
МЕТОДОЛОГИЯ И ТЕХНОЛОГИЯ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ / METHODOLOGY AND TECHNOLOGY OF VOCATIONAL EDUCATION

DOI: <https://doi.org/10.18454/PED.2024.3.1>**ИСПОЛЬЗОВАНИЕ ПРОГРАММ ГЕНЕРАТИВНОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ПРОЦЕССЕ НАЧАЛЬНОГО ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ В ВУЗЕ**

Научная статья

Долинский М.С.^{1,*}¹ ORCID : 0000-0002-3057-4051;¹ Гомельский государственный университет имени Ф. Скорины, Гомель, Беларусь

* Корреспондирующий автор (dolinsky[at]gsu.by)

Аннотация

В данной работе приводится обзор литературы по начальному обучению программированию в вузах и использованию программ генеративного искусственного интеллекта (ПГИИ) в этих целях.

При начальном обучении программированию в вузах выделяются задачи развития пяти навыков: абстракция (определение какую информацию о сущности/объекте необходимо сохранить, а какую игнорировать); обобщение (формулирование решения в общих терминах так, чтобы оно могло быть применено для решения различных задач); декомпозиция (разбиение сложных задач на меньшие подзадачи, которые легче понять и решить); алгоритмизация (определение шаг за шагом множества операций/действий, для того чтобы решить задачу); отладка (локализация и устранение ошибок). Отмечается также множество языков программирования, используемых в начальном обучении в вузах, как-то Pascal, C++, Python, C#, Java.

Обзор литературы по использованию ПГИИ при начальном обучении программированию в вузах указал на следующие основные проблемы и задачи: не допустить использования ПГИИ для получения незаслуженных оценок по предмету; разработать приёмы и примеры использования ПГИИ в учебном процессе для персонализации, повышения интенсивности и качества; исследовать новые направления использования ПГИИ, например, развитие навыков использования ПГИИ для решения реальных производственных задач (оператор ПГИИ).

Приводится описание авторского опыта использования ПГИИ для начального обучения программированию первокурсников факультета математики и технологий программирования Гомельского государственного университета им.Ф.Скорины.

Ключевые слова: генеративный искусственный интеллект, обучение программированию первокурсников, DL.GSU.BY.

USE OF GENERATIVE ARTIFICIAL INTELLIGENCE PROGRAMMES IN THE PROCESS OF INITIAL PROGRAMMING EDUCATION IN UNIVERSITIES

Research article

Dolinsky M.S.^{1,*}¹ ORCID : 0000-0002-3057-4051;¹ Francisk Skorina Gomel State University, Gomel, Belarus

* Corresponding author (dolinsky[at]gsu.by)

Abstract

This work provides a review of the literature on the initial teaching of programming in universities and the use of Generative Artificial Intelligence Programmes (GAIPs) for this purpose.

At the initial training of programming in universities, the tasks of developing five skills are highlighted: abstraction (determining what information about an entity/object should be kept and what should be ignored); generalization (formulating a solution in general terms so that it can be applied to solve different problems); decomposition (breaking complex problems into smaller subtasks that are easier to understand and solve); algorithmicization (defining step by step a set of operations/actions in order to solve a problem); debugging (localization and elimination of errors). It is also noted that there are many programming languages used in initial training at universities, such as Pascal, C++, Python, C#, Java.

The review of the literature on the use of GAIPs in initial programming training at universities indicated the following main problems and tasks: preventing the use of GAIPs to obtain undeserved grades in the subject; developing techniques and examples of using GAIPs in the teaching process to personalize, increase intensity and quality; exploring new directions of using GAIPs, for example, developing skills of using GAIPs to solve real production tasks (GAIPs operator).

The author's experience of using GAIPs for initial programming training of first-year students of the Department of Mathematics and Programming Technologies of Gomel State University named after F.Skorina is described.

Keywords: generative artificial intelligence, teaching programming to first-year students, DL.GSU.BY.

