

СИМУЛЯЦИОННОЕ ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА: УРОВНЕВАЯ МОДЕЛЬ И ИНСТРУМЕНТЫ

КОРОЛЁВ Владимир Владимирович

старший преподаватель

Ленинградский государственный университет им. А.С.Пушкина

г. Санкт-Петербург, Россия

Статья посвящена вопросам симуляционного обучения программированию с применением ИИ-технологий. Обосновывается переход к подготовке разработчиков в условиях ИИ-коллаборативного программирования. Представлена уровневая модель подготовки и соответствующие ей ИИ-симуляторы. Проведён анализ инструментов, выявлены проблемы фрагментарности, отсутствия платформенных решений.

Ключевые слова: симуляционное обучение, обучение программированию, искусственный интеллект в образовании, подготовка программистов, ИИ-симуляторы, коллаборативное программирование.

Современный этап цифровой трансформации характеризуется всё более активной интеграцией искусственного интеллекта (ИИ) в профессиональную деятельность программистов. Если первоначально ИИ-инструменты воспринимались как средства автоматизации рутинных операций, то сегодня они превращаются в коллаборативных агентов, участвующих в написании, отладке и оптимизации кода. Данный феномен, обозначаемый в научной литературе как переход к «Программированию 4.0», предполагает смену роли разработчика – от непосредственного кодирования к управлению интеллектуальными системами, что актуализирует пересмотр подходов к подготовке ИТ-специалистов [2].

Цель настоящего исследования – представить уровневую модель подготовки специалистов по разработке информационных систем, опирающуюся на идеи симуляционного обучения с применением ИИ-технологий, а также провести анализ существующих ИИ-симуляторов и обозначить ключевые методические проблемы их использования в учебном процессе.

Стоит признать, что сегодня процессы симуляционного обучения программированию переходят на новый уровень. Подобное становится возможным благодаря использованию ИИ-технологий. ИИ в программировании, по мнению Л. Мирабовой и М. Худайназаровой, вкуче с технологиями машинного обучения, упрощает решение типовых задач и позволяет создавать программный код, обрабатывать естественный язык, проводить

тестирование кода, его отладку и оптимизацию. Применение ИИ в обучении ИТ-специалистов видится авторами одной из перспективных задач погружения в новые условия разработки – написания кода [3, с. 66].

Позитивистских взглядов о применении ИИ в качестве тренажера по написанию кода придерживается М.С. Долинский, который отмечает перспективность моделей генеративного ИИ в решении указанной задачи. В частности, автор указывает на возможность создания учебных материалов средствами генеративного ИИ со стороны преподавателей для обучения ИТ-специалистов; также выделяется возможность организации самостоятельной работы студентов с ИИ в виде диалогового окна, что позволяет развивать алгоритмическое мышление и подготавливать к дальнейшему самостоятельному написанию кода. Однако внедрение ИИ должно происходить осторожно по причине возможных ошибок, галлюцинирования и проблем самостоятельного выполнения учебных задач [1, с. 89]. Поскольку ожидается, что ИИ постепенно заместит человека в разработке (написании кода), человеку нужно научиться руководить процессом, работать в команде и быть готовым критически оценивать (а также улучшать) код, созданный при работе с нейросетью. Поэтому обучение программированию с применением ИИ описывается в работе А.Г. Никольского, О.В. Виштак, Г.В. Очкур и Н.М. Виштак в качестве закономерного. ИИ, по мнению авторов, должен применяться в целях индивидуализации и выступать дополняющим

инструментом (с подготовкой материала для студента, оценки кода, объяснения сложных тем, автоматического контроля заданий, сопровождения в процессе решения стоящих задач в области программирования). Выделяется обширный спектр преимуществ-возможностей использования ИИ в обучении программированию, а именно: доступность, индивидуализация, сокращение времени на обучение основам, диалоговый формат взаимодействия. Недостатки конкретизируются авторами в ракурсе ошибок, потребности в контроле человека и проблемах этики [4, с. 195].

С учетом противоречивости применения ИИ в качестве симулятора программирования для обучения ИТ-специалистов, важной представляется ориентация на актуализацию модели подготовки будущих программистов к решению профессиональных задач при парной работе с ИИ. В русле идей о трансформации

профессиональных навыков в эпоху «Программирования 4.0», программист становится не столько автором кода, сколько «проводником» между замыслом и его реализацией через ИИ, что требует смещения акцента с сугубо технических умений на способность формулировать задачи, критически оценивать сгенерированные результаты и интегрировать их в итоговый проект [2]. Как никогда актуально развивать у студентов умения ставить задачи для ИИ (запросы), критически оценивать генерируемые результаты и подстраивать их под свою работу. Поскольку ИИ чат-бот рассматривается нами в качестве средства обучения – симулятора, позволяющего выполнять профессиональные задачи в условиях, приближенных к реальным, подобное послужило основой разработки уровневой модели подготовки специалистов по разработке информационных систем, представленной на рисунке 1.

