

Пономарев М.В., Рожина И.В.

ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ МЛАДШИХ ШКОЛЬНИКОВ В СРЕДЕ ПРОГРАММИРОВАНИЯ SCRATCH

Аннотация

В статье обосновывается применение Scratch при изучении раздела «Алгоритмизация и программирование» младшими школьниками. Рассматриваются возрастные психологические особенности в развитии мышления в данном возрасте. Обосновывается польза изучения информатики вообще, и алгоритмизации и программирования в частности. Рассматривается среда разработки Scratch, его особенности.

Ключевые слова: программирование, информатика, мышление детей, начальная школа, младшие школьники, начальное обучение информатике.

Ponomarev M.V., Rozhina I.V.

PROGRAMMING TRAINING JUNIOR SCHOOL STUDENTS IN SCRATCH PROGRAMMING ENVIRONMENT

Abstract

The article substantiates the use of Scratch when studying section «algorithmization and programming» by junior school pupils. Considered age psychological characteristics in the development of thinking in this age. Justifying the use of studying computer science in General, and an algorithmization and programming in particular. The Scratch development environment viewed and it is features.

Keywords: programming, computer science, thinking of children, elementary school, junior schoolchildren, elementary education in computer science.

ПРОБЛЕМА ИССЛЕДОВАНИЯ

Младший школьный возраст характеризуется активным переходом от практических действий в решении задач к решению их в уме, т. е. к более сложным формам мышления. И в современном мире становится важной уже не полученная информация из какого-либо источника, а применение этой информации в жизнедеятельности человека и общества. Современному человеку крайне необходимы умения думать, рассуждать, анализировать, делать выводы и строить «алгоритмы» своей деятельности, т. е. с раннего возраста должно развиваться логическое и алгоритмическое мышления.

В психологии, мышлением называют психический процесс отражения действительности, высшая форма творческой активности человека. Младший школьный возраст характеризуется активной «перестройкой» мышления ребенка. Младший школьник переходит от конкретно-образного мышления к словесно-логическому и рассуждающему мышлению [1]. Согласно положениям Л. С. Выготского о системном характере развития высших психических функций, в младшем школьном возрасте «системообразующей» функцией является мышление, и это сказывается на других психических функциях, которые интеллектуализируются, осознаются и становятся произвольными [2].

В школьном курсе информатики одним из важнейших разделов является «Алгоритмизация и программирование», что подтверждается как минимум тем, что в независимости от того, является ли информатика базовой, профильной или углубленной на этот раздел тратится в среднем треть всего учебного времени. При решении задач в разделе «Алгоритмизация и программирование», ученики должны [4]:

1. Четко понимать задачу.
2. Проанализировать задачу, чтобы понять, какими способами её можно решить.
3. Составить алгоритм решения задачи.
4. Исправить ошибки, которые могли возникнуть при выполнении алгоритма.

Вышеперечисленные пункты относятся практически к любой потенциальной задаче, с которой может встретиться человек в своей жизни, т.е. люди постоянно формируют алгоритмы своей деятельности.

Но возникает проблема средств обучения, при помощи которых мы можем начать детей учить алгоритмизации и программированию. Нам точно не подходят языки программирования, которые обычно предполагаются в обучении этого раздела, например, PascalABC.NET, VisualBasic.NET или Си/C++, т.к. эти языки рассчитаны на учеников старшей ступени обучения. Традиционно в пропедевтическом курсе информатики при изучении раздела «Алгоритмизация и программирование» ученики работают с исполнителями алгоритмов. Рассмотрим их:

- Чертежник. Предназначением данного исполнителя является построение рисунков, графиков, чертежей и т.д. Чертежник имеет перо, которое можно поднимать, опускать и перемещать. При перемещении опущенного пера за ним остается след – отрезок от старого положения пера до нового. Всего Чертежник умеет выполнять четыре команды: опустить перо, поднять перо, сместиться в точку (x, y) , сместиться на вектор (a, b) [6].

- Робот. Робот действует на прямоугольном клетчатом поле. Между некоторыми клетками могут быть расположены стены. Какие-то клетки могут быть закрашены. Сам Робот всегда занимает ровно одну клетку поля. Робот умеет выполнять всего 17 команд: 5 команд-приказов и 12 команд-вопросов. Команды – приказы: вверх, вниз, вправо, влево, закрасить. По командам вверх, вниз, влево, вправо Робот перемещается в соседнюю клетку в указанном направлении. Если на пути оказывается стена, команда не может быть выполнена. По команде закрасить Робот закрашивает клетку, в которой стоит. Если клетка уже была закрашена, она останется закрашенной, т.е. команда будет выполнена, но никаких видимых изменений не произойдет [6].

- Черепаха. Черепаха во многом похожа на чертежника. Разница в способе перемещения. В то время, как чертежнику необходимо задавать координаты, черепахе мы подаем команды вперед, назад и влево, вправо. Командам вперед и назад мы подаем число, отвечающее за количество шагов,

которые следует пройти, а у команд влево и вправо мы определяем угол поворота [5].

