### C:\Users\Tochka Rosta\Desktop\ВОСПИТАТЕЛЬНАЯ РАБОТА\ПРОГРАММЫ ВД\титульники сканы\основы программирования.jpg

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### Направленность дополнительной общеобразовательной общеразвивающей

**программы**

Дополнительная общеобразовательная программа «Основы программирования» имеет техническую направленность и осуществляется в соответствии с учебным планом дополни- тельного образования школы.

Дополнительная общеобразовательная программа «Основы программирования», да- лее Программа составлена в соответствии с Федеральным Законом от 29.12.2012 № 273-ФЗ

«Об образовании в Российской Федерации», Федеральным Законом от 31.07.2020 № 304 «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации» по во- просам воспитания обучающихся», Приказом министерства Просвещения РФ от 22.07.2022 № 629 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам», Постановлениями Главного госу- дарственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утвержде- нии санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к ор- ганизациям воспитания и обучения, отдыха и оздоровления детей и молодежи»», от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиени- ческие нормативы и требования к обеспечению безопасности и (или) безвредности для чело- века факторов среды обитания»», СП 2.4.3648-20 «Санитарно-эпидемиологические требова- ния к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», Уста- вом школы «Ступени».

Уровень освоения Программы – ознакомительный.

**Актуальность и педагогическая целесообразность Программы**

Информатика имеет большое и все возрастающее число междисциплинарных связей, причем как на уровне понятийного аппарата, так и на уровне инструментария. Многие пред- метные знания и способы деятельности (включая использование средств ИКТ), освоенные обучающимися на базе информатики, находят применение как в рамках образовательного процесса при изучении других предметных областей, так и в иных жизненных ситуациях, становятся значимыми для формирования качеств личности, т. е. ориентированы на формиро- вание метапредметных и личностных результатов. На протяжении всего периода становления школьной информатики в ней накапливался опыт формирования образовательных результа- тов, которые в настоящее время принято называть современными образовательными резуль- татами.

В настоящее время в России активно строится информационное общество. Одним из критериев признания общества информационным является снижение занятости в сфере про- изводства и увеличение в сфере услуг и информации (поскольку "сырьем" для нефизического труда является именно информация). Так, в Западной Европе, Японии и Северной Америке более 70% населения заняты в "информационной" сфере.

Для увеличения количества занятых в информационной сфере людей дополнительное образование может предложить корректировку к программе такого фундаментального школьного предмета, как информатика, добавив, например, достаточное количество часов по программированию. После такой подготовки ребенку намного проще адаптироваться к со- временным требованиям. Более того, некоторые современные информационные гиганты (например, Яндекс) считают, что школьники вполне могут принимать участие в разработках приложений, которые будут полезны многим пользователям. Таким образом, не нужно дока- зывать необходимость кружков, направленных на обучение программированию, в дополни- тельном образовании.

**Отличительные особенности** данной программы от уже существующих программ обучения программированию. В предлагаемой программе учащиеся получают законченный объем знаний определенного уровня по различным направлениям программирования. Затем происходит углубление и расширение знаний по каждому из направлений. Так, например, в направлении "Языки программирования" учащиеся сначала осваивают основные алгоритми- ческие конструкции языков программирования на примере Robot или Scratch, чтобы научиться писать простые программы, а затем учатся работать в команде, составляя одну большую программу из кусочков, написанных разными детьми.

**Цель программы:** обеспечение достижения планируемых результатов ООП, создание условий для развития проектно-исследовательской компетенции обучающихся, творческие способности в соответствии с интересами и склонностями учащихся.

### Задачи Программы

**Образовательные задачи:**

* Достижение планируемых результатов ООП в части формирования личностных, ме- тапредметных и предметных результатов обучения;
* ознакомить с основами моделирования и программирования;
* совершенствовать навыки программирования, экспериментирования и умения оцени- вать современные способы управления;
* обучить соблюдению правил техники безопасности при обращении персональным компьютером,
* научить использовать компьютер как инструмент для игр и приложений;
* сформировать общеучебные умения и навыки на основе средств и методов информа- тики и ИКТ, в том числе овладение умениями работать с различными видами инфор- мации, самостоятельно планировать и осуществлять индивидуальную и коллективную информационную деятельность, представлять и оценивать ее результаты;
* создать условия для пропедевтического (предварительного, вводного, ознакомитель- ного) изучения понятий основного курса школьной информатики, обеспечивающее целенаправленное формирование общеучебных понятий, таких как «объект», «систе- ма», «модель», «алгоритм» и др.;
* научить принципам работы с основными прикладными программами и средами про- граммирования.

