

1) Решением проблемы с разными версиями ОС и отсутствия смартфонов у школьников будет использование планшетов с предустановленными приложениями для учебников.

2) Сложность реализации сетевых функций можно решить с помощью привлечения квалифицированных людей.

Таким образом, использование мобильного дополнения к учебнику информатике позволит учителю облегчить проверку домашнего задания, посредством просмотра информации из личного кабинета. Школьникам такой подход использования своего смартфона в качестве инструмента учебной деятельности будет достаточно интересен.

Библиографический список

1. Мухина С.А. Нетрадиционные педагогические технологии в обучении/ Ростов н/Д: Издательство «Феникс», 2004.- 384 с.
2. Воронкова О.А. Информационные технологии в образовании. Интерактивные методы/ Ростов н/Д: Издательство «Феникс», 2010- 320с.
3. Боброва И.И. Информационные технологии в образовании. Практический курс/ Издательство: «Флинта», 2014.-196 с.
4. Войтович, И. К. Специфика создания электронных образовательных курсов // Вестник ТГПУ. – 2015. – №1 (154) – С. 138-143.
5. Белобородова, Т.Г. Использование информационно-коммуникационных технологий при организации самостоятельной работы студентов // Наука и образование: новое время (сетевое издание). – 2014. – №1.
6. Сиденко А.В. Игровой подход в обучении / Народное образование, 2000г., с. 134 - 137.
7. Обучение разработке мобильных приложений для Android [Электронный ресурс]// URL: <http://startandroid.ru/ru/>
8. Колисниченко Д. С. Программирование для Android. Самоучитель / СПб.: Санкт-Петербург, 2011. - 735 с.: ил.
9. Босова Л.Л. Информатика и ИКТ: учебник для 8 класса / 2-е изд. – М.: БИНОМ. Лаборатория знаний, 2014. – 160 с.

Апольских Е.И., старший преподаватель кафедры теоретических основ информатики

Жукова А. И., студент института физико-математического образования

Алтайский государственный педагогический университет
г. Барнаул

ВОЗМОЖНОСТИ ИЗУЧЕНИЯ ОСНОВ SCRATCH-ПРОГРАММИРОВАНИЯ В СРЕДНЕЙ ШКОЛЕ НА ПРОПЕДЕВТИЧЕСКОМ УРОВНЕ

Раннее введение основ программирования в школе способствует формированию у школьников алгоритмического мышления, что в современном обществе, прочно связанном с компьютерами и программируемой техникой, позволяет быстро адаптироваться к новым обстоятельствам, а значит быть востребованным в профессиональной деятельности. Алгоритмическое мышление формирует у школьника массу положительных качеств, таких как последовательность, внимательность, умение ставить перед собой цель и находить пути её достижения. Изучение программирования должно быть доступным, от простого к сложному, и состоять из задач, посильных для школьника, иначе исчезнет всякое желание действовать и разбираться с материалом. Все это можно начать реализовывать на пропедевтическом уровне.

В Современном толковом словаре Ефремовой Т.Ф. «ПРОПЕДЕВТИКА (от греч . προαίδεο - предварительно обучаю) - введение в какую-либо науку, предварительный

вводный курс, систематически изложенный в сжатой и элементарной форме».[1] Можно сказать, что пропедевтика является упрощённым обучением, «предобучением».

Существует достаточно много языков программирования, которые изначально создавались для изучения их детьми, такие как Лого, КуМир и другие. Основная цель этих языков заключалась в том, чтобы сформировать алгоритмическое мышление у детей, но в дальнейшем они не находили практического применения, приходилось переучиваться для того, чтобы иметь возможность работать с более «серьезными» языками.

Scratch - это полнофункциональный, но адаптированный под детское восприятие, язык. Он рассчитан на школьников начальных классов и средней школы, из чего следует возможность использования в качестве первого языка программирования. Если рассматривать его в качестве языка для изучения программирования на пропедевтическом уровне, то можно отметить его следующую особенность: при всей своей простоте и доступности - это полноценный язык, позволяющий решать достаточно сложные задачи. Ребёнок может создавать замысловатые программы, не мучаясь с запоминанием сложных конструкций или правописанием команд, уделяя больше времени возможности использовать свою. Также, отличительной чертой Scratch-программирования можно назвать лёгкость понимания: не обязательно знание основ, таких как виды алгоритмов, основные алгоритмические конструкции, типы данных. В тоже время Scratch, может быть достойным соперником более сложным языкам программирования, таким как Pascal, Java, СИ. Возможности Scratch довольно обширны: создание подобия презентаций, анимации, мультфильмы и игры.

