

Муниципальное бюджетное общеобразовательное учреждение г. Астрахани
«Средняя общеобразовательная школа №32 с углубленным изучением
предметов физико-математического профиля»

Принято на педагогическом совете
№ 2
от 9 января 2025г



«Утверждена»

Директор МБОУ г. Астрахани «СОШ № 32»
Сидорина О.Н.

Приказ №_01-10-253__
от «9» января 2025 года



КВАНТОРИУМ

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
Направленность: техническая

**Программирование для начинающих.
Программирование на языках Scratch и Python».**

Срок реализации программы: 1 год

Возраст детей: 11 - 14 лет

Автор составитель:
Замятин Виктор Сергеевич,
учитель информатики,
категория высшая

г. Астрахань
2025 г.

Пояснительная записка о реализации учебно-тематического плана

Программа составлена в соответствии с Федеральным законом «Об образовании в Российской Федерации» от 29.12.2012 № 273, с примерными требованиями к содержанию и оформлению образовательных программ дополнительного образования детей (письмо Минобрнауки РФ от 11.12.2006 № 06-1844), санитарно-эпидемиологическими требованиями к учреждениям дополнительного образования детей, с Приказом Министерства образования и науки Российской Федерации от 29.08.2013 года № 1008 г. Москва «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

Образовательная программа «Программирование для начинающих. Программирование на языках Scratch и Python» разрабатывалась на основе следующих материалов и документов: модуль «пропедевтика программирования со Scratch», Сорокина Т.Е.; «Творческие задания в среде Scratch: рабочая тетрадь для 5-6 классов», Ю.В. Пашковская; «Пропедевтика идей параллельного программирования в средней школе при помощи среды Scratch», В.Г. Рындак, В.О. Джинжер, Л.В. Денисова; «Раннее обучение программированию в среде Scratch», В.Г. Рындак, В.О. Джинжер, Л.В. Денисова; «Проектная деятельность школьника в среде программирования Scratch»/Учебно-методическое пособие/ В.Г. Рындак, В.О. Джинжер, Л.В. Денисова; Авторский курс «Поколение Python» (на сайте stepik.org), Автор Тимур Гуев. <https://stepik.org/course/58852/syllabus>

Научно-техническая направленность.

Введение. Содержание курса «Программирование для начинающих. Программирование на языках Scratch и Python» для учащихся возраста 11-14 лет рассчитано на обучение в объеме 102 учебных часа.

Актуальность и направленность программы.

В настоящее время прослеживается острый кризис квалифицированных кадров в различных сферах рынка труда. Работодатели, хоть и обращают внимание на область предметных знаний, но все же отдают предпочтения универсальным навыкам. Подобная позиция связана с постоянно изменяющимися условиями труда, технологическим прогрессом. Подобные универсальные навыки (способность работать в команде, многозадачность, творческий подход, критическое мышление, продуктивное мышление, нацеленность на результат) вырабатываются в процессе профессионального становления.

Характерной чертой развития общества на протяжении последних десятилетий является его все более расширяющаяся информатизация. Отражением и следствием этой тенденции явилась потребность в подготовке подрастающего поколения к вступлению в информационное общество, любая профессиональная деятельность в котором будет связана с информатикой и информационными технологиями. Изучение данного курса имеет важное значение для развития мышления подростков. В современной психологии отмечается значительное влияние изучения информатики и использования компьютеров в обучении на развитие у школьников теоретического, творческого мышления, а также формирование нового типа мышления, так называемого операционного мышления, направленного на выбор оптимальных решений; открывает новые возможности для овладения такими современными методами научного познания, как формализация, моделирование, компьютерный эксперимент и т.д. Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и некоторые общие мыслительные навыки, полезные и в более широком контексте.

