

Управление образования администрации
муниципального образования Кандалакшский район
Муниципальное автономное учреждение дополнительного образования
«Детско-юношеский центр «Ровесник»
имени Светланы Алексеевны Крыловой»
муниципального образования Кандалакшский район

ПРИНЯТА
педагогическим советом
от 30.05.2023 г.
Протокол № 6

УТВЕРЖДЕНА
приказом директора
от 31.05.2023 г. № 84
Директор  О.Ю. Савенкова



Дополнительная общеобразовательная общеразвивающая программа
технической направленности
«Начальное программирование компьютерных игр»
Возраст обучающихся: 10-14 лет
Срок реализации программы: 2 года (216 часов)
Уровень сложности: разноуровневая

Автор-составитель:
Забродин П.В.,
педагог дополнительного
образования

г. Кандалакша, 2023

Пояснительная записка

Настоящая дополнительная общеобразовательная общеразвивающая программа «Начальное программирование компьютерных игр» разработана с учетом:

- Федерального Закона Российской Федерации от 29.12.2012 г. № 273 «Об образовании в Российской Федерации»;
- Распоряжения Правительства Российской Федерации от 31.03.2022 г. № 678-р «Концепция развития дополнительного образования детей до 2030 года»;
- Приказа Министерства просвещения Российской Федерации от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Приказа Министерства просвещения Российской Федерации от 03.09.2019 г. № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.09.2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Постановления Главного государственного санитарного врача Российской Федерации от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
- Письма Министерства образования и науки Российской Федерации от 18.11.2015 г. № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»;
- Устава МАУДО ДЮОЦ «Ровесник» им. С.А. Крыловой

Направленность: техническая.

Актуальность программы. Занятия по программе «Начальное программирование компьютерных игр» помогут учащемуся сделать первые шаги в мире программирования, позволят познакомиться с сообществом таких же заинтересованных ребят, введут во все подробности и тонкости проектной деятельности. Овладевая навыками программирования, учащийся затрагивает и смежные сферы: логика, вычислительная математика, теория вероятности, а также и другие научные области: география, биология, физика, литература – в зависимости от интересов и выбора области развития собственного проекта.

Когда у учащегося сформирован необходимый набор знаний и умений, выполнен ряд задач и упражнений по разным темам, он может, используя их,

работать над собственным проектом. Это позволяет развивать творческие способности, проводить собственные исследования, работать в команде, и, что немаловажно, видеть результат собственной работы, вносить в неё коррективы и развивать её.

Отличительной особенностью программы является направленность на заинтересованность обучающихся в области программирования, в приобретении необходимых знаний, в предоставлении свободы в техническом творчестве, чтобы каждый выпускник имел по окончании готовый проект, который он сможет продемонстрировать, и который может стать основой для его дальнейшего развития в сфере программирования. Занимательные формы работы с использованием и компьютерных технологий воспринимаются детьми сначала на уровне игры, затем учащиеся вовлекаются в серьезную творческую работу, в ходе которой развивается личность ребенка.

Новизна программы заключается в комбинировании исследовательской деятельности с изучением основ программирования и создания проекта в программной среде Scratch. Аспект новизны заключается в том, что Scratch не просто язык программирования, а еще и интерактивная среда, где результаты действий визуализированы, что делает работу с программой понятной, интересной и увлекательной.

Особенность среды Scratch, позволяющая создавать в программе мультфильмы, анимацию, мини-игры, делает данную образовательную программу практически значимой для современного школьника, так как дает возможность увидеть практическое назначение алгоритмов и программ, что будет способствовать развитию интереса к профессиям, связанным с программированием.

С целью повышения эффективности образовательного процесса используются дистанционные и мультимедийные технологии. Созданные программы размещаются на Международном портале Scratch, одаренные учащиеся работают над индивидуальными (коллективными) проектами, с которыми участвуют в Международной конференции по программированию.

Уровень программы:

1 год обучения – стартовый,

2 год обучения – базовый.

Возраст обучающихся: 10-14 лет.

Содержание и объем стартовых знаний, необходимых для начального этапа освоения программы: для успешного прохождения программы учащемуся необходимо иметь первичные навыки работы на компьютере, уметь использовать клавиатуру, манипулятор типа «мышь», желательно уметь использовать поисковые системы.

Срок реализации программы: 2 года.

Объем программы:

1 год обучения – 72 часа,

2 год обучения – 144 часа.

Всего: 216 часов.

Режим занятий:

1 год обучения – 1 раз в неделю по 2 академических часа,

2 год обучения – 2 раза в неделю по 2 академических часа.

Продолжительность академического часа - 45 минут. Режим занятий соответствует санитарно-эпидемиологическим требованиям к обеспечению безопасных условий образовательной деятельности (СП 2.4. 3648-20, СанПиН 1.2.3685-21).

Формы организации учебной деятельности: групповая, парная.

Количество учащихся в группе: 12 человек.

Цель программы: способствовать формированию у обучающихся информационной и функциональной компетентности, развитию алгоритмического мышления при помощи работы в свободной среде визуального программирования.

Задачи:

обучающие:

- познакомить с общими идеями создания игровых приложений;
- ознакомить со средой программирования Scratch;
- сформировать представление о средствах разработки;
- научить создавать простейшие компьютерные игры;
- познакомить учащихся с понятиями: определениями алгоритма, цикла, условия, понятием компиляции проекта, подходами к поиску ошибок;
- познакомить учащихся с понятиями: «объект», «событие», «управление», «обработка событий», логика, процедуры;
- сформировать навыки составления алгоритмов;
- изучить функциональность работы основных алгоритмических конструкций;
- сформировать навыки практического программирования при решении поставленных технических задач и реализации творческих проектов в среде Scratch;
- сформировать навыки разработки комплексных проектов: интерактивных историй, мультфильмов, интерактивных презентаций.

- создать представление о проектно-исследовательской деятельности в области ИТ и методах организации творческого процесса при проектировании программных продуктов.

развивающие:

- способствовать расширению словарного запаса;
- способствовать развитию памяти, внимания, технического мышления, изобретательности;
- способствовать развитию алгоритмического мышления;
- способствовать формированию интереса к техническим знаниям;
- способствовать формированию умения практического применения полученных знаний;
- сформировать умение формулировать, аргументировать и отстаивать своё мнение;
- сформировать умение выступать публично с докладами, презентациями.

воспитательные:

- способствовать формированию интереса к освоению опыта познавательной, творческой, исследовательской деятельности;
- формировать умение работать в группах, обмениваться идеями;
- развивать личные качества учащихся
- воспитывать чувство патриотизма, гражданственности, гордости за достижения отечественной науки и техники.