### Развивающие задачи:

* развивать творческие способности владения компьютером;
* развивать навыки построения моделей и научить основам работы с оборудованием и программным обеспечением;
* обеспечить межпредметную интеграцию знаний и умений, рассматривая прикладные вопросы технической направленности;
* сформировать умение самостоятельно приобретать и применять знания;
* развивать алгоритмическое мышление и творческое воображение.

### Воспитательные задачи:

* воспитать уважение к точным наукам, понимание их жизненной необходимости, стремление к дальнейшему обучению.
* воспитание ответственного и избирательного отношения к информации; развитие по- знавательных, интеллектуальных и творческих способностей учащихся.

## Категория учащихся

Дополнительная общеобразовательная общеразвивающая программа «Программирование» рассчитана на контингент подростков 11-12 лет.

Исходя из специфики занятий по данной программе, предельная наполняемость учебной группы составляет не более 11 человек.

Срок реализации программы – сентябрь 2025 г. - май 2026 г., объем программы - 34 часа за срок обучения.

Программа рассчитана на 34 учебных часа в год (занятия планируется проводить 1 раз в неделю).

Форма обучения – очная.

Форма реализации: занятия – исследования, занятия – упражнения, творческие работы, игровые проекты.

Режим занятий:1 раз в неделю по 1 занятию, периодичность занятий – еженедельно. Продолжительность одного занятия 40 минут, включая непосредственно содержательный аспект в соответствии с учебно-тематическим планированием, а также с учетом организационных и заключительных моментов занятия. Время работы за компьютером не более 30 минут. Перерыв между занятиями – 10 минут.

Режим организации занятий по данной дополнительной общеобразовательной программе определяется календарным учебном графиком и соответствует нормам, утвержденным «СанПин к устройству, содержанию и организации режима работы образовательных организаций допол- нительного образования детей» № 41 от 04.07.2014 (СанПин 2.4.43172 -14, пункт 8.3, приложе- ние №3)

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ

Программа способствует формированию личностных, регулятивных, познавательных и коммуникативных учебных действий.

### Предметные результаты Обучающийся научится:

* знать принципы и структуру Scratch проектов, формы представления и управления ин- формацией в проектах в среде Scratch;
* уметь проектировать, изготовить и разместить в сети или подготовить для иной формы представления Scratch проекты;
* использовать основные алгоритмические конструкции: линейный алгоритм, циклический алгоритм, разветвляющийся алгоритм;
* владеть способами работы с изученными программами;
* применять при создании Scratch проектов основные принципы композиции и колористики;
* осуществлять рефлексивную деятельность, оценивать свои результаты, корректировать дальнейшую деятельность по разработке Scratch проектов;
* владеть специальными знаниями и практическими навыками в области программирования в среде Scratch;
* владеть приемами организации и самоорганизации работы по созданию проектов в среде Scratch;
* иметь положительный опыт коллективного сотрудничества при разработке проектов всреде Scratch;
* иметь опыт коллективной разработки и публичной защиты проектов в среде Scratch.

### Обучающийся получит возможность научиться:

* разрабатывать сценарии новых проектов и игр в среде Scratch;
* разрабатывать и создавать новые спрайты для собственных проектов в среде Scratch;
* владеть навыками структурного программирования для реализации сценариев собственных проектов и игр в среде Scratch;
* разрабатывать процедуры и функции в среде Scratch;
* использовать переменные для передачи данных из одной процедуры в другую;
* создавать собственные проекты Scratch, публиковать их в медиапространстве, представлять и защищать их.

### Личностные УУД:

* оценивать свои и чужие поступки, выбирать, как поступить, отвечать за свой выбор
* осознавать важность саморазвития, использовать это знание в жизненных ситуациях
* осознавать целостность мира и многообразие взглядов на него, вырабатывать свои мировоззренческие позиции
* понимать чувства других людей
* идти на взаимные уступки в разных ситуациях
* осознавать свои эмоции, мысли, черты характера, адекватно выражать и контролировать их в общении

### Познавательные УУД:

* находить ответы на вопросы (курса и для решения жизненных задач) в различных источниках информации
* анализировать свои и чужие поступки (действия, чувства; изменения в самом себе)
* обобщать (делать выводы, доказывать), устанавливать причинно-следственные связи (поступков, ситуаций)
* обогатить представления о собственных ценностях, эмоциональном мире человека, их роли в жизни.

