ISSN 2221-1993 • **Informatics in School** • 2025 • Volume 24 No 5

DOI: 10.32517/2221-1993-2025-24-5-72-77

А. А. Салахова
Московский педагогический государственный университет, г. Москва, Россия

КРЕАТИВНОЕ ПРОГРАММИРОВАНИЕ: ИЗ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В НЕПРЕРЫВНЫЙ КУРС ИНФОРМАТИКИ

Аннотация

Статья посвящена креативному программированию, освещенному в нетипичном ключе — через фундаментализацию понятия в школьном образовании как подхода, пришедшего из разработки программного обеспечения и сохраняющего свои ключевые характеристики: визуализацию результата, коллаборацию, работу с объектами, ограниченные ресурсы и быстрые циклы создания MVP (*англ.* Minimum Viable Product — минимально жизнеспособный продукт). Анализируются особенности интерпретации термина в образовательном процессе. Подчеркивается, что сегодня мы можем рассматривать креативное программирование как образовательную технологию. Показано, что оно выступает не только средством повышения мотивации обучающихся, но и инструментом формирования алгоритмического и проектного мышления, интегрированным в современные учебные модели и выполняющим пропедевтическую функцию через интуитивное введение новых концепций до их появления как понятий в курсе информатики, влияет на формы урочной и внеурочной деятельности. На основе анализа приводится модель внедрения креативного программирования в непрерывный курс информатики с I по XI класс, включая использование робототехники, мультимедиа, мобильной разработки, инструментов машинного обучения, введения в разработку игр, в том числе с применением визуальных сред. Отдельное внимание уделено роли искусственного интеллекта в групповых проектах и возможностям его интеграции в проектные группы. Вводится предложение рассматривать индивидуальные проекты обучающихся как модули коллаборативных (групповых) проектов. Статья подчеркивает значимость креативного программирования для предпрофессиональной подготовки школьников в условиях цифровой трансформации образования.

Ключевые слова: креативное программирование, непрерывный курс информатики, Scratch, искусственный интеллект в школе, проектный подход.

Информация об авторе

Салахова Алёна Антоновна, канд. пед. наук, доцент кафедры теории и методики обучения математике и информатике, Институт математики и информатики, Московский педагогический государственный университет, г. Москва, Россия; *e-mail:* aa.salakhova@mpgu.su

A. A. Salakhova
Moscow Pedagogical State University, Moscow, Russia

CREATIVE PROGRAMMING: FROM SOFTWARE DEVELOPMENT TO THE CONTINUOUS INFORMATICS SCHOOL COURSE

Abstract

The article explores creative programming in an atypical light, by fundamentalizing the concept in school education as an approach that originated from software development and retains its key characteristics: visualization of results, collaboration, working with objects, limited resources, and rapid MVP (Minimum Viable Product) development cycles. The article examines the specific interpretations of the term in the educational process. It emphasizes that today we can consider creative programming as an educational technology. It is shown that it acts not only as a means of increasing student motivation but also as a tool for developing algorithmic and design thinking, integrated into modern educational models, and fulfilling a propaedeutic function through the intuitive introduction of new concepts before they emerge as concepts in informatics courses. It also influences the forms of in-class and extracurricular activities. Based on the analysis, a model is presented for integrating creative programming into a continuous informatics curriculum for grades 1 through 11. This model incorporates robotics, multimedia, mobile development, machine learning tools, and an introduction to game development, including those using visual environments. Special attention is given to the role of artificial intelligence in group projects and the potential for its integration into project teams. The article proposes to consider individual student projects as modules of collaborative (group) projects. The article emphasizes the importance of creative programming for the pre-professional preparation of schoolchildren in the context of the digital transformation of education.

Keywords: creative programming, continuous informatics course, Scratch, artificial intelligence at school (AIEd), project approach.

