

DOI: 10.32517/2221-1993-2025-24-5-52-61

**М. С. Плотникова***Средняя общеобразовательная школа № 38, г. Таганрог, Россия*

ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ БЛОЧНЫХ ТЕХНОЛОГИЙ: ОБЗОР ПЛАТФОРМ ДЛЯ ОБУЧЕНИЯ ШКОЛЬНИКОВ

Аннотация

В условиях быстро развивающегося информационного общества вопрос эффективного обучения является как никогда актуальным, ведь многие традиционные подходы к обучению устаревают. В качестве дополнения к уже существующим подходам к обучению программированию и в рамках развития альтернативных подходов в статье рассматриваются методические аспекты обучения с применением блочных технологий на примере конкретных образовательных ресурсов. Согласно исследованиям, большинство информации человек получает через органы зрения, т. е. визуально, этим и обосновано главное достоинство блочных технологий программирования, где вместо статичного текста программы обучающийся создает код с помощью манипуляций графическими объектами — блоками. В статье анализируются достоинства визуального программирования для формирования вычислительного и алгоритмического мышления у обучающихся. Представлена характеристика таких платформ, как «ПиктоМир», Blockly.Ru, Scratch (Scratch Junior), EduBlocks, Minecraft: Education Edition, MIT App Inventor. Описаны особенности использования блочных сред на разных ступенях образования, а также перспективы интеграции таких технологий в учебный процесс.

Ключевые слова: программирование, блочные технологии, визуальное программирование, алгоритмическое мышление.

Информация об авторе

Плотникова Марина Сергеевна, учитель информатики, средняя общеобразовательная школа № 38, г. Таганрог, Россия; e-mail: marinapl-2001@yandex.ru

1. Введение

Согласно ФГОС, изучение информатики начинается не позднее седьмого класса и продолжается до выпуска из школы. На уровне основного общего образования на уроки информатики отводится в среднем один академический час в неделю. Одним из ключевых в курсе информатики является раздел «Алгоритмизация и программирование» [13], его изучение по утвержденному для школ УМК Л. Л. Босовой начинается со второго полугодия восьмого класса. Современное образование в области информатики ставит перед педагогами задачу формирования у обучающихся не только теоретических

знаний, но и практических навыков программирования. Программирование неразрывно связано с умением построения алгоритмов. Развитие алгоритмического мышления предполагает целостный подход, включающий в себя комплекс различных методов и упражнений в процессе всего обучения. В восьмом классе начинается изучение текстового языка программирования Pascal или Python. Это огромный пласт материала, не только включающий в себя использование базовых конструкций ввода-вывода, изучение на практике разных типов данных, но и подразумевающий работу с условиями, циклами и массивами. Однако традиционные текстовые языки на практике часто вызывают сложности у начи-

M. S. Plotnikova

School 38, Taganrog, Russia

BLOCK-BASED PROGRAMMING TECHNOLOGIES: A REVIEW OF PLATFORMS FOR TEACHING SCHOOLCHILDREN

Abstract

In a rapidly evolving information society, the issue of effective teaching is more relevant than ever, as many traditional approaches are becoming obsolete. To complement existing approaches to teaching programming and to develop alternative approaches, the article examines the methodological aspects of teaching using block-based technologies, using specific educational resources as examples. Research shows that people receive most information visually, which underlies the key advantage of block-based programming technologies. Instead of static program text, students create code by manipulating graphical objects — blocks. The article analyzes the benefits of visual programming for developing students' computational and algorithmic thinking. It presents the characteristics of platforms such as PiktoMir, Blockly.Ru, Scratch (Scratch Junior), EduBlocks, Minecraft: Education Edition, and MIT App Inventor. It describes the specifics of using block-based environments at different levels of education, as well as the prospects for integrating such technologies into the educational process.

Keywords: programming, block-based programming technologies, visual programming, algorithmic thinking.

Information about the author

Marina S. Plotnikova, an informatics teacher, School 38, Taganrog, Russia; e-mail: marinapl-2001@yandex.ru

нающих из-за необходимости запоминания синтаксиса и строгих правил написания кода.

Альтернативой выступают технологии блочного программирования, которые позволяют создавать программы путем визуального соединения блоков. Такой подход снижает когнитивную нагрузку на обучающихся, способствует быстрому пониманию ими алгоритмических структур и делает процесс обучения более доступным и увлекательным.

В рамках данной статьи рассмотрены методические основы обучения программированию с использованием блочных технологий и предложены рекомендации по их внедрению в образовательный процесс согласно возрастной дифференциации обучающихся [16].

2. Блочное программирование: понятие, особенности

Исторически визуальные средства для передачи информации являются одним из древнейших приемов, которыми пользуется человечество, ведь по разным данным от 70 до 90 % информации люди получают через органы зрения. В наши дни также везде постоянно используются символы для условного обозначения совершенно разных явлений — система пиктограмм является одной из наиболее универсальных для современного общества. Так или иначе, многие знаковые системы начинались с визуализации объектов, в них входящих, в связи с понятностью образов [1, 5]. Программирование также имеет тенденцию к упрощению и внедрению в него приемов наглядности.

Ф. Брукс, американский ученый в области вычислительных систем, писал о том, что основная проблема в обучении программированию заключается в том, что написанный код не материален, его невозможно увидеть. Помимо этого представить программу можно по-разному, ведь у каждого человека могут быть разные взгляды [3].

Под **визуальным программированием** понимают такой способ создания программы, в котором вместо написания текстового кода используют манипуляции графическими объектами. Некоторые представители научного сообщества считают визуальное программирование следующим этапом, или поколением, развития текстовых языков. Визуальное программирование — это основной прием наглядности в сфере информационных технологий.

Блочное программирование основано на принципе «drag and drop» (с *англ.* «перетащи и отпусти»). Действительно, каждый из элементов в такой среде программирования выглядит как блок, перемещаемый с помощью мыши, для составления кода из отдельных фрагментов. Нужный объект «захватывается» при помощи манипулятора, что является интуитивно понятной даже для начинающего пользователя механикой.

Блочные среды программирования рассматриваются в данной статье как частный случай визуальных средств. **Как инструмент обучения они обладают следующими особенностями:**

- **Визуальное представление кода** — программа собирается из готовых блоков, что исключает ошибки синтаксиса и не требует отдельного заучивания правил написания.

- **Интуитивная понятность** — блоки имеют четкую форму и семантику (циклы, условия, переменные), что облегчает понимание логики программы; также в большинстве языков можно встретить тематическое разделение блоков с помощью их деления по цветам.
- **Мультимедийная поддержка** — многие среды программирования позволяют работать с графикой, звуками и анимацией, повышая мотивацию обучающихся за счет разнообразия инструментов.
- **Быстрый результат** — даже новички могут создать работающую программу за короткое время; кроме того, любые изменения в коде влекут за собой мгновенное реагирование объектов в среде программирования.