Scratch можно описать как что-то среднее между Лого и конструктором Лего. По своим функциональным возможностям он похож на Лого, а по сбору фрагментов в алгоритм – на Лего, где каждый кубик имеет свои параметры и для того чтобы соединились два элемента, они должны быть подходящими друг другу.

Возможность заинтересовать школьников позволит в дальнейшем продолжить обучение в этом направлении и выбрать профессию, связанную с программированием. Рассмотрим языки программирования, изучаемые в школе, и сделаем сравнительный анализ по отношению к Scratch.

	Pascal ABC	Turbo Pascal	Python	КуМир	СИ	Java	Scratch
Программа создана специально в обучающих целях	+	+		+			+
Существует русскоязычная версия программы				+			+
Команды записываются на русском языке				+			+
Объектно-ориентированное программирование	+	+	+		+	+	+
Возможность работы без							+

знания основных алгоритмическ их конструкций и синтаксиса языка							
Программиров ание в игровой форме							+

Таблица 1. Сравнительный анализ различных языков программирования

Проанализировав таблицу, можно говорить о том, что Scratch является более подходящим для начального изучения программирования, так как он прост в освоении, русскоязычен, понятен, не имеет высокой планки ограничений по входным знаниям и, что немаловажно для обучения школьников, программирование реализуется в игровой форме. Кроме положительных характеристик, отмеченных в таблице, также можно выделить некоторое влияние Scratch-программирования на общее развитие школьника:

- Развивает творческое мышление. Свобода фантазии при создании алгоритмов для решения известных задач и проблем. Совершенствование навыков дизайна и рисования.
- Учит работать в команде. Scratch-онлайн кроме программирования, позволяет общаться всем, кто задействован в процессе, и вместе работать над проектом, а также выкладывать свои проекты в облачное хранилище.
- Учит мыслить критически, анализировать ситуацию, видеть общую картину. Работая в Scratch, дети учатся рассуждать и критически мыслить. Рассматривать взаимодействие нескольких алгоритмов, согласованность действий спрайтов (объектов).
- Развивает медиаграмотность. При создании проектов, школьники задействуют разные виды представления информации: текстовую, графическую, звуковую. Учатся использовать технологии, помогающие в создании проекта.
- Развивает коммуникативные навыки. Без умения общаться невозможно создание масштабных проектов, требующих работы многих людей, поэтому при взаимодействии дети учатся договариваться, слушать других и выражать свои мысли.
- Учит решать проблемы. Работа над проектами в Scratch требует умения ставить задачи, определять исходные данные и необходимые результаты, определять последовательность шагов для достижения цели.

Для более конкретного понимания того, что собой представляет Scratch, рассмотрим интерфейс программы (Рис.1).

