

Общество с ограниченной ответственностью

«Проектная школа - Самара»

443015 г. Самара, ул. Главная, д.3, оф.20, тел. 8(927)777-09-67

E-mail: mironov@100lines.ru



УТВЕРЖДАЮ

Директор

ООО «Проектная школа — Самара»

Е.Н.Миронов

15 декабря 2021 г.

**ДОПОЛНИТЕЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ
(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА
«ДЕТСКОЕ ПРОГРАММИРОВАНИЕ НА СКРЕТЧ»**

Возраст детей: от 7 лет

Срок обучения: 72 академических часа

Автор-составитель:

Мальцев С.А., педагог ДО

г.о. Самара, 2021

Введение

Программа «Детское программирование на Скретч» разработана в соответствии с требованиями к дополнительным общеобразовательным общеразвивающим программам технической направленности и направлена на формирование у детей младшего школьного возраста основ алгоритмического мышления и навыков визуального программирования.

Визуальная среда программирования Scratch, созданная в MIT Media Lab, представляет собой уникальный образовательный инструмент, позволяющий детям создавать интерактивные истории, анимации, игры и музыкальные композиции без необходимости изучения сложного синтаксиса текстовых языков программирования. Блочное программирование делает процесс создания программ интуитивно понятным и доступным для детей 7-11 лет.

Актуальность программы обусловлена необходимостью раннего формирования цифровой грамотности, развития алгоритмического мышления и творческих способностей в условиях стремительной цифровизации общества. Программа отвечает социальному заказу родителей и образовательных учреждений на формирование IT-компетенций у подрастающего поколения.

1. Пояснительная записка

Актуальность программы. В условиях стремительной цифровизации общества и перехода к экономике знаний, навыки работы с информацией и понимание принципов работы алгоритмов становятся базовыми компетенциями XXI века. Среда визуального программирования Scratch, разработанная в MIT Media Lab, является идеальным инструментом для первичного погружения детей в мир IT. Она позволяет снизить порог входа в программирование, избавляя ребенка от необходимости написания синтаксического кода, и фокусирует его внимание на логике, алгоритмическом мышлении и творческой составляющей. Программа актуальна, так как формирует не только hard skills (навыки кодинга), но и критически важные soft skills (креативность, решение проблем, командная работа).

Отличительные особенности программы. Программа строится на принципах геймификации и проектного обучения. Вместо сухого изучения синтаксиса, дети с первых занятий создают интерактивные истории, анимацию и игры. Отличительной чертой является раннее внедрение элементов геймдизайна и UX/UI (создание меню, навигация, баланс игры), что переводит ребенка из разряда «потребителя контента» в разряд «создателя контента».

Учебные дисциплины:

Основы алгоритмизации и логики (понятие алгоритма, линейные, разветвляющиеся и циклические структуры). Компьютерная графика и анимация (работа с растровыми изображениями, покадровая анимация, создание костюмов). Основы геймдизайна (механики игр, баланс, проектирование уровней, пользовательский опыт).

Практические дисциплины:

Визуальное конструирование скриптов (сборка блоков, отладка кода). Мультимедийное проектирование (создание интерактивных сцен, работа со звуком). Проектная деятельность (разработка, тестирование и защита собственных IT-проектов).

Цель образовательной программы: формирование у обучающихся основ алгоритмического и логического мышления, развитие творческого потенциала и способностей к созданию собственных мультимедийных и игровых проектов в среде визуального программирования Scratch.

Задачи программы:

Обучающие:

- познакомить с интерфейсом среды Scratch 3.0 и базовыми понятиями объектно-ориентированного программирования (спрайт, сцена, скрипт);
- научить составлять линейные, циклические и разветвляющиеся алгоритмы;
- сформировать навыки работы с координатной плоскостью, переменными, списками и событиями;
- обучить основам отладки кода и оптимизации скриптов.

