

УДК 378.147:004.42

СОВРЕМЕННЫЕ ПОДХОДЫ В ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ БУДУЩИХ УЧИТЕЛЕЙ ИНФОРМАТИКИ

И.А. Яшина (Красноярск, Россия)

Аннотация

Постановка проблемы. В условиях цифровой трансформации образования сохраняется ряд системных проблем в предметной подготовке будущих учителей информатики по программированию. Среди них выделяются недостаточный уровень мотивации и алгоритмического мышления у абитуриентов, применение устаревших методов обучения, сосредоточенных на синтаксисе и вычислительных задачах, неоднозначный выбор языка программирования, а также дефицит педагогической интеграции, не позволяющий студентам научиться адаптировать материал и диагностировать ошибки школьников.

Цель статьи заключается в анализе современных методических подходов к обучению программированию студентов педагогических вузов и описании подходов формирования их готовности к преподаванию программирования в школе.

Методология (материалы и методы) исследования. В качестве методологии исследования применен анализ современных научных публикаций за последние шесть лет, посвященных практике обучения программированию в педагогических вузах. Также обобщен собственный опыт реализации образовательных программ в Красноярском государственном педагогическом университете.

Результаты. На основе анализа успешных практик (сквозные проекты, итерационное, персонализированное, STEM-обучение и др.) синтезированы ключевые подходы в подготовке: предметно-методическая интеграция, практико-ориентированный и контекстный характер обучения, осознанное формирование педагогического инструментария и развитие метапредметных компетенций.

Заключение. Установлено, что готовность будущего учителя к преподаванию программирования формируется в комплексной образовательной среде, обеспечивающей синергию глубоких предметных знаний (на базе школьно-релевантных технологий, таких как Python), инженерно-проектного мышления и педагогической компетентности. Ключевым фактором является не внедрение отдельных инноваций (в т.ч. ИИ), а их интеграция в образовательную экосистему под руководством критически мыслящего преподавателя. Перспективы видятся в углублении фундаментальной подготовки, усилении методической рефлексии и разработке моделей эффективного взаимодействия «педагог-технология-методология».

Ключевые слова: обучение программированию, подготовка учителей информатики, педагогические подходы, Python, профессиональная готовность.

Яшина Ирина Александровна – кандидат педагогических наук, доцент кафедры информатики и информационных технологий в образовании, Красноярский государственный педагогический университет им. В.П. Астафьева; e-mail: kulakova_do@kspu.ru

Постановка проблемы. В современном мире информационно-коммуникационных технологий знание языка программирования является одной из базовых составляющих информационной культуры человека. Известная цитата академика Ершова о том, что «программирование – вторая грамотность», становится все более актуальной в наше время, тем более что информационные технологии проникли практически во все сферы деятельности

человека. Особенно актуальным становится знание языка программирования, если человек желает связать свою деятельность с IT-индустрией [Мартынук, 2018].

Основная задача учителя информатики в этом контексте – формирование ИКТ-компетентности обучающихся, включая тех, кто ориентирован на современные IT-профессии.

Ключевым элементом фундаментальной предметной подготовки самого педагога, при-

званного решать эту задачу, выступает обучение программированию. При этом содержание и методы этого обучения должны адекватно отражать актуальный уровень развития профессиональной области программирования.

Несмотря на достаточную разработанность теоретических и учебно-методических основ обучения программированию в литературе, в предметной подготовке студентов педагогических вузов сохраняется ряд существенных проблем.

Анализ современных исследований [Пугач, Тюжина, Макарова, 2019; Розов, Подсадников, 2019; Гафуанов, Поднебесова, 2020; Круподеровы, 2020; Джураева, 2021; Сайфурова, Рагулина, Федорова, 2022] позволяет выделить следующие ключевые проблемы.

Во-первых, наблюдается недостаточный уровень мотивации к изучению языков программирования высокого уровня, а также низкий уровень развития алгоритмического мышления у абитуриентов, поступающих в педагогические вузы.

Во-вторых, устаревшие методы обучения, акцентирующие внимание на вычислительных задачах или формальном изучении синтаксиса, не способствуют формированию навыков решения реальных педагогических задач.

В-третьих, выбор языка программирования и среды разработки остается спорным вопросом, что может привести к проблемам технологического устаревания.

Наконец, существует дефицит педагогической интеграции. Студенты не обучаются адаптировать сложность материала, диагностировать ошибки школьников или проектировать мотивирующие задания.

