УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ НАДЕЖДИНСКОГО МУНИЦИПАЛЬНОГО РАЙОНА
Индивидуальный предприниматель Жураковская Евгения Сергеевна

УТВЕРЖДАЮ:

_____Е.С. Жураковская

1 августа 2024г.

ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Дополнительная общеобразовательная общеразвивающая программа технической направленности

Возраст обучающихся: 9–10 лет
Срок реализации программы: 1 год

Пушин Руслан Алексеевич, методист
Бережная Татьяна Станиславовна, методист
Лобанов Андрей Сергеевич, методист

п. Зима Южная
2024

Раздел № 1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1.1. Пояснительная записка

Начало XXI века ознаменовалось бурным развитием IT-технологий. Рост и развитие таких компаний как Google, Apple, Facebook подтверждают это. Мировые лидеры IT-индустрии периодически обращаются к школьникам с призывом изучать программирование. Становится понятно, что чем раньше обучающийся начнёт овладевать навыками программирования, тем больший запас знаний, умений и навыков он получит к моменту выбора основного рода деятельности. Даже если в будущем карьерный путь ребёнка не будет связан с программированием, умение разбираться в сложных системах и взаимодействовать с новыми технологиями пригодится ему в любой сфере, ведь цифровые технологии используются повсеместно.

Кроме того, изучение программирования позволит ребёнку познакомиться с сообществом таких же заинтересованных ребят, введёт в подробности и тонкости проектной деятельности. Овладевая навыками программирования, ребёнок затрагивает и смежные направления, такие как логика, математика, теория вероятности, а также и другие области знаний: география, биология, физика, литература – в зависимости от интересов ребёнка и выбора вектора развития собственного проекта.

Когда у ребёнка сформирован необходимый набор знаний, умений и навыков, выполнен ряд задач и упражнений по разным темам, он может, используя их, работать над собственным проектом. Это позволяет развивать такие важные компетенции XXI века, как критическое мышление, креативность и любознательность, коммуникация и работа в команде.

Программа имеет техническую направленность, которая заключается в популяризации и развитии интереса детей младшего школьного возраста к программированию, формировании у них знаний основ программирования,

умения составлять план деятельности, развитии познавательной активности, исследовательских, прикладных способностей, алгоритмического, логического и критического мышления.

При разработке программы учитывался опыт аналогичных образовательных инициатив в России и мире, в том числе разработки Harvard Graduate School of Education, MIT Media Lab, Code.org и других лидеров в области обучения программированию.

Направленность программы – техническая.

Язык реализации программы: государственный язык РФ – русский.

Уровень освоения – стартовый.

Адресат программы – программа рассчитана на обучающихся в образовательных организациях Надеждинского муниципального района в возрасте от 9 до 10 лет.

Организация образовательного процесса

Срок освоения программы – 1 год. Объём программы – 72 часа.

Занятия проходят 1 раз в неделю и длятся 2 академических часа. Длительность академического часа 40 минут. Дети занимаются в кабинете с педагогом, самостоятельно работая за компьютером (непрерывно – не более 20 минут), предусмотрен перерыв.

Набор на программу осуществляется на добровольной основе и не требует наличия дополнительных знаний. Зачисление производится на основании заявления родителя/законного представителя. Количество обучающихся в группе от 6 до 12 человек.

Программа реализуется с помощью учебной платформы Алгоритмика.

1.2. Цель и задачи программы

Цель программы: развитие технического мышления у обучающихся Надеждинского муниципального района 9-10 лет посредством визуального программирования.

Задачи программы

Воспитательные:

1. Воспитывать обучающегося, обладающего информационной культурой.
2. Воспитывать стремление к познавательной активности, исследовательским и прикладным способностям.
3. Воспитывать у обучающихся самостоятельность, инициативность, целеустремлённость.

Развивающие:

1. Развивать логическое, креативное и пространственное мышление.
2. Развивать навык публичного выступления и презентации.
3. Развивать и расширять специфический словарный запас, как на русском, так и на английском языках.
4. Развивать память, внимание, воображение.
5. Формировать навык использования итерационного подхода при решении различных задач.

Обучающие:

1. Знакомить с основными понятиями алгоритмизации (исполнитель, алгоритм, программа, цикл).
2. Обучать способам разработки алгоритмов с использованием последовательностей, событий, циклов и условий.
3. Знакомить с основными принципами визуального программирования.
4. Обучать определять наиболее целесообразный алгоритм для решения поставленной задачи и оптимизировать текущий алгоритм.

5. Обучать способам использования инструментов виртуальной среды Scratch для решения задач.

6. Научать тестировать код, находить и исправлять ошибки в нём.

7. Обучать способам разработки собственных программ с использованием чужого кода.