Направленность дополнительной образовательной программы — техническая. Она заключается в развитии технического и творческого мышления у обучающихся среднего школьного

возраста через разработку алгоритмов в цифровой среде, взаимодействие с приложениями, сервисами и инструментами вне зависимости от платформы или интерфейса, формирование навыка определения различных видов ошибок (логическая, синтаксическая, ошибка среды и ошибка взаимодействия) и принципах их исправления в процессе разработки с помощью процедур отладки. Также развитие у детей умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, формирование навыков сотрудничества и коллективной работы в цифровой среде.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного мышления и навыков проектной деятельности.

Программа составлена с учетом задач национального проекта «Цифровая экономика» (федерального проекта «Кадры для цифровой экономики») и национального проекта «Образование» (федерального проекта «Молодые профессионалы»): создание условий для формирования у учащихся основ цифровых компетенций, необходимых для дальнейшего участия в разработке и поддержке инфраструктур передачи, обработки и хранения данных, а также цифровых продуктов для граждан, бизнеса и власти; разработка, внедрение цифровых технологий и платформенных решений.

Сегодня наука и технология развиваются столь стремительно, что образование зачастую не успевает за ними. Одним из самых известных неформальных способов организации внеучебной образовательной деятельности является метод проектов. Самым подходящим инструментом для организации такой деятельности является среда Scratch, которая есть серьезное и современное направление компьютерного дизайна и анимации. Овладев даже минимальным набором операций, самый неискушённый пользователь может создавать законченные проекты. Scratch - это самая новая среда, которая позволяет детям создавать собственные анимированные и интерактивные истории, презентации, модели, игры и другие произведения. Работа в среде Scratch позволяет, с одной стороны, организовать среду для самореализации и самоутверждения учащихся, и, с другой стороны, сформировать у них тягу к творчеству и знаниям и дать подходящие средства её реализации. Быть успешным в такой среде становится проще.

Scratch можно рассматривать как инструмент для творчества, оставив программирование на втором плане. Школьники могут сочинять истории, рисовать и оживлять на экране придуманных ими персонажей, учиться работать с графикой и звуком. Применений возможностям Scratch можно найти множество: в этой среде легко создавать анимированные открытки, мини-игры, мультфильмы. В результате выполнения простых команд может складываться сложная модель, в которой будут взаимодействовать множество объектов, наделенных различными свойствами. Начальный уровень программирования настолько прост и доступен, что Scratch рассматривается в качестве средства обучения не только старших, но и младших школьников.

Когда ученики создают проекты в Scratch, они осваивают множество навыков XXI века, которые будут необходимы для успеха:

- творческое мышление,
- ясное общение,
- системный анализ,
- беглое использование технологий,
- эффективное взаимодействие,
- проектирование,
- умение обучаться и самообучаться,
- самостоятельное принятие решений.

Изучение Scratch может серьезно помочь школьникам освоить азы алгоритмизации и программирования, а полученные знания пригодятся для дальнейшего и более серьезного изучения программирования. Работа в среде Scratch ведется, так же как средство подготовки учащихся к всевозможным конкурсам и выставкам по данной тематике, которые в настоящее время набирают

большие обороты.

Язык программирования Python принято считать одним из самых простых в освоении. Обучаться языку возможно имея лишь базовые навыки использования компьютера, установка необходимого ПО не сложнее установки любой другой программы, а синтаксис языка не перегружен и интуитивно понятен. Данные факторы раскрывают Python максимально эффективным инструментом при обучении детей программированию.

При этом Python является современным, востребованным и перспективным языком разработки, позволяющим не только заложить фундамент знаний будущих программистов и инженеров, но получить востребованные и актуальные навыки, необходимые в учебной, проектной и повседневной деятельности.

Направленность программы
Дополнительная общеобразовательная общеразвивающая программа технической направленности «Программирование для начинающих. Программирование на языках Scratch и Python» имеет техническую направленность, что позволяет обучающимся приобщиться к инженерно-техническим знаниям в области инновационных технологий, сформировать техническое мышление.

Объем программы: 102 тематических часа. Из них 48 часов - теория и 54 часа - практика.