Указанные проблемы свидетельствуют о том, что подготовка будущих учителей информатики к преподаванию программирования в школе является критически важной и сложной задачей, особенно в условиях цифровой трансформации образования. Для ее решения необходимы эффективные подходы, обеспечивающие как глубокое понимание принципов программирования, так и развитие педагогических умений.

Цель работы заключается в анализе современных подходов в обучении программированию студентов педагогических вузов и описании подхода к формированию готовности учить программированию, основанного на фундаментальном понимании принципов программирования и развитии учебной самостоятельности будущих учителей информатики.

Методология (материалы и методы) исследования. В качестве основных методов исследования определен анализ научной психолого-педагогической литературы, обобщен собственный опыт реализации образовательных программ в Красноярском государственном педагогическом университете им. В.П. Астафьева. Синтез методических подходов обучения программированию в педагогических вузах позволил сформулировать обобщенные подходы к обучению и формированию готовности будущих учителей информатики к преподаванию программирования в школе.

Обзор литературы. Вопросы эффективного обучения программированию на различных ступенях обучения волнуют исследователей и не перестают быть актуальными в настоящее время. Анализ проблематики логично начать с рассмотрения общих вызовов, с которыми сталкивается высшее образование в этой области.

В обучении программированию студентов в целом, а не только будущих учителей можно выделить следующие проблемы: высокая когнитивная нагрузка обучаемых, отрыв от практики, высокая нагрузка на преподавателей, проблема качественной и своевременной обратной связи, отставание образовательных программ от потребностей рынка труда [Telesko et al., 2023; Лавина, Ильина, 2025]. Пандемия COVID-19 заставила образовательные учреждения экстренно переходить на дистанционное обучение, что создало множество дополнительных проблем, но в то же время способствовало развитию дистанционных форм обучения и новых методических подходов в обучении.

Ответом на эти вызовы становятся новые методологии. Успешный опыт зарубежных авторов [Kularbphettong et al., 2024] по разработке

и аprobации модели смешанного обучения заслуживает внимания. Ключевыми факторами успеха стали не технологии сами по себе, а методический подход, эффективность которого напрямую зависит от продуманности среды обучения, обеспечивающей синергию аудиторной работы и онлайн-компонентов. Подходы, подобные Model-based Agile Programming (MBAD) [Telesko et al., 2023], предлагают переосмыслить саму суть преподавания программирования – от натаскивания в синтаксисе к развитию компетенций в области анализа, проектирования и моделирования.

Одним из самых значимых современных трендов, трансформирующих образовательный процесс, является использование искусственного интеллекта (ИИ). Анализ научных исследований в области обучения программированию последних лет [Цепляев, 2023; Sheese et al., 2024; Budhiraja et al., 2024; Ahmed et al., 2025] показывает смещение научного интереса в сторону использования искусственного интеллекта в обучении. На наш взгляд, это обусловлено в первую очередь эффективным использованием ИИ-агентов в профессиональной разработке. Интеграция больших языковых моделей (LLM) в интегрированную среду разработки (IDE) представляет собой значительный шаг в эволюции инструментов разработки, предлагая существенные преимущества в автоматизации и качестве кода.

Использование ИИ-инструментов в обучении программированию носит двойственный характер. С одной стороны, они открывают новые педагогические возможности, смешая фокус на алгоритмическое мышление и анализ кода. С другой – требуют осторожного внедрения, чтобы не подменить глубокое обучение поверхностным использованием готовых решений. Будущее обучение, вероятно, будет делать больший акцент на чтении, оценке и модификации кода, чем на его написании с нуля [Цепляев, 2023].

ChatGPT стал неотъемлемой частью учебного процесса, выступая в роли помощника в программировании, «мозгового центра» и ре-

петитора. Однако его широкое и долгосрочное использование сдерживается проблемами точности, юзабилити и этическими вопросами [Budhiraja et al., 2024]. Исследование [Ahmed et al., 2025] проверяет целесообразность использования LLM для ассистентов преподавателей в курсе программирования. Вместо того чтобы выдавать студентам готовые ответы ИИ, авторы предложили гибридную модель: ИИ генерирует черновик обратной связи, который человек-ассистент проверяет, редактирует и затем отправляет студенту. Гибридная модель для обратной связи в программировании технически возможна, но не привела к ожидаемому улучшению результатов. Основной проблемой оказалась пассивность людей, которые перестали критически проверять выводы ИИ. Авторы делают вывод, что ИИ должен не заменять человеческое участие, а интегрироваться в учебный процесс, чтобы реально поддерживать, а не подрывать работу педагогов.

