language-Model Assisted Learning How to Program?,” *Artificial Intelligence. ECAI 2023 International Workshops*, pp. 425–438, 2024; doi:10.1007/978-3-031-50485-3_41
 47. C. Hocquettea, J. Langerb, A. Croppera, and U. Schmidb, “Can Humans Teach Machines to Code?,” in *arxiv.org*, 2024. [Online]. Available: <https://arxiv.org/pdf/2404.19397>
 48. L. Jacques, “Teaching CS-101 at the Dawn of ChatGPT,” *ACM Inroads*, vol. 14, no. 2, pp. 40–46, 2023; doi:10.1145/3595634

Received 05-06-2024, the final version — 26-06-2024.

Michael Dolinsky, Cand. of Sciences (Tech.), Associate Professor of the Department Mathematical Problems of Management and Computer Science, Faculty of Mathematics and Technologies of Programming, Francisk Skorina Gomel State University, Gomel, Republic of Belarus,
 ✉ dolinsky@gsu.by