

УДК 378:004.42
**НЕПРЕРЫВНОЕ ИТ-ОБРАЗОВАНИЕ ДЛЯ НОВЫХ ВОЗМОЖНОСТЕЙ: ОПЫТ
ВЫСТРАИВАНИЯ ТРАЕКТОРИИ «ШКОЛА – АИШ – БГУИР»**

Прохоренко Д.М.

преподаватель АИШ при БГУИР, г.Минск
multidem@gmail.com

В статье рассматривается практика реализации концепции непрерывного ИТ-образования на примере выстраивания индивидуальной образовательной траектории «школа → Академия информатики для школьников при БГУИР → БГУИР». Описана трёхэтапная модель обучения технологическим стекам (на примере веб-разработки и языка Python), включающая изучение ядра языка, освоение парадигм программирования и разработку проектов с использованием современных фреймворков. Обоснована важность преемственности между уровнями обучения, гибкости форматов (очный, онлайн, гибридный) и адаптации учебных программ под интересы и возрастные особенности школьников. Отдельное внимание уделено роли искусственного интеллекта в образовательном процессе: предложена поэтапная модель интеграции ИИ-инструментов с акцентом на развитие критического мышления и способности обучающегося объяснять и модифицировать полученные решения. Подчёркивается, что цель непрерывного ИТ-образования — не только формирование профессиональных компетенций junior-разработчика, но и воспитание мотивированного, адаптивного специалиста, готового к lifelong learning в условиях быстро меняющихся технологий.

Ключевые слова: непрерывное ИТ-образование, образовательная траектория, школа – АИШ – БГУИР, программирование на Python, веб-разработка, трёхэтапная модель обучения, искусственный интеллект в образовании, дополнительное образование школьников, алгоритмическое мышление, геймификация обучения.

Сегодня реализацию концепции непрерывного обучения можно начинать с самого раннего школьного возраста, выстраивая образовательную траекторию: школа → Академия информатики → БГУИР [1, 6]. Задача педагогов-практиков АИШ как важного звена передовой линии образования – создавать поэтапные условия для осознанного выбора школьниками будущей профессиональной направленности в ИТ-индустрии.

Кратко рассмотрим реализацию этой задачи на примере одного из наиболее востребованных сегодня технологических стеков: веб-разработка + программирование на языке Python. Не секрет, что на первоначальный выбор курса обучения в АИШ влияют современные ИТ-тренды. Ещё недавно максимальной популярностью пользовались HTML, JavaScript, PHP, Java и другие языки. Сегодня в лидерах – Python, и тенденции постоянно меняются. Однако ни одна из перечисленных технологий не исчезает с рынка окончательно: они пересекаются и успешно применяются на современном этапе, каждая – для своих задач. Выбор первого курса – это уверенный шаг в долгосрочном обучении школьника, и он должен обеспечивать равный доступ к качественному дополнительному образованию для учащихся из всех социальных групп [2].



Рисунок 1 – Образовательная траектория.

Обучение каждой из упомянутых технологий строится по схожей трёхэтапной модели. Первый этап – изучение core (ядра языка, базовых конструкций, которые идентичны в разных языках и отличаются лишь синтаксисом; на этом этапе закладываются основы алгоритмического мышления). Второй шаг – освоение стилей программирования, где безусловным лидером остаётся объектно-ориентированное программирование (ООП) как важное дополнение к функциональному и другим парадигмам. Завершающим этапом курса становится разработка проектов с использованием популярных API и фреймворков [2, 3]. Для Python это, например, Flask или Django с перспективой изучения фронтенда (HTML, TypeScript). Для веб-разработки – Angular или React с заделом на последующее изучение бэкенда через дополнительный курс Python.

Каждая из трёх ступеней обеспечивает преемственность и мотивирует двигаться дальше. В результате, пройдя курс по Python, ученик понимает, что следующий логический шаг – это веб-разработка. И наоборот, изучив веб-технологии, можно перейти к Python, Java или C#, которые реализуются в рамках отдельных курсов. Так приходит осознание, что профессиональная специализация в IT-секторе – понятие достаточно условное, требующее постоянного саморазвития [6]. Трёхэтапная модель полностью укладывается в концепцию базового, углублённого и продвинутого уровней обучения.



Рисунок 2 – Трёхэтапная модель

На рынке дополнительного образования сегодня представлены очный, онлайн и гибридный форматы, летние интенсивы, пробные занятия и системы скидок. Обучение проводится по выходным или в будни после школьных уроков. Все эти возможности повышают комфорт обучения: родители могут самостоятельно выбрать оптимальный для ребёнка формат и вписать занятия в плотный график развития школьника. Онлайн и гибридные форматы обеспечивают равный доступ к качественному дополнительному образованию для учащихся из регионов и разных социальных групп [1].