Information about the author

Alena A. Salakhova, Candidate of Sciences (Education), Associate Professor at the Department of Theory and Methods of Teaching Mathematics and Informatics, Institute of Mathematics and Informatics, Moscow Pedagogical State University, Moscow, Russia; *e-mail:* aa.salakhova@mpgu.su

ISSN 2221-1993 • **Информатика в школе** • 2025 • Том 24 № 5

1. Введение

В школе понятие «креативное программирование» чаще всего употребляется в контексте внеурочных занятий, связанных с пропедевтикой программирования с применением визуальных языков. Под *креативностью* в контексте подобной деятельности понимают особенности формы занятий и содержания задач, требующих более творческого подхода [7].

Важно отметить, что как термин это понятие чаще всего не поясняется. Большинство статей, посвященных креативному программированию, направлены на исследование его прикладных особенностей и возможных форм, а не его сущности в качестве одного из направлений программирования как профессиональной деятельности в сфере информационных технологий.

Обобщая опубликованные практические исследования, можно сказать, что *для педагогического контекста креативное программирование* — это раздел обучения программированию, для которого характерны:

- наличие визуального (реже — физического) исполнителя;
- большая творческая составляющая заданий;
- мультимедийность (отображение на экране различных форматов мультимедиа: интерактивные игры, викторины, мультфильмы; воспроизведение звуков и т. д.);
- использование упрощенного синтаксиса в языках программирования (за счет фреймворков или специальных учебных языков) или интуитивное объяснение некоторых сложных элементов (например, работа со списками по индексам, применение понятия «очередь» без его озвучивания в Scratch);
- акцент на игровых формах;
- преобладание проектного обучения и групповой работы;
- быстрое достижение результата (проект реализуется за один-два урока).

В своей статье [6] Н. Н. Самылкина описывает образовательную робототехнику как одну из педагогических технологий. По аналогии можно привести аргументацию в пользу того, что сегодня креативное программирование в школе также может рассматриваться в подобной роли, поскольку оно:

- реализует самостоятельную надпредметную цель — в контексте не только инженерного мышления, но и проектной деятельности (и культуры) обучающихся;
- выходит на транспредметную интеграцию — за счет привлечения контента и контекста проектов, в том числе гуманитарной составляющей;
- изменяет формат взаимодействия субъектов образовательного процесса — за счет реализации, например, виртуальных проектных мини-офисов и команд;
- изменяет характеристики компонентов методической системы обучения: содержания, форм, методов и средств обучения.

Нельзя не отметить популярность этого словосочетания у родителей и коммерческую успешность образовательных проектов, связанных с креативным программированием, за счет эмоционального отклика потребителя.

2. Креативное программирование в школе и в профессиональной разработке ПО

Педагогическое понимание креативного программирования довольно близко к концепции, используемой в ИТ-сфере в рамках гибкой методологии разработки (Agile) [2].

Креативное программирование (*англ.* Creative Programming) — это один из подходов внутри существующих методологий разработки программного обеспечения наряду, например, с экстремальным программированием (*англ.* Extreme Programming). В профессиональной сфере нет единого определения этого понятия, чаще всего оно используется интуитивно. Не существует и никаких действующих международных или региональных регламентов и стандартов к подходу в целом, а также ограничений к применяемым языкам.

Креативное программирование как подход определяется по ключевой особенности — созданию выразительного минимального реализуемого функционального продукта (*англ.* Minimum Viable Product, MVP — минимально жизнеспособный продукт) [3]. Это означает, что созданный продукт может демонстрировать работу только конкретного блока или модуля системы/большого проекта, при этом результат работы виден конечному пользователю, а реализация может быть выполнена неоптимальным способом.

Цель креативного программирования — тестирование гипотез на возможность их реализации [2]. Для того чтобы стимулировать генерацию идей командами, часто вводятся искусственные ограничения для создания нетипичной обстановки и предоставляются нетипичные инструменты [1]. Последнее мы можем добавить в педагогическое понимание креативного программирования как важный ограничитель.

Решение задачи разделяется на два направления:

- тестирование идей и поиск алгоритма — помогает развивать алгоритмическое мышление, навыки создания универсальных алгоритмов;
- использование уникальных особенностей конкретного инструмента — конкретные языки становятся источником дополнительного вдохновения за счет их специфических свойств/возможностей, что позволяет понять идею и логику языка программирования, а не только запомнить его синтаксис.