Однако нам необходимы более красочные и интересные средства, которые мотивировали бы учеников заниматься, побудили бы у детей желание к саморазвитию, а также были более богаты в плане обучения программированию и алгоритмизации.

ПРИМЕНЕНИЕ SCRATCH В ОБУЧЕНИИ

Таким средством может служить среда разработки и язык Scratch. Он был разработан Массачусетским технологическим институтом для обучения программированию с 8 до 16 лет. На 2017 год в онлайн-сообществе Scratch было зарегистрировано более 22 миллионов пользователей, он поддерживает более 70 языков, в том числе и русский [8].

Scratch является визуальной событийно-ориентированной средой программирования для школьников младших и средних классов. Целью создания этого языка было позволить детям, у которых нет опыта программирования, изучить основные принципы императивного, объектно-ориентированного и многопоточного программирования [3]. Этот язык удобно использовать как начальный язык программирования, потому что создавать проекты достаточно легко, а полученные навыки могут применяться в таких языках, как Python и Java [8], т. к. в Scratch присутствуют [3]:

- стандартные для языков процедурного типа: следование, ветвление, циклы, переменные, типы данных (целые и вещественные числа, строки, логические, списки – динамические массивы), псевдослучайные числа;
- объектно-ориентированные средства: объекты (их поля и методы), передача сообщений и обработка событий;
- интерактивные средства: обработка взаимодействия объектов между собой, с пользователем, а также событий вне компьютера (при помощи подключаемого сенсорного блока);
- средства параллельного выполнения: запуск методов объектов в параллельных потоках с возможностью координации и синхронизации;
- создание простого интерфейса пользователя.

Однако, прежде чем однозначно сказать, что Scratch является идеальной средой для обучения программированию в младшем школьном возрасте, нам необходимо обосновать его применение в младшем школьном возрасте. Для этого, давайте рассмотрим подробнее саму среду.

Как было сказано выше Scratch является визуальной событийно-ориентированной средой программирования. Его применение ограничено исключительно образовательными и развлекательными целями. разработчики позиционируют Scratch, как среду для создания различных игр, историй, где оживают созданные детьми персонажи, где первоочередной задачей ставится реализация детьми какой-либо истории, а уже затем обучение программированию. Так что идеальным вариантом будет постановка развлекательно-образовательной задачи на уроках, в которой учащиеся могли бы реализовы-

вать собственные задумки и учиться программировать, соответственно развивать творческое, логическое и алгоритмическое мышление.

Например, мы можем в качестве практического задания дать детям собрать вечеринку и позвать на неё всех, кого бы они хотели на ней видеть и запрограммировать всех гостей на различные танцы. Вот пример выполненного задания [7].



Рис. 1. Сцена и танцоры



Рис. 2. Код спрайта «Спрайт 2»

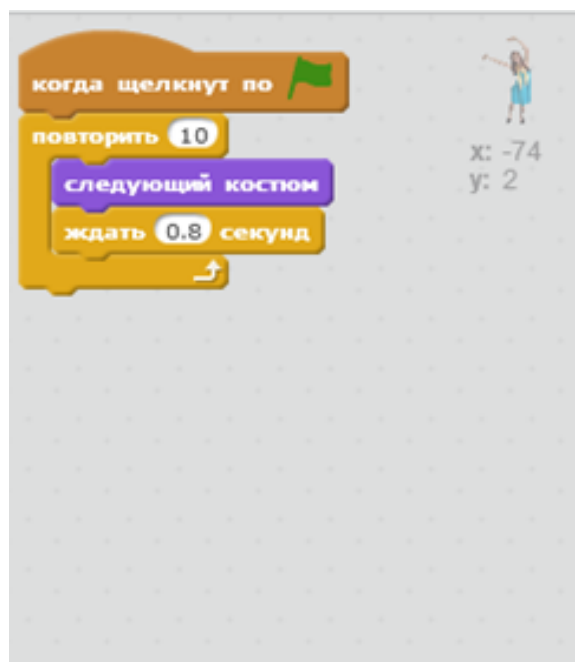


Рис. 3. Код спрайта «Catherine»

На рисунке выше представлен программа спрайта «Спрайт 2» (рыжий кот). В этой программе он по нажатию зеленого флажка в течении 2 секунд высвечивает слово «Hello» и затем выполняет цикл, в котором описан «танец» – игра барабанов и хождение туда-сюда.

У данного спрайта в библиотеках среды есть несколько «костюмов» – набор спрайтов с другими движениями. В коде этого объекта можно увидеть, что в течении 10 раз данный спрайт меняет свой «костюм» с интервалом 0.8 секунд. Также из стандартной библиотеки был выбран фон для этой программы.