### Регулятивные УУД:

* определять цель, проблему в учебной деятельности (на занятии)
* выдвигать версии в группе и индивидуально
* определять цель, проблему в жизненно-практической деятельности, оценивать степень и способы достижения цели в жизненных ситуациях, самостоятельно исправлять ошибки
* планировать свою деятельность в учебной и жизненной ситуациях
* прогнозировать последствия собственных и чужих поступков

### Коммуникативные УУД:

* излагать свое собственное мнение и позицию с аргументами, фактами
* понимать позицию другого (выраженную в явном и неявном виде: чувства, причины действий, поступков), корректировать свое мнение, достойно признавать его ошибочность
* самостоятельно решать проблемы в общении с взрослыми и друзьями
* преодолевать конфликты, толерантно относиться к другому человеку

### Метапредметные результаты:

* умение анализировать;
* умение работать с текстом;
* развитие воображения; развитие внимания и памяти.

## СОДЕРЖАНИЕ ПРОГРАММЫ

### Учебный (тематический) план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название раздела, темы** | **Количество часов** | | | **Формы атте- стации (кон- троля)** |
| **Теория** | **Пра кти ка** | **Всего** |
| 1. | Правила техники безопасности. Интерфейс  среды программирования Scratch | 1 |  | 1 | Предваритель-  ный контроль |
| 2. | Сцена. Редактирование фона | 1 |  | 1 |  |
| 3. | Понятие спрайтов |  | 1 | 1 |  |
| 4. | Основные скрипты программы Scratch |  | 1 | 1 |  |
| 5. | Синий ящик – команды движения |  | 1 | 1 |  |
| 6. | Назначение сенсоров |  | 1 | 1 |  |
| 7. | Создание программ для передвижения спрай-  тов по сцене. |  | 1 | 1 | тест |
| 8. | Создание программ для рисования различных  фигур | 1 |  | 1 |  |
| 9. | Фиолетовый ящик – внешний вид объекта |  | 1 | 1 |  |
| 10. | Оживление объекта с помощью добавления  костюмов |  | 1 | 1 |  |
| 11. | Переупорядочивание костюмов |  | 1 | 1 |  |
| 12. | Назначение сенсоров *костюм* и *размер* | 1 |  | 1 |  |
| 13. | Понятие раскадровки движения |  | 1 | 1 | Промежуточ- ный контроль |
| 14. | Создание программы для управления внеш-  ним видом объекта |  | 1 | 1 |  |
| 15. | Создание Scratch-историй |  | 1 | 1 |  |
| 16. | Лиловый ящик – добавление звуков |  | 1 | 1 |  |
| 17. | Озвучивание Scratch-историй |  | 1 | 1 |  |
| 18. | Базовая конструкция ветвление |  | 1 | 1 |  |
| 19. | Циклы с фиксированным числом повторений | 1 |  | 1 |  |
| 20. | Создание программ с использованием циклов |  | 1 | 1 |  |
| 21. | Зеленый ящик – операторы |  | 1 | 1 |  |
| 22. | Синхронизация работы объектов |  | 1 | 1 | Текущий кон-  троль |
| 23. | Последовательность и параллельность вы-  полнения |  | 1 | 1 |  |
| 24. | Последовательные и параллельные потоки | 1 |  | 1 | Шестнадцать |
| 25. | Взаимодействие между спрайтами |  | 1 | 1 |  |
| 26-27. | Компьютерные игры – вред или польза. Виды  компьютерных игр | 1 | 1 | 2 |  |
| 28. | Разработка базовых спрайтов для игры |  | 1 | 1 |  |
| 29. | Логика создания персонажей для игры |  | 1 | 1 |  |
| 30. | Разработка и создание основных спрайтов и  их костюмов для будущей игры |  | 1 | 1 |  |
| 31. | Синхронизация работы скриптов для разных  спрайтов |  | 1 | 1 |  |
| 32. | Переход из одной сцены в другую. Создание  интерфейса игры |  | 1 | 1 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 33. | Разработка и защита творческого проекта |  | 1 | 1 | Итоговый контроль  Зачет |
|  | Всего | **7** | **26** | **33** |  |