Чтобы достичь поставленных целей в обучении программированию, важно выделить **этапы, на которых в процесс обучения могут быть интегрированы визуальные среды**. Так, наиболее эффективным приемом будет внедрение их в начале обучения, при формировании основных понятий, после чего может быть осуществлен более плавный переход к разработке простейших программ и изучению классических алгоритмов. Благодаря этому процесс обучения можно организовать так, чтобы реализовать основную содержательную задачу — моделирование алгоритмизации [14].

Блочное программирование способствует развитию ключевых компонентов вычислительного мышления. Обучающиеся учатся разбивать задачу на последовательность шагов, что закладывает основные принципы построения алгоритмов. Без навыка абстрагирования невозможно грамотно выделить главные элементы задачи с помощью игнорирования второстепенных. Благодаря пониманию декомпозиции сложная задача разделяется на более мелкие подзадачи. Также обучающиеся начинают адаптироваться и «привыкать» писать алгоритмы, а значит, учатся использовать некоторые готовые решения для новых задач [6, 12].

Так или иначе, большинство крупных и сложных проектов по-прежнему разрабатывается на каком-либо из существующих языков программирования, поэтому «блоки» на данный момент не исключают необходимости изучения «текста» при желании развиваться в данной профессиональной сфере. Это обеспечивает реализацию одной из парадигм современного образования, а именно его непрерывности.

При работе обучающихся в любой из сред программирования необходимо соблюдать определенные этапы обучения работе с этой средой. Так, начинать всегда следует со знакомства с интерфейсом среды программирования. На первых занятиях важно дать обучающимся возможность попробовать управление и манипуляцию блоками самостоятельно, без большого количества подсказок со стороны педагога, ведь большинство элементов составления скриптов являются интуитивно понятными. Часто на практике даже младшие школьники с помощью метода проб и ошибок или через визуальные подсказки самостоятельно адаптируются к составлению программ. Далее образовательная траектория является преимущественно стандартной — как и при изучении привычных для школьного образования языков программирования, происходит плавный пере-

ход от линейных программ к таким темам, как события, данные, условия, циклы [15].

Обязательным этапом следует выделить создание творческих проектов — как в рамках промежуточного контроля при переходе между темами, так и в качестве итогового продукта. Здесь важно со стороны педагога сформулировать строгий набор требований к программам на проверку: от возможного набора тем до минимального количества используемых спрайтов и скриптов, обязательных конструкций, а для оценки проекта опираться на критериальную базу, с которой обучающиеся заранее ознакомлены. Таким образом, в процессе работы роль педагога смещается в сторону наставничества над творческим проектом обучающегося.

3. Обзор платформ блочного программирования

Уже сейчас блочное программирование активно используется в образовании благодаря своей наглядности и доступности. Рассмотрим некоторые из известных инструментов, которые могут быть использованы в школьном курсе информатики с учетом возраста обучающихся.

3.1. «ПиктоМир»

Это разработанная в России свободно распространяемая программная система для изучения азов программирования дошкольниками и младшими школьниками [10]. Даже для малышей, которые еще не освоили чтение и письмо, информатика окажется понятным и увлекательным предметом. В «ПиктоМире» перед пользователем стоит главная задача — переместить робота, составив для него программу. Пример такой программы представлен на рисунке 1. В «ПиктоМире» существует несколько уровней сложности, на каждом из

которых ребята знакомятся с новыми видами алгоритмов: от простых линейных до циклических. Достоинство этой платформы в том, что существует ограничение на количество команд на каждом уровне, а значит, написанный алгоритм необходимо оптимизировать: без этого переход на следующий уровень недоступен.

Данная среда программирования является «младшим братом» «КуМира», но каждая из них работает независимо друг от друга. Тем не менее программу, разработанную в «ПиктоМире», можно сохранить с расширением *.kum, что позволит продолжить программирование в «КуМире», если возникнет такая необходимость.

Работа с простыми исполнителями — отличный прием для развития алгоритмического мышления, ведь действия таких исполнителей достаточно легко представить и смоделировать в жизни.

«ПиктоМир» разработан в Федеральном научном центре Научно-исследовательский институт системных исследований Национального исследовательского центра «Курчатовский институт» (НИЦ «Курчатовский институт» — НИИСИ) по заказу Российской академии наук. С 2013 года разработчики поддерживают две поставки «ПиктоМира» — классическую и мобильную, причем последняя имеет достаточно хороший рейтинг в магазинах приложений и хорошие отзывы от пользователей. Данное программное обеспечение также отличается кроссплатформенностью.

Отдельным достоинством «ПиктоМира» является наличие разнообразных открытых разработок на русском языке, представленных на официальном сайте, что упрощает методическую работу учителя.

Опыт использования «ПиктоМира» педагогами позволяет сделать вывод о том, что визуальное программирование актуально внедрять даже на начальных этапах развития алгоритмического мышления для самой младшей возрастной группы будущих программистов [7, 11].