Авторский коллектив [Sheese et al., 2024] провел исследование по использованию инновационного инструмента на базе LLM, который предоставляет помощь в программировании по запросу, не раскрывая решения напрямую. Результаты данного исследования демонстрируют, что студенты в основном используют ИИ-помощника для сиюминутной тактической помощи (исправить ошибку, написать код), а не для глубокого концептуального обучения. В то время как обучение программированию не ограничивается решением насущных задач в области программирования, но также предполагает создание прочной основы базовых концепций.

На фоне общих трендов и вызовов особую актуальность приобретает анализ подходов к подготовке будущих учителей информатики, которые должны не только освоить программирование сами, но и научиться его преподавать. Важно понимание проблем обучения программированию на различных уровнях. Л.Л. Босовой проведен анализ современных подходов к обучению программированию в школах России и других стран. Автор рассматривает

историю, текущее состояние, проблемы и перспективы развития этой области, а также выделяет ключевые международные тенденции [Босова, 2018]:

раннее, обязательное и непрерывное обучение программированию;

использование разнообразных современных сред (Scratch, Python, облачные сервисы);

проектный подход и командная работа для развития практических навыков;

интеграция с робототехникой и реальными задачами.

Сравнительный анализ систем обучения программированию в разных странах позволяет сделать вывод о необходимости реформирования российского образования в области программирования с учетом мирового опыта, чтобы развивать у школьников алгоритмическое мышление и практические навыки, соответствующие требованиям ХХI в.

Анализ современных исследований за последние шесть лет в области обучения программированию будущих учителей информатики [Пугач, Тюжина, Макарова, 2019; Розов, Подсадников, 2019; Гафуанов, Поднебесова, 2020; Круподерова, Круподерова, 2020; Джираева, 2021; Сайфурова, Рагулина, Федорова, 2022] выявляет ряд значимых методических подходов (табл.), которые не только развивают предметные компетенции студентов, но и целенаправленно формируют их педагогическое мышление, включая умение адаптировать сложный материал для школьной аудитории и критически оценивать образовательные технологии, тем самым внося ключевой вклад в профессиональную готовность будущих учителей информатики. В контексте данного анализа представляет интерес и опыт Красноярского государственного педагогического университета им. В.П. Астафьева, который также был обобщен и включен в таблицу.

Методические подходы обучения программированию в различных педагогических вузах

Methodological approaches to teaching programming in various pedagogical universities

Подход	Суть методики	Решаемая проблема
1	2	3
Комплексный подход с фокусом на преемственности и педагогическую адаптацию (КГПУ)	Двухуровневая структура: изучение Python и C++. Курс Python выполняет роль выравнивающего курса – от основ до ООП. Интеграция с дисциплинами: «робототехника», «школьный практикум», «численные методы» и другие, что закрепляет знания и демонстрирует студентам межпредметные связи и педагогический контекст применения программирования. Ключевой элемент – интеграция современных образовательных технологий: облачные среды, LMS Moodle с автоматизированной проверкой заданий (плагин CodeRunner) и внешние обучающие платформы (Питонтыор). Активно исследуется и внедряется использование ИИ-инструментов (LLM) для поддержки обучения	Недостаточная базовая подготовка абитуриентов; разрыв между академическими знаниями и требованиями школьного преподавания; необходимость формирования не только технических, но и педагогических компетенций; высокая нагрузка на преподавателя при проверке заданий; необходимость обеспечения оперативной обратной связи
Сквозные проекты со спиральным усложнением	Эволюционное развитие одного проекта (например, игры) на протяжении курса: от простой консольной версии к сложной (с GUI, внешними данными, рефакторингом). Этапы синхронизированы с темами курса	Низкая мотивация из-за разрозненности задач; отсутствие связи с практикой; «отложенный старт» проектной деятельности
Итерационное обучение	Организация учебного процесса как пошагового уточнения и расширения программной реализации. Лабораторные работы: теория → модификация готового кода → самостоятельное задание. Лекции синхронизированы с практикой	Формальное изучение теории, оторванное от практики; быстрое забывание материала; трудности в самостоятельном создании программ с нуля

Окончание табл.