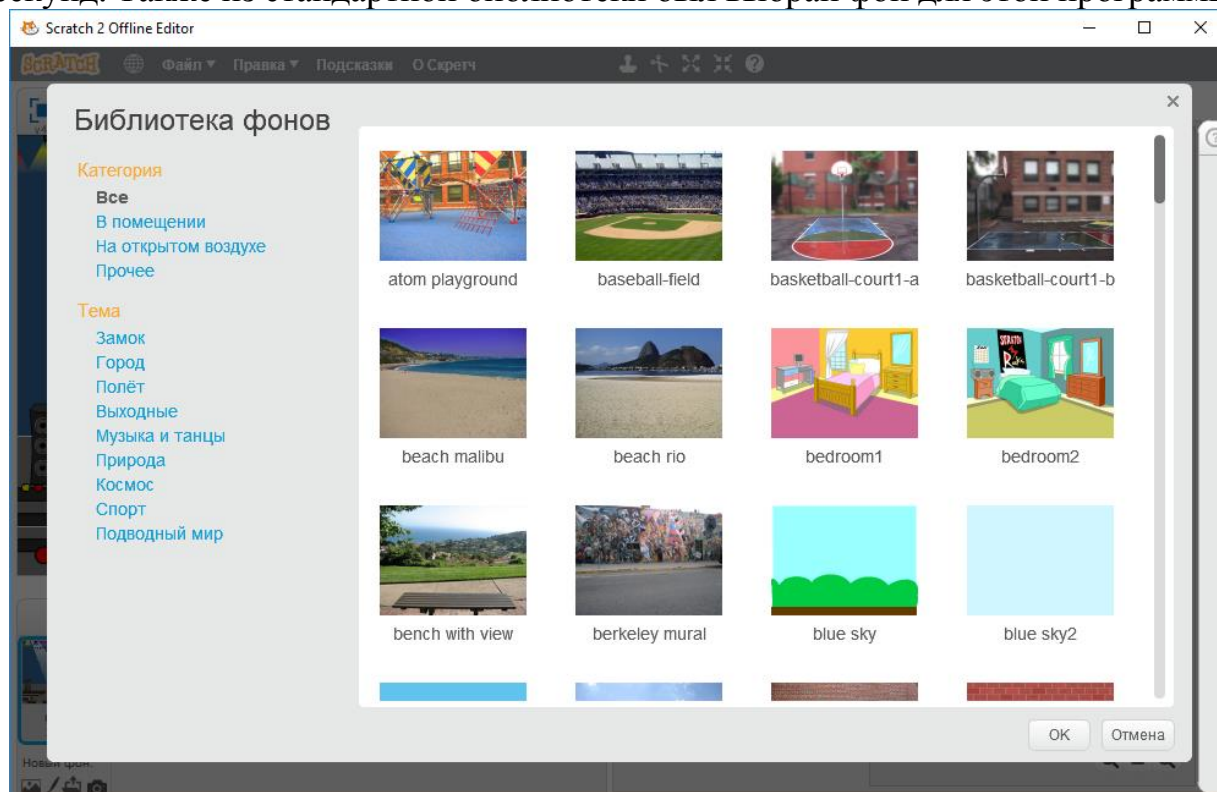


Рис. 4. Библиотека фонов Scratch

Однако это был лишь один из возможных вариантов решения задачи. Учащиеся могли, например, увеличить количество танцоров, изменить игровую музыку, загрузить собственные фоны в программу.

Существует положительный опыт применения Scratch. Также есть достаточно большое количество научных работ и статей, где Scratch успешно применялся во время занятий информатики или на дополнительных занятиях, связанных с программированием. Все исследователи отмечали, что дети крайне заинтересованы программированием в данной среде. Но также подмечали, что Scratch можно использовать не только для преподавания программирования. В частности, Scratch возможно использовать при изучении графики [9] и в различных научно-исследовательских проектах [10].

Подводя итог, можно сказать, что язык программирования Scratch является средой программирования, которая отлично подойдет для преподавания программирования в рамках пропедевтического курса информатики, т. к.:

- имеет красочный дизайн;
- среда проста в усвоении;
- позволяет развивать творческие способности учеников;

- поддерживает основные парадигмы программирования.

ЛИТЕРАТУРА:

1. Мухина В. С. Возрастная психология. Феноменология развития. 10 изд. М.: «Академия», 2006.
2. Шаповаленко И. В. Возрастная психология (психология развития и возрастная психология). М.: «Гардарики», 2005.
3. Scratch // Progopedia. URL: <https://goo.gl/9Z4Ehj> (дата обращения: 05.04.2018).
4. Алгоритмизация и программирование. URL: <https://goo.gl/9VTiSM> (дата обращения: 05.04.2018).
5. Исполнители // kpolyakov.spb.ru. URL: goo.gl/pGSL6Z (дата обращения: 12.04.2018).
6. Исполнители алгоритмов // sgl.s.ru. URL: <https://goo.gl/spfwwB> (дата обращения: 12.04.2018).
7. Твое первое руководство Scratch // Scratch.mit. URL: <https://goo.gl/kByz7c> (дата обращения: 10.04.2018).
8. Скретч (язык программирования) // Википедия. URL: goo.gl/J7kcNc (дата обращения: 10.04.2018).
9. Сорокина Т. Е. От stem к steam-образованию через программную среду Scratch // Современные информационные технологии и ИТ-образование. 2015.
10. Нажимова Ю. В. Обучение программированию младших школьников в рамках системы дополнительного образования // Интерактивная наука. 2017.