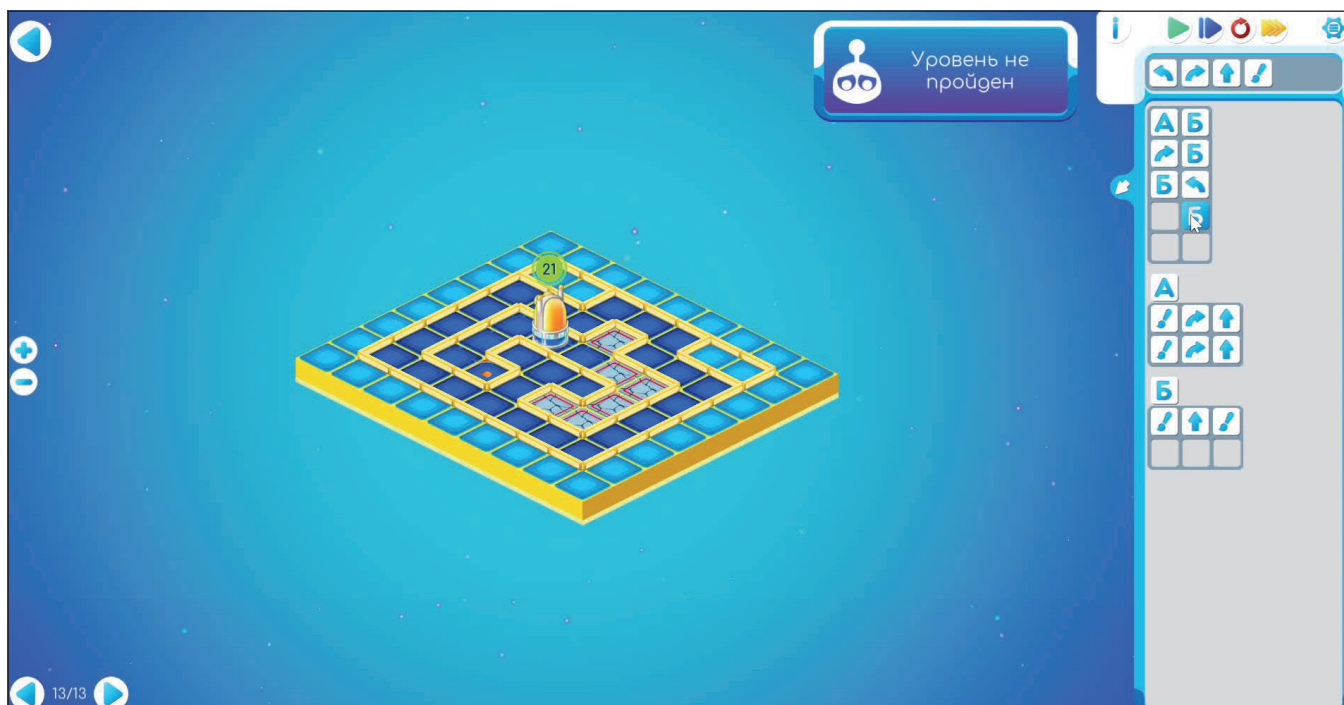


Рис. 1. Среда программирования «ПиктоМир»

3.2. Blockly.Ru

Следующий ресурс, весьма обширный по своему функционалу, — это «образовательный проект для будущих программистов» (как его называют авторы) Blockly.Ru [17]. Сайт содержит несколько разделов — «Среда программирования Blockly», «Blockly — HTML», «Игры» и др., — и в нем есть задания для самых разных возрастов: упражнения по интересам найдут все — от младших школьников до выпускников и продвинутых пользователей, поэтому обращаться к нему учитель сможет в любом классе.

Так, самое простое задание на знакомство с блочной средой «фауна» отлично подходит для тех, кто первый раз сталкивается с языком визуального программирования

Blockly и еще не знаком с приемом соединения блоков. Также на сайте представлены игры, чем-то похожие на систему «ПиктоМира»: необходимо помочь кролику Банни дойти домой, составляя для него маршрут из кусочков кода, как это видно на рисунке 2.

Для учащихся основной школы существует блок с программированием черепашки и соответствующий набор заданий.

Более старшая возрастная группа в свою очередь тоже не осталась обделена: создатели Blockly.Ru показывают, как разработанная ими среда программирования Blockly может помочь школьникам визуализировать структуру программ, которые они пишут на привычных текстовых языках. В качестве примера на рисунке 3 представлен язык разметки гипертекста HTML. Особое



Рис. 2. Сайт Blockly.Ru. Игра «Банни идет домой»

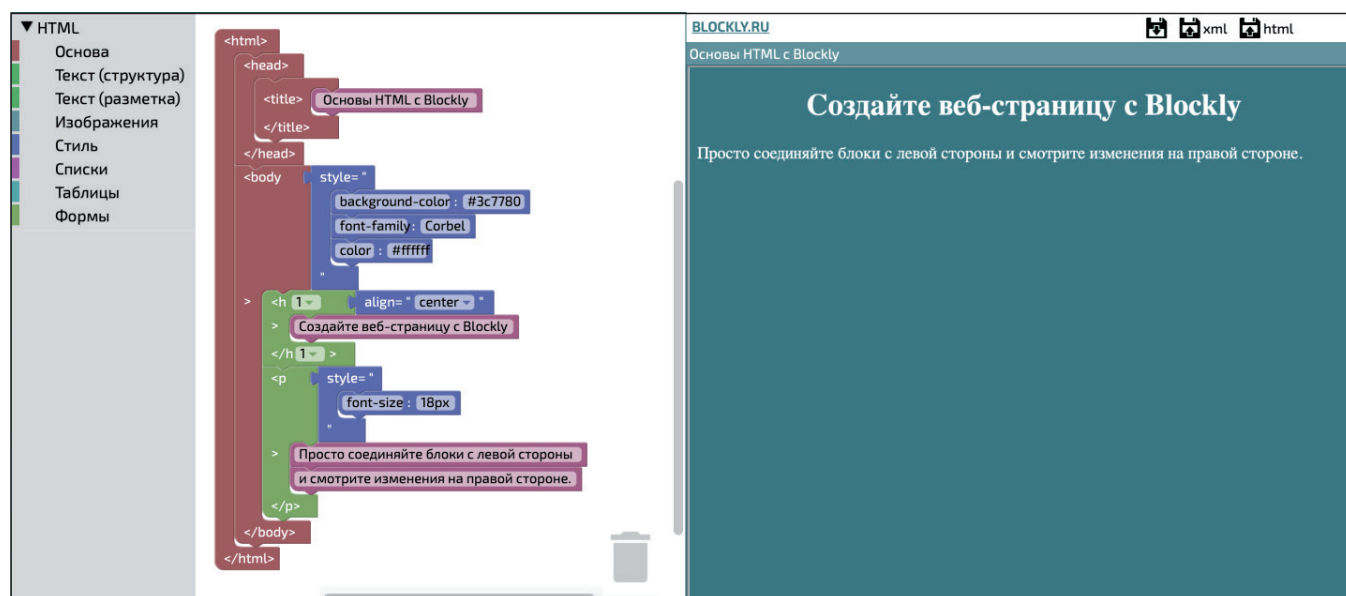


Рис. 3. Сайт Blockly.Ru. Среда программирования Blockly — HTML

внимание здесь заслуживает режим разделения экрана, в котором справа отображается сгенерированная с помощью кода интернет-страница.

Один из главных плюсов данной платформы — это большое разнообразие статей и методических разработок на русском языке в открытом доступе. Многие платформы также используют визуальный язык программирования Blockly в своих проектах, разработчики открыты к новым дополнениям, а также к переводам на различные языки.