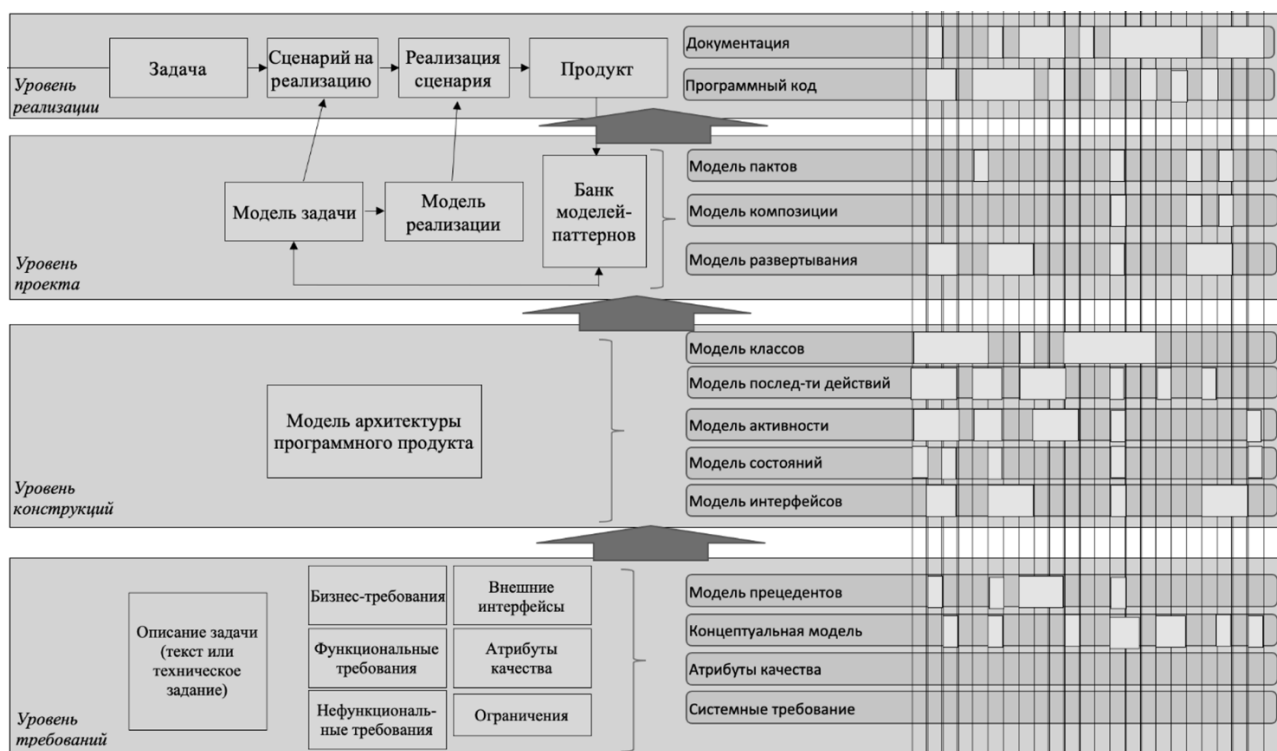


Рисунок 1. Модель подготовки специалистов по разработке информационных систем

В разработанную модель включено четыре уровня: уровень требований, уровень конструкций, уровень модели реализации и уровень реализации. В соответствии с

разработанной моделью были выделены различные соответствующие требованиям симуляторы с использованием ИИ-технологий, представленные на рисунке 2.