1	2	3
Персонализированное обучение на основе ИОМ и мобильных технологий	Построение индивидуальных образовательных маршрутов (ИОМ) с помощью интерактивной инфографики. Использование мобильных технологий (чат-боты, облачные среды, Web 2.0) для поддержки выбора уровня сложности, ресурсов, динамической коррекции траектории. Этапы: индивидуализированное → командное → метаобъектное обучение	Неоднородный уровень подготовки студентов; недостаток индивидуальных консультаций; низкая мотивация и трудности в понимании объектно ориентированного программирования
Интеграция STEM + «Перевернутый класс»	Сочетание проблемно-семиотического подхода (Science), современных технологий (Technology), инженерного проектирования (Engineering), математического моделирования (Math). Использование видеозаписей работы профессионалов («Перевернутый класс») для демонстрации реального процесса разработки ПО	Разрыв между академическим обучением и профессиональной практикой; технологическое устаревание знаний; недостаток времени на аудиторную практику
Модульная система с инженерным уклоном и внеаудиторной деятельностью	Структуризация на 2 модуля: «Математические основы» (алгоритмы, анализ сложности) и «Программирование» (ООП, Web). Акцент на портфолио алгоритмов, сравнительных проектах, хакатонах, командной разработке. Использование альтернативных методов оценки (SWOT-анализ жизненного цикла ПО)	Недостаток системного мышления; слабые навыки полного цикла разработки; дефицит soft skills (коллaborация, презентация)
Python-ориентированный курс с дидактическими связями	Обучение на Python с акцентом на его дидактических преимуществах (лаконичность) и преемственности (от основ к ИИ). Использование Turtle для связи блочного (Scratch) и текстового программирования. Раннее введение строк, работа с файлами, SymPy для математиков. Критический анализ рисков (динамическая типизация)	Неактуальный выбор языка (Pascal/Delphi); трудности перехода от блочного к текстовому коду; слабая связь с современными технологиями (ИИ)

Результаты исследования. Проведенный анализ современных исследований позволяет выделить *три ключевых направления современных подходов в обучении программированию: системность и непрерывность – раннее начало, обязательность обучения, сквозные линии от школы к вузу; практикоориентированность – переход от теоретического обучения к активным методам: кейс-технологии, проектный подход, командная работа, деятельностный подход; интеграция технологий ИИ – использование языковых моделей для снижения когнитивной нагрузки, фокусировки на алгоритмическом мышлении, развития навыков анализа и оценки кода.*

Современные подходы требуют пересмотра образовательных парадигм с акцентом на формировании не только технических, но и системных, критических и творческих компетенций. Успешная интеграция ИИ требует не только совершенствования моделей, сколько разработки четких педагогических протоколов их использования и критического взаимодействия с искусственным интеллектом.

Синтез рассмотренных методических подходов обучения программированию в педагогических вузах позволяет сформулировать следующие обобщенные принципы построения образовательного процесса:

Во-первых, необходима предметно-методическая интеграция: освоение языка (предпочтительно Python, учитывая его школьную релевантность) должно сопровождаться постоянной рефлексией его дидактических преимуществ, ограничений и типичных трудностей учащихся. Конечно, методическая сторона обучения программированию рассматривается в рамках дисциплины «Методика обучения информатике», однако даже в предметной подготовке необходима такая интеграция.

Во-вторых, критически важен практико-ориентированный и контекстный характер обучения. Проектная деятельность, будь то сквозные «спиральные» проекты или проекты в рамках модулей, должна максимально приближаться к реальным задачам, с которыми столкнутся учителя: создание учебных игр, разработка инструментов автоматизации проверки заданий, интеграция с актуальными образовательными платформами и технологиями. Это формирует не только навыки программирования, но и способность видеть педагогический потенциал программных решений.

В-третьих, требуется осознанное формирование педагогического инструментария. Будущий учитель должен не только владеть Python, но и уметь использовать различные библиотеки (Turtle, Tkinter), а также современные технологии, в том числе и технологии ИИ.

В-четвертых, необходима развитая система формирования метапредметных и профессиональных компетенций. Сюда входит развитие системного мышления через анализ алгоритмов и проектирование (сильная сторона модульного подхода), soft skills через хакатоны и командную работу, навыков рефакторинга и

адаптации сложности (практика «спиральных задач»), критического мышления при выборе технологий. Построение индивидуальных маршрутов обучения и инновационные методы оценки (SWOT-анализ, кейсы, балльно-рейтинговая оценка) также вносят вклад в профессиональное становление.