3.3. Scratch

Визуальный событийно-ориентированный язык Scratch [20], разработанный в 2007 году Массачусетским

технологическим институтом, давно отлично зарекомендовал себя как эффективный инструмент в работе учителей информатики. Это более свободная среда разработки мультфильмов и игр по сравнению с Blockly, ведь пользователь всегда имеет доступ к полному набору команд при работе с проектом. Scratch идеально подходит для того, чтобы начинающий программист познакомился с алгоритмами, научился их составлять, а также разобрался с базовыми понятиями в программировании. Зачастую на практике для детей 7–13 лет Scratch является наиболее актуальным выбором в качестве первого этапа изучения программирования. С помощью блоков намного проще сопоставить взаимосвязь команд в алгоритме, цветовое

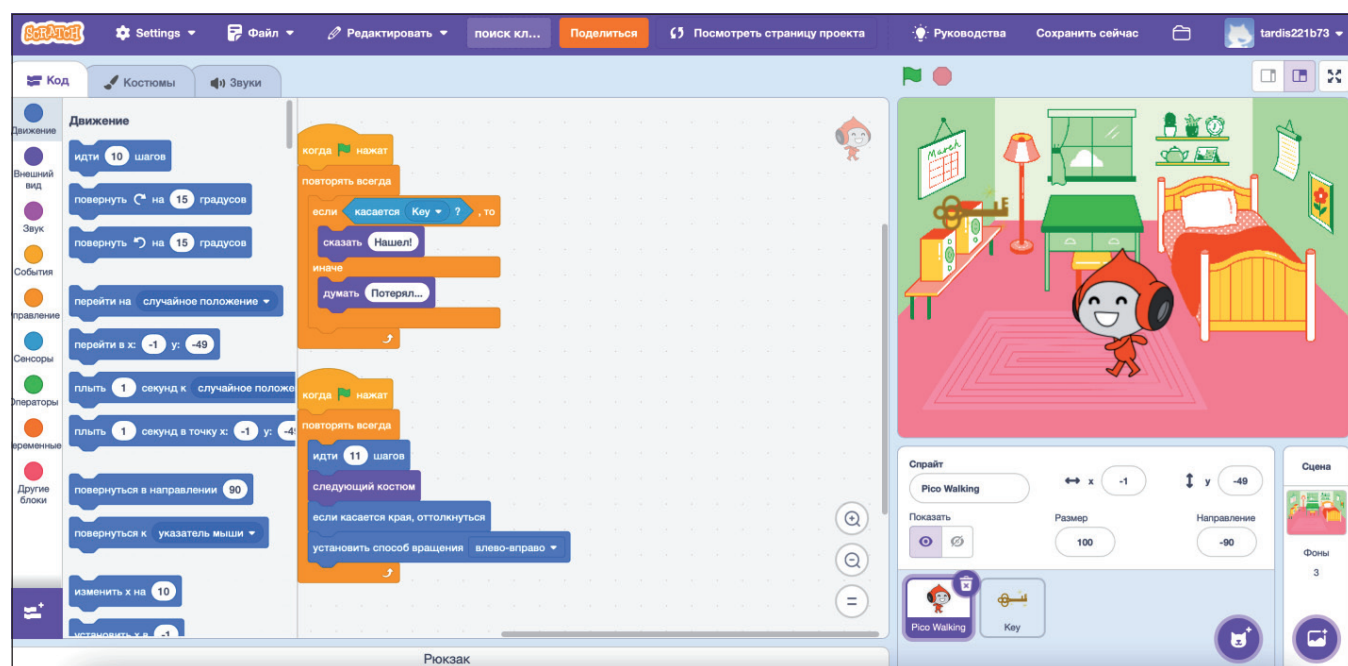


Рис. 4. Среда программирования Scratch

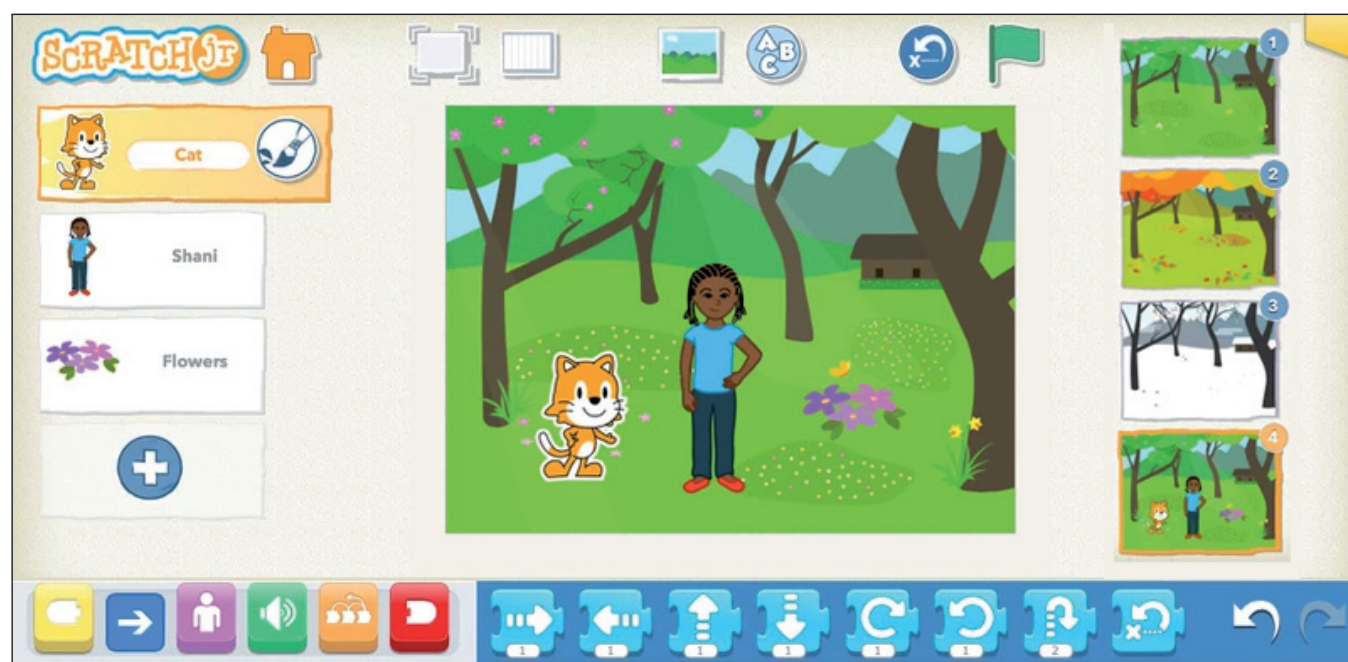


Рис. 5. Среда программирования Scratch Junior

решение расставляет акценты при изучении разных тем, отступы помогают визуально закрепить основные принципы при составлении условий и работе с циклами, что наглядно видно на рисунке 4.

Также обучающийся сразу видит результат своих действий: изменение разных фрагментов кода и порядка команд незамедлительно влечет за собой изменения в поведении спрайтов. В игровой форме происходит также первое знакомство с двумерной системой координат. Подключаемые библиотеки позволяют пользователю активно использовать в своих проектах внешние устройства компьютера, в том числе поддерживается работа с микрофоном, колонками и даже камерой. Кроме того, поддерживается интеграция с облачными ресурсами, например, с Google Переводчиком. Возможности Scratch используются и в робототехнике — при применении робототехнических наборов, таких как Makey Makey, Micro:bit, LEGO BOOST и др.