Воспитывающие:

- воспитывать культуру работы за компьютером и правила цифровой гигиены;
- формировать уважение к интеллектуальной собственности и авторскому праву (умение корректно использовать чужие ассеты с указанием авторства или создание собственных);
- воспитывать настойчивость в достижении цели, умение доводить начатое дело до конца, толерантность при работе в команде.

Развивающие:

- развивать логическое, алгоритмическое и пространственное мышление;
- развивать творческое воображение, креативность и способность к нестандартному решению задач;

- развивать навыки самопрезентации, публичных выступлений и аргументации своей точки зрения.

2. Принципы, формы и методы обучения

Основными **принципами обучения** являются:

1. **Научность.** Этот принцип предопределяет сообщение учащимся только достоверных, проверенных практикой сведений. Даже в визуальной среде Scratch преподаются настоящие концепции программирования (переменные, циклы, условия, координаты), которые являются базой для любых текстовых языков.
2. **Доступность.** Предусматривает соответствие объема и глубины учебного материала уровню общего развития учащихся. Сложные концепции объясняются через жизненные аналогии и визуальные образы (блочное программирование).
3. **Связь теории с практикой.** Обязывает вести обучение так, чтобы учащиеся могли сознательно применять приобретенные ими знания на практике. Каждая изученная категория блоков немедленно применяется для создания мини-игры или анимации.
4. **Воспитательный характер обучения.** Процесс обучения является воспитывающим, учащийся не только приобретает знания, но и развивает усидчивость, умение преодолевать трудности при отладке кода, морально-духовные качества.
5. **Активность обучения.** В процессе обучения все действия должны быть обоснованы. Активность предполагает самостоятельность: ребенок не просто копирует код за педагогом, а понимает, почему блок работает именно так.
6. **Наглядность.** Объяснение техники программирования на конкретных готовых изделиях (играх, мультфильмах). Цветовая кодировка блоков в Scratch сама по себе является мощным наглядным пособием.
7. **Систематичность и последовательность.** Учебный материал дается по определенной системе: от простого движения к сложным условиям и переменным. От частного к общему.
8. **Индивидуальный подход в обучении.** Педагог исходит из индивидуальных особенностей учащегося (темп работы, уровень логического мышления, усидчивость) и, опираясь на сильные стороны, доводит его подготовленность до уровня общих требований с перспективой на творческое опережение.

Формы организации образовательного процесса: сочетание фронтально-групповой и индивидуальной форм организации учебной деятельности (групповое обсуждение идей, индивидуальное выполнение практических заданий по созданию проектов, коллективное тестирование и оценка полученных результатов).

Методы обучения:

1. Объяснительно-иллюстративный – предоставление информации (объяснения, рассказ, инструктаж, работа с технологическими картами, демонстрация готовых проектов).
2. Эвристический – метод творческой деятельности (создание собственных игровых механик, уникальных сюжетов).
3. Проблемный – постановка проблемы (например, «как сделать так, чтобы герой не проваливался сквозь пол?») и самостоятельный поиск ее решения через подбор блоков.
4. Программированный – набор операций, который необходимо выполнить в ходе практических работ по пошаговым алгоритмическим картам.
5. Репродуктивный – воспроизведение знаний и способов деятельности (сборка скриптов по образцу, упражнения по аналогу).
6. Поисковый – решение проблемных задач с помощью педагога и самостоятельный поиск оптимального алгоритма.

Главный метод, который используется при изучении программирования – это метод проектов. Основные этапы разработки проекта – игры/мультфильма:

1. Освоение теоретического базиса (изучение новой категории блоков), необходимого для дальнейшей продуктивной работы;
2. Определение идеи проекта (сюжет, жанр, механика);
3. Цель и задачи представляемого проекта (что должен уметь делать игрок);
4. Создание графики, звуков и написание кода (сборка скриптов);
5. Детальный разбор принципа работы полученного алгоритма;
6. Тестирование, устранение неисправностей (отладка/дебаггинг) и доработка проекта.