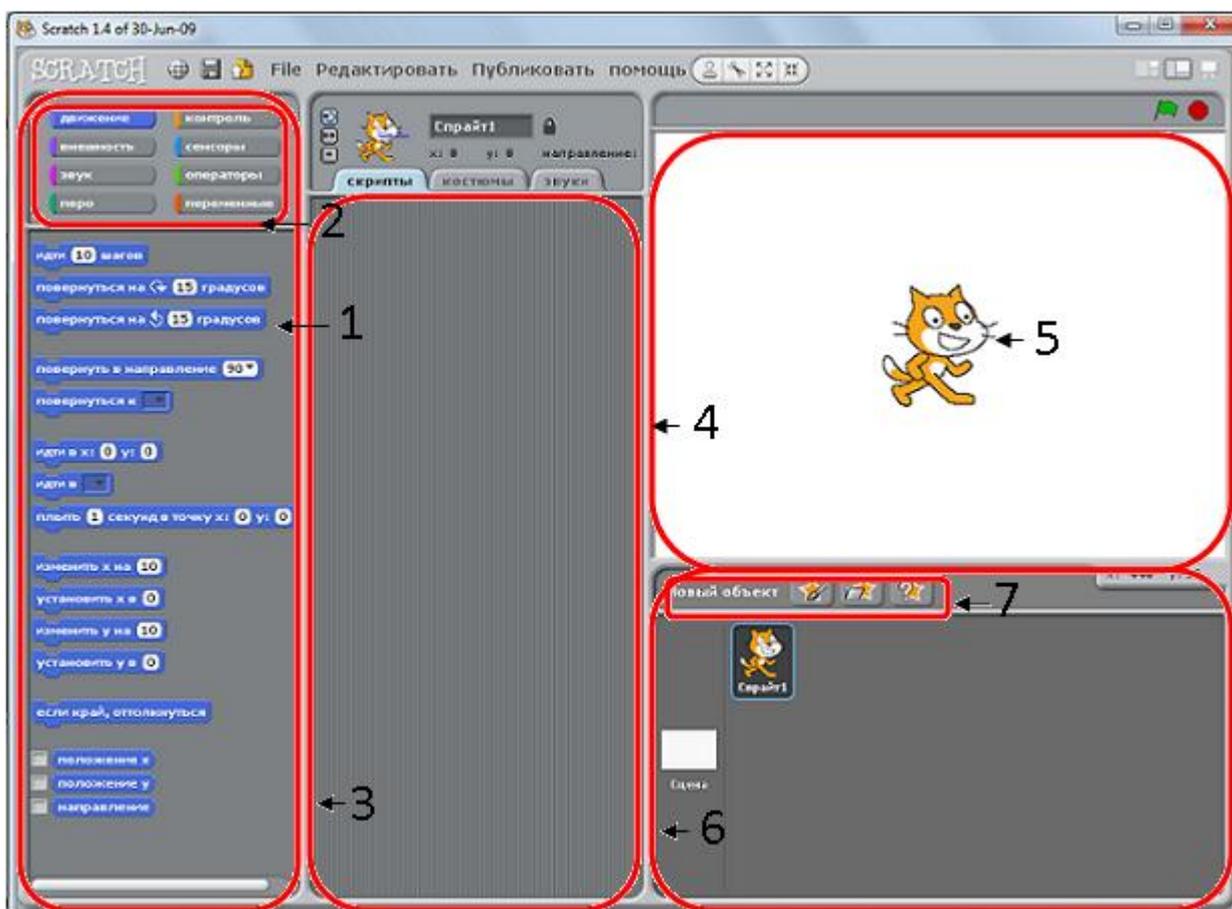


Рис. 1. Окно программы

На рисунке 1 можно увидеть, что блоки команд расположены в отдельной части экрана (1) и разбиты на семь категорий по своей функциональности (2), каждая из которых обозначена определённым цветом. Условно блоки можно разделить на две категории – те, в которые нельзя записать дополнительную информацию и те, в которых это сделать можно. Кроме того, есть область, в которой эти блоки составляются (3) (их необходимо переносить из своей области и «цеплять» в нужном порядке), причём в одном поле можно составлять неограниченное число программ, которые будут работать одновременно или последовательно, начиная с момента выполнения определённой команды. Также, на экране располагается сцена (4), а в ней объекты – спрайты (5), с которыми происходят действия, задаваемые программой. Кроме того, каждому спрайту можно создать свой костюм, для этого есть своя программа, по своим возможностям похожая на Paint. Программа может писаться для каждого спрайта в отдельности, либо для всех, кто находится на сцене, которых также может быть несколько. Выбор сцены можно производить в специальной части окна программы (6), там же можно выбрать добавить сцену или спрайт из галереи объектов (7), которая загружается при установке программы, кроме того можно создать своего героя или сцену и сохранить на диске, чтобы потом вновь вернуться к ним. При желании во время выполнения программы могут проигрываться звуковые файлы, некоторое количество которых также представлено в галерее.

Составим в Scratch программу, которая покажет, что программирование – это не сложно. Пусть спрайт ходит по сцене и спрашивает как дела.

В качестве спрайта будем использовать стандартного кота, у которого есть два костюма (Рис.2 и Рис.3).

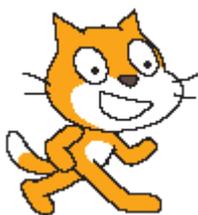


Рис.2. Костюм 1



Рис.3. Костюм 2

Сначала нам необходим блок, с которого будет начинаться программа, он находится в категории «управление» и может быть представлен одним из трёх, изображённых на рис.4.



Рис.4. Блок начала программы

Пусть нами будет выбран первый вариант. Затем, нам необходим цикл, в котором будут происходить все действия, ведь кот должен произвести множество шагов, пока программист не выключит программу. В Scratch множество различных циклов, кроме итерационного и различных условных, также есть бесконечный цикл, которым мы воспользуемся (Рис.5).



Рис.5. Бесконечный цикл

Наполним цикл действиями: спрайт должен двигаться, это будет происходить с помощью специального блока, в котором программистом задаётся количество шагов (Рис.6) (кроме этого, с помощью другого блока, можно изменять значение координат в плоскости сцены).



Рис.6. Блок передвижения

При каждом движении кот будет менять костюм с помощью блока, изображённого на Рис.7, что сделает передвижение более похожим на ходьбу.