### Содержание Программы

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Название разде- ла, кол-во часов** | **Содержание раздела** |
| 1 | **Интерфейс про- граммы Scratch (1 ч)** | Введение. Что такое Scratch. Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch. Теория. История создания среды Scratch. Основные базовые алгоритмические конструкции (линейные алгоритмы, с условным оператором, циклического типа с предусловием и постусловием) и их исполнение в среде Scratch. Понятие исполнителя, алгоритма и программы, их назначение, виды и использование. Виды управления исполнителем. Способы записи алгоритма. Основные характеристики исполнителя. Система ко- манд исполнителя. Понятие проект, его структура и реализация в среде Scratch. Основные компоненты проекта Scratch: спрайты и скрипты. Принцип создания анимации и движения объектов. Листинг программы. Сцена. Текущие данные о спрайте. Стиль поворота. Закладки. Панель инструментов, Новый спрайт. Координаты мышки. Режим представления. Окно скриптов. Окно блоков. Блоки стека. Блоки заголовков.  Блоки ссылок. Самодостаточные и открытые скрипты. |
| **2** | **Начало работы в среде Scratch**  **(2 ч)** | Сцена. Редактирование фона. Добавление фона из файла. Теория. Сцена. Широта и высота сцены. Текущие координаты объекта. Редактирование текущего фона. Вставка нового фона из файла. Вставка стандартного фона из библиотечного моду- ля среды. Рисование фона в графическом редакторе. Создание нескольких фонов в одной сцене.  Практика. Создание фона сцены на выбранную учащимся тему.  **3.** Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов.  Теория. Стандартный объект. Спрайты. Список спрайтов. Редактор рисования для создания новых спрайтов. Инструменты рисования (кисточка, линия, текст, эллипс) и редактирования объекта (ластик, заливка, поворот, выбор, печать, пипетка). Центрирование костюма. Масштабирование спрайта. Загрузка на сцену спрайтов из стандартной коллекции среды Scratch. Вставка спрайтов из файлов форматов JPG, BMP, PNG, GIF. Выбор случайного спрайта. Удаление спрайтов.  Практика. Создание фона сцены и прорисовка основных спрайтов для Scratch-истории. |
| **3** | **Основные скрипты про- граммы Scratch (18 ч).** | Синий ящик – команды движения. Темно-зеленый ящик – ко- манды рисования.  Теория. Команды – *идти*; *повернуться направо (налево)*; *по- вернуть в направлении*; *повернуться к; изменить х (у) на; установить х (у) в; если край, оттолкнуться.* Принципиальное различие действия команд *идти в* и *плыть в*. Назначение сенсоров *положение х, положение у* и *направлении.* Команды – *очистить, опустить перо, поднять перо, установить цвет пера, изменить цвет пера на, установить цвет пера, изменить тень пера, установить тень пера, изменить размер пера на, установить размер пера, печать.*  Практика. Создание программ для передвижения спрайтов по сцене. Создание программ для рисования различных фигур.   1. Фиолетовый ящик – внешний вид объекта. Оживление объекта с помощью добавления костюмов.   Теория. Костюмы спрайта. Копирование и редактирование костюма спрайта с помощью редактора рисования. Переупорядочивание костюмов. Команды – *перейти к костюму, следующий костюм, говорить…в течении…секунд, сказать, думать, думать…секунд, изменить… эффект на, установить эффект…в значение, убрать графические эффекты, изменить размер на, установить размер, показаться, спрятаться, перейти в верхний слой, перейти назад на…1 слоев.* Назначение сенсоров *костюм* и *размер.* Понятие раскадровки движения. Изменение костюма спрайта для имитации движения*.* Практика. Создание программы для управления внешним видом объекта. Создание Scratch-историй с имитацией хождения и движения объектов.   1. Желтый ящик – контроль. Лиловый ящик – добавление звуков. Теория. Кнопка с зеленым флажком и ее назначение. Управление последовательностью выполнения скриптов. Понятие управляющих сообщений. Команды – *передать, пере- дать и ждать, когда я получу.* Скрипты для создания условных конструкций программы – *если, если…или*. Скрипты для управления циклами – *всегда, повторить, всегда, если, повторять до.*. Команды – *когда клавиша…нажата, когда щелкнут по, ждать…секунд, ждать до, остановить скрипт, остановить все.* Загрузка звуков из стандартной коллекции и из файлов жесткого диска. Запись звука через микрофон. Принципиальная разница работы команд *играть звук* и *играть звук до завершения.* Команды – *остановить все звуки, барабану играть…тактов, оставшиеся…тактов, но- ту…играть…тактов, выбрать инструмент, изменить гром- кость, установить громкость, изменить темп на, установить темп.* Назначение сенсоров *громкость* и *темп.* Практика. Создание программ с элементами управления объектом. Озвучивание Scratch-историй. 2. Использование в программах условных операторов.   Теория. Базовая конструкция ветвление, назначение, виды (полная и неполная форма). Понятие условия. Изменение по- рядка выполнения скриптов в зависимости от условия. Раз- ветвление листинга программы. Скрипты условных операторов. Использование неполной формы ветвления в системе Scratch.  Практика. Создание программ с изменением последовательного выполнения скриптов при наличии условий.  Функциональность работы циклов. Цикличность выполнения действий в зависимости от поставленных условий. Теория. Циклы с фиксированным числом повторений. Заголовок цикла. Тело цикла. Циклы с условным оператором. Заголовок цикла. Тело цикла. Предусловие и постусловие. Зацикливание.  Практика. Создание программ с использованием циклов с фиксированным числом повторений. Создание программ с использованием циклов с предусловием и постусловием.   1. Зеленый ящик – операторы. Использование арифметических и логических блоков вместе с блоками управления. Теория. Числа. Стринги. Логические величины. Логические выражения. Арифметические операции. Логические операции. Операции сравнения. Команды для работы со стрингами – *слить, буква…в, длинна строки.* Команда *выдать случайное от…до.* Использование арифметических и логических блоков в листинге программы. Просмотр полученного результата. Практика. Создание программ с использованием операций сравнения данных. Создание программ с использованием арифметических данных и логических операций. 2. События. Оранжевый ящик – переменные.   Теория. События в проектах Scratch. Понятие переменных и необходимость их использования в листинге программы. Гло- бальные и локальные переменные. Имя переменной и правила его формирования. Команды для переменных - поставить*…в, изменить…на, показать переменную, спрятать переменную.* Удаление переменных. Создание счетчиков с помощью переменных.  Практика. Разработка сценария Scratch-историй с несколькими событиями. Создание проектов с использование глобальных и локальных переменных.   1. Списки.   Теория. Создание списков и необходимость их использования в проектах Scratch. Добавление в список данных. Удаление данных из списка. Удаление списка. Команды работы со списками – *добавить…к*, *удалить…из, поставить…в…из, заме- нить элемент…в…на, элемент…из, длина списка*.  Практика. Создание программ-тестов по принципу сравнения данных из нескольких списков.   1. Голубой ящик – сенсоры. Ввод-вывод данных.   Теория. Понятие сенсора. Правила применения и область действия команд *касается, касается цвета* и *цвет.касается.* Функционал команды *спросить…и ждать.* Сенсоры *мышка по х, мышка по у, мышка нажата?, клавиша…нажата?, рас- стояние до, перезапустить таймер.* Сенсоры, значение которых можно выводить на экран – *ответ, таймер, громкость, громко? …значение сенсора* и *сенсор….* Необходимость ввода данных для их обработки в программе. Ввод данных с помощью команды *спросить.* Вывод конечного результата обработки с помощью команд *говорить* и *сказать.*  Практика. Создание проектов с использованием значений сенсоров и команды *спросить.* Создание программ для обработки данных пользователя с выводом на экран конечного результата. |

