

## Применение онлайн-сред визуального программирования на уроках информатики как образовательный мейнстрим

Величко А.И., учитель  
математики и информатики  
МКОУ «Маминская СОШ»

«Каждый в этой стране должен научиться программировать, потому что это учит людей думать» (Стив Джобс)

В образовании сегодня часто можно услышать термин «мейнстрим», означающий основное направление и главную тенденцию, обусловленную требованием времени. Каков же сегодняшний мейнстрим на уроках информатики в общеобразовательной школе, с чем связан и как долго продлится?

Дело в том, что сегодняшние ученики являются поколением Z (рожденные в 2000-2020 годы). Это первое поколение, родившееся в эпоху Интернета, так называемые цифровые дети. Они с большим трудом представляют себе мир без электронных средств, умело используют приложения. То, что предыдущие поколения называли «технологиями будущего», для поколения Z уже настоящее. Современные дети привыкли, что все должно быть ярким, зрелищным и сенсорным. Компьютеры они предпочитают книгам и во всём стремятся к немедленным результатам. Интернет формирует их взгляды. Мотивация обучающихся сейчас напрямую зависит от того, насколько хорошо они понимают, как и где смогут применить полученные знания.

Но тогда, как же работать педагогам с поколением Z? Ответы на этот вопрос постоянно держат учителей в состоянии творческого поиска и влекут существенные изменения в их работе. Наглядность - главное условие. Выход в Интернет в школе делает равными возможности всех российских школьников и учителей по доступу к ЦОР и изменяет процесс обучения.

Разделы информатики, касающиеся алгоритмизации и программирования, исторически с самого начала входили в школьную информатику и несли очень важную образовательную нагрузку. Алгоритмическое мышление является значительной составляющей интеллектуального развития человека.

Основные термины по разделу «Алгоритмизация»: Алгоритм управления. Алгоритмический язык (учебный). Блок-схема. Вспомогательный алгоритм. ГРИС. Дискретность алгоритма. Зацикливание. Исполнитель алгоритма управления. Команда ветвления. Команда цикла. Конечность алгоритма. Подпрограмма. Понятность алгоритма. Последовательная детализация алгоритма. Программа. Программное управление. Прямая связь. Система команд исполнителя (СКИ). Среда исполнителя. Структура алгоритма управления. Точность алгоритма.

Согласно ФГОС ООО по информатике выпускник 8 класса должен знать:

- что такое алгоритм управления; какова роль алгоритма в системах управления;
- в чем состоят основные свойства алгоритма;

- способы записи алгоритмов: блок-схемы, учебный алгоритмический язык;
- основные алгоритмические конструкции: следование, ветвление, цикл; структуры алгоритмов;
- назначение вспомогательных алгоритмов; технологии построения сложных алгоритмов: метод последовательной детализации и сборочный (библиотечный) метод.

Должен уметь:

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
- выполнить трассировку алгоритма для известного исполнителя;
- составлять линейные, ветвящиеся и циклические алгоритмы управления одним из учебных исполнителей;
- выделять подзадачи; определять и использовать вспомогательные алгоритмы.

И, согласно сложившейся методике, нецелесообразно сразу начинать обучение школьников программированию на языках высокого уровня.

Идея визуального (графического) программирования, как показывает практика, хорошо применима при обучении школьников, поскольку этот способ создания программ не требует знания всех тонкостей синтаксиса языка программирования, а заключается в манипулировании графическими объектами. В этом случае программа строится (собирается) из готовых графических блоков, внутри которых уже находится все необходимое для корректной работы программы. Главное - правильно воспользоваться имеющимся «строительным материалом». Немаловажное значение имеет возможность создания собственных проектов на платформе.

В данной статье будут рассмотрены два доступных образовательных ресурса-лидера.

**1. CodeStudio** (<https://studio.code.org/>) – содержит пять 20-часовых курсов по основам алгоритмизации; коллекцию различных заданий для Часа кода; галерею ранее созданных учениками рисунков и игр. Каждый курс (всего их пять) учитывает возраст и возможности обучающихся, состоит из 18-21 этапа, на каждом этапе предлагается от двух до 15 заданий. Этап - это отдельная игра с постепенно усложняющимися заданиями (Рис.1).

Все программы в CodeStudio создаются из разноцветных блоков, обозначающих различные действия исполнителя. При создании программы блоки последовательно присоединяются друг к другу. В отдельных заданиях часть нужного кода уже составлена и ее нужно только дополнить несколькими блоками - в данной среде ученики учатся не только составлять собственную программу, но и понимать чужую. Почти на всех этапах последнее задание - творческого характера, в нем ученик должен составить произвольный алгоритм на основе изученного в предыдущих заданиях. После выполнения таких заданий ученики могут поделиться ссылкой в социальной сети на получившиеся у них приложение, игру, анимацию или рисунок. Также на основе любой работы можно сделать свою: скопировать понравившуюся работу и внести в нее свои изменения.