Введение**1.1. Начальные курсы программирования**

Предмет «программирование» является базовым для первых курсов всех инженерных специальностей университетов и очень сложным для большинства студентов [1]. Сравнительный анализ подходов к обучению программированию во вводных курсах приводится в [2]. Сравнительный анализ состояния обучения пяти базовым

понятиям во вводных курсах программирования (абстракции, декомпозиции, алгоритмизации, отладке и решении задач) в странах Европы, Северной Америки, Океании, Азии, Латинской Америки и Африки приводится в [3]. В 2020 году IEEE и ACM – две ведущих ассоциации специалистов в области компьютеров и компьютерных наук создали Computing Curricula, который определил стратегию обучения компьютерным наукам, в том числе и вводным курсам программирования [4].

Среди предлагаемых специалистами средств повышения эффективности обучения программированию на первом курсе университета встречаются:

- визуальные среды программирования [5], [6];
- использование специально разработанных WEB-платформ [7], [8], [9];
- применение проектно-ориентированного подхода [10], [11];
- специальные обучающие средства обучению программированию [12];
- авторские методики обучения программированию [13], включающие современные подходы к обучению, модифицированную педагогика, создание привлекательной и реалистичной среды обучения программированию [14];
- система автоматического оценивания решений студентов [15];
- геймификация [16], [17];
- специальное программное средство для контроля прогресса студентов в течение семестра [18].

1.2. Влияние на обучение программ искусственного интеллекта

В [19] отмечается, что AI-программы, такие как ChatGPT в состоянии выполнять задания вводных курсов по программированию вместо студентов, при этом ChatGPT понимает не только текстовые описания заданий, но и программы, написанные на различных языках программирования таких как Python, Java, C++ и др. Соответственно, перед преподавателями стоит задача изменения соответствующих заданий.

В [20] представлено один из подходов к преподаванию вводных курсов программирования с помощью искусственного интеллекта (AI). Среди прочего, предлагается давать задания на использование AI генерирующего программы по различным представлениям задач: текстовые представления (алгоритмы, написанные на языке программирования или псевдокоде), визуальные представления (например, UML-диаграммы), числовые и формульные представления. Другой вариант заданий – генерация нескольких решений одной и той же задачи с последующим сравнительным анализом реализаций. Третий – объяснение студентом сгенерированных AI решений.

В [21] предлагается использовать ChatGPT для оценивания программ, разработанных студентами, отмечая, что исторически это делалось либо проверкой вручную, либо автоматической проверкой по тестам.

Традиционные подходы к обучению программированию фокусируются на практике написания кода. Но с появлением AI-генераторов кода требуется изменять эти подходы в сторону развития навыков формирования эффективных текстовых запросов к AI-генераторам кодов [22].

В [23] описывается использование ChatGPT в обработке данных, указывая на качества достоинств такого подхода персонализированное обучение, объяснений концепций, генерация кода с пояснениями, оценивание работ. Обработка данных включает три существенных обязательных курса: базы данных, анализ данных, машинное обучение.

ChatBot, который помогает студентам осваивать оформление программ на языке Python в соответствии со стандартом PEP-8 представлен в [24].

Анализ всех этих приводит к следующим выводам:

1. Применение программ генеративного искусственного интеллекта неотвратимо и может существенно повысить качество учебного процесса самыми разнообразными и неожиданными пока способами.
2. В настоящее время одной из важнейших задач является защита от использования ПГИИ недобросовестными студентами для сдачи лабораторных и практических работ.

Основные результаты

2.1. Цель семестрового курса по программированию и основам алгоритмизации

Автор много лет обучает первокурсников вводному курсу программирования [25], [26]. На первый курс факультета математики и технологий программирования Гомельского государственного университета им. Ф. Скорины поступают студенты, которые, в большинстве своём, не умеют писать программы, в то же время встречаются первокурсники, которые начали самостоятельное обучение ещё до поступления в университет, сделав собственный выбор языка программирования (в 2023-2024 учебном году это один из 5 языков: Pascal, C++, Python, Java, C#).