Scratch имеет свое постоянно развивающееся активное сообщество юных программистов, где каждый может поделиться своими разработками с другими, обсудить проекты и найти друзей по интересам. Такое общение и возможность обмениваться опытом является значительной мотивацией при обучении программированию. Базовые руководства и уроки-знакомства также помогают начинающим не потеряться в новой среде разработки. Несмотря на то что Scratch часто называют «детским» языком программирования, его функционал позволяет создавать большие и сложные проекты, прототипы игр со сложной графикой и механикой. Scratch также имеет «младшего брата» **Scratch Junior** для юных программистов без навыков чтения (рис. 5).

3.4. EduBlocks

Образовательная онлайн-платформа EduBlocks — это синтез Scratch и Python, прорыв в визуальном программировании [18]. С помощью данной среды можно

работать с HTML 5, Python 3, а также применять ее в робототехнике. На разноцветных блоках, принцип соединения и внешний вид которых аналогичны Scratch, написаны команды языка Python. Отдельно находятся блоки импорта библиотек, работы с переменными, выражениями, логические операторы, циклы, списки, функции, а также непосредственно команды из разных библиотек. Эта визуализация позволяет отделить одни понятия от других, сделать обучение более структурированным. Проверка работы созданной программы производится при нажатии на кнопку «Run» во встроенном онлайн-компиляторе Trinket. EduBlocks также поддерживает кроссплатформенность. Код, который автоматически генерируется при работе с блоками в EduBlocks, можно в каждый момент времени сохранить в файл с расширением *.py и открыть в любой среде программирования, поддерживающей язык Python. Ограничения при программировании на EduBlocks можно снять с помощью пустой команды `#your own code`, как в примере на рисунке 6. То же справедливо и при работе с HTML.

Основное достоинство платформы EduBlocks — возможность отображения кода в виде блоков и в виде текстового описания на Python/HTML. Платформа также предлагает режим параллельного просмотра, где все изменения в программе мгновенно отображаются. Это позволяет легко сопоставить блоки с настоящим кодом и делает переход на чисто текстовый язык более плавным.

Благодаря визуализации каждой команды и связей между командами обучающимся проще понять основные принципы составления алгоритмов и порядка выполнения команд в нем. Также при изучении Python часто возникает сложность при расстановке нужного количества пробелов в коде — а блоки в EduBlocks позволяют акцентировать на этом внимание, что особенно важно в работе с условиями и циклами.

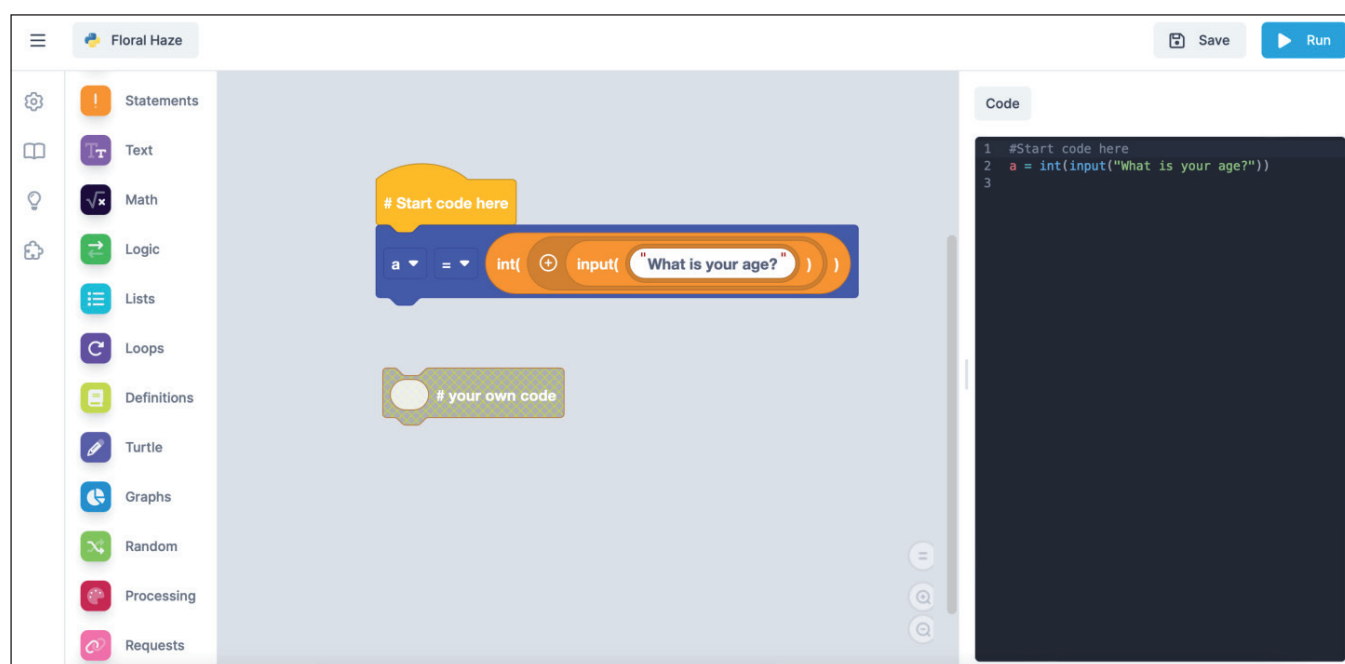


Рис. 6. Среда программирования EduBlocks — Python

Таким образом, работа с онлайн-платформой EduBlocks будет полезна как переходный этап к изучению текстовых языков: программируя в уже знакомой по работе в Scratch блочной среде, обучающийся запоминает названия новых команд на языке Python/HTML и намного быстрее осваивает их. Все, что нужно для работы на платформе EduBlocks, — это доступ в интернет.

Подводя итог, стоит сказать о том, что с помощью блоков намного проще сопоставить взаимосвязь команд в алгоритме, цветовое решение расставляет акценты при изучении разных тем, отступы помогают визуальнo закрепить основные принципы при составлении условий и работе с циклами. Также обучающийся сразу видит результат своих действий: изменение разных фрагментов кода и порядка команд незамедлительно влечет за собой изменение в поведении программы [2].

3.5. Minecraft: Education Edition

Minecraft — одна из самых популярных игр современности, покори́вшая сердца миллионов пользователей по всему миру. Она разработана в жанре песочницы студией Mojang в 2011 году и известна своим уникальным стилем. Все объекты в Minecraft состоят из достаточно крупных пикселей, игровой мир строится из кубиков. Отличительной особенностью этой игры является возможность добывать самые разнообразные ресурсы и крафтить новые предметы.