Прогнозируемые результаты

Предметные результаты:

По окончании 1 года обучения обучающиеся должны:

знать:

- виды основных компьютерных игр;
- основы визуального языка программирования Scratch;
- принципы ввода и вывода данных;
- принципы управления внутри игровыми объектами;
- основную терминологию в области алгоритмизации, программирования, разработки компьютерных игр, компьютерной графики.

уметь:

- составить и записать алгоритм для конкретного исполнителя с алгоритмической линейной структурой;
- анализировать игровые приложения;

- пользоваться основной терминологией в области алгоритмизации, программирования, разработки компьютерных игр, компьютерной графики.

По окончании 2 года обучения обучающиеся должны:

знать:

- основные предметные понятия («информация», «алгоритм», «исполнитель», «модель») и их свойства;
- виды основных компьютерных игр;
- основы визуального языка программирования Scratch;
- основные алгоритмические конструкции;
- принципы управления внутри игровыми объектами;
- некоторые принципы разработки игровых программ;
- основную терминологию в области алгоритмизации, программирования, разработки компьютерных игр, компьютерной графики.

уметь:

- составить и записать алгоритм для конкретного исполнителя с алгоритмической линейной, условной и циклической структурой;
- проектировать простые игровые приложения с использованием конструктора компьютерных игр;
- пользоваться основной терминологией в области алгоритмизации, программирования, разработки компьютерных игр, компьютерной графики;
- использовать методы разработки простейших компьютерных игр;
- представлять свой проект.

Личностные результаты:

- 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 иллюстрированной среде программирования, мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам.

Метапредметные результаты:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы
- своей познавательной деятельности;

- умение оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

Формы подведения итогов реализации дополнительной программы: создание проектов, участие в олимпиаде, публикации проектов, тестирование.

Формы демонстрации результатов: обучения на основании тестирований и результатов участия обучающихся в олимпиадах и соревнованиях

**Учебно-тематический план
1 года обучения**

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Введение в компьютерное проектирование	2	1	1	Беседа Тестирование
2.	Знакомство со средой Scratch	8	2	6	Итоговое задание по теме
3.	Линейные алгоритмы. Управление спрайтами	10	2	8	Итоговое задание по теме
4.	Циклические алгоритмы Понятие цикла. Команда повторить. Рисование узоров и орнаментов	14	6	8	Итоговое задание по теме
5.	Алгоритмы ветвления Соблюдение условий. Сенсоры. Блок если	14	6	8	Итоговое задание по теме
6.	Переменные. Их создание. Использование счётчиков	10	2	8	Итоговое задание по теме
7.	Свободное проектирование	10	2	8	Итоговое задание по теме

8.	Итоговое повторение	2	1	1	Итоговое задание по теме
9.	Заключительное занятие	2	0	2	Презентация проекта
	Итого	72	22	50	

Содержание программы 1 года обучения

1. Введение в компьютерное проектирование

Теория (1 час). Введение. Цель, задачи программы. План работы на учебный год. Режим занятий. Первичный инструктаж. Первичный инструктажи по темам: «Правила поведения в ДЮОЦ «Ровесник», «Охрана жизни и здоровья учащихся на учебных занятиях». Свободное программное обеспечение.

Авторы программной среды Scratch.

Практика (1 час). Выполнение тестовой работы

2. Знакомство со средой Scratch

Теория (2 часа). Знакомство со средой Scratch. Понятие спрайта и объекта.

Практика (6 часов). Ознакомление с учебной средой программирования Scratch.

Элементы окна среды программирования. Спрайты. Хранилище спрайтов. Понятие команды. Разновидности команд. Структура и составляющие скриптов - программ, записанных языком Scratch. Понятие анимации. Команды движения и вида.

Анимация движением и изменением вида спрайта. Создание самого простого проекта, его выполнения и сохранения. Хранилище проектов. Создание и редактирование скриптов. Перемещение и удаление спрайтов.

3. Линейные алгоритмы. Управление спрайтами

Теория (2 часа). Использование простейших растровых и векторных редакторов для создания и редактирования изображений; изменение центра изображения; изменения в изображения из встроенной библиотеки; создание сложных графических объектов путем копирования и модификации простых объектов и их фрагментов, использование возможности работы с цветом.

Практика (8 часов). Управление спрайтами: команды идти, повернуться на угол, опустить перо, поднять перо, очистить. Координатная плоскость. Точка отсчёта, оси координат, единица измерения расстояния, абсцисса и ордината. Навигация в среде Scratch. Определение координат спрайта. Команда идти в точку с заданными координатами. Создание проекта «Кругосветное путешествие Магеллана». Команда плыть в точку с заданными координатами. Режим презентации. Управление спрайтами: команды идти, повернуться на угол, опустить перо,

поднять перо, очистить. Изменение костюма спрайта. Понятие операции и выражения. Арифметические операции. Выполнение упражнений в среде программирования Scratch. Выполнение собственных проектов. Скретч арифметические выражения.

4. Циклические алгоритмы Понятие цикла. Команда повторить. Рисование узоров и орнаментов

Теория (6 часов). Составление и отлаживание программного кода; использовать конструкции программной среды для создания линейных разветвленных и циклических алгоритмов; организовывать параллельные вычисления; организовывать последовательность событий программы, передачу управления от одних исполнителей другим. Многократное повторение команд как организация цикла. Особенности использования цикла в программе. Упрощение программы путём сокращения количества команд при переходе от линейных алгоритмов к циклическим. Схематическая запись циклического алгоритма. Типы циклических алгоритмов. Основные конструкции программной среды, используемые для написания программ исполнителям с применением циклов.

Практика (8 часов). Конструкция всегда. Список как упорядоченный набор однотипной информации. Создание списков. Добавление и удаление элементов. Создание проектов в среде программирования Scratch. Создание проектов «Берегись автомобиля!» и «Гонки по вертикали». Команда если край, оттолкнуться. Ориентация по компасу. Управление курсом движения. Команда повернуть в направление. Проект «Полёт самолёта». Анимация. Создание проектов «Осьминог». Анимация. Создание проектов «Девочка, прыгающая на скакалке». Анимация. Создание проектов «Движение планет». Анимация. Создание проектов «Дисотека». Создание мультипликационного сюжета «Кот и птичка». Проект «Хожение по коридору». Проект «Слепой кот». Выполнение собственных проектов.