Срок освоения: 1 год - 2 раза в неделю по 1.5 учебных часа (102 часов в год)

Цели программы:

Развитие логического мышления, формирование базовых понятий структурного программирования, развитие творческого и познавательного потенциала подростка.

Задачи:

Образовательные:

Овладение базовыми понятиями объектно-ориентированного программирования и применение их при создании проектов в визуальной среде программирования Scratch;

Формирование базовых понятий структурного программирования на языке Python;

Приобщение обучающихся к новым технологиям, способным помочь им в реализации собственного творческого потенциала;

Развитие познавательной деятельности учащихся в области новых информационных технологий;

Совершенствование навыков работы на компьютере и повышение интереса к программированию.

Воспитательные:

Формирование культуры и навыков сетевого взаимодействия;

Способствование развитию творческих способностей и эстетического вкуса подростков;

Способствование развитию коммуникативных умений и навыков обучающихся.

Развивающие:

Способствование развитию логического мышления, памяти и умению анализировать;

Создание условия для повышения самооценки обучающегося, реализации его как личности;

Формирование потребности в саморазвитии;

Способствование развитию познавательной самостоятельности.

Основные личностные результаты, формируемые в процессе освоения программы – это:
формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершённых творческих учебных проектов;

формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе, приобретённой благодаря иллюстративной среде программирования мотивации к обучению и познанию;

развитие опыта участия в социально значимых проектах, повышение уровня самооценки, благодаря реализованным проектам;

формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;

формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;

формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;

развитие эстетического сознания через творческую деятельность на базе иллюстрированной среды программирования.

К основным метапредметным результатам (осваиваемым обучающимися межпредметным понятиям и универсальным учебным действиям, способности их использования как в учебной, так и в познавательной и социальной практике), формируемые в процессе освоения программы, можно отнести: умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;

умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата, понимая, что в программировании длинная программа не значит лучшая программа;

умение оценивать правильность решения учебно-исследовательской задачи;

умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;

владение основами самоконтроля, принятия решений;

умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;

ИКТ-компетенцию;

умение сотрудничества и совместной деятельности со сверстниками в процессе проектной и учебно-исследовательской деятельности.

Основные предметные результаты, формируемые в процессе изучения программы направлены на:

осознание значения математики и информатики в повседневной жизни человека;

формирование представлений об основных предметных понятиях — «информация», «алгоритм», «модель» и их свойствах;

развитие логических способностей и алгоритмического мышления, умений составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;

развитие представлений о числах, числовых системах;

овладение символьным языком алгебры, умение составлять и использовать сложные алгебраические выражения для моделирования учебных проектов, моделировать реальные ситуации на языке алгебры;

развитие пространственных представлений, навыков геометрических построений и моделирования таких процессов, развитие изобразительных умений с помощью средств ИКТ;

формирование информационной и алгоритмической культуры, развитие основных навыков использования компьютерных устройств и программ;

формирование умения соблюдать нормы информационной этики и права.

В результате освоения дисциплины обучающийся должен:

уметь:

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- настраивать рабочую среду Python;

знать:

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;

- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования;
- основы создания приложений в Python;
- инструментальные средства Python.

Формы аттестации: опрос, защита проекта.

Форма отслеживания и фиксации результатов: фронтальный и индивидуальный опрос, проектная деятельность, практические занятия по каждой теме.

Промежуточный контроль включает: выполнение практического задания.

Материально-техническое обеспечение:

Электронно-программное обеспечение:

- специализированные цифровые инструменты учебной деятельности (компьютерные программы);

Технические средства обучения:

- ноутбук – 15 шт.
- компьютеры с учебным программным обеспечением;
- демонстрационный экран;
- интерактивная доска.

СОДЕРЖАНИЕ ПРОГРАММЫ

I. Начало работы в среде Scratch (8 ч).

1. 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 команд 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. Стилль 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. Сцена. Редактирование фона. Добавление фона из файла.