В 2016 году корпорация Microsoft приобрела права на игру и поддержала разработку ее образовательной версии — Minecraft: Education Edition (Minecraft EDU). Главным отличием этой версии является наличие специального мода Агента, управление которым реализуется через среду блочного программирования. Агент может заниматься строительством масштабных объектов (пример программы представлен на рисунке 7). При написа-

нии алгоритмов для его работы очень важен этап проектирования будущей модели и понимание циклических конструкций. В игровой форме проверяется понимание координат и масштабов, развивается пространственное мышление, ведь Агент может столкнуться с ограничениями, понимание которых невозможно без приема «поставь себя на место исполнителя». Мир, материалы и mobs, уже знакомые из стандартной версии любимой игры, увлекают и помогают обучающимся сохранять устойчивую мотивацию.

Minecraft: Education Edition помогает проводить уроки по различным школьным дисциплинам — от математики до химии. Здесь легко моделировать обучающие квесты с заданиями, проводить опыты и опросы. Также можно организовать коллективное обучение на заранее созданных серверах.

Платформа не только позволяет использовать уже готовые материалы, но и дает возможность разрабатывать свои. Основной минус — стоимость работы. С сентября 2016 года школам стала доступна годовая подписка на Education Edition стоимостью \$1–5 за учащегося в зависимости от выбранного пакета [9].

3.6. MIT App Inventor

В качестве одного из инструментов можно также рассматривать платформу MIT App Inventor [19], разработанную и поддерживаемую Массачусетским технологическим институтом, которая является визуальной средой программирования для разработки мобильных приложений. За время ее существования с помощью данного инструмента было создано более 85 миллионов программ, 18 миллионов пользователей являются постоянными. В среднем в течение месяца миллион человек работают с App Inventor. Платформа активно поддерживается и обновляется, количество разработок за день может достигать 75 тысяч.

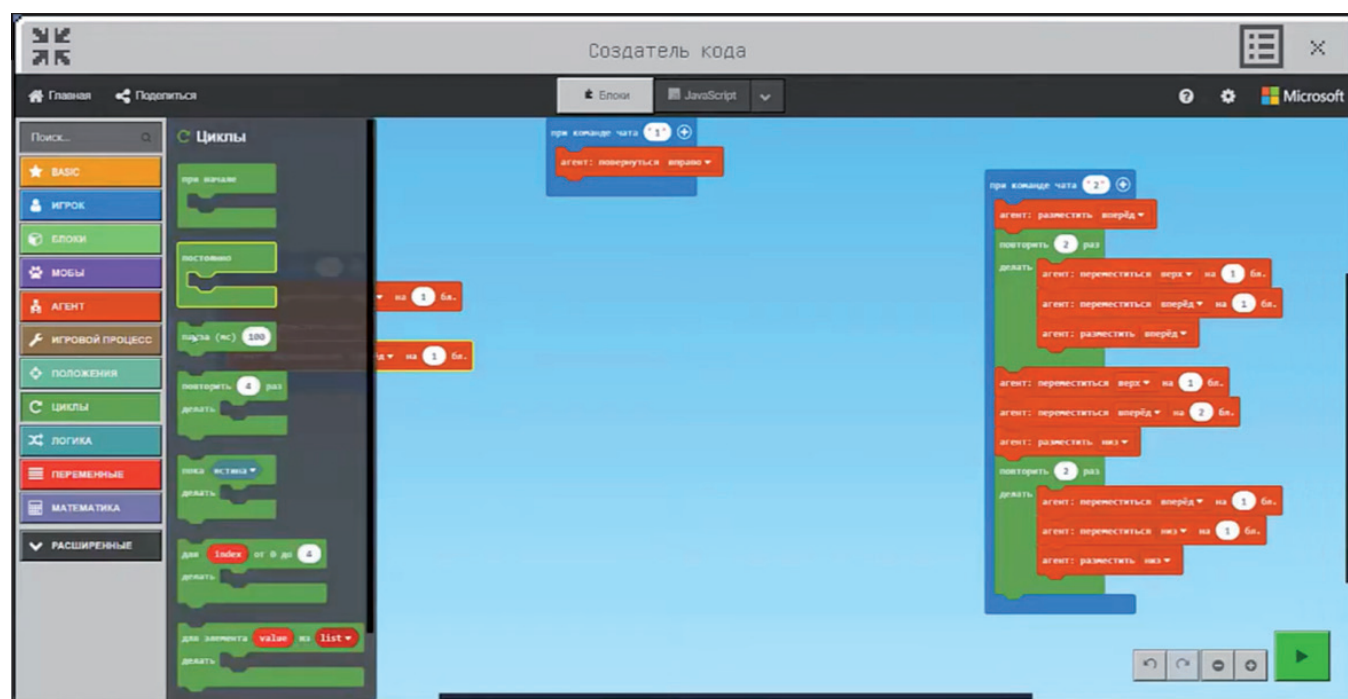


Рис. 7. Среда программирования Minecraft: Education Edition

Создание мобильных приложений — это актуальное направление, которое представляет из себя трудоемкий продолжительный процесс, требующий от программиста знания тестовых языков и умения работать со специализированными подключаемыми библиотеками, а также подразумевает наличие навыков использования графических элементов. Это действительно сложно, но благодаря платформе App Inventor каждый может почувствовать себя настоящим разработчиком и создать свое работающее приложение. Наличие готового продукта, который можно использовать в обычной жизни, — это высокая мотивация для обучающихся, ведь для современного поколения очень важно иметь практическое воплощение результатов своей работы в наиболее сжатые сроки.

Для того чтобы работать с App Inventor, необходимо только иметь доступ в интернет и браузер. App Inventor основан на облачных технологиях, поэтому в начале работы необходимо привязать Google-аккаунт. Это позволяет иметь доступ к своим данным с любого устройства и продолжать проекты в удобное время.

Несмотря на то что сайт встречает английским интерфейсом, после входа появляется возможность выбрать язык, в том числе разработчики официально представляют поддерживаемую русскую версию для работы с программируемыми блоками.

При открытии режима создания перед пользователем появляется область настройки интерфейса будущего приложения (как показано на рисунке 8). С помощью встроенных инструментов (они расположены слева) можно добавить разнообразные мультимедиакомпоненты (фото, видео, звуки), а также элементы взаимодействия с контентом (кнопки, флажки, формы ответов, поля для рисования и т. д.).