5. Алгоритмы ветвления Соблюдение условий. Сенсоры. Блок если

Теория (6 часов). Использование ветвления при написании программ. Короткая форма. Полная форма условного оператора. Конструкции ветвления для моделирования. Цикл пока. Повторение команд исполнителя при выполнении определенного условия. Типы исполнителей программной среды Scratch. Системы команд исполнителей. Различные системы команд для разных типов исполнителей. Управление событиями. Передача сообщений исполнителям для выполнения определенной последовательности команд. Передача управления между различными типами исполнителей.

Практика (8 часов). Управляемый стрелками спрайт. Создание проектов в среде программирования Scratch Создание коллекции игр: «Лабиринт». Создание коллекции игр: «Лабиринт», «Кружащийся котёнок». Пополнение коллекции игр:

«Опасный лабиринт». Составные условия. Проект «Тренажёр памяти». Датчик случайных чисел. Проекты «Разноцветный экран», «Хаотичное движение». Проекты «Кошки-мышки», «Вырастим цветник». Циклы с условием. Проект «Будильник». Запуск спрайтов с помощью мыши и клавиатуры. Проект «Переодевалки». Проект «Дюймовочка». Самоуправление спрайтов. Обмен сигналами. Блоки передать сообщение и когда я получу сообщение. Проекты «Лампа» и «Диалог». Датчики. Проекты «Котёнок-обжора», «Презентация».

6. Переменные. Их создание. Использование счётчиков

Теория (2 часа). Управление событиями. Передача сообщений исполнителям для выполнения определенной последовательности команд.

Передача управления между различными типами исполнителей. Цикл пока. Повторение команд исполнителя при выполнении определенного условия.

Типы исполнителей программной среды Scratch. Системы команд исполнителей. Различные системы команд для разных типов исполнителей.

Практика (8 часов). Проект «Голодный кот». Ввод переменных. Проект «Цветы». Доработка проекта «Лабиринт» - запоминание имени лучшего игрока. Ввод переменных с помощью рычажка. Проекты «Цветы» (вариант-2), «Правильные многоугольники». Список как упорядоченный набор однотипной информации. Создание списков. Добавление и удаление элементов. Проект «Гадание». Проект «Назойливый собеседник». Создание игры «Угадай слово». Поиграем со словами. Строковые константы и переменные.

7. Свободное проектирование

Теория (2 часа). Программный код. Конструкции программной среды для создания линейных, разветвленных и циклических алгоритмов. Параллельные вычисления. Последовательность событий программы, передача управления от одних исполнителей другим.

Практика (8 часов). Мультимедийный проект. Описание сюжетных событий. Создание тестов – с выбором ответа и без. Создание проектов по собственному замыслу. Анимация. Создание эффекта анимации с помощью последовательной смены изображений. Имитационные модели. Интерактивные проекты. Игры. Музыкальные возможности. Сказка, иллюстрация к басне. Игровой проект. Скретч-квест.

8. Итоговое повторение

Теория (1 час). Программный код. Использование конструкции программной среды для создания линейных, разветвленных и циклических алгоритмов. Параллельные вычисления. Организация последовательности событий программы, передача управления от одних исполнителей другим.

Практика (1 час). Моделирование ситуации. Интерактивность исполнителей. Мини-проект «Водолей».

Заключительное занятие

Практика (2 часа). Итоговое тестирование. Защита проекта.

Учебно-тематический план 2 года обучения

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Введение в образовательную программу, техника безопасности.	4	2	2	Беседа, участие в работе групп
3.	Кейс 1: «Реклама»	36	8	28	Демонстрация решений кейса
4.	Кейс 2: «Конкурс энтузиастов».	48	8	40	Демонстрация решений кейса
5.	Кейс 3: «ПДД».	56	8	48	Демонстрация решений кейса Тестирование
	Итого	0	0	0	

Содержание программы 2 года обучения

1. Введение в образовательную программу, техника безопасности.

Теория (2 часа). Инструктаж по технике безопасности и безопасному поведению. Организационные вопросы. Цели и задачи объединения. Обсуждение плана работы. Понятие об алгоритмизации и программировании. Правила внутреннего распорядка.

Практика (2 часа). Техника безопасности на занятиях. Организация рабочего места. Знакомство с программами для создания игр. Практическая работа по постановке сцен, созданию спрайтов.

Кейс 1: «Реклама»

Теория (8 часов). Метод координат. Движение спрайта. Повороты на угол. Градусы. Алгоритм. Понятие алгоритма как формального описания последовательности действий исполнителя, приводящих от исходных данных к конечному результату. Схематическая запись алгоритма. Использование геометрических фигур для схематической записи алгоритма. Линейные алгоритмы.

Основные признаки линейного алгоритма. Схематическое описание линейного алгоритма. Геометрические примитивы, используемые для описания линейного алгоритма.

Практика (28 часов). Программное управление исполнителем. Создание программ для перемещения исполнителя по экранному полю. Понятие «поворот исполнителя в определённое направление». Угол поворота 90° . Поворот исполнителя на 90° по часовой или против часовой стрелки. Создание программ для рисования линий. Изменение цвета и толщины рисуемой линии. Особенности пунктирной линии. Программа для исполнителя, чтобы он оставлял пунктирную линию при перемещении по экранному полю. Прямоугольник, квадрат – основные черты. Написание программ для движения исполнителя вдоль сторон квадрата, прямоугольника. Редактирование программы рисования квадрата для получения квадрата с другим размером стороны. Прерывание программы. Циклические алгоритмы. Организация циклического алгоритма при необходимости многократного повторения команд. Особенности использования цикла в программе. Оптимизация программы путём сокращения количества команд при переходе от линейных алгоритмов к циклическим. Схематическая запись циклического алгоритма. Типы циклических алгоритмов. Основные конструкции программной среды, используемые для написания программ исполнителем с применением циклов. Конечный цикл. Оптимизация программы для исполнителя, рисующего линии, квадраты, прямоугольники при использовании цикла. Программа исполнителя для рисования нескольких однотипных геометрических фигур, например, нескольких квадратов из одной вершины, но с различным значением стороны. Конструкции программной среды «спрятаться/показаться». Выполнение программы исполнителем, не показанным в поле выполнения программы. Написание и отладка программ с применением конструкции «цикл в цикле». Бесконечный цикл. Повторяющаяся смена внешности исполнителя для имитации движения персонажа. Использование бесконечного цикла для создания анимации. Получение различных эффектов воспроизведения программы за счёт изменения костюмов исполнителей Scratch.