Целью семестрового курса по программированию и основам алгоритмизации (в общей сложности 10 академических часов в неделю) – заложить основы базовых теоретических знаний и практических навыков по следующим направлениям:

По текстовой постановке задачи сделать математическую/алгоритмическую постановку

Исходный текст задания может содержать от нескольких строчек до нескольких страниц. Студент должен уметь выделить полезную информацию от несущественной и в завершение этого процесса переформулировать условие задачи в виде математической или алгоритмической постановки, используя в качестве предполагаемых объектов обработки и результатов такие понятия как числа, символы, строки, одномерные и многомерные массивы чисел, символов и строк.

Разработать алгоритм

Разработка алгоритма предполагает текстовое представление последовательности действий, приводящей к получению требуемого по постановке задачи результата по заданным исходным данным. Алгоритм должен быть описан так, чтобы его мог использовать любой студент потока для получения правильного результата по предлагаемым входным данным.

Написать программу

Обучение большинства студентов ведётся на языке программирования Паскаль по следующим причинам:

· Паскаль был задуман как язык для обучения программированию, и потому проще других усваивается студентами, не имеющими никакой предварительной подготовки.

· С 1999 года под руководством автора разрабатывается система дистанционного обучения DL.GSU.BY [27], [28], в которой построена эффективная среда автоматизированного персонализированного обучения программированию на языке Паскаль «с чистого листа» школьников различных возрастных диапазонов: 1-4 классы, 5-8 классы, 9-11 классы [29]. Использование данной среды, несмотря на сильную дифференциацию в предварительной подготовке первокурсников, позволяет обеспечить практическую реализацию главной парадигмы авторского обучения «на каждом занятии каждую минуту работает каждый ученик». Ученику всегда предлагается посильное задание, и он сам может регулировать уровень роста сложности предлагаемых к выполнению заданий. В данном случае уместно привести аналогию с множеством лестниц, ведущих на одну и ту же высоту, но ступеньки, у которых сильно разнятся, могут быть очень маленькими, больше, ещё больше, вплоть до очень больших.

· Для студентов имеются подготовленные автором конспекты опорных схем лекций по темам «Введение в программирование», «Одномерный массив – стандартные алгоритмы», «Одномерный массив – технология разработки нестандартных алгоритмов», «Двумерный массив», «Геометрия», «Строки», «Сортировка», «Пользовательские функции и процедуры», «Очередь», которые иллюстрируются фрагментами программ на Паскале.

· Изучается минимальное подмножество синтаксиса языка программирования Паскаль, такие как: встроенные типы данных (символ, строка, целое и вещественное число), одномерный и двумерный массивы встроенных типов, операторы присваивания, условия и циклов FOR и WHILE. Поскольку основная цель не изучение языка программирования Паскаль, а развитие навыков алгоритмизации и программирования.

Поддерживается также и обучение на других языках программирования (C++, Python, C#, Java): имеются конспекты опорных схем лекций для этих языков, выполненные и развивающиеся заинтересованными студентами, осуществляется автоматическая проверка решений на всех этих языках программирования, поддерживается форум, где можно задать вопрос и в течение суток получить ответ от более квалифицированного одногруппника или старшекурсника.

Отправить на автоматическую проверку

Важная особенность выстроенной системы персонализированного обучения заключается в том, что условия задачи предъявляются автоматически; студент, написав решение, может отправить его на проверку и в течение нескольких секунд узнать результат. Для автоматической проверки решение запускается на каждом из подготовленных автором задания входных данных и ответы студента и автора задания сравниваются, в случае неоднозначности проверка ответов решения студента осуществляется специальными программами.