Чтобы начать разработку кода, необходимо в правой верхней части экрана перейти в раздел «Блоки» (его

можно видеть на рисунке 9). Рабочая область напоминает Scratch, что сразу дает характеристику основным принципам работы в App Inventor: на разноцветных блоках, сгруппированных по области применения, написаны команды на русском языке. Для построения кода необходимо собрать скрипты из элементов так, будто бы это кусочки пазла, что очевидно прослеживается из дизайна.

Одно из главных преимуществ App Inventor по сравнению со Scratch заключается в том, что App Inventor при наведении курсора на элемент в библиотеке дает пользователю подсказку: для чего предназначен этот блок. Это значительно упрощает процесс программирования и экономит время при обращении к справочным материалам.

MIT App Inventor также предлагает поддержку, необходимую разработчикам приложений, по следующим пунктам: инструкции для установки и первичного тестирования, дизайн блоков в режиме редактирования, обучающие материалы для начинающих и описание последовательности действий для установки разработанных приложений. Несмотря на то что на данной платформе весь код создается с помощью цветных блоков, результат можно рассматривать как настоящее прикладное программное обеспечение: устанавливать приложения можно на разные устройства.

Отдельно стоит отметить качество представляемого на платформе комплекта учебно-методических материалов. К сожалению, он представлен только на английском языке, родном для создателей, но современные технологии перевода текста и видео позволяют не считать это серьезной проблемой. Сообщество MIT App Inventor очень активно и постоянно поддерживает создание новых уроков, всегда открыто к обсуждению идей и помогает в решении вопросов и возникающих при программировании трудностей. Каждый из обучающихся