Заключение. Проведенный анализ позволяет утверждать, что современные успешные практики обучения программированию будущих учителей информатики конвергируют к модели комплексной образовательной среды. Ее ключевая задача – обеспечить неразрывный синтез глубокого предметного знания (на базе школьно-релевантных технологий, таких как Python), инженерно-проектного мышления и педагогической компетентности, формируя у будущего педагога умение адаптировать сложность, диагностировать ошибки и проектировать мотивирующие задания.

Таким образом, готовность учить программированию формируется не просто знанием языков, а способностью видеть за технической задачей – педагогическую, за синтаксисом – дидактический потенциал, а за учебным проектом – модель профессиональной деятельности.

Перспективы развития видятся в углублении этой интеграции через усиление методической рефлексии на всех этапах обучения, тесную связь с реальной школьной практикой и осознанное использование технологического стека (включая ИИ) как инструмента для решения педагогических задач. Ключевым фактором остается не внедрение отдельных инноваций, а создание целостной образовательной экосистемы, где критически мыслящий преподаватель управляет технологиями для достижения педагогических целей.

Библиографический список

- Босова Л.Л. Как учат программированию в XXI веке: отечественный и зарубежный опыт обучения программированию в школе. Информатика в школе. 2018. № 6. С. 3–11. URL: <https://school.infojournal.ru/jour/article/view/254/254> (дата обращения: 20.10.2025).
- Гафуанов Я.Ю., Поднебесова Г.Б. Формирование профессиональной ИКТ-компетентности при обучении программированию будущих учителей информатики и ИТ-специалистов // Вестник Томского государственного университета. 2020. № 455. С. 175–182. DOI: 10.17223/15617793/455/24

3. Губашева Х.А., Магамедова Д.М., Магазиева З.А. Инновационные методы обучения программированию и ИТ в российских вузах // Международный научно-исследовательский журнал. 2022. Ч. 3, № 5 (119). С. 69–71. URL: <https://research-journal.org/archive/5-119-2022may/innovacionnye-metody-obucheniya-programmirovaniyu-i-it-v-rossijskix-vuzax> (дата обращения: 20.10.2025).
4. Джураева Д.Р. Средства и формы организации обучения программированию в высшей школе будущих учителей информатики // Экономика и социум. 2021. № 5-2 (84). URL: <https://cyberleninka.ru/article/n/sredstva-i-formy-organizatsii-obucheniya-programmirovaniyu-v-vysshey-shkole-buduschihih-uchiteley-informatiki> (дата обращения: 20.10.2025).
5. Круподерова Е.П., Круподерова К.Р. Обучение языкам и технологиям программирования как компонент предметной подготовки будущих учителей информатики // Проблемы современного педагогического образования. 2020. № 69-3. URL: <https://cyberleninka.ru/article/n/obuchenie-yazykam-i-tehnologiyam-programmirovaniya-kak-komponent-predmetnoy-podgotovki-buduschihih-uchiteley-informatiki> (дата обращения: 20.10.2025).
6. Лавина Т.А., Ильина Л.А. Современные подходы к обучению программированию при подготовке кадров в области информационной безопасности в условиях формирования национальной системы высшего образования // Вестник Череповецкого государственного университета. 2025. № 1. С. 219–230. URL: <https://cyberleninka.ru/article/n/sovremennoye-podhody-k-obucheniyu-programmirovaniyu-pri-podgotovke-kadrov-v-oblasti-informatsionnoy-bezopasnosti-v-usloviyah> (дата обращения: 20.10.2025).
7. Мартынюк А.А. Интегрированный подход при обучении программированию в средней школе // Интерактивная наука. 2018. № 2 (24). URL: <https://cyberleninka.ru/article/n/integrirovannyyu-podhod-pri-obuchenii-programmirovaniyu-v-sredney-shkole> (дата обращения: 20.10.2025).
8. Пугач В.И., Тюжина И.В., Макарова Е.Л. Обучение программированию будущих учителей информатики: задачи со спирально повышающейся сложностью // Самарский научный вестник. 2019. № 4 (29). URL: <https://cyberleninka.ru/article/n/obuchenie-programmirovaniyu-buduschihih-uchiteley-informatiki-zadachi-so-spiralno-povyshayuscheysya-slozhnostyu> (дата обращения: 20.10.2025).
9. Розов К.В., Подсадников А.В. Язык программирования Python в педагогическом вузе: от основ до искусственного интеллекта // Информатика и образование. 2019. № 6. С. 26–33. DOI: 10.32517/0234-0453-2019-34-6-26-33
10. Сайфурова И.О., Рагулина М.И., Федорова Г.А. Персонализированное обучение будущих учителей информатики программированию на основе применения мобильных технологий // Наука о человеке: гуманитарные исследования. 2022. № 4 (16). С. 116–125. DOI: io.570i5/issni998-5320.2022.i6.4.i4
11. Цепляев А.Ф. Использование языковых моделей искусственного интеллекта для изучения основ программирования // Символ науки. 2023. № 5-2. С. 58–60. URL: <https://cyberleninka.ru/article/n/ispolzovanie-yazykovyh-modeley-iskusstvennogo-intellekta-dlya-izucheniya-osnov-programmirovaniya> (дата обращения: 25.10.2025).
12. Яшина И.А. Искусственный интеллект в обучении программированию студентов педагогического вуза // Открытое образование. 2024. № 4 (28). С. 23–32. DOI: 10.21686/1818-4243-2024-4-23-32
13. Ahmed, U., Sahai, S., Leong, B., & Karkare, A. (2025). Feasibility study of augmenting Teaching Assistants with AI for CS1 Programming Feedback. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSETS 2025)* (pp. 11–17). Association for Computing Machinery, NY, USA. DOI: 10.1145/3641554.3701972