Уступка тестов

Огромное подспорье для начинающих изучение программирования – выдача им тестов: то есть входных данных, на которых их программа выдала неверный ответ вместе с авторским ответом. Это позволяет быстрее и проще понять студентам, в каких именно случаях их программа работает неверно. Последний тест задачи не отдаётся студентам, что позволяет с одной стороны, развивать у студентов навыки самостоятельной разработки тестов, а с другой стороны, обеспечивает невозможность написания «обманых» решений типа «если ввод такой, ответ такой».

Сделать локализацию ошибки, исправить её

Это самая сложная и работа и, по мнению автора, соответствующие навыки достигаются многократными упражнениями в локализации и устранении ошибок.

Заметим, что это всё является основой для дальнейшего обучения языкам и технологиям программирования на старших курсах.

2.2. Система оценивания, исключающая возможность использования ГИИ для незаслуженного повышения оценки

Авторская система оценивания базируется на автоматически формируемой ведомости оценок, в которую включаются отдельными колонками все виды деятельности студента в течение учебного семестра, включающие контроль теории, контроль практики, обучение, техминимум, индивидуальные задания, пропуски занятий. Вместе они формируют фактически оценку отношения студента к учебному процессу. Множители значений в колонках подобраны таким образом, что все студенты, добросовестно относящиеся к занятиям, получают высокие оценки – до 10 и выше. Другой важной оценкой является оценка по контрольному срезу. Оценка по предмету выставляется как минимум из этих двух оценок («Отношение» и «Умение»). Напомню, что в Беларуси 10-балльная система оценивания.

Контрольный срез – это занятие, на котором студенту предъявляется 10 индивидуальных заданий на 1.5 часа, выбранных случайно из имеющегося банка заданий, решённых первокурсниками прошлых лет. Оценка равна количеству полностью решённых задач за эти 1.5 часа. Таким образом, студент должен для 10 различных задач (следовательно, 10 раз) выполнить всё, чему он учился в семестре: сформулировать для себя формальную постановку задачи по исходной текстовой, разработать алгоритм, написать программу на выбранном им языке программирования, в случае ошибок – найти их и исправить. При этом первые 5 задач выбираются из заданий «Техминимум», которые студенты могут решать в течение семестра. Техминимум содержит задания на темы: «Введение в программирование», «Одномерный массив», «Двумерный массив», «Геометрия», «Строки». Таким образом, оценка 5 гарантируется каждому, кто добросовестно занимается в течение семестра, вне зависимости от уровня предварительной подготовки до поступления в университет. Затем следуют 4 задания на темы «Одномерный массив», «Двумерный массив», «Геометрия», «Строки». Здесь задания берутся из банков заданий, недоступных студентам в течение семестра. Последнее 10-е задание на тему «Очередь» берётся из банка заданий, доступных студентам в течение семестра.

С середины семестра, когда изучена вся необходимая базовая теория, каждый студент имеет право писать контрольный срез еженедельно. При этом оценка по контрольному срезу меняется только в сторону увеличения.

Важно отметить, что технически обеспечена ситуация, когда студент должен решать задачи опираясь только на собственные знания. Закрыт доступ в Интернет, равно как и ко всем дискам и папкам локальной сети, кроме специальной папки, из которой автоматически удаляются все файлы при входе студента в систему. Преподаватель осуществляет визуальный контроль, чтобы студенты не пользовались конспектами и телефонами.

Другой важной особенностью системы обучения является необходимость последовательного решения сотен и тысяч маленьких задач в процессе обучения, что с одной стороны, многократно тренирует нужные навыки, а с другой – делает нерациональным обращение к ГИИ тех студентов, которые могут сами решить задачу.

2.3. Факторы повышения мотивации студентов к качественному обучению

Следующие факторы повышают мотивацию студентов к качественному обучению:

Автоматическое персонализированное обучение

Для каждой темы теории задачи организованы в древовидную структуру, с набором «базовых» задач на стволе дерева. В случае невозможности решить задачу студент может нажать кнопку «Не знаю» и попасть на поддерево обучающих заданий, к каждому из которых может быть своё дерево подводящих заданий.