1.3. Содержание программы

Учебный план

№	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1	**Введение в программу**	**10**	**4**	**6**	
1.1	Линейный алгоритм	2	2	0	Классический тест
1.2	Циклы	2	0,5	1,5	Сопоставление
1.3	Начальная расстановка	2	0,5	1,5	Заполнение пропусков
1.4	События	2	0,5	1,5	Опрос
1.5	Проект. Визитка	2	0,5	1,5	Проект
2	**Пространство**	**10**	**3**	**7**	
2.1	Координаты	2	1	1	Опрос
2.2	Повороты в направление	2	0,5	1,5	Сопоставление
2.3	Вращения и градусы	2	0,5	1,5	Заполнение пропусков
2.4	Сообщения	2	0,5	1,5	Сортировка
2.5	Проект. Мультфильм	2	0,5	1,5	Проект
3	**Создание игры**	**12**	**3**	**9**	
3.1	Условия и оператор выбора	2	0,5	1,5	Классический тест
3.2	Изменение координат	2	0,5	1,5	Сопоставление
3.3	Процедуры	2	0,5	1,5	Сортировка
3.4	Планирование игр	2	0,5	1,5	Опрос
3.5	Тестирование игр	2	0,5	1,5	Заполнение пропусков
3.6	Презентация игр	2	0,5	1,5	Проект
4	**Логика**	**12**	**3**	**9**	
4.1	Операторы И, ИЛИ, НЕ	2	0,5	1,5	Классический тест
4.2	Цикл с условием	2	0,5	1,5	Сопоставление
4.3	Случайные числа и диапазоны	2	0,5	1,5	Сортировка
4.4	Области координат	2	0,5	1,5	Опрос
4.5	Групповой проект	4	1	3	Презентация проекта
5	**Переменные**	**12**	**3**	**9**	

5.1	Переменные и циклы	2	0,5	1,5	Опрос
5.2	Типы данных	2	0,5	1,5	Сопоставление
5.3	Счет в играх	2	0	2	Сортировка
5.4	Переменные как параметр	2	1	1	Классический тест
5.5	Проект	4	1	3	Презентация проекта
6	**Клонирование в программировании**	**6**	**2**	**4**	
6.1	Клоны	3	1	2	Классический тест
6.2	Локальные и глобальные переменные	3	1	2	Сопоставление
7	**Подведение итогов**	**10**	**1**	**9**	
7.1	Игра «Зеркала»	4	1	3	Наблюдение
7.2	Финальный проект	4	0	4	Наблюдение
7.3	Визуальное (креативное) программирование. Выпускной офлайн	2	0	2	Презентация проекта
	Итого:	**72**	**19**	**53**	

Содержание учебного плана

1. Раздел: 1: Введение в программу

1.1. Тема: Линейный алгоритм

Теория. Программа. Основные разделы программы. Понятие «Линейный алгоритм». Среда Scratch. Интерфейс среды Scratch.

1.2. Тема: Циклы

Теория. Понятие «Цикл». Что такое спрайты, зачем нужны и что с ними можно делать.

Практика. Смена внешнего вида спрайтов.

1.3. Тема: Начальная расстановка

Теория. Начальная расстановка. Начальные параметры внешнего вида спрайта. Анимации. Планирование.

Практика. Создание расстановки, анимации, планирования.

1.4. Тема: События

Теория. События. Способы запуска скриптов. Бесконечные скрипты.

Практика. Создание событий, запуск скриптов.

1.5. Тема: Проект. Визитка

Теория. Что такое проект. Составляющие проекта. Структура. Как описать и представить проект. Презентация проекта.

Практика. Создание проекта из пройденного материала. Презентация проектов.

2. Раздел: Пространство

2.1. Тема: Координаты

Теория. Координаты спрайта. Способы задания положения спрайта.

Практика. Практическая работа по заданию положения спрайта.

2.2. Тема: Повороты и направления

Теория. Поворот. Направление поворота.

Практика. Изменение ориентации спрайта относительно других объектов.

2.3. Тема: Вращения и градусы

Теория. Вращение. Градусы. Ориентация спрайтов.

Практика. Практическое применение вращения.

2.4. Тема: Сообщения

Теория. Сообщения. Взаимодействие объектов и переходов между сценами с помощью сообщений.

Практика. Программирование взаимодействия объектов и переходов между сценами с помощью сообщений.

2.5. Тема: Проект. Мультфильм

Теория. Что такое проект? Способы создания и представления проектов. Использование инструментов визуального программирования при создании мультфильмов.

Практика. Создание мультфильма. Презентация.

3. Раздел: Создание игры

3.1. Тема: Условия и оператор выбора

Теория. Условия. Оператор выбора.

Практика. Использование оператора выбора при решении задач в играх.

3.2. Тема: Изменение координат

Теория. Изменение координат.