Выполнение практических заданий: движение спрайта, повороты на углы, движения по координатной сетке, линейный алгоритм, алгоритм ветвление, циклический алгоритм, передача сообщений. Работа над решением кейса.

2. Кейс 2: «Конкурс энтузиастов»

Теория (8 часов). Компьютерная графика. Векторные и растровые графические редакторы. Встроенный растровый графический редактор. Основные инструменты графического редактора – кисточка, ластик, заливка (цветом или градиентом), рисование линий, прямоугольников, квадратов, эллипсов и окружностей, выбор фрагмента изображения и отражение его по горизонтали или вертикали,

использование инструмента «Печать» для копирования выделенной области изображения, работа с текстом. Масштаб фрагмента изображения. Палитра цветов, установка цвета переднего плана и фона, выбор цвета изображения с помощью инструмента «Пипетка». Изменение центра костюма. Изменение размера костюма. Кнопочное управление. Джойстик. Счетчик. Соприкосновение спрайтов. Условия срабатывания счетчика.

Практика (40 часов). Создание растрового и векторного рисунка. Создание и обработка спрайтов. Работа с текстом, графическими примитивами применение их в создании фонов. Создание презентации. Создание анимации. Создание мультфильма.

Практические задания. Работа над решением кейса.

3. Кейс 3: «ПДД»

Теория (8 часов). Параллелизм в программной среде. Использование нескольких исполнителей. Копирование программ исполнителей. Выполнение одинаковых программ разными исполнителями при использовании различных начальных условий. Параллельное выполнение однотипных действий. Принцип суперкомпьютерных технологий. Таймер для вычисления времени выполнения программы. Интерактивность программ. Возможность организации диалога между исполнителями. Операторы для слияния текстовых выражений. Взаимодействие исполнителей. Сенсоры касания объектов или цвета. Использование сенсоров при взаимодействии исполнителей. Задержка выполнения программы. Слои изображения. Выполнение команд исполнителями в разных слоях изображения. Ветвление в алгоритмах. Использование ветвления при написании программ. Короткая форма. Полная форма условного оператора. Конструкции ветвления для моделирования ситуации. Цикл «пока». Повторение команд исполнителя при выполнении определенного условия. Последовательное выполнение фрагментов программы разными исполнителями. Типы исполнителей программной среды Scratch. Системы команд исполнителей. Различные системы команд для разных типов исполнителей. Имитационные модели.

Практика (48 часов). Управление событиями. Передача сообщений исполнителям для выполнения определенной последовательности команд. Описание сюжетных событий. Анимация. Создание эффекта анимации с помощью последовательной смены изображений. Практические задания. Работа над решением кейса.

Комплекс организационно-педагогических условий

Календарный учебный график (Приложение 1).

Материально-техническое обеспечение:

Аппаратное обеспечение:

- персональные компьютеры (ноутбуки) –12 шт.;

- проектор;
- экран;
- доступ к сети Интернет.

Программное обеспечение:

- операционная система Windows 10 professional;
- браузер google chrome;
- доступ к сайту scratch.mit.edu;
- программа MO PowerPoint.

Информационно-методическое обеспечение

Дидактический материал (раздаточный материал по темам занятий программы, наглядный материал, мультимедийные презентации, технологические карты). Медиатека (познавательные игры, музыка, энциклопедии, видео). Видеоуроки. Методические разработки занятий.

Методы организации учебного процесса:

Информационно-рецептивный метод – предъявление педагогом информации и организация восприятия, осознания и запоминание учащимися данной информации.

Репродуктивный метод – составление и предъявление педагогом заданий на воспроизведение знаний и способов умственной и практической деятельности, руководство и контроль за выполнением; воспроизведение воспитанниками знаний и способов действий по образцам, произвольное и непроизвольное запоминание.

Метод проблемного изложения – постановка педагогом проблемы и раскрытие доказательно пути его решения; восприятие и осознание учащимися знаний, мысленное прогнозирование, запоминание.

Эвристический метод – постановка педагогом проблемы, планирование и руководство деятельности учащихся; самостоятельное решение учащимися части задания, непроизвольное запоминание и воспроизведение.

Исследовательский метод – составление и предъявление педагогом проблемных задач и контроль за ходом решения; самостоятельное планирование учащимися этапов, способ исследования, самоконтроль, непроизвольное запоминание.

В организации учебной познавательной деятельности используются также словесные, наглядные и практические методы.

Словесные методы. Словесные методы педагог применяет тогда, когда главным источником усвоения знаний учащимися является слово (без опоры на наглядные способы и практическую работу). К ним относятся: рассказ, опрос, объяснение и т.д.

Наглядные методы. К ним относятся методы обучения с использованием наглядных пособий.

Практические методы. Методы, связанные с процессом формирования и совершенствования умений и навыков учащихся. Основным методом является практическое занятие.

Дидактические средства. В ходе реализации образовательной программы педагогом используются дидактические средства: учебные наглядные пособия, демонстрационные устройства, технические средства.

Формирование и совершенствование умений и навыков (изучение нового материала, практика).

Познавательный – восприятие, осмысление и запоминание учащимися нового материала с привлечением наблюдения готовых примеров, моделирования, изучения иллюстраций, восприятия, анализа и обобщения демонстрируемых материалов.

Метод проектов – при усвоении и творческом применении навыков и умений в процессе разработки собственных моделей).

Систематизирующий метод – опрос по теме, составление систематизирующих таблиц, графиков, схем и т.д.

Контрольный метод – при выявлении качества усвоения знаний, навыков и умений и их коррекция в процессе выполнения практических заданий.

Рефлексия – возможность обдумать то, что учащиеся запрограммировали, помогает им более глубоко понять идеи, с которыми они сталкиваются в процессе своей деятельности на предыдущих этапах. Размышляя, учащиеся устанавливают связи между полученной ими новой информацией и уже знакомыми им идеями, а также предыдущим опытом.

Творческие задачи, представляющие собой адекватный вызов способностям ребёнка, наилучшим образом способствуют его дальнейшему обучению и развитию. Радость свершения, атмосфера успеха, ощущение хорошо выполненного дела – всё это вызывает желание продолжать и совершенствовать свою работу.

Формы обучения:

- проектная деятельность самостоятельная работа;
- работа в парах, в группах;
- творческие работы;
- индивидуальная и групповая исследовательская работа;
- знакомство с научно-популярной литературой.

Формы подведения итогов: промежуточные проекты, тестирования, участие в олимпиадах.

Формы отслеживания и фиксации результатов

В течение учебного года для определения уровня усвоения программы учащимися осуществляются диагностические срезы:

– входная диагностика – тестирование, где выясняется стартовый уровень ЗУН учащегося.