По своей сущности креативное программирование в гибкой методологии разработки рассматривается как аналог брейншторма, под который выделяется отдельный короткий спринт (креативный спринт; обычно длится пять дней или неделю, что отличает его от обычных спринтов длиной девять недель), а его результаты могут даже не включаться в итоговый проект [2].

Также важно отметить, что креативное программирование предусматривает обязательную коллаборацию. Однако *в учебных проектах* мы допускаем индивидуальное выполнение (в том числе в рамках Олимпиады по креативному программированию), хотя это противоречит самой концепции подхода.

В сравнительной таблице 1 представлены особенности креативного программирования и их проявление в ИТ-сфере и в школе.

ISSN 2221-1993 • **Informatics in School** • 2025 • Volume 24 No 5

Таблица 1

Креативное программирование в ИТ-сфере и в школе

№ п/п	Особенность	В разработке ПО	В школе
1	Коллаборация	Обязательная работа в команде либо применение нейросети как компаньона	Преобладает групповая работа, реже — парная. Иногда допускается создание индивидуальных проектов
2	Высокий уровень интерактивности конечного результата	Требуется для получения наибольшей обратной связи путем тестирования продукта. Это позволяет позднее снизить затраты при реализации уже «настоящего» (целевого) продукта путем быстрого тестирования сразу нескольких идей (например, различных пользовательских интерфейсов или ключевых технологий)	Создание игр, викторин или интерактивных мультфильмов — виртуальное взаимодействие с конечным пользователем [7]. Взаимодействие с робототехническими средствами — физическое взаимодействие с конечным пользователем, иногда с виртуальными роботами и мирами [4]. Создание преимущественно визуализированных решений, включая добавление графического интерфейса программам на Python или веб-интерфейса для обработчиков баз данных. Создание игр с графическим интерфейсом на Unity или графических новелл в рамках проектной деятельности. Разработка графических интерфейсов для исследовательских проектов и проектов с машинным обучением [8–10]. Стоит отметить, что основная цель визуализации здесь иная по сравнению с ИТ-сферой — мотивация обучающихся, видящих результат своей работы (визуальное закрепление успеха)
3	Объектно-ориентированное и функциональное программирование	Работа производится с объектами и их свойствами, для которых затем описываются функции. Объект (или класс) первостепенны	Например, в Scratch производится работа со спрайтами и персонажами, являющимися объектами, для которых описываются функции. Физический робот и его датчики также являются объектами, для которых описывается программа. В визуальных языках объект первостепенен. Наследование рассматривается через клоны спрайтов
4	Применение скриптовых языков программирования	Помогает работать с мультимедиа и обеспечивать производительность системы (последовательная загрузка и исполнение с защитой от обрыва связи, цель — представление хотя бы части результата при ошибке системы). Обычно это языки Lua, Python, Processing, JavaScript	Применение таких языков, как Scratch, Python (реже — JavaScript) [10]. Традиционно скриптовые языки считаются интерпретируемыми, однако современные технологии размывают границы между компиляцией и интерпретацией. Указанные выше языки — интерпретируемые, как и JavaScript, который может быть применен для веб-разработки. В школе применяется и Wiring (для программирования для Arduino, основан на Processing), являющийся компилируемым, но программы на нем исторически также называют скриптами. Многие языки, связанные с робототехникой, интерпретируются при управлении оператором и компилируются при загрузке в робота (или интерпретируются исполнителем, без возможности вмешаться в процесс извне)
5	Короткий срок жизни проекта	Необязательная завершенность проекта. Упор на тестирование идеи, а не на дальнейшую эксплуатацию. Однако продукт обязан быть минимально жизнеспособным	Проект завершается итоговым продуктом, однако зачастую к нему выдвигаются высокие требования. Следует учитывать, что учебные проекты по своей сущности чаще всего не предусматривают дальнейшей эксплуатации
6	Преднамеренное ограничение ресурсов	Введение запрета на использование привычных инструментов, включая библиотеки, для стимуляции поиска неординарного решения	Учитель может ограничивать набор доступных команд, создавая хорошо контролируемую «песочницу» для достижения образовательных целей и задач

ISSN 2221-1993 • **Информатика в школе** • 2025 • Том 24 № 5

Таким образом, мы видим, что *креативное программирование в школе — это, в сущности, тот же подход, который применяется в профессиональной сфере разработки ПО, но адаптированный к условиям школы*. Следовательно, новации, вносимые в производство, могут находить отображение (разумеется, в несколько измененном виде) и при подготовке содержания или изменении формата занятий креативным программированием со школьниками. Например, внедрение ИИ-компаньонов (в том числе как части интеллектуальной среды разработки) возможно в качестве инструмента реализации коллаборации при подготовке индивидуальных проектов, но потребует ограничений («ИИ — помощник (ментор), а не решатель»).