Уровень требований	Уровень архитектуры	Уровень модели реализации	Уровень реализации
<ul style="list-style-type: none"> • GPT (OpenAI) • Gemini (Google) • Claude3 (Anthropic) • IBM Watson Assistant • Notion AI • AIPRM for ChatGPT • LearningProgramming • AI TA 	<ul style="list-style-type: none"> • GPT (OpenAI) • Gemini (Google) • Llama (Meta) • GitHub Copilot • CodeWhisperer • ArchitectAI • Hugging Face AutoTrain • CodeAid • NotebookGPT 	<ul style="list-style-type: none"> • GPT (OpenAI) • Gemini (Google) • Claude3 (Anthropic) • MMAPR • TutorBot+ • Figma with AI plugins • while True: learn() • Runway ML • DeepMind AlphaFold 	<ul style="list-style-type: none"> • GPT (OpenAI) • Gemini (Google) • Llama (Meta) • Codeboard • Coding Step • GPTutor • CodeHelp • Apprentice Tutor Builder • GitHub Copilot • Tabnine • DeepCode • Replit AI • Kite • Codeium

Рисунок 2. ИИ-симуляторы, соответствующие уровням модели подготовки

Критически оценивая представленные симуляторы обучения программированию, стоит отметить, что в целом они позволяют оптимизировать многие задачи, возникающие в ходе обучения программированию. Однако обратной стороной представленных симуляторов является то, что они нацелены на решение отдельного пласта задач – отсутствует единое платформенное решение, которое бы в полной мере (в качестве «сквозного инструмента») соответствовало идеям симуляционного обучения. Сложной представляется организация совместной параллельной работы в команде при использовании ИИ-симуляторов (на текущем этапе их развития). Также немаловажной видится проблема отсутствия унифицированного способа формализации задач, поскольку применение естественного языка для взаимодействия с ИИ-технологиями и инструментами не в полной мере соответствует методическим основам обучения программированию. Требуется выработать единый подход к обучению, сформировать систему дифференцированных по уровню сложности задач, обеспечить объективность оценивания работы и при этом сохранить самостоятельность обучающихся, развивать у них алгоритмическое мышление при работе с ИИ-симуляторами (что становится перспективным направлением будущих исследований).

Таким образом, результаты проведённого исследования позволяют подчеркнуть, что современные симуляторы программирования способны обеспечить не только технологическую, но и методическую поддержку в обучении и оценке навыков студентов ИТ-специальностей. Применение симуляторов программирования, в том числе основанных на ИИ-технологиях и инструментах, способствует обогащению опыта осуществления профессиональной деятельности, поддержанию интереса, а также позволяет совершенствовать методику обучения. При этом важно вырабатывать единые критерии оценки, стандартизировать формализацию учебных задач, сочетать применение ИИ-решений с традиционной практикой написания кода, обеспечивать постепенное усложнение заданий, развивать алгоритмическое мышление. Важным направлением дальнейших исследований в области симуляционного обучения программированию становится направление разработки решений, предназначенных для объединения различных инструментов в единую систему. Также уточнения требуют методическое сопровождение симуляционного обучения, поскольку необходимы рекомендации по организации практических занятий, выбору форм контроля и критериев оценивания.

ЛИТЕРАТУРА

1. Долинский, М. С. Направления использования генеративного искусственного интеллекта при начальном обучении программированию в университетах / М. С. Долинский // Компьютерные инструменты в образовании. – 2024. – № 2. – С. 85-96. DOI: 10.32603/2071-2340-2024-2-85-96.
2. Королев, В. В. ИИ-коллаборативное программирование: трансформация профессиональных навыков программиста и образовательные вызовы / В. В. Королев // XXIX Царскосельские чтения : Материалы международной научной конференции. В 2-х томах, Санкт-Петербург, 22–23 апреля 2025 г. – Санкт-Петербург : Ленинградский государственный университет им. А.С. Пушкина, 2025. – С. 306–311. – EDN ENCYBN.
3. Мирабова Л. Искусственный интеллект и машинное обучение в программировании: тенденции и перспективы / Л. Мирабова, М. Худайназарова // CETERIS PARIBUS. – 2023. – № 10. – С. 65-67.
4. Никольский, А. Г. Искусственный интеллект как инструмент изучения программирования / А. Г. Никольский, О. В. Виштак, Г. В. Очкур, Н. М. Виштак // Известия ТулГУ. Технические науки. – 2024. – № 2. – С. 191-197. DOI: 10.24412/2071-6168-2024-2-191-192.

SIMULATION-BASED PROGRAMMING TRAINING USING ARTIFICIAL INTELLIGENCE TECHNOLOGIES: A LAYER-BASED MODEL AND TOOLS

KOROLEV Vladimir Vladimirovich

Senior Lecturer

Pushkin Leningrad State University

St. Petersburg, Russia

The article is devoted to the issues of simulation training in programming using AI technologies. The transition to training developers in the conditions of AI collaborative programming is substantiated. A level model of training and corresponding AI simulators is presented. The analysis of tools is carried out, the problems of fragmentation, the lack of platform solutions are revealed.

Keywords: simulation training, programming training, artificial intelligence in education, training of programmers, AI simulators, collaborative programming.