Низкий порог входа

Фактически от студента не требуется никакой предварительной подготовки, есть даже пакет заданий, позволяющий выучить слова program, var, longint, begin, readln, writeln, end (включая расположение соответствующих латинских букв на клавиатуре) и их перевод, которые используются при написании программ на Паскале

Множество точек входа

Студент может начинать обучение в каждой из тем либо в разделе «Техминимум», где находятся простейшие задания, либо в разделе «Обучение», где находятся более сложные задания, либо в разделе «Индивидуальные задания», где для каждой темы имеется набор заданий, из которого студент может выбирать интересное для себя и посильное задание.

Персонализация

Фактически система обучения ориентирована на выбор каждым студентом оптимального для себя маршрута обучения. В разделах «Техминимум» и «Обучение» обеспечивается автоматическое предложение задач к решению в зависимости от того, успешно или нет была решена предыдущая задача.

Интенсификация

Автоматическая выдача условий задач и проверка решений в течение минуты сильно интенсифицируют учебный процесс, а всевозможные таблицы результатов, отражающие текущие достижения студентов, мотивируют многих студентов работать лучше и больше.

Свобода в выборе языка разработки программ

В настоящее время студенты выбирают на практике между пятью языками программирования: Pascal, Python, Java, C#, C++. Если возникнет потребность, могут быть добавлены новые языки.

2.4. Направления использования ГИИ в обучении студентами

С 2023-2024 учебного года автор предложил студентам активно пользоваться программами генеративного искусственного интеллекта. Для этого на форуме системы DL.GSU.BY для первокурсников, изучающих программирование, была заведена специальная тема «Учимся с помощью искусственного интеллекта», в первом сообщении которой приводятся ссылки на различные программы генерирующие тексты программ, с предложением пользоваться ими в случае следующих затруднительных ситуаций:

- если не работает решение – загружаете решение – получаете советы
- если не можете придумать решение – загружаете условие – получаете советы и тексты решений
- если не можете придумать тест, на котором не работает решение, загружаете условие задачи и решение, получаете тест, на котором решение не работает.

О результатах использования предложено писать ответными сообщениями в этой же теме: что спрашивали, какой ответ получили, получена ли помощь?

Уже имеются первые отклики от студентов, как отрицательные, так и положительные. Вот некоторые фрагменты из них:

Позитивный отклик:

«Пользуюсь ChatGPT 3.5 уже более месяца, очень полезная штука. Помогает учиться самостоятельно и не только. Любого рода вопросы, даже если совсем трудно понять, может объяснить по многу раз, самыми разными путями, языками и т.д. Сложные задачи решает плохо, однако если написать четкий алгоритм сложной задачи, то вполне может решить. Очень хорошо помогает осваивать новые темы тем, кто изучает другие языки программирования (C++, Java, Python, C# и др.)

Если нужен аналог-конспекта по основным темам, и не только, пишешь ему сообщение вида:

****скопированный конспект для PascalABC****

Привет, переделай, пожалуйста, конспект для ... (ваш язык программирования) переписываешь в тетрадь, учишься.

Так же помогает разобраться с "тупиком" в задаче, найти ошибку в коде, указать на неправильную логику решения и найти правильный путь.

Что лично мне понравилось, так это, что он отвечает без эмоций. Даже если ты совсем уж "не понимаешь", бот пытается максимально помочь даже с простыми вопросами. Приятным бонусом является то, что бот подстраивается под твои умения и решает задачи только теми методами, которые тебе известны (для этого в начале общения лучше указать, что ты умеешь, чтобы бот не использовал неизвестные тебе способы решения)».

Негативный отклик:

«Данный бот очень далек от совершенства по некоторым пунктам:

1) Часто пишет полную ерунду, так как не может понять, что ему нужно сделать, касается задач типа дан двумерный массив из каждой его строки найти максимальное значение и занести это значение в одномерный массив, и потом из одномерного массива вывести минимальное (то есть задачи с 2-мя и более заданиями).