3. Креативное программирование в непрерывном курсе информатики

Рассмотрим, как креативное программирование реализуется в рамках непрерывного курса информатики (табл. 2). Поскольку затронута в большей степени внеурочная деятельность, примеры приведены для I—XI классов без исключений. В столбце «Комментарий» таблицы 2 дается пояснение о месте в курсе и о пропедевтике конкретных понятий из программирования. Отметим, что предлагается применение отечественной реализации среды Scratch от «Роббо» (https://scratch.ru/). В таблице 2 звездочкой (*) отмечены дополнительные инструменты, являющиеся иностранными решениями (российские аналоги на момент написания статьи отсутствуют), однако все они находятся либо в открытом доступе, либо вовсе являются Open Source решениями.

Отметим, что особенно актуально подобное внедрение креативного программирования в начальное общее образование, где сегодня мы также наблюдаем курс на усиление субъективности обучающихся и усиление роли дифференцированного (разноуровневого) подхода за счет внеурочной деятельности [1].

4. Применение искусственного интеллекта в групповых проектах

В качестве одного из вариантов групповых учебных проектов можно рассматривать **кейс-стади с открытым финалом** (после изучения необходимо создать собственный продукт с добавочной пользой, не рассмотренной ранее в проекте, например, с новой функцией или с неким преимуществом).

В этом случае **формируется команда**, состоящая из двух-трех человек и агента (модели нейронной сети), также являющегося субъектом деятельности. Перед ними ставится задача создания конечного минимально жизнеспособного продукта (MVP).

Тема проекта выбирается случайным образом или темы распределяются педагогом — указываются область исследования и вводимые ограничения (например, использование в проекте определенного языка программирования). Конкретную тему на основе анализа области исследования и выбранных задач формулируют обучающиеся. При этом важно отметить, что учитель может ввести дополнительные ограничения на ресурсы, включая временны́е и контекстные (например, ограничить целевую аудиторию продукта).

Далее возникают **два варианта применения искусственного интеллекта:**

1) если применяются *чат-бот* и конкретная языковая модель (т. е. ИИ не находится в контексте всего обсуждения и работы и отвечает только на активные запросы в свою сторону), то обучающиеся обращаются к ней как к недостоверному эксперту/специалисту — это означает, что все предоставляемые решения подвергаются критической проверке остальными членами команды;
2) если применяется *ИИ-агент* как виртуальный участник в среде командной работы (работает с контекстом и способен обращаться к участникам группы без прямого вызова), то на него перекладывается роль менеджера проекта (project manager, или SCRUM-master).

Таблица 2

Креативное программирование в непрерывном курсе информатики

Уровень	Класс	Предлагаемые инструменты или подходы	Комментарий
Начальное общее образование	I, II	Создание мультфильмов в Scratch вместе с учителем + «ПиктоМир»	Только внеурочная деятельность. В основном упор делается на закрепление понимания связи последовательных действий (только линейные алгоритмы). *Нет кода даже на блоках, использование пиктограмм.* *Все действия воспроизводятся исполнителем визуально (результат исполнения каждого блока кода видимый)*
	III, IV	Создание простых мультфильмов в Scratch самостоятельно + TRIK Junior для средств робототехники	Введение ветвления и простых условий. Управление физическим исполнителем, включая получение обратной связи (от датчиков) без точного измерения (проверка пограничного значения). *Возможно появление простых текстовых команд, но преимущественно используются блоки с подписанными пиктограммами*