Теория. Сцена. Ширина и высота сцены. Текущие координаты объекта. Редактирование текущего фона. Вставка нового фона из файла. Вставка стандартного фона из библиотечного модуля 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.

II. Основные скрипты программы Scratch (18 ч).

4. Синий ящик – команды движения. Темно-зеленый ящик – команды рисования.

Теория. Команды – *идти; повернуться направо (налево); повернуть в направлении; повернуться к; изменить x (y) на; установить x (y) в; если край, оттолкнуться.* Принципиальное различие действия команд *идти в* и *плыть в*. Назначение сенсоров *положение x, положение y и направление.* 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

5. 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, изменитьэффект 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.

6. Желтый ящик – контроль. Лиловый ящик – добавление звуков.

Теория. Кнопка с зеленым флажком и ее назначение. Управление последовательностью выполнения скриптов. Понятие управляющих сообщений. Команды – *передать, передать 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 щелкнул 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, ноту...играть...тактов, выбрать инструмент, изменить громкость, установить громкость, изменить темп на, установить темп.* Назначение сенсоров *громкость и темп.*

Практика. Создание программ с элементами управления объектом. Озвучивание Scratch-историй.

7. Использование в программах условных операторов.

Теория. Базовая конструкция ветвление, назначение, виды (полная и неполная форма). Понятие условия. Изменение порядка выполнения скриптов в зависимости от условия. Разветвление листинга программы. Скрипты условных операторов. Использование неполной формы ветвления в системе Scratch.

Практика. Создание программ с изменением последовательного выполнения скриптов при наличии условий.

8. 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. Заикливание.

Практика. Создание программ с использованием циклов с фиксированным числом повторений. Создание программ с использованием циклов с предусловием и постусловием.

9. Зеленый ящик – операторы. Использование арифметических и логических блоков вместе с блоками управления.

Теория. Числа. Строинги. Логические величины. Логические выражения. Арифметические операции. Логические операции. Операции сравнения. Команды для работы со строингами – *слить,*

буква...в, длина 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.

10. События. Оранжевый ящик – переменные.

Теория. События в проектах Scratch. Понятие переменных и необходимость их использования в листинге программы. Глобальные 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 использованием глобальных и локальных переменных.

11. 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, заменить элемент...в...на, элемент...из, длина списка*.

Практика. Создание программ-тестов по принципу сравнения данных из нескольких списков.

12. 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 x, мышка по y, мышка нажата?, клавиша...нажата?, расстояние 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

III. Работа с несколькими объектами. Синхронизация их работы (4ч).

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.

IV. Использование программы Scratch для создания мини-игр (8 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.

V. Разработка творческого проекта (4ч)

19. Разработка и защита творческого проекта. Разработка и создание программы с использованием подготовленных материалов. Тестирование и отладка проекта. Защита проекта. Опубликование проекта.

VI. Начало работы в Python (10 ч)

Теория. Общие сведения о языке Python. История языка программирования Python и его возможности. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python.

Практика. Установка программы Python. Знакомство с интерфейсом. Сохранение и запуск python-программ. Установка модулей.

VII. Условный оператор (7ч)

Теория. Логические операторы: and, or, not. Порядок выполнения операций. Переменные без значения – None. Понятие «условный оператор», «вложенные команды», «оператор сравнения». Конструкция if и её синтаксис. Операторы сравнения: <, >, >=, <=, !=, ==. Структура программы. Конструкция if-else. Команды if и elif.

Практика. Решение задач на отработку условного оператора и операторов сравнения.

VIII. Типы данных (7ч)

Теория. Ввод-вывод данных в программе. Типы данных, примеры работы с типами данных. Метод map.

Практика. Укрепление навыка работы с переменными на примерах. Выполнение практических заданий.

IX. Циклы for и while (15ч)

Теория. Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Заикливание и выход из цикла с помощью команды break. Примеры использования циклов. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа.

Практика. Написание программы по определенным условиям. Решение задач на применение циклов for и while.