Благодаря возможности создания личного кабинета обучающиеся могут продолжить работу дома над завершением этапа, все достижения сохраняются и высвечивают-

ся зеленым (если алгоритм составлен оптимально) или желтым (не оптимально) цветом (Рис.2). По окончании курса обучающийся получает именной сертификат.



Название урока	Прогресс
1. Урок без компьютера: Танграмы	Урок без компьютера
2. Лабиринт и пчела	1 2 3 4 5 6 7 8 9
3. Художник	1 2 3 4 5 6 7 8 9 10 11 12 13 14
4. Урок без компьютера: Конверт...	Урок без компьютера
5. Урок без компьютера: игра «П...	Урок без компьютера
6. Художник: Переменные	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
7. Лаборатория игр: Переменные	1 2 3 4 5 6 7 8
8. Урок без компьютера: Игры с ...	Урок без компьютера
9. Пчела: Циклы с параметром	1 2 3 4 5 6 7 8 9 10 11
10. Художник: Циклы с параметр...	1 2 3 4 5 6 7 8 9 10 11 12

Рисунок 1

Этот 20-часовой курс охватывает основы информатики и программирования в курсах 2-4. Курс предназначен для учеников в возрасте 10-18 лет. Проверьте свое знание курсов 2-4 для более глубокого освоения!

Название урока	Прогресс
1. Введение в информатику	Урок без компьютера
2. Лабиринт	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
3. Вычислительное мышление	Урок без компьютера
4. Программирование на бумаге ...	Урок без компьютера
5. Художник	1 2 3 4 5 6 7 8 9 10
6. Алгоритмы	Урок без компьютера
7. Художник 2	1 2 3 4 5 6 7 8 9 10 11

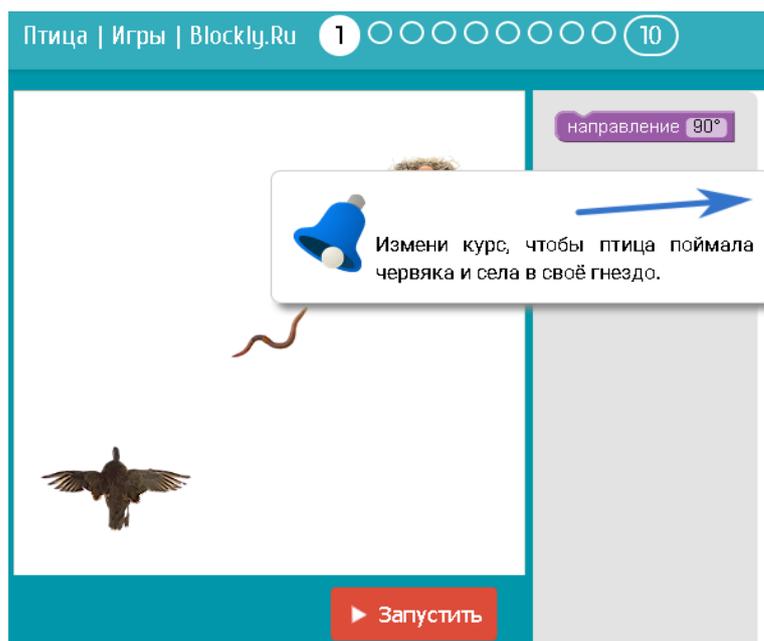
Рисунок 2

## 2.Blockly (<http://blockly.ru/games.html>)

Blockly-игры по структуре и организации представляют собой аналог рассмотренной выше CodeStudio. Blockly - это серия игр, которые являются проектом Google с открытым исходным кодом и созданы для привлечения будущих программистов.

Состоит из следующих игр-тренажеров: «Головоломка» (знакомит учеников с интерфейсом программы Blockly), «Лабиринт» (10 постепенно усложняющихся заданий для выбранного исполнителя на алгоритмические конструкции), «Птица» (Рис.3) (набор заданий с препятствиями, последние задания в наборе достаточно сложны, для их правильного выполнения нужно применять составные условия), «Черепашка» (появляется много новых блоков: математические операции, переменные, работа с подпрограммами, логические блоки, которые ученикам предлагается изучить самостоятельно в процессе создания своего проекта), «Фильм» ( набор заданий по созданию анимации, где вводится новая переменная «время»), «Пруд» (комплексная игра).

Но в отличие от предыдущей среды Blockly работает по принципу «здесь и сейчас», так как не имеет возможности создания личного кабинета и сохранения результатов.



**Рисунок 3**

Таким образом, возможности современных ЦОР и средств ИКТ позволят учителю информатики реализовать на уроке сразу несколько педагогических технологий согласно ФГОС ООО: игровая, разноуровневой дифференциации, самостоятельной работы (учитель выступает в роли тьютора), индивидуализации обучения, творческого мышления, проектная, опережающего и проблемного обучения. Не маловажен для учителя учет темпа работы, автоматизированный контроль, высокая мотивация, выявление одаренных ребят. Реализуя педагогический принцип «от простого к сложному» игры - среды специально разработаны для самостоятельного обучения в собственном темпе для ребят, которые не имели предварительного опыта программирования. И этот образовательный мейнстрим, без сомнения, имеет будущее.