3. Механизм отслеживания результатов

Контроль освоения учебного материала учащимися проходит в три этапа:

1. Входной мониторинг сформированности базовой компьютерной грамотности и представлений об алгоритмах (устный опрос, мини-тест на знание устройства ПК).
2. Проведение промежуточных (текущих) контрольных тестов в вопросно-ответной форме и в виде защиты мини-проектов (например, защита игры «Лабиринт» после изучения сенсоров и условий).
3. Итоговый мониторинг сформированности компетенций учащихся; создание и публичная защита собственных итоговых проектов на основе пройденного материала.

В конце курса проводится анализ качества данной программы (содержания, методической подачи и организационных моментов) и, по необходимости, проводится коррекция программы для следующего набора.

4. Прогнозируемый результат

В конце обучения по данному курсу ученики должны знать:

- общенаучные и технические базовые термины информатики (алгоритм, программа, баг, отладка, интерфейс);
- элементную базу среды Scratch (спрайт, сцена, костюм, фон, скрипт);
- категории блоков и их назначение (Движение, Внешность, Звук, События, Управление, Сенсоры, Операторы, Переменные);
- основы декартовой системы координат (оси X и Y);
- порядок взаимодействия объектов между собой через события и сообщения;
- назначение и принцип работы циклов и условий (ветвлений);
- назначение и принцип работы переменных и списков;
- правила цифровой гигиены и техники безопасности при работе за компьютером.

В конце обучения по данному курсу ученики должны уметь:

- создавать интерактивные проекты и игры без написания синтаксического кода;
- управлять движением, внешностью и звуком объектов с помощью алгоритмов;
- пользоваться логическими операторами и условиями (Если... то... иначе);
- создавать и использовать переменные для подсчета очков, жизней и таймеров;
- использовать инструмент «Перо» для рисования геометрических фигур с помощью кода;
- применять клонирование для создания множества однотипных объектов (пули, враги);
- самостоятельно находить и исправлять логические ошибки в своем коде (отладка);
- презентовать свою работу, аргументировать принятые проектные решения.

5. Учебно-тематический план обучения

№ п/п	Тема	часы		
		всего	теория	практ.
1	Вводное занятие. Техника безопасности и цифровая гигиена. Интерфейс Scratch.	2	0,5	1,5
2	Тема 1. Сцена и спрайты. Управление объектами.	2	0,5	1,5
3	Тема 2. Движение. Алгоритмы и исполнители.	2	0,5	1,5
4	Тема 3. Координатная плоскость X и Y.	2	0,5	1,5
5	Тема 4. Повороты, направления и углы.	2	0,5	1,5

6	Тема 5. Внешность. Костюмы и покадровая анимация.	2	0,5	1,5
7	Тема 6. Эффекты внешности. Цвет, размер, прозрачность.	2	0,5	1,5
8	Тема 7. Работа со звуком. Запись и редактирование.	2	0,5	1,5
9	Тема 8. Фоны сцены. Смена декораций.	2	0,5	1,5
10	Тема 9. События. Запуск программ. Клавиатура.	2	0,5	1,5
11	Тема 10. События. Мышь и касания.	2	0,5	1,5
12	Тема 11. Циклы. Блок «Повтори N раз».	2	0,5	1,5
13	Тема 12. Бесконечные циклы. Блок «Повторять всегда».	2	0,5	1,5
14	Тема 13. Условия. Блок «Если... то».	2	0,5	1,5
15	Тема 14. Расширенные условия. «Если... то... иначе».	2	0,5	1,5
16	Тема 15. Сенсоры. Касания спрайтов и цветов.	2	0,5	1,5
17	Тема 16. Сенсоры. Расстояние до указателя мыши.	2	0,5	1,5
18	Тема 17. Мини-проект «Лабиринт» (комплексное применение).	2	0,2	1,8
19	Тема 18. Инструмент «Перо». Основы рисования кодом.	2	0,5	1,5
20	Тема 19. Геометрия в Scratch. Рисование фигур пером.	2	0,5	1,5
21	Тема 20. Переменные. Понятие, создание, типы.	2	0,5	1,5
22	Тема 21. Переменные. Таймеры, жизни и уровни.	2	0,5	1,5
23	Тема 22. Операторы. Математические и логические.	2	0,5	1,5
24	Тема 23. Случайные числа в играх.	2	0,5	1,5
25	Тема 24. Клонирование. Создание копий спрайтов.	2	0,5	1,5
26	Тема 25. Механика клонов. Удаление и взаимодействие.	2	0,5	1,5
27	Тема 26. Отправка сообщений. Взаимодействие сцен.	2	0,5	1,5
28	Тема 27. Мои блоки. Создание собственных функций.	2	0,5	1,5
29	Тема 28. Основы геймдизайна. Механики и баланс.	2	0,5	1,5
30	Тема 29. Интерфейс игры. Меню и навигация.	2	0,5	1,5
31	Тема 30. Отладка и оптимизация кода.	2	0,5	1,5
32	Тема 31. Итоговый проект. Генерация идеи и прототип.	2	0,2	1,8
33	Тема 32. Итоговый проект. Создание ассетов и кода.	2	0,0	2,0
34	Тема 33. Итоговый проект. Сборка и тестирование.	2	0,0	2,0
35	Тема 34. Подготовка к защите. Создание презентации.	2	0,5	1,5
36	Тема 35. Итоговая защита проектов. Рефлексия.	2	0,0	2,0
	ИТОГО	72	15,9	56,1