14. Budhiraja, R., Joshi, I., Challa, J., Akolekar, H., & Kumar, D. (2024). "It's not like Jarvis, but it's pretty close!" – Examining ChatGPT's usage among undergraduate students in Computer Science. In *Proceedings of the 26th Australasian Computing Education Conference (ACE '24)* (pp. 124–133). Association for Computing Machinery, NY, USA. DOI: 10.1145/3636243.3636257
15. Chiodini, L., Bevilacqua, J., & Hauswirth M. (2025). Surveying upper-secondary teachers on programming misconceptions. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V. 1 (ICER '25)* (pp. 166–180). Association for Computing Machinery, NY, USA. DOI: 10.1145/3702652.3744227
16. Faherty, R., Quille, K., Vivian, R., McGill, M., Becker, B., & Nolan, K. (2021). Comparing programming self-esteem of upper secondary school teachers to CS1 students. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE '21)* (pp. 554–560). Association for Computing Machinery, NY, USA. DOI: 10.1145/3430665.3456372
17. Kularbphettong, K., Raksuntorn, N., Kumchoo, S., & Boonseng, Ch. (2024). Assessing programming language teaching during COVID19 with blended learning. In *Proceedings of the 2024 10th International Conference on e-Society, e-Learning and e-Technologies (ICSLT) (ICSLT '24)* (pp. 20–24). Association for Computing Machinery, New York, USA. DOI: 10.1145/3678610.3678614
18. Sheese, B., Liffiton, M., Savelka, J., & Denny, P. (2024). Patterns of student help-seeking when using a Large Language Model-powered Programming Assistant. In *Proceedings of the 26th Australasian Computing Education Conference (ACE '24)* (pp. 49–57). Association for Computing Machinery, NY, USA. DOI: 10.1145/3636243.3636249
19. Telesko, R., Spahic-Bogdanovic, M., Hinkelmann, K., & Pande, C. (2023). A new approach for teaching programming: Model-based Agile Programming (MBAD). In *Proceedings of the 8th International Conference on Information and Education Innovations (ICIEI '23)* (pp. 13–18). Association for Computing Machinery, NY, USA. DOI: 10.1145/3594441.3594445

MODERN APPROACHES TO TEACHING PROGRAMMING TO FUTURE COMPUTER SCIENCE TEACHERS

I.A. Yashina (Krasnoyarsk, Russia)

Abstract

Statement of the problem. In the context of the digital transformation of education, a number of systemic problems persist in the subject-specific training of future computer science teachers in programming. These include insufficient levels of motivation and algorithmic thinking among applicants, the use of outdated teaching methods focused on syntax and computational tasks, ambiguous choice of programming language, and a lack of pedagogical integration, which prevents students from learning how to adapt material and diagnose school students' errors.

The purpose of the article is to analyze modern methodological approaches to teaching programming to students of pedagogical universities and to describe approaches to developing their readiness to teach programming at school.