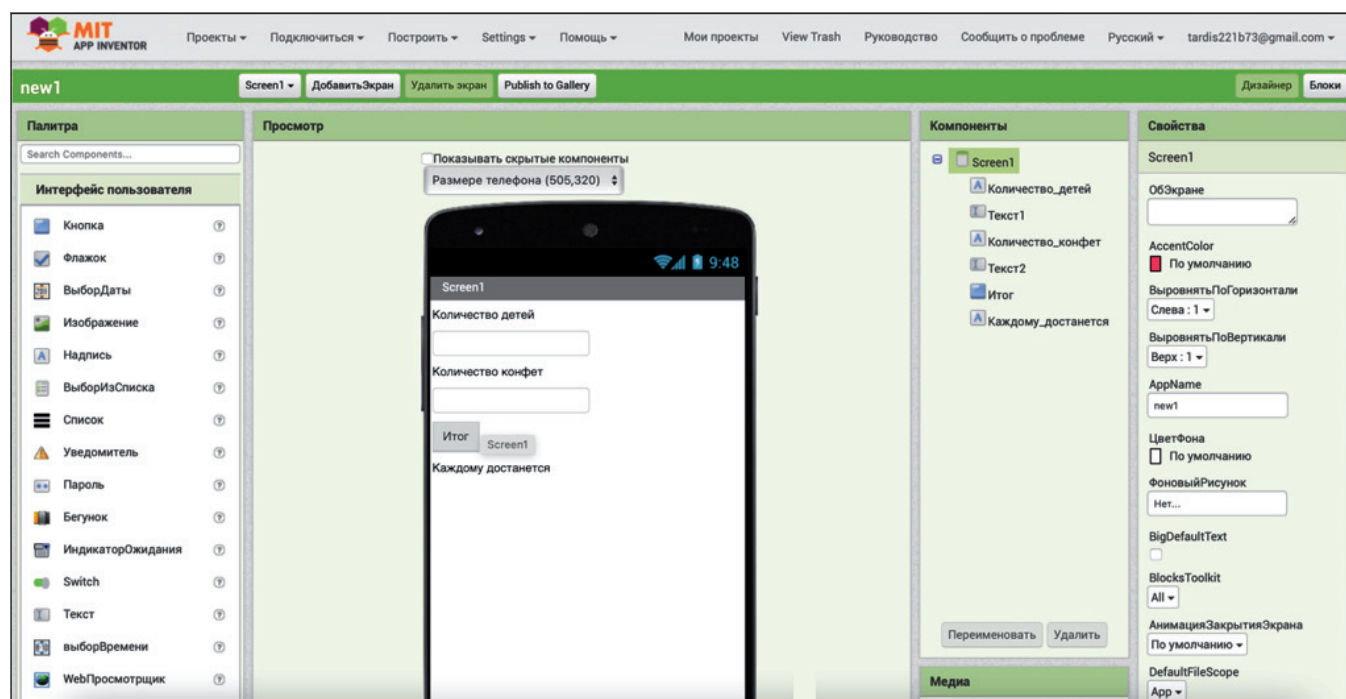


Рис. 8. Интерфейс режима разработчика MIT App Inventor

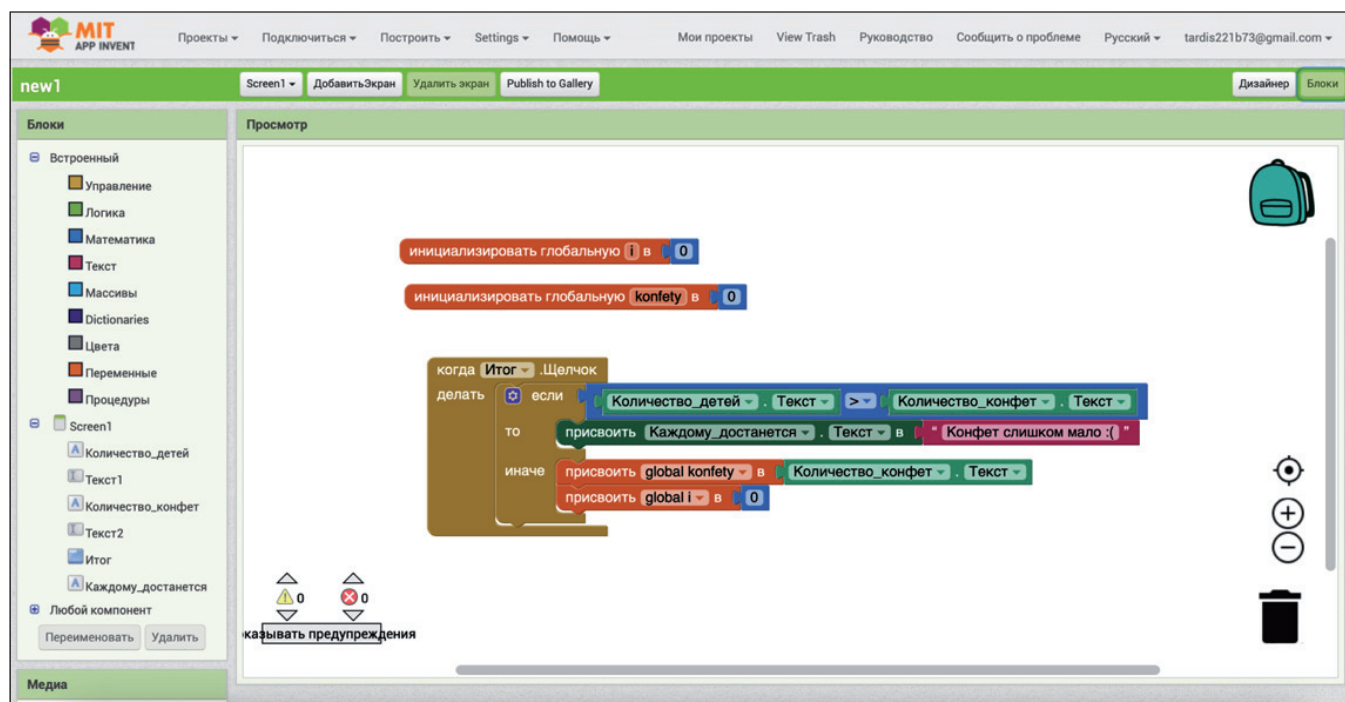


Рис. 9. Среда программирования MIT App Inventor

материалов разделен по уровням сложности — простой (*англ.* basic), средний (*англ.* intermediate), продвинутый (*англ.* advanced) — и оснащен тегами основных используемых компонентов. Это позволяет пользователю подобрать тот материал, который будет соответствовать способностям и интересам обучающегося [8, 11].

4. Заключение

Важно, что понятийный аппарат, используемый в визуальном программировании, совпадает с понятийным аппаратом текстовых языков, что позволяет обеспечить более плавный переход между этапами обучения. Прием визуализации с помощью геометрических фигур и их взаимных расположений относительно друг друга в свою очередь обеспечивает более высокий уровень понимания обучающимися основных принципов программирования из-за того, что более активно задействуются глаза как часть системы органов чувств [4].

«Лучше один раз увидеть, чем несколько раз услышать» — это основной принцип визуального программирования. Блоки, выделенные разными цветами в зависимости от использованной темы, помогают более ярко закрепить создаваемый образ и более четко разделить используемые понятия. Данный прием является отличным инструментом для создания четкой структуры алгоритмов как на начальных этапах изучения программирования, так и на этапе закрепления ранее полученных знаний и навыков.

Рассмотренные в статье примеры платформ демонстрируют тот факт, что визуальное программирование актуально для разных возрастных групп и позволяет решать широкий спектр практических задач в обучении алгоритмизации и программированию.

Развитые навыки алгоритмического мышления являются одной из ключевых характеристик современного

человека — гражданина информационного общества. Вряд ли визуальное программирование когда-нибудь полностью заменит текстовое, но за ним однозначно будущее в обучении информатике, потому что современное поколение — поколение визуалов и носителей клипового мышления. Это новый этап, к которому должны быть готовы люди, организующие образовательный процесс.

Список источников

1. *Артемов В. А.* Психология наглядности при обучении. М.: Просвещение, 1998. 268 с.
2. *Белоконова С. С., Плотникова М. С.* Блочный Python как переходный этап в обучении программированию // Информатика в школе. 2021. № 7 (170). С. 57–62. EDN: FJUIIS. DOI: 10.32517/2221-1993-2021-20-7-57-62.
3. *Брукс Ф.* Мифический человек-месяц, или Как создаются программные системы. СПб.: Питер, 2021. 368 с.
4. *Занков Л. В.* Наглядность и активизация учащихся в обучении. М.: ГУПИ МП РСФСР, 1960. 312 с.
5. *Коменский Я. А.* Великая дидактика. Педагогическое наследие. М.: Педагогика, 1939. 123 с.
6. *Куликова Н. Ю., Пономарева Ю. С.* Возможности интерактивных сетевых средств при обучении информатике и ИКТ в школе // Continuum. Математика. Информатика. Образование. 2020. № 2 (18). С. 96–106. EDN: GOLGOY. DOI: 10.24888/2500-1957-2020-18-2-96-106.
7. *Леонов А. Г., Первин Ю. А., Зайдельман Я. Н.* Программные исполнители в цифровых образовательных средах «ПиктоМир», «Роботландия» и «КуМир» // Информатика в школе. 2019. № 9 (152). С. 54–61. EDN: ZCBQIC. DOI: 10.32517/2221-1993-2019-18-9-54-61.
8. *Малова А. И.* Обучение программированию школьников с использованием визуальной среды AppInventor при создании компьютерных игр для мобильных устройств // Молодежь и наука: шаг к успеху. Сборник научных статей IV Всероссийской научной конференции перспективных разработок молодых ученых (г. Курск, 19–20 марта 2020 года). Курск: Изд-во Юго-Западного гос. ун-та, 2020. С. 76–80. EDN: CUANKI.
9. *Мишин В. А.* Использование платформы Minecraft: Education Edition на уроках информатики // Информатика

в школе. 2021. № 3 (166). С. 50–59. EDN: SJCQSM. DOI: 10.32517/2221-1993-2021-20-3-50-59.

10. ПиктоМир. <https://piktomir.ru>

11. Самылкина Н. Н., Етова А. В. Создание мобильных приложений в MIT App Inventor с использованием элементов геймификации при изучении программирования в основной школе // Информатика в школе. 2020. № 8 (161). С. 37–45. EDN: SRMOKO. DOI: 10.32517/2221-1993-2020-19-8-37-45.

12. Сиденко А. Г. Использование игровых сред для обучения программированию // Информатизация непрерывного образования. Материалы Международной научной конференции. В 2 т. / под общ. ред. В. В. Гриншуна. М.: РУДН, 2018. С. 491–496.

13. Федеральный государственный образовательный стандарт основного общего образования. <https://edsoo.ru/>

wp-content/uploads/2023/08/Приказ-№-287-от-31.05.2021-ФГОС_ООО.pdf

14. Флойд Р. Парадигмы программирования // Лекции лауреатов премии Тьюринга за первые двадцать лет 1966–1985. М.: Мир, 1993. С. 86–98.

15. Широких А. А., Костарева Е. А. Графические среды для обучения школьников программированию // Информатика в школе. 2016. № 6 (119). С. 33–38. EDN: WMILX.

16. Якиманская И. С. Образное мышление и его место в обучении // Советская педагогика. 1968. № 12. С. 62–71.

17. Blockly.Ru — Для будущих программистов — For tomorrow's programmers. <http://blockly.ru/index.html>

18. EduBlocks. <https://app.edublocks.org>

19. MIT App Inventor. <https://appinventor.mit.edu>

20. Scratch — Imagine, Program, Share. <https://scratch.mit.edu>