

*Лавринова Ирина Игоревна
учитель математики и информатики
ОГБОУ “Смоленская специальная
(коррекционная) общеобразовательная
школа I и II видов”
Центр дистанционного образования
г. Смоленск*

Изучение визуальных сред программирования в ЦДО

Работая учителем информатики в смоленском Центре дистанционного образования для детей-инвалидов, я имею дело с различным контингентом обучающихся. Ученики с сохранным интеллектом должны изучать темы «Алгоритмизация и программирование» по обычным общеобразовательным программам.

Зачастую данный раздел информатики дается учащимся с трудом. Одновременно приходится разбираться со всеми тонкостями алгоритмизации и учить формы представления алгоритмов, вникать в особенности условных операторов и циклов, обучающимся не всегда понятно, как это реализуется на практике.

Как же привить обучающимся навыки алгоритмизации и программирования? Учитель, конечно, познакомит их с алгоритмическими структурами, командами, операторами, но нужно научить детей мыслить особым образом, понимать суть команд. Следовательно преподавание должно вестись максимально наглядно, а ученики должны видеть результат своих действий.

Кроме того, сейчас предлагается вводить в школьный курс такой предмет, как Робототехника, но я считаю, что успешное освоение его, без основ алгоритмизации будет довольно затруднительно, если вообще возможно. Поэтому мне кажется просто необходимым изучение визуальных сред программирования, где при помощи простого Drag&Drop можно создавать программы.

Графическая среда программирования как раз позволяет уйти от синтаксической рутины: весь синтаксис "запаян" в блоки, из которых составляется программа. Блоки включают в себя ключевые слова и оформляют управляющие конструкции.

Пример таких сред – Scratch, Snap!, Blockly.

Думаю, многие слышали о **Scratch** [1] - это визуальная объектно-ориентированная среда программирования для обучения школьников младших и средних классов. Идея программы возникла в 2003 году, вышла в релиз в 2006 году в качестве инструмента, который делает программирование простым и понятным. Цель создания программы — позволить обучающимся без опыта программирования изучить основные принципы императивного, объектно-ориентированного программирования.

При изучении Scratch учащиеся:

1. увидят практическое назначение алгоритмов и программ;

2. научатся самостоятельно принимать решения;

3. научатся составлять алгоритмы.

4. синтаксис Scratch напоминает Паскаль, но запомнить его гораздо проще. Следовательно, в дальнейшем изучать Паскаль будет проще;

5. изучат функциональность работы циклов и операторы.

Изучение Scratch может серьезно помочь обучающимся в освоении тем алгоритмизации и программирования.

Осваивать Scratch можно уже с обучающимися начальной школы на простых упражнениях, знакомясь постепенно с интерфейсом, создавая анимации и мультфильмы и так далее, постепенно усложняя задания. Причем выполнение заданий будет происходить не пассивно, путем зубрежки и механических упражнений, а осознано – глубоко и основательно.

Интерфейс Scratch максимально прост, поскольку рассчитан на использование детьми, и содержит несколько графических закладок: скрипты, костюмы, звуки. В этих закладках выбираются нужные команды, действия, отображение, звуковое оформление и другие параметры. Результат готового проекта показывается в общем окне.

Хотя прошло довольно много времени с первого релиза программы, учебные и методические пособия на русском языке [2] только начали появляться. Это не мешает ученикам играть, изучать, делиться и исследовать код программ в сообществе Scratch. [1]

Некоторое время назад была разработана альтернатива Scratch — веб-система **Snap!** [3] Интерфейс программы близок к Scratch, но работает она в браузере (ее можно установить на сервер школы).

Snap! и Scratch требуют свободы, творческих идей, полета фантазии, но школа зачастую требует таких упражнений, которые можно конкретно оценивать. Тогда на помощь приходит язык Blockly.

Blockly [4] — визуальный язык программирования с веб-интерфейсом. Создан при поддержке Google в 2012 году. Главное его отличие состоит в том, что он позволяет создавать учебные задания, проходя которые, ученик постепенно осваивает какую-либо тему.

В чем состоит новизна:

1. сильный акцент на визуальной составляющей
2. возможность компиляции на более традиционные языки.

Например, отдельные части программы могут очень быстро создаваться на Google Blockly, а затем компилироваться, например, в Python. Исходный код такой программы может быть доработан уже на Python, или даже вставлен в уже готовый проект в целях его расширения.

Google предоставляет пользователям не очень большой, но достаточно содержательный стартовый набор приложений, реализованных на базе Blockly. В этот набор на момент написания статьи входят следующие приложения.

Головоломка – для знакомства с интерфейсом Blockly. Среда программирования Blockly устроена достаточно просто: пользователь собирает программу из блоков и вставок. Этот процесс напоминает сбор головоломки из фигурных частей. Блоки – аналог управляющих конструкций, процедур и операторов языка программирования. Вставки служат параметрами процедур и аргументами функций, а также представляют значения, возвращаемые функциями. Естественно, детям во время знакомства обо всех этих абстракциях говорить необязательно, достаточно познакомить их с приложением Головоломка.

Лабиринт – служит для знакомства с азами программирования: последовательным исполнением команд, циклами и условными операторами. Приложение разбито на 10 уровней по уровню возрастания сложности. Экран разбит на 3 части:

- левая, где программа выполняется,
- средняя, содержащая "кассу" блоков, из которых можно составить программу,
- правая - рабочая область, в которой программа собирается.

Черепашка – известная рисующая рептилия. Приложение Черепашка содержит гораздо более развитые управляющие конструкции. Так, в группе Циклы присутствуют 4 вида циклов и оператор выхода из цикла. Используя блоки из группы Процедуры можно создать процедуры и функции.

В группе Логические представлен конструктор для условного оператора с произвольным числом "иначе если".

В группе Математика – ряд математических функций.

Птица, Пруд, Пруд JS — на данный момент находятся в стадии разработки и частично не русифицированы.

Фильм — позволяет создавать простые анимации и фильмы.

Один из главных разработчиков данного языка Нил Фрейзер [5] так описывает главную цель его создания: «Эта популярная концепция прошла очень длинный путь. Это попытка подняться на ещё более высокий уровень абстракции с каждым шагом вперед, чтобы лишить программирование закрытости, сделать его более естественным и доступным для как можно более широкого круга лиц».

Ввиду специфики организации образовательного процесса дистанционным урокам зачастую не хватает наглядности, которая особенно важна для детей с ОВЗ. На мой взгляд, использование визуальных сред могло бы добавить интереса к урокам информатики и показать, что программирование может быть интересным и востребованным в школе.

Источники:

1. Официальный сайт сообщества Scratch. URL: https://scratch.mit.edu/scratch_1.4/ (дата обращения: 19.01.15)
2. Ю. Пашковская. Творческие задания в среде Scratch. 5-6 класс. Рабочая тетрадь. Издательство: Бинوم. Лаборатория знаний. 2014 г.
3. Официальный сайт Snap! URL: <http://snap.berkeley.edu/snapsource/snap.html> (дата обращения: 19.01.15)
4. Официальный сайт Blockly URL: <https://blockly-games.appspot.com/> (дата обращения: 19.01.15)
5. Neil Fraser. URL: <https://neil.fraser.name/> (дата обращения: 19.01.15)