|  |  |  |
| --- | --- | --- |
| **4** | **Работа с не**  **сколькими объектами. Синхронизация их работы (4 ч).** | 13. Последовательность и параллельность выполнения скриптов.  Теория. Последовательные и параллельные потоки в программах Scratch. Одновременная и попеременная работа нескольких исполнителей.  Практика. Создание Scratch-историй с одновременной и попеременной работой нескольких исполнителей.  14. Взаимодействие между спрайтами. Управление через обмен сообщениями.  Теория. Решение проблемы появления новых исполнителей  только после того, как старые исполнители выполнили свои действия. Взаимодействие спрайтов с неподвижными объектами с помощью команд *касается* и *касается цвета.* Взаимодействие спрайтов с помощью команд *передать* и *когда я получу.* Использование сообщений для создания событий.  Практика. Создание Scratch-историй с взаимодействием нескольких исполнителей и неподвижных объектов. Создание  Scratch-историй с взаимодействием нескольких исполнителей. |
|  |
|  |
|  |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **5** | **Использование**  **программы**  **Scratch для со-**  **здания мини-игр**  **(7 ч).** | 15. Виды компьютерных игр. Алгоритмическая разработка  листинга программы.  Теория. Компьютерные игры – вред или польза. Виды компьютерных игр. Этапы разработки игр программистами.  Практика. Алгоритмическая разработка проекта, запись на естественном языке событий и точек взаимодействия героев  будущей игры.  16. Разработка базовых спрайтов для игры. Формирование базовых скриптов.  Теория. Логика создания персонажей для игры. Перевод алгоритма, написанного на естественном языке, в коды Scratch.  Практика. Разработка и создание основных спрайтов и их костюмов для будущей игры. Разработка скриптов для спрайтов и объектов.  17. Синхронизация работы скриптов для разных спрайтов.  Практика. Доработка основного листинга программы с целью установления связей между спрайтами. Тестирование и отладка программы.  18. Переход из одной сцены в другую. Создание интерфейса игры. |
|  |
|  |
|  |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  | Теория. Односторонний (без возможности вернуться назад) переход из одного пространства в другое. Понятие интерфейса. Элементы интерфейса. Основные принципы дизайна интерфейсов. Обратная связь. Необходимые элементы меню.  Практика. Создать программу для перемещения объекта по игровой карте и разработать интерфейс для Scratch-проекта.  19. Сообщество Scratch в Интернете. Просмотр и публикация проектов. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  | Теория. Правила работы в сети. Интернет-сообщества. Сообщество Scratch. Регистрация на сайте. Использование заимствованных кодов и объектов. Авторские права. Публикацияпроектов Scratch.  Практика. Регистрация на сайте сообщества Scratch. Просмотр проектов сообщества и публикация собственных проектов. |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **5** | **Разработка**  **творческого проекта (1 ч)** | 20. Разработка и защита творческого проекта. Разработка и создание программы с использованием подготовленных мате- риалов. Тестирование и отладка проекта. Зачет. |