6. Обеспечение программы

Для организации образовательного процесса по программе необходимы следующие ресурсы.

Кадровые: Специалист, имеющий высшее или среднее профессиональное образование (педагогическое, техническое или IT-направление), владеющий знаниями, навыками и методикой преподавания курса по визуальному программированию, понимающий возрастную психологию детей.

Материально-технические: Компьютерный класс, оснащенный персональными компьютерами или ноутбуками (1 место на 1 учащегося); настенная доска (маркерная или интерактивная); мультимедийный проектор и экран; система стабилизированного электроснабжения; доступ к сети Интернет; наушники для каждого учащегося (для работы со звуком). Помещение должно соответствовать санитарно-эпидемиологическим требованиям (СанПиН) к организации работы с ЭВМ.

Программно-методические:

1. Оффлайн-версия среды визуального программирования Scratch 3.0 (Scratch Desktop), установленная на каждый ПК.
2. Веб-браузеры с настроенной системой фильтрации контента для доступа к официальному сообществу Scratch.
3. Методическое пособие (поурочные планы, конспекты занятий).
4. Раздаточный материал (алгоритмические карты, шпаргалки по категориям блоков, шаблоны Game Design Document).
5. Мультимедийные презентации к каждому занятию.

7. Литература

Нормативно-правовая база:

1. Федеральный закон РФ «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
2. СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».
3. Концепция развития дополнительного образования детей до 2030 года.

Для педагога (учебная и методическая литература):

4. Резник М. Детский сад будущего: как мы учим творчеству в эпоху цифровых технологий. — М.: Манн, Иванов и Фербер, 2020.

5. Яковлев Н.В. Scratch для детей. Наглядное программирование. — СПб.: БХВ-Петербург, 2021.
6. Раков С.А. Программирование в среде Scratch: учебное пособие. — М.: ИНФРА-М, 2019.
7. Папуловская М.С. Информатика. 5-6 классы. Проектные задачи. — М.: Просвещение, 2020.

Для обучающихся:

8. Сэндфорд Дж. Scratch для детей. Самоучитель по программированию. — М.: Эксмо, 2021.
9. Королёв Д.В. Scratch 3.0. Программирование для детей и их родителей. — М.: Наука и техника, 2020.
10. Фарр Дж. Создаем игры в Scratch. — М.: Манн, Иванов и Фербер, 2020.

Электронные ресурсы:

11. Официальное сообщество Scratch: <https://scratch.mit.edu>
12. Scratch Wiki (русская версия): <https://ru.scratch-wiki.info>