– промежуточная диагностика позволяет выявить достигнутый на данном этапе уровень ЗУН учащихся, в соответствии с пройденным материалом программы. Предлагается тестирование, а также учитывается участие в соревнованиях и проектная деятельность учащихся.

– итоговая диагностика проводится в конце учебного года (демонстрация и публикация проектов) и предполагает комплексную проверку образовательных результатов в виде теста по всем ключевым направлениям, а также учитывается участие в соревнованиях и проектная деятельность учащихся. Данный контроль позволяет проанализировать степень усвоения программы учащимися.

Итоговые и промежуточные результаты контроля фиксируются в диагностической карте (Приложение 3).

Виды контроля

Виды контроля	Содержание	Методы	Сроки контроля
Входной	Начальный уровень подготовки учащихся, имеющиеся знания, умения и навыки, связанные с предстоящей деятельностью.	Тестирование	Сентябрь
Промежуточные	Выполнение итоговых практических заданий	Практическая работа	По мере изучения разделов программы
Итоговый	Проектная деятельность Освоение учебного материала за учебный год, предполагает комплексную проверку образовательных результатов по всем ключевым направлениям	Тестирование, защита проекта	Май

Оценка уровней освоения программы

Уровни /%	Параметры	Показатели
Высокий уровень	Теоретические знания	Оценка теоретических знаний на основе тестирования. Учащийся освоил материал в полном объеме. Знает и понимает значение терминов, самостоятельно ориентируется в содержании материала по темам.
	Практические умения	Способен свободно применять в практической работе полученные знания. Учащийся проявляет устойчивое внимание к выполнению заданий, сосредоточен во время практической работы, получает результат своевременно. Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.
	Навыки ведения проектной деятельности	Обучающийся хорошо работает со всеми членами команды. Всегда справляется с поставленной задачей в группе. Свободно генерирует идеи. Легко применяет полученные знания и умения в решении поставленной задачи.
Средний уровень	Теоретические знания	Оценка теоретических знаний на основе тестирования. Обучающийся освоил базовые знания, но слабо ориентируется в содержании материала по некоторым темам.
	Практические умения	Владеет базовыми навыками и умениями, но не всегда может в полном объеме выполнить практическое самостоятельное задание, затрудняется и просит помощи педагога. В работе допускает небрежность, делает ошибки, но может устранить их после наводящих вопросов или самостоятельно. Оценить результаты своей деятельности может с подсказкой педагога. Обучающийся заинтересован, но не всегда проявляет устойчивое внимание к

		выполнению задания.
	Навыки ведения проектной деятельности	Обучающийся слабо сосредоточен во время работы в группе, не всегда умеет находить общий язык с членами команды. Справляется с поставленной задачей в группе, но просит помощи и подсказки педагога. Не всегда умеет генерировать идеи. Применяет полученные знания и умения в решении поставленной задачи, но с некоторыми подсказками педагога или товарищей.
Низкий уровень	Теоретические знания	Оценка теоретических знаний на основе тестирования. Владеет минимальными знаниями, слабо ориентируется в содержании материала.
	Практические умения	Обучающийся способен выполнять каждую операцию практической работы только с подсказкой педагога или товарищей. Не всегда правильно применяет в практической работе необходимые знания или не использует вовсе. В работе допускает грубые ошибки, не может их найти даже после указания. Не способен самостоятельно оценить результаты своей работы.
	Навыки ведения проектной деятельности	Обучающийся слабо контактирует в работе с членами команды. Не умеет генерировать идеи. Не всегда умеет справиться с поставленной задачей в группе. Решение задачи происходит исключительно с подсказкой педагога. Слабо применяет полученные знания и умения в решении поставленной задачи, исключительно с подсказками педагога или товарищей.

Список литературы для педагога

1. Великович Л., Цветкова М. Программирование для начинающих. – Москва: Бинум, 2007

2. Голиков Д. В. Scratch для юных программистов. — СПб.: БХВ-Петербург, 2017. - 192 с.: ил.
3. Окулов С. М. Основы программирования. - Москва: Лаборатория Базовых Знаний, 2002.- 424 с.
4. Патаракин Е.П. Учимся готовить в среде Скретч. Версия 2.0
5. Полат Е.С. Новые педагогические и информационные технологии в системе образования. – Москва: Академия. – 2006.
6. Рындак, В.Г. «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.
7. Шпынева С. М. Методическое пособие Технологии Scratch. – Тамбов, 2014. – 29с: ил.
8. Scratch | Home | imagine, program, share [сайт]. URL: <http://scratch.mit.edu>
Scratch | Галерея | Gymnasium №3 [сайт]. URL: <http://scratch.mit.edu/galleries/view/54042>

Литература для обучающихся и родителей

1. Беляев М.А. и др. «Основы информатики». URL: <http://www.vuithelp.ru/files/538.html>
2. Голиков, Д.В. Книга юных программистов на Scratch: учебное пособие/Д.В. Голиков, А.Д. Голиков. – Москва: ВHV, 2017 – 192 с.
3. Голиков Д., Голиков А. Программирование на Scratch 2. Часть 1. Делаем игры и мультики. Подробное пошаговое руководство для самостоятельного изучения ребёнком. — Scratch4russia.com, 2014.
4. Голиков Д., Голиков А. Программирование на Scratch 2. Часть 2. Делаем сложные игры. Подробное пошаговое руководство для самостоятельного изучения ребёнком. — Scratch4russia.com, 2014.
5. Олейникова, О. Н. Социальное партнерство в профессиональном образовании [Электронный ресурс] / О. Н. Олейникова. - URL: <http://www.cvets.ru/SPart/SocPartnership.pdf>
6. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). Москва : Интуит.ру, 2008.
7. Скретч [Электронный ресурс] // Материал с Wiki-ресурса Letopisi.Ru — «Время вернуться домой». URL: <http://letopisi.ru/index.php/Скретч>.

Сетевые ресурсы

1. <http://scratch.mit.edu> – сетевое сообщество для обмена проектами, сайт, где можно скачать свободно распространяемую программу Scratch

2. <http://scratch.mit.edu> - официальный сайт проекта Scratch
3. Trening.self-promo.ru (Дистанционный курс в поддержку ОП «Школа программирования»);
4. Scratch.mit.edu — официальный сайт Скретч (язык программирования)
5. ScratchED (англ.). Гарвардский университет. — Глобальное сообщество учителей Скретч

**«Начальное программирование компьютерных игр»
1 год обучения**

Педагог д/о:

Кол-во часов: 72

Режим проведения занятий: 1 раз в неделю по 2 академических часа.