ISSN 2221-1993 • **Informatics in School** • 2025 • Volume 24 No 5

Окончание табл. 2

Уровень	Класс	Предлагаемые инструменты или подходы	Комментарий
Основное общее образование	V	Интерактивные истории и мультфильмы в Scratch. TRIK Studio для робототехники (возможно, виртуальной). Программирование виртуальных миров, например, в Kodu GameLab* [4]	Введение сложных условий (логических связок в условиях) и итераторов (циклов с заданным количеством повторений). Закрытый контур управления физическим роботом (по заданной программе) либо на основе простых триггеров — изменения пограничного состояния датчика. *Код на блоках с цветовой дифференциацией по группам команд.* *«Пазлы», демонстрирующие возможность соединения разных блоков команд или данных*
	VI	Создание викторин Scratch. TRIK Studio для робототехники (возможно, виртуальной)	Использование циклов с проверкой условия. Открытый контур управления физическим роботом (управление оператором дистанционно или принятие решений автономно на основе показаний датчиков после обработки полученных данных). Введение переменных. Применение асинхронного программирования без введения формального понятия. *Создание функций как именованных групп повторяющихся блоков кода. Постепенный уход от цветовой палитры блоков в пользу понимания текстовых формулировок и особенностей команд*
	VII	Scratch или Blockly: MIT App Inventor 2*. На сегодня MIT App Inventor 2* поддерживает не только разработку мобильных приложений для Android, но и запуск созданных программ на iOS	Пропедевтика работы со списками и абстракциями. Работа с очередями без формального введения понятия. Создание собственных функций и подпрограмм. Работа с внешними устройствами и датчиками. Создание мобильных приложений. Основной упор на сбор и обработку данных. *Использование собственных функций. Параллельное использование блочных и текстовых языков*
	VIII, IX	Scratch: Machine Learning For Kids* Плавный переход на Python с подключением Machine Learning For Kids* через API (как библиотеки, без объяснения понятия API и использования термина). Желательно использовать Machine Learning For Kids* в тестовом режиме без регистрации для доступа ко всем функциям, включая распознавание текста. Однако это затрудняет сохранение проектов	Применение языка программирования Scratch с фреймворками и аддонами (возможно плавное введение этих понятий). Знакомство с принципами машинного обучения, включая компьютерное зрение и нейронные сети [5]. *Переход на текстовые языки программирования. Пропедевтика чтения логов и дебаггинга через работу с результатами обучения модели, проверки состояния переменных во время исполнения программы.* *Пропедевтика понятия «стек технологий» путем сочетания разных языков в одном проекте*
Среднее общее образование	X, XI	*Для углубленного уровня:* Применение стека технологий (например, Unity* + 3D-моделирование для создания игр, либо Python + JavaScript для проектов в области веб-технологий [10]). В робототехнике — использование Python для Raspberry Pi. Возможно использование Raspberry Pi (интеллектуальная обработка) + Arduino (управление движением, автоматика и автоматизация). *Для базового уровня:* применение нейронных генеративных сетей как субъекта внутри группы для генерации кода по сформированному техническому заданию, использование ИИ-помощника для модерации обсуждения в проектных группах	Введение дополнительного языка программирования высокого уровня и создание сложных долговременных проектов с широким применением мультимедиатехнологий. Групповые проекты, где индивидуальные проекты рассматриваются как отдельные модули системы/проекта, созданного в коллаборации. Введение понятия «стек технологий». *Сочетание различных инструментов мультимедиа*

ISSN 2221-1993 • **Информатика в школе** • 2025 • Том 24 № 5

В первом случае обучающимся необходимо распределить роли, затем составить четкое техническое задание после осуществления предварительных этапов (разработка области исследования, анализ доступных и возможных ресурсов, описание ожидания пользовательского опыта, функционала продукта, выбор стека технологий) и далее, используя корректные запросы к нейросети, получить код для решения и создать графику. Желательно при этом не использовать полностью готовые конструкторы приложений, поскольку они ограничивают область работы. Также можно использовать генеративную модель для наполнения «рыбы» — текстовых описаний, необходимых для верстки. Для представления интерактивных элементов и графики, включая дизайн макета и прочие картинки, тоже могут использоваться генеративные сети.