Практика. Перемещение спрайтов по сцене.

3.3. Тема: Процедуры

Теория. Понятие «процедуры» как подпрограммы, которая только выполняет операции, без возврата значения. Имя, параметры, результат.

Практика. Использование процедур при решении задач, в т. ч. в играх.

3.4. Тема: Планирование игры

Теория. Виды и способы планирования игры.

Практика. Планирование игры.

3.5. Тема: Тестирование игр

Теория. Способы и виды тестирования игр.

Практика. Тестирования созданной игры.

3.6. Тема: Презентация игр

Теория. Способы и структура презентации игры.

Практика. Презентация созданной игры.

4. Раздел: Логика

4.1. Тема: Операторы И, ИЛИ, НЕ

Теория. Логические операторы И, ИЛИ, НЕ. Приоритет логических операторов.

Практика. Использование логических операторов.

4.2. Тема: Цикл с условием

Теория. Понятие «Цикл с условием».

Практика. Использование цикла с условием.

4.3. Тема: Случайные числа и диапазоны

Теория. Случайные числа. Диапазон случайных чисел.

Практика. Использование случайных чисел в программах.

4.4. Тема: Области координат

Теория. Понятие «Область координат».

Практика. Настройка области сцены для появления и движения объектов.

4.5. Тема: Групповой проект

Теория. Что такое групповой проект. Способы планирования и реализации.

Практика. Презентация группового проекта.

5. Раздел: Переменные

5.1. Тема: Переменные и циклы

Теория. Переменные. Создание и вызов переменных. Типы данных в переменных.

Практика. Сохранение информации в переменных.

5.2. Тема: Типы данных

Теория. Понятие «Типы данных».

Практика. Операции с типами данных.

5.3. Тема: Счёт в играх

Практика. Создание счёта в играх. Организация счёта в играх.

5.4. Тема: Переменные как параметр

*Теория. Понятие «*Переменная как параметр».

Практика. Операции с переменными.

5.5. Тема: Проект

Теория. Способы создания проекта технической направленности.

Практика. Презентация проекта.

6. Раздел: Клонирование в программировании

6.1. Тема: Клоны

Теория. Клонирование. Однотипные объекты (клоны): создание объекта, принадлежащего классу, наследование свойств и события.

Практика. Взаимодействие клонов одного класса и разных классов друг с другом.

6.2. Тема: Локальные и глобальные переменные

Теория. Локальные и глобальные переменные. Организация хранения данных в глобальных и локальных переменных.

Практика. Организация хранения данных в глобальных и локальных переменных.

7. Раздел: Подведение итогов

7.1. Тема: Игра «Зеркала»

Теория. Использование курсора для создания интерфейса игры. Составление плана игры с помощью mind map. Улучшение интерфейса игры, меняя форму курсора.

Практика. Создание плана игры и скриптов для спрайтов. Разработка скриптов для зеркала, блика и лучиков. Создание плана на платформе. Программирование курсора в «Scratch. Создание игры. Добавление возможности запустить игру заново после показа финального экрана. Доработка игры. Создание скрипта кристалла, переключение на финальный экран. Программирование кнопки «Новая игра» в «Scratch». Тестирование игры.

7.2. Тема: Финальный проект

Практика. Выбор проекта, который будет презентован на финальном занятии: планирование и программирование новых проектов; работа над проектами, начатыми на прошлом занятии; дополнение любых проектов, созданных в процессе обучения. Обсуждение возникших вопросов. Продолжение программирования своих проектов. Обсуждение, что можно добавить в проект. Завершение программирования своих проектов. Проведение тестирования проектов, которые будут презентовать на финальном занятии. Публикация проектов в «Зале славы» и оценка проектов других ребят: поставить реакцию или комментарий.

7.3. Тема: Визуальное (креативное) программирование. Выпускной офлайн

Практика. Ролевая мини-игра «Возвращение на землю».

1.4. Планируемые результаты

Личностные результаты

Обучающийся будет способен:

- самостоятельно планировать и осуществлять деятельность, связанную с цифровой средой.

У обучающегося будет сформировано:

- понятие цифровой культуры;

- устойчивый интерес к проектной деятельности.

Метапредметные результаты

Обучающийся будет уметь:

- эффективно планировать деятельность;

- применять изученные технологии в повседневной жизни;

- выполнять логические операции: сравнение, анализ, обобщение, установление аналогий.

Предметные результаты

Обучающийся будет знать:

- функции, события, типы событий, обработчики событий;
- координатную плоскость, установку и изменение координат;
- алгоритмические конструкции, логические значения и операции.

Обучающийся будет уметь:

- использовать итерационный подход;
- применять на практике полученные знания.

Обучающийся будет владеть:

- языком программирования Scratch и основными алгоритмическими структурами: линейной, условной и циклической.