№	Дата	Время	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
1.			Теория/практика	2	Инструктаж по технике безопасности. Введение в компьютерное проектирование		Тестирование. Опрос
2.			Теория/практика	2	Знакомство со средой Скретч. Понятие спрайта и объекта. Хранилище спрайтов. Понятие команды. Разновидности команд.		Практическая работа
3.			Теория/практика	2	Структура и составляющие скриптов - программ, записанных языком Скретч. Понятие анимации. Команды движения и вида. Анимация движением и изменением вида спрайта		Практическая работа
4.			Практика	2	Создание самого простого проекта, его выполнения и сохранения. Перемещение и удаление спрайтов.		Практическая работа
5.			Теория/практика	2	Линейные алгоритмы. Управление спрайтами. Управление спрайтами: команды идти, повернуться на угол, опустить перо, поднять перо, очистить. Создание текстовой надписи		Практическая работа

6.		Теория/практика	2	Управление спрайтами: команды идти, повернуться на угол, опустить перо, поднять перо, очистить. Создание проекта «Аквариум»		Практическая работа
7.		Практика	2	Управление спрайтами: команды идти, повернуться на угол, опустить перо, поднять перо, очистить. Изменение костюма спрайта		Практическая работа
8.		Практика	2	Создание проекта «Кругосветное путешествие Магеллана»		Практическая работа
9.		Практика	2	Арифметические операции. Выполнение упражнений в среде программирования Scratch. Выполнение собственных проектов. Скретч арифметические выражения		Практическая работа
10.		Теория/ практика	2	Составление и отлаживание программного кода; использовать конструкции программной среды для создания линейных разветвленных и циклических алгоритмов		Практическая работа
11.		Теория/ практика	2	Упрощение программы путём сокращения количества команд при переходе от линейных алгоритмов к циклическим. Схематическая запись циклического алгоритма. Типы циклических алгоритмов. Основные конструкции программной среды, используемые для написания программ исполнителям с применением циклов.		Практическая работа
12.		Теория/ практика	2	Создание проектов «Берегись автомобиля!» и «Гонки по вертикали»		Практическая работа
13.		Теория/ практика	2	Проекты «Полёт самолёта», «Осьминог»		Создание проекта
14.		Теория/ практика	2	Проекты «Девочка, прыгающая на скакалке». «Движение планет»		Практическая работа

15.		Теория/ практика	2	Создание мультипликационного сюжета «Кот и птичка»		Практическая работа
16.		Практика	2	Проект «Хождение по коридору». Проект «Слепой кот». Выполнение собственных проектов.		Практическая работа
17.		Теория/ практика	2	Использование ветвления при написании программ. Типы исполнителей программной среды Scratch		Практическая работа
18.		Теория/ практика	2	Управляемый стрелками спрайт. Создание проектов в среде программирования Scratch Создание коллекции игр: «Лабиринт». Создание коллекции игр: «Лабиринт», «Кружащийся котёнок».		Практическая работа
19.		Теория/ практика	2	Пополнение коллекции игр: «Опасный лабиринт». Составные условия. Проект «Тренажёр памяти».		Практическая работа
20.		Теория/ практика	2	Датчик случайных чисел. Проекты «Разноцветный экран», «Хаотичное движение».		Практическая работа
21.		Теория/ практика	2	Проекты «Кошки-мышки», «Вырастим цветник». Циклы с условием. Проект «Будильник».		Практическая работа
22.		Теория/ практика	2	Запуск спрайтов с помощью мыши и клавиатуры. Проект «Переодевалки». Проект «Дюймовочка».		Практическое занятие
23.		Практика	2	Самоуправление спрайтов. Обмен сигналами. Блоки передать сообщение и когда я получу сообщение. Проекты «Лампа» и «Диалог». Датчики. Проекты «Котёнок-обжора», «Презентация».		Создание проекта
24.		Теория/ практика	2	Управление событиями. Передача сообщений		Создание проекта

			практика		исполнителям для выполнения определенной последовательности команд. Проект «Голодный кот». Ввод переменных.		
25.			Теория/ практика	2	Проект «Цветы». Доработка проекта «Лабиринт» - запоминание имени лучшего игрока. Ввод переменных с помощью рычажка.		Создание проекта
26.			Практика	2	Проекты «Цветы» (вариант-2), «Правильные многоугольники».		Создание проекта
27.			Практика	2	Список как упорядоченный набор однотипной информации. Создание списков. Добавление и удаление элементов. Проект «Гадание».		Создание проекта
28.			Практика	2	Проект «Назойливый собеседник». Создание игры «Угадай слово». Поиграем со словами. Строковые константы и переменные.		Создание проекта
29.			Теория/ практика	2	Программный код. Конструкции программной среды для создания линейных, разветвленных и циклических алгоритмов. Параллельные вычисления.		Создание проекта
30.			Теория/ практика	2	Мультимедийный проект. Описание сюжетных событий. Создание тестов – с выбором ответа и без.		Создание проекта
31.			Практика	2	Создание проектов по собственному замыслу. Анимация. Создание эффекта анимации с помощью последовательной смены изображений.		Создание проекта
32.			Практика	2	Имитационные модели. Интерактивные проекты		Создание проекта
33.			Практика	2	Игры. Музыкальные возможности. Сказка, иллюстрация к басне. Игровой проект. Скретч-квест.		Создание проекта

34.		Теория/ практика	2	Программный код. Использование конструкции программной среды для создания линейных, разветвленных и циклических алгоритмов.		Создание проекта
35.		Практика	2	Параллельные вычисления. Мини-проект «Водолей».		Создание проекта
36.		Практика	2	Итоговое тестирование. Защита проекта.		Создание проекта

Календарный учебный график
«Начальное программирование компьютерных игр»
2 год обучения

Педагог д/о:

Кол-во часов: 144

Режим проведения занятий: 2 раза в неделю по 2 академических часа.