X. Строковый тип данных (9ч)

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практика. Применение методов строк. Применение срезов к строкам. Решение задач со строками.

XI. Списки (12ч)

Теория. Списочные выражения. Методы split, join. Другие методы списков и строк. Знакомство со срезами и диапазонами. Работа со списками, срезами. Примеры решения задач со спискам.

Практика. Создание списка, операции со списками и элементами списков, перебор списка. Применение методов списков. Применение срезов к спискам.

№ п/ п	НАИМЕНОВАНИЕ ТЕМЫ	Дата		ТЕОРИЯ	ПРАКТ.	ВСЕГО
		По плану	фактич			
I. Начало работы в среде Scratch (8 ч)						
1	Введение. Что такое Scratch.			1	0	1
2	Основные алгоритмические конструкции. Знакомство с интерфейсом программы Scratch.			1	0	1
3	Сцена. Редактирование фона. Добавление фона из файла.			1	1	2
4	Понятие спрайтов. Добавление новых спрайтов. Рисование новых объектов.			1	1	2
5	Создание и редактирование спрайтов и фонов для сцены			1	1	2
II. Основные скрипты программы Scratch (18 ч)						
4	Синий ящик – команды движения. Темно-зеленый ящик – команды рисования.			1	1	2
5	Фиолетовый ящик – внешний вид объекта. Оживление объекта с помощью добавления костюмов.			1	1	2
6	Желтый ящик – контроль. Лиловый ящик – добавление звуков.			1	1	2
7	Использование в программах условных операторов.			1	1	2
8	Функциональность работы циклов. Цикличность выполнения действий в зависимости от поставленных условий.			1	1	2
9	Зеленый ящик – операторы. Использование арифметических и логических блоков вместе с блоками управления.			1	1	2
10	События. Оранжевый ящик – переменные.			1	1	2
11	Списки.			1	1	2
12	Голубой ящик – сенсоры. Ввод-вывод данных.			1	1	2

III. Работа с несколькими объектами. Синхронизация их работы (4 ч)						
13	Последовательность и параллельность выполнения скриптов.			1	1	2
14	Взаимодействие между спрайтами. Управление через обмен сообщениями.			1	1	2
IV. Использование программы Scratch для создания мини-игр (8 ч)						
15	Виды компьютерных игр. Алгоритмическая разработка листинга программы.			1	1	2
16	Разработка базовых спрайтов для игры. Формирование базовых скриптов.			1	1	2
17	Синхронизация работы скриптов для разных спрайтов.			1	1	2
18	Переход из одной сцены в другую. Создание интерфейса игры.			1	1	2
V. Разработка творческого проекта (4 ч)						
19	Разработка и защита творческого проекта			0	4	4
VI. Начало работы в Python (10 ч)						
20	Введение. Знакомство с Python			1	0	1
21	Ввод-вывод данных. Команды print() и input()			1	1	2
22	Необязательные параметры команды print(): sep, end. Переменные. PEP8. Комментарии			1	1	2
23	Работа с целыми числами. Преобразование строки к числу. Основные операции.			1	1	2
24	Работа с целыми числами. Нахождение остатка от деления. Целочисленное деление. Обработка цифр числа.			1	1	2
25	Обобщающее занятие по вводу-выводу данных.			0	1	1
VII. Условный оператор (7ч)						
26	Условный оператор if-else. Отступы. Операторы сравнения.			1	1	2
27	Логические операторы and, or, not			1	1	2
28	Условный оператор. Вложенные условия. Каскадные условия.			1	1	2
29	Обобщающее занятие по условному оператору.			0	1	1
VIII. Типы данных (7ч)						
30	Числовые типы данных. Целочисленный тип данных int. Числа с плавающей точкой float. Встроенные функции max(), min(), abs()			1	1	2