На уровне отдельного курса, для удержания интереса подростков на протяжении всего обучения, материал должен подаваться от простого к сложному, логически последовательно, быть тщательно проработан, наглядно иллюстрирован и содержать множество практических примеров [7]. Программа требует периодического рефакторинга и должна оперативно следовать новым трендам. Практические задания должны соответствовать интересам ребёнка, отражать его внутренний мир и мотивировать на учёбу: вместо программирования условного «калькулятора» для изучения типов данных лучше написать код, в котором Бэтмен и Робин на бэтмобиле спасают жителей Готэма [5]. К ученику необходимо входить в доверие, выступать в роли старшего товарища, говорить с ним на одном языке, зажигать юные сердца и просветлять умы, не забывая при этом о педагогической ответственности. Важно также адаптировать программы для детей с особенностями развития [1].

Не следует ограничивать процесс обучения только техническими темами – необходимо расширять кругозор и прививать любовь к профессии [6]. Такой подход позволяет не просто подготовить junior-разработчика, но и воспитать достойного члена общества.

Для самых юных участников неоспоримую ценность сохраняют игровые форматы (Scratch, Kodu Game Lab, Minecraft Edu и др.). Они в увлекательной форме обучают основам программирования и создают прочную базу для перехода от визуальных алгоритмов к профессиональному коду [5, 7].

В условиях быстрых технологических изменений необходимо оперативно обновлять учебные программы [2]. Ещё недавно популярными казались no-code и low-code системы, которые, казалось, заменят классическую разработку приложений. Однако с приходом искусственного интеллекта визуальное программирование перешло в сферу компетенций ИИ. Сегодня компании понимают: проще попросить нейросеть написать черновой код сложной системы и передать его разработчику на доработку, чем собирать решение из ограниченных по возможностям визуальных конструкторов.

Несколько слов о роли ИИ в современном образовательном процессе. Что говорит по этому поводу сам искусственный интеллект? Приведу краткие тезисы, с которыми трудно не согласиться: «Главная идея обучения в эпоху ИИ – не запрещать его, а интегрировать как инструмент с контролируруемыми ограничениями. Обучение следует строить поэтапно: на начальных курсах ИИ либо запрещается, либо используется исключительно для пояснения теории в дополнение к преподавателю. На последующих этапах – контролируемое использование ИИ с обязательным разбором и объяснением результата. На старших курсах ИИ становится полноценным инструментом, в том числе для генерации кода. Граница проходит не по тому, кто написал код, а по тому, может ли студент объяснить, модифицировать и защитить это решение. Как избежать списывания: устные защиты, вариативные задания, задачи на критику ответов ИИ, ограниченное время на написание кода и др.» [4, 8]. Тем не менее, вопросы эффективной реализации запрета и контроля на начальных этапах остаются открытыми.

Неоспоримый потенциал ИИ заключается в его оперативной поддержке «здесь и сейчас»: он выступает источником лучших практик, виртуальным наставником и помощником в рутинных задачах. Когда ученик вырастет, поступит в вуз и успешно его окончит, изученные сегодня технологии, скорее всего, трансформируются в новые. Но благодаря работе с ИИ ребёнок «научится учиться», будет мотивирован на дальнейшее непрерывное образование и сможет эффективно взаимодействовать с преподавателями вуза на качественно новом уровне [4, 6].

Литература

1. Концепция развития непрерывного образования в Республике Беларусь. – Минск : Министерство образования Республики Беларусь, 2020. – 32 с.
2. Downey, A. B. Think Python: How to Think Like a Computer Scientist. – 2nd ed. – Sebastopol : O'Reilly Media, 2015. – 260 p.
3. Grinberg, M. Flask Web Development: Developing Web Applications with Python. – 2nd ed. – Sebastopol : O'Reilly Media, 2018. – 412 p.
4. UNESCO. Guidance for generative AI in education and research. – Paris : UNESCO, 2023. – 48 p.
5. Копылов, В. В. Геймификация в IT-образовании: от визуальных конструкторов к профессиональной разработке / В. В. Копылов, Н. А. Смирнова // Информатика и образование. – 2022. – № 4. – С. 45–52.
6. Бабанский, Ю. К. Оптимизация учебно-воспитательного процесса: методологические основы / Ю. К. Бабанский. – М. : Просвещение, 2019. – 192 с.
7. Papert, S. Mindstorms: Children, Computers, and Powerful Ideas. – New York : Basic Books, 1980. – 216 p.
8. Selwyn, N. Should We Replace Teachers with AI? On the Limits of Education Technology. – Cambridge : Polity Press, 2019. – 184 p.