The research methodology (materials and methods). The research methodology involved the analysis of modern scientific publications from the last six years, devoted to the practice of teaching programming in pedagogical universities. The author's own experience in implementing educational programs at the Krasnoyarsk State Pedagogical University is also generalized.

Research results. Based on the analysis of successful practices (scaffolded projects, iterative, personalized, and STEM learning, etc.), key approaches in training are synthesized: subject-methodological integration, practice-oriented and contextual nature of learning, conscious formation of pedagogical tools, and the development of meta-subject competencies.

Conclusion. It is established that the future teacher's readiness to teach programming is formed in a comprehensive educational environment that ensures the synergy of deep subject knowledge (based on school-relevant technologies such as Python), engineering-project thinking, and pedagogical competence. The key factor is not the implementation of individual innovations (including AI), but their integration into the educational ecosystem under the guidance of a critically thinking teacher. Prospects are seen in deepening fundamental training, enhancing methodological reflection, and developing models of effective 'teacher-technology-methodology' interaction.

Keywords: *programming instruction, computer science teacher training, pedagogical approaches, Python, professional readiness.*

Yashina, Irina A. – PhD (Pedagogy), Associate Professor, Department of Informatics and Information Technology in Education, Krasnoyarsk State Pedagogical University named after V.P. Astafyev (Krasnoyarsk, Russia); e-mail: kulakova_do@kspu.ru

References

1. Bosova, L.L. (2018). How to teach programming in the 21st Century: Russian and foreign experience in teaching programming at school. *Informatika v shkole* [Informatics at School], 6, 3–11. URL: <https://school.infojournal.ru/jour/article/view/254/254> (access date: 20.10.2025).
2. Gafuanov, Y.Y., & Podnebesova, G.B. (2020). The formation of professional ICT-competence in teaching programming to future teachers of Informatics and IT experts. *Vestnik Tomskogo gosudarstvennogo universiteta* [Tomsk State University Journal], 455, 175–182. DOI: 10.17223/15617793/455/24
3. Gubasheva, H.A., Magamedova, D.M., & Magazieva, Z.A. (2022). Innovative methods of teaching programming and it in Russian Universities. *Mezhdunarodniy nauchno-issledovatel'skiy zhurnal* [International Research Journal], 3–5 (119), 69–71. URL: <https://research-journal.org/archive/5-119-2022may/innovacionnye-metody-obucheniya-programmirovaniyu-i-it-v-rossijskix-vuzax> (access date: 20.10.2025).
4. Dzhuraeva, D.R. (2021). Means and forms of organization of teaching programming in higher school for future teachers of informatics. *Ekonomika i sotsium* [Economy and Society], 5-2 (84). URL: [https://cyberleninka.ru/article/n/sredstva-i-formy-organizatsii-obucheniya-programmirovaniyu-v-vysshay-shkole-buduschih-uchiteley-informatiki](https://cyberleninka.ru/article/n/sredstva-i-formy-organizatsii-obucheniya-programmirovaniyu-v-vysshey-shkole-buduschih-uchiteley-informatiki) (access date: 20.10.2025).