2) Пишет непонятно и очень мудрено, так же оформление страдает, код размашистый и постоянно с большими буквами так же не удобно его читать.

3) Если задача довольно сложная, то он часто использует функции и если сразу ему не сказать, чтобы он их не использовал, то в итоге получишь непонятное, неработающее нечто.

Но у бота есть и плюсы:

1) Бот дает хорошие советы по коду, а также его объяснения кратки и понятны (меня его подсказки много раз выручали).

2) Бот помогает учиться, если при работе в техминимуме и в обучении сотрудничать с ботом, то разбор новой темы идет намного легче, но для каждого все фломастеры разные, то мне было бы проще работать по книжке, так как свой первый язык программирования я учил по книжке.

ТАКЖЕ: бот подходит для помощи в изучении, а не для сдачи задач, так как если все задачи будут очень грязными с длинными названиями переменных, то это явно бот.

В итоге: потенциал есть, но пока что бот очень сырой, плюс поиск в ошибок в коде я не тестил, но при условии, что он код еле пишет, то, думаю, ошибки он будет искать плоховато».

По предварительному мнению автора, программы генеративного искусственного интеллекта могут сильно помочь в изучении языка программирования, в особенности, отличного от Паскаля, при условии, что студент умеет учиться самостоятельно.

Обобщённо говоря, студент может получить товарища или персонального учителя, который в состоянии помочь ему на всех этапах решения задачи, в том числе:

- по текстовой постановке задачи сделать математическую/алгоритмическую постановку;
- разработать алгоритм;
- написать программу по алгоритму;
- сделать локализацию ошибки, исправить её;
- придумать тест, на котором не работает программа при условии вдумчивого отношения студента к вопросам и ответам.

2.5. Направления использования ПГИИ в обучении преподавателем

В настоящее время автор планирует:

- для «проблемных» задач собирать и систематизировать «общение студентов с ПГИИ» (видеозапись и/или скрины диалога, результаты)
- по результатам анализа собранных материалов разработать методические рекомендации по работе с ПГИИ для решения учебных задач
- разработать рекомендации по обучению специальности «Оператор ПГИИ».

Заключение

В настоящей работе приведен обзор литературы по вводным курсам обучения программированию и проблемам и достижениями при использовании в этих курсах программ генеративного искусственного интеллекта. Далее приведен авторский опыт обучения программированию и использования программ генеративного искусственного интеллекта на факультете математики и технологий программирования Гомельского государственного университета им. Ф.Скорины. Автор считает, что в перспективе программы генеративного искусственного интеллекта могут существенно повысить персонализацию, а, следовательно, и качество обучения. Однако для этого требуется не только развитие самих программ, но корректировка методик в сторону подготовки студентов к обучению с помощью таких программ.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы на английском языке / References in English

1. Aldalur I. Improving Programming Learning in Engineering Students through Discovery Learning / I. Aldalur and X. Sagarna // IEEE Revista Iberoamericana de Tecnologías del Aprendizaje (Early Access). — DOI: 10.1109/RITA.2023.3301409
2. Shanmugam D.B. Approaches to Teaching Programming: a comprehensive review and analysis / D.B. Shanmugam, Dr.N. Vijayalakshmi, N. Revathi // chapter in book "Research in Multidisciplinary Subjects" (Volume-2) / under ed. Biplab Auddy. — 2023
3. Belmar H.M. Review on the Teaching of Programming and Computational Thinking in the World / H.M. Belmar // Front. Comput. Sci. — 4, 997222. — DOI: 10.3389/fcomp.2022.997222
4. CC 2020: Computing Curricula 2020: Paradigms for Global Computing Education. — URL: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf> (accessed: 10.12.2023)