**Требование к оценке и критерии оценки достижения планируемых результатов**

|  |  |
| --- | --- |
| **Уровни освоения Программы** | **Результат** |
| Высокий уровень освоения Программы | Учащиеся демонстрируют высокую заинтересованность в учебной, познавательной и творческой деятельности, составляющей содержание Программы. На итоговом тестировании показывают отличное знание теоретического материала, практическое применение знаний воплощается в качественный продукт |
| Средний уровень освоения Программы | Учащиеся демонстрируют заметную заинтересованность в учеб- ной, познавательной и творческой деятельности, составляющей содержание Программы. На итоговом тестировании показывают хорошее знание теоретического материала, допуская незначительные ошибки, практическое применение знаний воплощается в качественный продукт |
| Низкий уровень освоения Программы | Учащиеся демонстрируют низкий уровень заинтересованности в учебной, познавательной и творческой деятельности, составляющей содержание Программы. На итоговом тестировании показывают недостаточное знание теоретического материала, практическая работа не соответствует требованиям |

### Литература

1. Голиков Д.И. «Scratch для юных программистов», «БХВ-Петербург», Санкт- Петербург, 2017.
2. Краля Н. А. Метод учебных проектов как средство активизации учебной деятельности учащихся: Учеб-но-методическое пособие / Под ред. Ю. П. Дубенского. Омск: Изд-во ОмГУ, 2005. 59 с.
3. Матвеева Н. В. Информатика и ИКТ.: методическое пособие / Н. В. Матвеева, Е. Н. Челак, Н. К. Конопатова, Л. П. Панкратова. М.: БИНОМ. Лаборатория знаний, 2009. 420 с.
4. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008. 61 с.
5. Скретч [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вер- нуться домой». URL: <http://letopisi.ru/index.php/Скретч>
6. Скретч онлайн[Электронный ресурс] // Онлайн среда программирования Scratch — URL: <https://scratch.mit.edu/>

### Электронные ресурсы

* 1. Методические разработки (презентации, видеоуроки, flash-ролики).
  2. Сетевые ресурсы Scratch <https://scratch.mit.edu/>.