5. Krupoderova, E.P., & Krupoderova, K.R. (2020). Teaching programming languages and technologies as a component of subject-specific training for future computer science teachers. *Problemy sovremennoogo pedagogicheskogo obrazovaniya* [Problems of Modern Pedagogical Education], 69 (3). URL: <https://cyberleninka.ru/article/n/obuchenie-yazykam-i-tehnologiyam-programmirovaniya-kak-komponent-predmetnoy-podgotovki-buduschih-uchiteley-informatiki> (access date: 20.10.2025).
6. Lavina, T.A., & Ilina, L.A. (2025). Programming training in the field of information security in the context of the development of the national higher education system. *Vestnik Cherepovetskogo gosudarstvennogo universiteta* [Cherepovets State University Bulletin], 1 (124), 219–230. URL: <https://cyberleninka.ru/article/n/sovremennye-podhody-k-obucheniyu-programmirovaniyu-pri-podgotovke-kadrov-v-oblasti-informatsionnoy-bezopasnosti-v-usloviyah> (access date: 20.10.2025).
7. Martynyuk, A.A. (2018). The integrated approach to teaching programming in secondary school. *Interaktivnaya nauka* [Interactive Science], 2 (24). URL: <https://cyberleninka.ru/article/n/integrirovannyy-podhod-pri-obuchenii-programmirovaniyu-v-sredney-shkole> (access date: 20.10.2025).
8. Pugach, V.I., Tyuzhina, I. V., & Makarova, E. L. (2019). Training programming to prospective it teachers: tasks with a spiral-increasing complexity. *Samarskiy nauchnyi vestnik* [Samara Scientific Bulletin], 4 (29). URL: <https://cyberleninka.ru/article/n/obuchenie-programmirovaniyu-buduschih-uchiteley-informatiki-zadachi-so-spiralno-povyshayuscheysya-slozhnostyu> (access date: 20.10.2025).
9. Rozov, K.V., & Podsadnikov, A.V. (2019). Python programming language in pedagogical university: from the bases to artificial intelligence. *Informatika i obrazovanie* [Computer Science and Education], 6, 26–33. DOI: 10.32517/0234-0453-2019-34-6-26-33
10. Saifurova, I.O., Ragulina, M. I., & Fedorova, G. A. (2022). Personalized programming training of future Computer Science teachers using the mobile technologies. *Nauka o cheloveke: gumanitarnie issledovaniya* [Russian Journal of Social Sciences and Humanities], 4 (16), 116–125. DOI: io.570i5/issni998-5320.2022.i6.4.i4
11. Tseplyaev, A.F. (2023) Using artificial intelligence language models to learn the basics of programming. *Simvol nauki* [Symbol of Science], 5 (2), 58–60. URL: <https://cyberleninka.ru/article/n/ispolzovanie-yazykovyh-modeley-iskusstvennogo-intellekta-dlya-izucheniya-osnov-programmirovaniya> (access date: 20.10.2025).
12. Yashina, I.A. (2024). Artificial intelligence in teaching programming to students of pedagogical university. *Otkrytoe obrazovanie* [Open Education], 28 (4), 23–32. DOI: 10.21686/1818-4243-2024-4-23-32
13. Ahmed, U., Sahai, S., Leong, B., & Karkare, A. (2025). Feasibility study of augmenting Teaching Assistants with AI for CS1 Programming Feedback. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (SIGCSETS 2025) (pp. 11–17). Association for Computing Machinery, NY, USA. DOI: 10.1145/3641554.3701972
14. Budhiraja, R., Joshi, I., Challa, J., Akolekar, H., & Kumar, D. (2024). “It’s not like Jarvis, but it’s pretty close!” – Examining ChatGPT’s usage among undergraduate students in Computer Science. In *Proceedings of the 26th Australasian Computing Education Conference* (ACE ’24) (pp. 124–133). Association for Computing Machinery, NY, USA. DOI: 10.1145/3636243.3636257
15. Chiodini, L., Bevilacqua, J., & Hauswirth M. (2025). Surveying upper-secondary teachers on programming misconceptions. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1* (ICER ‘25) (pp. 166–180). Association for Computing Machinery, NY, USA. DOI:10.1145/3702652.3744227
16. Faherty, R., Quille, K., Vivian, R., McGill, M., Becker, B., & Nolan, K. (2021). Comparing programming self-esteem of upper secondary school teachers to CS1 students. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (ITiCSE ‘21) (pp. 554–560). Association for Computing Machinery, NY, USA. DOI: 10.1145/3430665.3456372

17. Kularbphettong, K., Raksuntorn, N., Kumchoo, S., & Boonseng, Ch. (2024). Assessing programming language teaching during COVID19 with blended learning. In *Proceedings of the 2024 10th International Conference on e-Society, e-Learning and e-Technologies (ICSLT)* (ICSLT '24) (pp. 20–24). Association for Computing Machinery, New York, USA. DOI: 10.1145/3678610.3678614
18. Sheese, B., Liffiton, M., Savelka, J., & Denny, P. (2024). Patterns of student help-seeking when using a Large Language Model-powered Programming Assistant. In *Proceedings of the 26th Australasian Computing Education Conference (ACE '24)* (pp. 49–57). Association for Computing Machinery, NY, USA. DOI: 10.1145/3636243.3636249
19. Telesko, R., Spahic-Bogdanovic, M., Hinkelmann, K., & Pande, C. (2023). A new approach for teaching programming: Model-based Agile Programming (MBAD). In *Proceedings of the 8th International Conference on Information and Education Innovations (ICIEI '23)* (pp. 13–18). Association for Computing Machinery, NY, USA. DOI: 10.1145/3594441.3594445