РАЗДЕЛ № 2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

2.1. Условия реализации программы

1. Материально-техническое обеспечение:

- учебный кабинет с необходимой мебелью для ведения образовательного процесса (парты/столы, стулья, шкаф для методических материалов);
- персональные компьютеры или ноутбуки по количеству обучающихся;
- мультимедийный проектор;
- доска или флипчарт, мел или маркер;
- бумага А4, ручки/карандаши;
- приложение или доступ на платформу «Алгоритмика»;
- логины и пароли для доступа на обучающую платформу «Алгоритмика».

Требования к ПК:

- обязательно: колонки, микрофон, монитор диагональю не менее 15" 1366*768;
- опционально: наушники, веб-камера;
- операционная система Windows 7,8,10/ MacOS;

- браузер Google Chrome версии 76.0.3809.132. и старше.

2. Учебно-методическое и информационное обеспечение:

- поурочные методические рекомендации к занятиям издательства «Алгоритмика»;
- тематические презентации;
- задания на платформе «Алгоритмика» с автопроверкой;
- справочный теоретический материал, размещённый на платформе «Алгоритмика»;
- бонусные задания на платформе;
- задачи со свободным решением для реализации в среде разработки на базе платформы «Алгоритмика».

2.2. Оценочные материалы и формы аттестации

Контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов:

- на каждом занятии: выполнение заданий на платформе, взаимоконтроль учеников в парах, самоконтроль учащихся;

- в конце каждого раздела: проведение презентации финальных проектов и их оценка;

- после третьего раздела: промежуточная аттестация в форме опроса обучающихся;

- после освоения всей программы: итоговый контроль в форме презентации проекта.

Используются следующие формы аттестации:

- **классический тест** – выбор одного или нескольких правильных ответов в шаблоне с вопросами;
- **заполнение пропусков** – шаблон текста с пропусками для заполнения;

- **классификация** – шаблон с элементами для группировки объектов в соответствии с их общими признаками;

- **сопоставление** – шаблон с элементами для сравнения двух элементов для того, чтобы показать их сходства или различия;

- **сортировка** – шаблон с элементами для последовательного расположения или формирования в группы элементов в зависимости от выбранного критерия;

- **ответ в свободной форме**, в т. ч. прикрепление файла – шаблон для внесения ответа в свободной форме;

- **опрос** – шаблон для проверки знаний посредством устных или письменных вопросов;

- **презентация** – представление информации в форме текста, таблицы, слайдов.

- **проект** – уникальный индивидуальный или групповой план, направленный на достижение конкретной цели, практического применения полученных знаний.

Возможны следующие **формы деятельности** учащихся: учебная дискуссия, занятие-проект, практическое занятие. При решении одной задачи учащиеся могут работать индивидуально, затем в парах, а при проверке задачи – в паре с преподавателем.

2.3. Методические материалы

Методические материалы предоставляются платформой «Алгоритмика» и состоят из:

- поурочных методических рекомендаций к занятиям;
- тематических презентаций;
- заданий на платформе;
- справочного и теоретического материала, размещённого на платформе;

- бонусных заданий на платформе;

- задач со свободным решением для реализации в среде разработки на базе платформы «Алгоритмика».

Дополнительная методическая литература:

1. Анеликова Л.А., Гусева О.Б. Программирование на алгоритмическом языке КуМир. М.: Солон-Пресс, 2011.

2. Вордерман К., Вудкок Дж., Макманус Ш. и др. Программирование для детей. М.: Манн, Иванов и Фербер, 2015.

3. Кушниренко А.Г., Лебедев Г.В. 12 лекций о том, для чего нужен школьный курс информатики и как его преподавать: Метод. пособие. М.: Лаборатория Базовых Знаний, 2000.

4. Кушниренко А.Г., Лебедев Г.В. Основы информатики и вычислительной техники. М.: Просвещение, 1990.

2.4. Календарный учебный график

Этапы образовательного процесса		1 год
Продолжительность учебного года, неделя		36
Количество учебных дней		34
Продолжительность учебных периодов	1 полугодие	16.09.2024-29.12.2024
	2 полугодие	09.01.2025-31.05.2025
Возраст детей, лет		7-9
Продолжительность занятия, академический час		2
Режим занятия		1 раз/нед
Годовая учебная нагрузка, час		72

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Гуриков С.Р. Информатика. М.: Инфра-М, 2023.

2. Гусева Е.Н. Информатика: Уч. пособие. М.: Флинта, 2023.

3. Лопатин В.М. Информатика. М.: Лань, 2021

4. Семакин И.Г., Залогова Л.А., Русаков С.В. Информатика: Уч. Для 7 класса. М.: Просвещение/Бином, 2021.