№	Дата	Время	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
37.			Теория/ практика	2	Введение в образовательную программу, техника безопасности		Беседа
38.			Теория/ практика	2	Знакомство с программами для создания игр. Практическая работа по постановке сцен, созданию спрайтов.		Беседа, практическая работа
39.				2	Кейс 1: «Реклама». Метод координат. Движение спрайта. Повороты на угол. Градусы. Алгоритм.		практическая работа
40.				2	Кейс 1: «Реклама». Понятие алгоритма как формального описания последовательности действий исполнителя, приводящих от исходных данных к конечному результату.		практическая работа
41.				2	Кейс 1: «Реклама». Схематическая запись алгоритма. Использование геометрических фигур для схематической записи алгоритма. Линейные алгоритмы. Основные		практическая работа

					признаки линейного алгоритма. Схематическое описание линейного алгоритма.		
42.				2	Кейс 1: «Реклама». Геометрические примитивы, используемые для описания линейного алгоритма.		практическая работа
43.				2	Кейс 1: «Реклама». Программное управление исполнителем. Создание программ для перемещения исполнителя по экранному полю. Понятие «поворот исполнителя в определённое направление». Угол поворота 90° . Поворот исполнителя на 90° по часовой или против часовой стрелки.		практическая работа
44.				2	Кейс 1: «Реклама». Создание программ для рисования линий. Изменение цвета и толщины рисуемой линии. Особенности пунктирной линии. Программа для исполнителя, чтобы он оставлял пунктирную линию при перемещении по экранному полю.		практическая работа
45.				2	Кейс 1: «Реклама». Прямоугольник, квадрат – основные черты. Написание программ для движения исполнителя вдоль сторон квадрата, прямоугольника. Редактирование		практическая работа

					программы рисования квадрата для получения квадрата с другим размером стороны.		
46.				2	Кейс 1: «Реклама». Прерывание программы. Циклические алгоритмы. Организация циклического алгоритма при необходимости многократного повторения команд		Тест
47.				2	Кейс 1: «Реклама». Особенности использования цикла в программе. Оптимизация программы путём сокращения количества команд при переходе от линейных алгоритмов к циклическим. Схематическая запись циклического алгоритма. Типы циклических алгоритмов. Основные конструкции программной среды, используемые для написания программ исполнителям с применением циклов. Конечный цикл		Демонстрация решений кейса
48.				2	Кейс 1: «Реклама». Оптимизация программы для исполнителя, рисующего линии, квадраты, прямоугольники при использовании цикла. Программа исполнителя для рисования нескольких однотипных геометрических фигур, например,		практическая работа

					нескольких квадратов из одной вершины, но с различным значением стороны.		
49.			2	Кейс 1: «Реклама». Конструкции программной среды «спрятаться/показаться». Выполнение программы исполнителем, не показанным в поле выполнения программы..			практическая работа
50.			2	Кейс 1: «Реклама». Написание и отладка программ с применением конструкции «цикл в цикле». Бесконечный цикл.			практическая работа
51.			2	Кейс 1: «Реклама». Повторяющаяся смена внешности исполнителя для имитации движения персонажа. Использование бесконечного цикла для создания анимации.			практическая работа
52.			2	Кейс 1: «Реклама». Получение различных эффектов воспроизведения программы за счёт изменения костюмов исполнителей Scratch.			практическая работа
53.			2	Кейс 1: «Реклама». Выполнение практических заданий: движение спрайта, повороты на углы, движения по координатной сетке.			практическая работа
54.			2	Кейс 1: «Реклама». Выполнение			практическая

					практических заданий: линейный алгоритм.		работа
55.			2	Кейс 1: «Реклама» Выполнение практических заданий: алгоритм ветвление			практическая работа
56.			2	Кейс 1: «Реклама» Выполнение практических заданий: циклический алгоритм, передача сообщений.			практическая работа
57.			2	Кейс 2: «Конкурс энтузиастов». Компьютерная графика. Векторные и растровые графические редакторы.			практическая работа
58.			2	Кейс 2: «Конкурс энтузиастов». Встроенный растровый графический редактор. Основные инструменты графического редактора – кисточка, ластик, заливка (цветом или градиентом), рисование линий, прямоугольников, квадратов, эллипсов и окружностей, выбор фрагмента изображения и отражение его по горизонтали или вертикали, использование инструмента «Печать» для копирования выделенной области изображения, работа с текстом.			практическая работа
59.			2	Кейс 2: «Конкурс энтузиастов». Масштаб фрагмента изображения. Палитра цветов, установка цвета			практическая работа

					переднего плана и фона, выбор цвета изображения с помощью инструмента «Пипетка». Изменение центра костюма.		
60.			2	Кейс 2: «Конкурс энтузиастов». Изменение размера костюма. Кнопочное управление. Джойстик.		практическая работа	
61.			2	Кейс 2: «Конкурс энтузиастов». Соприкосновение спрайтов. Условия срабатывания счетчика.		практическая работа	
62.			2	Кейс 2: «Конкурс энтузиастов». Создание растрового и векторного рисунка		практическая работа	
63.			2	Кейс 2: «Конкурс энтузиастов». Создание растрового и векторного рисунка		практическая работа	
64.			2	Кейс 2: «Конкурс энтузиастов» Создание растрового и векторного рисунка.		практическая работа	
65.			2	Кейс 2: «Конкурс энтузиастов». Создание и обработка спрайтов.		практическая работа	
66.			2	Кейс 2: «Конкурс энтузиастов». Работа с текстом, графическими примитивами применение их в создании фонов.		практическая работа	
67.			2	Кейс 2: «Конкурс энтузиастов» Создание презентации		Тест Промежуточная диагностика	

68.			2	Кейс 2: «Конкурс энтузиастов» Создание презентации		Демонстрация решений кейса
69.			2	Кейс 2: «Конкурс энтузиастов» Создание презентации		практическая работа
70.			2	Кейс 2: «Конкурс энтузиастов» Создание анимации.		практическая работа
71.			2	Кейс 2: «Конкурс энтузиастов» Создание анимации.		практическая работа
72.			2	Кейс 2: «Конкурс энтузиастов» Создание мультфильма		практическая работа
73.			2	Кейс 2: «Конкурс энтузиастов» Создание мультфильма		практическая работа
74.			2	Кейс 2: «Конкурс энтузиастов» Создание мультфильма		практическая работа
75.			2	Кейс 2: «Конкурс энтузиастов» Создание мультфильма		практическая работа
76.			2	Кейс 2: «Конкурс энтузиастов» Создание мультфильма		практическая работа
77.			2	Кейс 2: «Конкурс энтузиастов»		практическая работа
78.			2	Кейс 2: «Конкурс энтузиастов» Создание мультфильма		практическая работа
79.			2	Кейс 2: «Конкурс энтузиастов» Создание мультфильма		практическая работа
80.			2	Кейс 2: «Конкурс энтузиастов» Создание мультфильма		практическая работа
81.			2	Кейс 3: «ПДД».		практическая