5. Cheng Y.P. Applying Computational Thinking and Formative Assessment to Enhance the Learning Performance of Students in Virtual Programming Language / Y.P. Cheng, S.C. Cheng, M. Yang [et al.]; In: Huang, Y.M., Rocha, T. (eds) // *Innovative Technologies and Learning. ICITL 2023. Lecture Notes in Computer Science*, vol 14099. Springer, Cham. — DOI: 10.1007/978-3-031-40113-8_13
6. Xu X. Teaching Reform of Virtual Instrument Technology and Application Course Based on OBE Concept / X. Xu // *Open Access Library Journal*. — 2023. — 10. — 1-9. — DOI: 10.4236/oalib.1110491
7. Fonseca I. A Web-based Platform and a Methodology to Teach Programming Languages in Electrical Engineering Courses / I. Fonseca, N.C. Martins and F. Lopes // *2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)*. — Coimbra, Portugal, 2022. — 1-6. — DOI: 10.1109/EAEEIE54893.2022.9820594
8. Mamman M. Computerized E-learning Platform for Teaching Python Programming Using Hausa Language / M. Mamman, R.S. Rdadi, M.A. Sadiq // *World Journal of Advanced Engineering Technology and Sciences*. — 2023. 09(02). — 116–123. — DOI: 10.30574/wjaets.2023.9.2.0214
9. Zhang S. Exploring the Applications of EduCoder Platform in Blended Teaching for Computer Major / S. Zhang, J. Yang, X. Sang // *Journal of Education and Educational Research*. — 2023. — 4(2). — URL: <https://drpress.org/ojs/index.php/jeer/article/view/10819/10528> (accessed: 10.12.2023)
10. Grindei L. Project-oriented Approach in Teaching Programming to the First Year Undergraduate Students in Electrical Engineering / L. Grindei, C. Constantinescu, A. Bojita [et al.] // *2023 10th International Conference on Modern Power Systems (MPS)*. — Cluj-Napoca, Romania, 2023. — 1-4. — DOI: 10.1109/MPS58874.2023.10187591
11. Lewis J.E. Comparing First-Year Student Programming Confidence Perceptions Between Different Hands-On Projects / J.E. Lewis, N. Hawkins, B.S. Robinson // *Paper presented at 2023 ASEE Annual Conference & Exposition*. — Baltimore, Maryland. — URL: <https://peer.asee.org/43274> (accessed: 10.12.2023)
12. Vakaliuk T. The Use of ICT of Educational Purposes in the Independent Work of Students in the Study of the Basics of Programming by Future Software Engineers / T. Vakaliuk, O. Chyzhmotria, D. Antoniuk [et al.] // *In Proceedings of the 2nd Myroslav I. Zhaldak Symposium on Advances in Educational Technology (AET 2021)*. — 644-657 — DOI: 10.5220/0012066700003431
13. Mosobalaje O.O. Introducing Python Coding to Petroleum Engineering Undergraduates: Excerpts from a Teaching Experience / O.O. Mosobalaje and O.D. Orodu // *Paper presented at the SPE Nigeria Annual International Conference and Exhibition*. — Lagos, Nigeria, July 2023. — DOI: 10.2118/217148-MS
14. Othman J. Factors Affecting Engineering Students in Learning the Programming Subject at UiTM Pulau Pinang: A Study on Educators' Ability / J. Othman, S. B. Mahlan, R. Kadar [et al.] // *International Journal of Academic Research in Progressive Education and Development*. — 2023. — 12(2). — 944–953. — DOI: 10.6007/IJARPED/v12-i2/17359
15. Nguyen A. Teaching DevOps and Software Engineering Practices Using an Automated Programming Assessment System / A. Nguyen, V. Ha. — 20Jul.2023. — URL: <https://journal.tvu.edu.vn/index.php/journal/article/view/2104> (accessed: 10.12.2023)
16. Puztai K. Education supported by games even in university? / K. Puztai // *IntechOpen*. — DOI: 10.5772/intechopen.1002208
17. Zakiah N. A Review of Gamification for Learning Programming Language / N. Zakiah, D. Khalid, W. Hidayah // *International Conference on Business Studies and Education (ICBE 2023)*. — URL: <https://www.icbe.my/wp-content/uploads/2023/06/5-1.pdf> (accessed: 10.12.2023)
18. Zhuang Y. OPTES: A Tool for Behavior-based Student Programming Progress Estimation / Y. Zhuang, L. Wang, M. Zhang [et al.] // *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. — 122-131. — DOI: 10.1109/COMPSAC57700.2023.00025.
19. Oosterwyk G. Beyond the Hype: A Cautionary Tale of ChatGPT in the Programming Classroom / G. Oosterwyk, P. Tsiobolane, P. Kautondokwa [et al.] // *Conference: SACLA2023*. — URL: https://www.researchgate.net/publication/372495260_Beyond_the_Hype_A_Cautionary_Tale_of_ChatGPT_in_the_Programming_Classroom/link/64ba588fc41fb852dd8d4b11/download (accessed: 10.12.2023)
20. Jacques L. Teaching CS-101 at the Dawn of ChatGPT / L. Jacques // *ACM Inroads*. — 2023. — 14(3). — 40–46. — DOI: 10.1145/3595634
21. Bengtsson D. Assessment Accuracy of a Large Language Model on Programming Assignments Degree Project in Computer Science and Engineering / D. Bengtsson, A. Kaliff. — 2023. — URL: <https://www.diva-portal.org/smash/get/diva2:1779792/FULLTEXT01.pdf> (accessed: 10.12.2023)
22. Denny P. Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators / P. Denny, A. Luxton-Reilly, J. Leinonen [et al.] — arXiv:2307.16364v1 [cs.HC] 31 Jul 2023. — URL: <https://arxiv.org/pdf/2307.16364.pdf> (accessed: 10.12.2023)
23. Zheng Y. ChatGPT for Teaching and Learning: An Experience from Data Science Education / Y. Zheng // *In The 23rd Annual Conference on Information Technology Education (SIGITE '23)*. — URL: <https://arxiv.org/pdf/2307.16650.pdf> (accessed: 10.12.2023)
24. Farah J. Harnessing Rule-Based Chatbots to Support Teaching Python Programming Best Practices International Conference on Interactive Collaborative Learning (ICL2023) / J. Farah, S. Ingram, A. Purohit [et al.] — Madrid, Spain, September 26-29, 2023. — URL: <https://infoscience.epfl.ch/record/303628> (accessed: 10.12.2023)
25. Dolinsky M. An Approach to Teaching Introductory-Level Computer Programming / M. Dolinsky // *Olympiads in Informatics*. — 2013. — 7. — 14-22. — URL: <https://iainformatics.org/journal/INFOL114.pdf> (accessed: 10.12.2023)