Рис.7. Смена костюма

Кроме того, если мы хотим изменить скорость передвижения спрайта с быстрой на среднюю, нам потребуется блок, в котором мы можем установить значение ожидания 0.1с (Рис.8).



Рис.8. Время ожидания

Внимательный читатель задастся вопросом, что будет, если спрайт дойдёт до края сцены, на этот случай мы используем блок «если край, оттолкнуться» (Рис.9) вместе с тем,

выберем свойство спрайта «поворачивать только влево - вправо» (Рис.10), что не даст ему передвигаться в неестественной позе.

если край, оттолкнуться

Рис.9. Перемещение от края



Рис.10. Движение влево - вправо

Последним элементом, который потребуется, будет блок, с помощью которого спрайт будет «говорить», в который можно записать текст (Рис.11). Также есть блоки, которые предусматривают ожидание письменного ответа, но в данной программе мы их использовать не будем.

сказать Как дела?

Рис.11. Блок общения

Определим последовательность блоков: первым будет блок начала программы, затем цикл, в котором будут заключены все остальные блоки. Для того, чтобы спрайт не вышел за край сцены, первым поставим блок перемещения от края, затем блок передвижения, смена костюма, в этом случае после каждого перемещения спрайт будет как бы «менять ногу», затем блок ожидания, позволяющий спрайту идти не столь быстро, и в конце – блок общения, который будет выполняться каждый «прогон» цикла, таким образом, спрайт будет постоянно спрашивать «как дела?».

Теперь, когда выбраны все необходимые блоки и их последовательность, соберём программу (Рис.12).

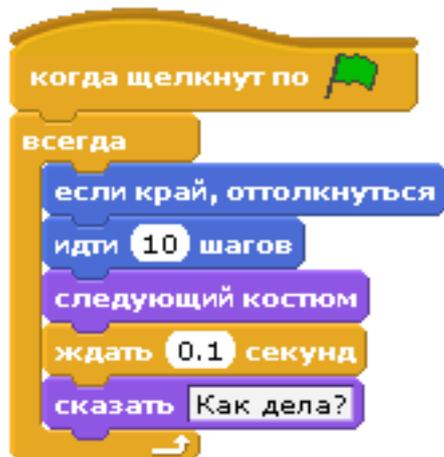


Рис.12. Полная программа

В начальный момент времени спрайт будет находиться в костюме 1 без вопроса «как дела?» (Рис.13).

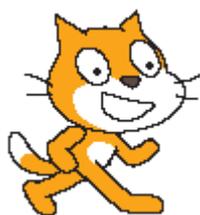


Рис.13. Начальное положение спрайта

После первого раза «прогона» блоков цикла, спрайт будет выглядеть иначе (Рис.14) и располагаться правее относительно своего предыдущего местоположения на сцене.

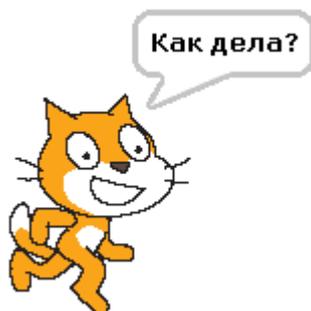


Рис.14. Первый «прогон» цикла

После того, как спрайт оттолкнётся от края сцены, он зеркально отобразится и продолжит свой путь (Рис.15).

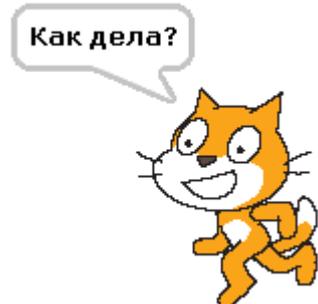


Рис.15. Поворот по достижению края

Таким образом, мы создали простую программу передвижения спрайта.

Из всего рассмотренного, можно сделать выводы:

- Scratch – один из немногих языков программирования, полностью реализованный на русском языке.
- Scratch возможно внедрять с возраста, более раннего, нежели тот, с которого вводится программирование у большинства авторов учебников, т.е он подходит для любой возрастной группы школьников.
- Нет необходимого порога для начала изучения Scratch, кроме умения читать.
- Программы реализуются в объектно-ориентированном режиме и игровой форме.
- Возможно составление сложных алгоритмов, с помощью которых будут созданы игра или фильм.
- Создание проектов функционально посилено для школьника.
- С помощью разработки игровых программ, у школьника формируется алгоритмическое мышление.

Можно сказать, что Scratch является языком программирования, на который стоит обратить внимание каждому учителю информатики.

Библиографический список

Ефремова, Т.Ф. Современный толковый словарь [Текст] / Т. Ф. Ефремова – Москва: Большая Советская Энциклопедия. Т.2 2