					Параллелизм в программной среде. Использование нескольких исполнителей. Копирование программ исполнителей.		работа
82.				2	Кейс 3: «ПДД». Выполнение одинаковых программ разными исполнителями при использовании различных начальных условий. Параллельное выполнение однотипных действий.		практическая работа
83.				2	Кейс 3: «ПДД». Принцип суперкомпьютерных технологий.		практическая работа
84.				2	Кейс 3: «ПДД». Таймер для вычисления времени выполнения программы.		практическая работа
85.				2	Кейс 3: «ПДД». Интерактивность программ. Возможность организации диалога между исполнителями.		практическая работа
86.				2	Кейс 3: «ПДД». Операторы для слияния текстовых выражений		практическая работа
87.				2	Кейс 3: «ПДД». Взаимодействие исполнителей.		практическая работа
88.				2	Кейс 3: «ПДД». Сенсоры касания объектов или цвета. Использование сенсоров при взаимодействии исполнителей.		практическая работа
89.				2	Кейс 3: «ПДД». Задержка выполнения программы.		практическая работа

90.			2	Кейс 3: «ПДД». Слои изображения. Выполнение команд исполнителями в разных слоях изображения.		практическая работа
91.			2	Кейс 3: «ПДД». Ветвление в алгоритмах. Использование ветвления при написании программ.		практическая работа
92.			2	Кейс 3: «ПДД». Короткая форма.		Итоговый тест
93.			2	Кейс 3: «ПДД». Полная форма условного оператора.		Демонстрация решений кейса
94.			2	Кейс 3: «ПДД» Конструкции ветвления для моделирования ситуации.		практическая работа
95.			2	Кейс 3: «ПДД» Цикл «пока».		практическая работа
96.			2	Кейс 3: «ПДД». Повторение команд исполнителя при выполнении определенного условия		практическая работа
97.			2	Кейс 3: «ПДД». Последовательное выполнение фрагментов программы разными исполнителями		практическая работа
98.			2	Кейс 3: «ПДД» Типы исполнителей программной среды Scratch		практическая работа
99.			2	Кейс 3: «ПДД». Системы команд исполнителей.		практическая работа
100.			2	Кейс 3: «ПДД». Различные системы команд для разных типов исполнителей.		практическая работа
101.			2	Кейс 3: «ПДД». Имитационные модели.		практическая работа

102.			2	Кейс 3: «ПДД». Управление событиями. Передача сообщений исполнителям для выполнения определенной последовательности команд.		практическая работа
103.			2	Кейс 3: «ПДД». Описание сюжетных событий.		практическая работа
104.			2	Кейс 3: «ПДД». Анимация. Создание эффекта анимации с помощью последовательной смены изображений		практическая работа
105.			2	Кейс 3: «ПДД». Создание эффекта анимации с помощью последовательной смены изображений.		практическая работа
106.			2	Кейс 3: «ПДД». Создание эффекта анимации с помощью последовательной смены изображений.		практическая работа
107.			2	Кейс 3: «ПДД». Защита проекта.		практическая работа
108.			2	Итоговое занятие		Итоговый тест, защита проекта

Входная диагностика
Тестирование

Ф.И. _____

1. Что такое Scratch?
 - a. Сообщество, в котором собираются люди и обсуждают свои проблемы.
 - b. Язык программирования
 - c. Программа по созданию приложений на телефон.
 - d. Затрудняюсь ответить.
2. Что такое спрайт?
 - a. Фирма, логотип, работодатель.
 - b. Напиток.
 - c. Действующее лицо в программе Scratch.
 - d. Затрудняюсь ответить.
3. Для чего нужны языки программирования?
 - a. Для того, чтобы компьютер понимал и выполнял команды человека.
 - b. Для того, чтобы компьютер работал сам как ему нравится без участия человека.
 - c. Для того, чтобы у человека была работа.
 - d. Затрудняюсь ответить.
4. Кто такой программист?
 - a. Человек, который придумывает и собирает робота.
 - b. Человек, который следит за роботом.
 - c. Человек, который пишет алгоритм (шаги) для деятельности робота.
 - d. Затрудняюсь ответить.
5. Кто такой инженер?
 - a. Человек, который придумывает и собирает робота.
 - b. Человек, который следит за роботом.
 - c. Человек, который пишет алгоритм (шаги) для деятельности робота.
 - d. Затрудняюсь ответить.

Промежуточная диагностика

Тестирование

Ф.И. _____

1. Минимальная смысловая единица (команда) языка Скретч, которая служит для создания скриптов (сценариев) - это

- a. скрипт
- b. блок
- c. алгоритм
- d. действие
- e. стек

2. Алгоритм (или сценарий), составленный из блоков языка Скретч для какого-либо объекта - это

- скрипт
- блок
- алгоритм
- действие
- стек

3. Подвижный графический объект, который действует на сцене проекта, выполняя разнообразные алгоритмы (сценарии). Исполнитель алгоритмов, которому доступны ВСЕ команды языка Скретч - это

- скрипт
- спрайт
- сцена
- интерфейс
- стек

4. Это неподвижный графический объект, который изображает место действия проекта. Является исполнителем алгоритмов, которому доступны почти все команды Скретч, кроме команд движения и рисования.

- скрипт
- спрайт
- сцена
- интерфейс
- стек

5. Понятная компьютеру система символов для точной записи алгоритмов и их выполнения компьютером.

интерфейс
язык программирования
скрипт
блок-схемы
сценарий

6. Алгоритм, выраженный на языке программирования.

интерфейс
язык
программирования програм
ма
блок-схемы
сценарий

Итоговая диагностика

Тестирование

Ф.И. _____

1. Форма блоков, которые служат для остановки выполнения одного скрипта или всех скриптов проекта. В Скретче всего 2 блока этого вида.

стек
заголовок
заглушка логический
С-блок

2. Самая многочисленная форма блоков, из которых создаются стопки (предложения) - последовательности блоков в скриптах.

стек
заголовок
заглушка
логический
С-блок

3. Форма блоков, которые содержат логическое значение (истина-ложь), вставляются в шестиугольные и прямоугольные окна других блоков, позволяют создавать логические условия для ветвлений и циклов.

стек
заголовок
заглушка
логический
С-блок

4. Форма блоков, которые содержат данные и вставляются в прямоугольные и овальные окна других блоков.

стек
заголовок
заглушка
логический
С-блок

5. Форма блоков, которые всегда стоят в начале скриптов. Указывают на способ, событие запуска скрипта.

стек
заголовок
заглушка
логический
С-блок

6. Форма блоков, которые служат для создания циклов и ветвлений.

стек
заголовок
заглушка
логический
С-блок