31	Строковый тип данных. Строковый тип данных str. Функции len() и str(). Конкатенация строк. Умножение строки на число. Оператор in.			1	1	2
32	Подключение модулей в Python. Модуль math.			1	1	2
33	Обобщающее занятие по типам данных.			0	1	1
IX. Циклы for и while (15ч)						
34	Цикл for. Переменная цикла.			1	1	2
35	Цикл for. Функции range() с одним, двумя и тремя параметрами.			1	1	2
36	Частые сценарии с использованием цикла for.			1	1	2
37	Цикл while. Считывание данных до стоп значения. Бесконечный цикл.			1	1	2
38	Использование цикла while для обработки цифр числа.			1	1	2
39	Оператор break. Оператор continue. Бесконечные циклы.			1	1	2
40	Вложенные циклы. Операторы break и continue во вложенных циклах.			1	1	2
41	Обобщающее занятие по циклам			0	1	1
X. Строковый тип данных (9ч)						
42	Индексация строк. Итерирование строк.			1	1	2
43	Срезы строк. Изменение символов строки.			1	1	2
44	Методы и функции строк. Конвертация регистра.			1	1	2
45	Методы и функции строк. Поиск и замена.			1	1	2
46	Обобщающее занятие по строковым типам данных			0	1	1
XI. Списки (12ч)						
47	Создание списков. Пустые списки. Встроенная функция list(). Вывод списков.			1	1	2
48	Встроенные функции len(), sum(), min(), max(). Оператор принадлежности in. Индексация и срезы. Конкатенация и умножение на число. Отличие списков от строк.			1	1	2
49	Методы добавления и удаления элементов. Метод добавления элемента append(). Метод расширения списка extend(). Оператор del.			1	1	2

50	Вывод списка с помощью for. Вывод списка с помощью распаковки. Вывод строки с помощью распаковки.			1	1	2
51	Строковые методы split() и join().			1	1	2
52	Обобщающее занятие по спискам			0	1	1
53	Итоговое занятие			1	0	1
ИТОГО:				48	54	102

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Печатные пособия:

1. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch: учебно-методическое пособие - В. Г. Рындак, В. О. Дженжер, Л. В. Денисова. — Оренбург: Оренб. гос. ин-т. менеджмента, 2009. — 116 с.: ил.
2. «Информатика. Математика. Программы внеурочной деятельности для начальной и основной школы: 3-6 класс.» - М.С.Цветкова, О.Б.Богомолова. – М.: БИНОМ. Лаборатория знаний, 2013.
3. Иллюстрированное руководство по языкам Scratch и Python «Программирование для детей»/К. Вордерман, Дж.Вудкок, Ш.Макаманус и др.; пер. с англ.С.Ломакин. – М.: Манн, Иванов и Фербер, 2015.
4. Scratch для юных программистов. Денис Голиков ; - БХВ-Петербург , 2019
5. Васильев, А. Н. Python на примерах [Текст]:практ. курс /А. Н. Васильев - Наука и Техника, 2019 - 432 с.
6. Прохоренок, Н. А. Python 3: самое необходимое [Текст]:практ. курс / Н. А. Прохоренок, В. А. Дронов - БХВ-Петербург, 2019 - 608 с.
7. Гэддис, Т. Начинаем программировать на Python [Текст]: учебник/Т. Гэддис - БХВ-Петербург, 2019 - 768 с.
8. Седжвик, Р. Программирование на языке Python /Р. Седжвик, К. Уэйн, Р. Дондеро - Вильямс, 2017 - 736 с.
9. Харрисон, М. Как устроен Python. [Текст]:практ. курс / М. Харрисон - Питер, 2002 - 272 с.

Электронные ресурсы

1. <http://scratch.mit.edu/pages/source> – страница разработчиков, где выложен код и примеры.
2. <http://scratch.mit.edu> – официальный сайт Scratch
3. <http://letopisi.ru/index.php> /Скретч - Скретч в Летописи.ру
4. <http://setilab.ru/scratch/category/commun> - Учитесь со Scratch
5. Курс Поколение Python: курс для начинающих на платформе Stepik. (<https://stepik.org/course/58852/syllabus>)