**Календарный учебный график программы "Программирование"**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Месяц** | **Число по плану** | **Число по факту** | **Форма занятия** | **Количество часов** | **Тема занятия** | **Место про- ведения** |
| 1. | Сентябрь | 3 | 17 | Лекция | 1 | Правила техники безопасности. Интер-  фейс среды программирования Scratch | Каб. 36 |
| 2. | Сентябрь | 10 | 17 | Лекция | 1 | Сцена. Редактирование фона | Каб. 36 |
| 3. | Сентябрь | 17 | 24 | Практика | 1 | Понятие спрайтов | Каб. 36 |
| 4. | Сентябрь | 24 | 24 | Выполнение упражнений | 1 | Основные скрипты программы Scratch | Каб. 36 |
| 5. | Октябрь | 1 |  | Выполнение упражнений | 1 | Синий ящик – команды движения | Каб. 36 |
| 6. | Октябрь | 8 |  | Практика | 1 | Назначение сенсоров | Каб. 36 |
| 7. | Октябрь | 15 |  | Выполнение упражнений | 1 | Создание программ для передвижения  спрайтов по сцене. | Каб. 36 |
| 8. | Октябрь | 22 |  | Лекция | 1 | Создание программ для рисования раз-  личных фигур | Каб. 36 |
| 9. | Ноябрь | 5 |  | Выполнение  упражнений | 1 | Фиолетовый ящик – внешний вид объ-  екта | Каб. 36 |
| 10. | Ноябрь | 12 |  | Выполнение  упражнений | 1 | Оживление объекта с помощью добав-  ления костюмов | Каб. 36 |
| 11. | Ноябрь | 19 |  | Выполнение упражнений | 1 | Переупорядочивание костюмов | Каб. 36 |
| 12. | Ноябрь | 26 |  | Лекция | 1 | Назначение сенсоров *костюм* и *размер* | Каб. 36 |
| 13. | Декабрь | 3 |  | Контроль | 1 | Понятие раскадровки движения | Каб. 36 |
| 14. | Декабрь | 10 |  | Практика | 1 | Создание программы для управления  внешним видом объекта | Каб. 36 |
| 15 | Декабрь | 17 |  | Выполнение упражнений | 1 | Создание Scratch-историй | Каб. 36 |
| 16. | Декабрь | 24 |  | Выполнение упражнений | 1 | Лиловый ящик – добавление звуков | Каб. 36 |
| 17. | Январь | 14 |  | Выполнение упражнений | 1 | Озвучивание Scratch-историй | Каб. 36 |
| 18. | Январь | 21 |  | Выполнение упражнений | 1 | Базовая конструкция ветвление | Каб. 36 |
| 19. | Январь | 28 |  | Лекция | 1 | Циклы с фиксированным числом повто-  рений | Каб. 36 |
| 20. | Февраль | 4 |  | Выполнение упражнений | 1 | Создание программ с использованием  циклов | Каб. 36 |
| 21. | Февраль | 11 |  | Выполнение упражнений | 1 | Зеленый ящик – операторы | Каб. 36 |
| 22. | Февраль | 18 |  | Контроль | 1 | Синхронизация работы объектов | Каб. 36 |
| 23. | Февраль | 25 |  | Выполнение упражнений | 1 | Последовательность и параллельность  выполнения | Каб. 36 |
| 24. | Март | 4 |  | Лекция | 1 | Последовательные и параллельные по-  токи | Каб. 36 |
| 25. | Март | 11 |  | Практика | 1 | Взаимодействие между спрайтами | Каб. 36 |
| 26-27. | Март | 18 |  | Лекция | 2 | Компьютерные игры – вред или польза.  Виды компьютерных игр | Каб. 36 |
| Апрель | 1 |
| 28. | Апрель | 8 |  | Выполнение упражнений | 1 | Разработка базовых спрайтов для игры | Каб. 36 |
| 29. | Апрель | 15 |  | Выполнение упражнений | 1 | Логика создания персонажей для игры | Каб. 36 |
| 30. | Апрель | 22 |  | Выполнение упражнений | 1 | Разработка и создание основных спрай-  тов и их костюмов для будущей игры | Каб. 36 |
| 31. | Апрель | 29 |  | Выполнение упражнений | 1 | Синхронизация работы скриптов для  разных спрайтов | Каб. 36 |
| 32. | Май | 6 |  | Контроль | 1 | Переход из одной сцены в другую. Со-  здание интерфейса игры | Каб. 36 |
| 33-34. | Май | 13, 20 |  | Разработка проекта | 2 | Разработка и защита творческого проек-  та | Каб. 36 |