26. Dolinsky M. Teaching Algorithms and Programming First Year University Students on Base of Distance Learning System DL.GSU.BY / M. Dolinsky // WSEAS Transactions on Advances in Engineering Education. — 2022a. — 19. — 52-57. — URL: <https://wseas.com/journals/articles.php?id=5611> (accessed: 10.12.2023)
27. Dolinsky M. Instrumental System of Distance Learning DL.GSU.BY and Examples of its Application / M. Dolinsky // Global Journal of Computer Science and Technology Interdisciplinary. — 2022e. — 22(1). — 45-53. — URL: https://globaljournals.org/GJCST_Volume22/6-Instrumental-System-of-Distance-Learning.pdf (accessed: 10.12.2023)
28. Dolinsky M. A New Generation Distance Learning System for Programming and Olympiads in Informatics / M. Dolinsky // Olympiads in Informatics. — 2017. — 11. — 29-39. — URL: https://ioinformatics.org/journal/v11_2017_29_39.pdf (accessed: 10.12.2023)
29. Dolinsky M. Gomel Training School for Olympiads in Informatics / M. Dolinsky // Olympiads in Informatics. — 2016. — 10. — 237-24. — URL: https://ioinformatics.org/journal/v10_2016_237_247.pdf (accessed: 10.12.2023)