Во втором случае необходимо вести всю коммуникацию по проекту непосредственно в среде командной работы, например, в «Битрикс24» (российское ПО, бесплатное для небольших команд), либо в мессенджере с подключенным чат-ботом, имеющим доступ к переписке для ее анализа. Такой агент должен анализировать количество задач, нагрузку каждого участника, проверять сроки исполнения и перераспределять задачи, а также предоставлять отчеты (дашборды — от *англ.* dashboard) капитану команды, что помогает при составлении сводной документации проекта.

Оба варианта внедрения предполагают формирование и развитие у обучающихся компетенций, специфических для командной работы в современной ИТ-сфере, включая критическую оценку кода при промпт-программировании (*англ.* prompt-programming).

Отметим, что при этом описание документации ложится все равно на обучающихся, что требует от них понимания принципа работы их продукта и используемых в нем технологий.

Результатом проекта является не только и не столько минимально жизнеспособный продукт, сколько документация с описанием критериев к каждому этапу жизни проекта (детальное техническое задание, описание стека технологий, лог работ, включая тестирование, и описание итогового продукта).

5. Заключение

Креативное программирование в школе не ограничено изучением Scratch и не является исключительно игровой формой проведения занятий. Формирование легкого отношения к терминам из-за их созвучности с досуговыми активностями ввиду влияния массмедиа может ограничить мировоззрение обучающихся и лишить их интересного опыта.

В статье был рассмотрен подход, пришедший из разработки программного обеспечения в рамках гибкой методологии (Agile), который занял прочные позиции в школьном образовании, включая содержание углубленного курса информатики основного и среднего общего образования. Была показана еще одна сторона креативного программирования и подчеркнута значимость и актуальность курса информатики в современных реалиях, в том числе за счет интеграции новых образовательных технологий.

Важно отметить, что креативный подход в программировании — это часть предпрофессиональной подготовки будущих специалистов в ИТ-сфере, позволяющая вводить в содержание образования сложные понятия и концепции, актуальные для цифровых профессий.

Список источников

1. *Босова Л. Л., Самылкина Н. Н., Мишин В. А.* О разноуровневом обучении программированию в курсе информатики основной школы в условиях дифференциации содержания обучения // Преподаватель XXI век. 2024. № 1-1. С. 253–273. EDN: SVVVUO. DOI: 10.31862/2073-9613-2024-1-253-273.

2. *Гласс Р.* Креативное программирование 2.0: пер. с англ. С. Маккавеева. СПб.: Символ-Плюс, 2009. 352 с.

3. *Грунефелд В.* Креативный программист. СПб.: Питер, 2024. 272 с.

4. *Даниелян С. Д.* Разработка элективного курса языка программирования Kodu для младших школьников // Вопросы педагогики. 2022. № 4-2. С. 80–83. EDN: FELJST.

5. *Лейн Д.* Машинное обучение для детей. Практическое введение в искусственный интеллект: учебное пособие: пер. с англ. М. А. Федотенко. М.: Лаборатория знаний, 2023. 291 с.

6. *Самылкина Н. Н.* Образовательная робототехника — от модного тренда до педагогической технологии. Что дальше? // Информатика в школе. 2018. № 6 (139). С. 52–55. EDN: XZOONN.

7. *Тарапата В. В.* Scratch. Программировать сможет каждый: учебное пособие. М.: Лаборатория знаний, 2025. 299 с.

8. *Хорошевич П. А.* Использование высокоуровневых языков программирования для реализации креативного программирования // Актуальные проблемы методики обучения информатике и математике в современной школе. Материалы международной научно-практической интернет-конференции (Москва, 24–28 апреля 2023 года). М.: МПГУ, 2023. С. 791–796. EDN: LQYKUK.

9. *Хорошевич П. А.* Применение инструментов креативного программирования в ходе изучения алгоритмов машинного обучения // Информационно-коммуникационные технологии в педагогическом образовании. 2023. № 4 (85). С. 15–19. EDN: WNNUVX.

10. *Янцев В. В.* JavaScript. Креативное программирование: учебное пособие для СПО. 2-е изд., стер. СПб.: Лань, 2024. 232 с.