

Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа № 3 с. Чермен»

«Рассмотрено»
На заседании МО
Протокол № 1
От «30» 08 2024г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Информатика»

направление: техническое

для 5-7 классов

срок реализации: 2024-2025 учебный год (68 часов)

Составитель:
Газдиева Д.М

педагог дополнительного образования

с. Чермен 2024 г.

СОДЕРЖАНИЕ

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ..	4
1.1. Пояснительная записка.....	4
1.2. Цель и задачи программы.....	6
1.3. Содержание программы.....	7
1.3.1. Учебно-тематический план.....	7
1.3.2. Содержание учебно-тематического плана.....	9
1.4. Планируемые результаты.....	17
РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ.....	19
2.1. Календарный учебный график.....	19
2.2. Условия реализации программы.....	21
2.3. Формы аттестации / контроля.....	21
2.4. Методические материалы.....	22
2.5. Список литературы.....	22
2.6. Приложения.....	24

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

1.1. Пояснительная записка

Дополнительная общеразвивающая программа «Информатика» имеет техническую направленность. Программа курса предполагает знакомство с основными понятиями, используемыми в языках программирования высокого уровня, решение большого количества творческих задач, многие из которых моделируют процессы и явления из таких предметных областей, как информатика, алгебра, геометрия, география, физика, русский язык и др. Большинство заданий встречаются в разных темах для того, чтобы показать возможности решения одной и той же задачи или проблемы различными средствами, обеспечивающими достижение требуемого результата, что в итоге приведет к способности выбирать оптимальное решение данной задачи или проблемы.

Программа разработана согласно нормативным и правовым документам:

- Закон Российской Федерации «Об образовании» (Федеральный закон от 29 декабря 2012 г. №273-ФЗ);
- Государственная программа РФ «Развитие образования» на 2018 – 2025 гг. (постановление Правительства Российской Федерации от 26 декабря 2017 г. № 1642);
- Концепция развития дополнительного образования детей (Распоряжение Правительства РФ от 31 марта 2022 г. № 678-Р);
- Приказ Министерства просвещения РФ от 9 ноября 2018 г. № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Письмо Министерства образования и науки РФ от 18.11.2015 № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы);

- Письмо Минобрнауки РФ от 11.12.2006 N 06-1844 «О примерных требованиях к программам дополнительного образования детей»;
- Постановление 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»;
- Устав и локальные акты учреждения.

Актуальность программы. В настоящее время всё более актуальным для человека становится владение новыми информационными технологиями. Развитие цифровых технологий идёт быстрыми темпами. Очевидно, что вслед за «демократизацией» сетевых технологий (рост доступности Интернета) и технологий связи (введение в широкий обиход мобильных телефонов) наступает время «демократизации» телевидения. Постепенно телевидение превращается в средство коммуникации, доступное для всех желающих.

Данная программа даёт возможность использовать навыки, полученные во время обучения основам программирования, включая детей в систему средств массовой коммуникации общества. Создание мультфильмов рассматривается в рамках данной программы как сфера социальной практики детей и подростков, в рамках которой они развивают свои творческие способности, приобретают культурный, нравственный, мировоззренческий и социальный опыт.

Отличительной особенностью программы является её практика-ориентированный характер. Подростки проходят обучение в процессе работы над реальным телевизионным продуктом – мультфильмами. Такая работа в составе детской редакции имеет четкие сроки, накладывает на весь коллектив и каждого человека большую ответственность за выполнение взятых на себя обязательств. Занятия курса направлены на развитие мышления, логики, творческого потенциала учеников.

Программа ориентирована на использование получаемых знаний для разработки реальных проектов. Метод проектов позволяет формировать активную, самостоятельную и инициативную позицию школьников; формировать не просто умения, а компетенции, он приоритетно направлен на развитие познавательного интереса учащихся. Всю работу над проектом осуществляют сами школьники, а учителю отводится роль куратора, помощника, наставника. Курс содержит большое количество творческих заданий (именуемых Кейсами).

Адресат программы: учащиеся 13- 15 лет

Объем и срок освоения программы: Программа рассчитана на 34 часов, на 1 год обучения.

Режим занятий, периодичность и продолжительность: Занятия проводятся 1 раз в неделю по 1 академическому часу.

Форма обучения: очная.

Количество детей в группах: 5 - 15 человек.

Особенности организации образовательного процесса. Рабочая программа может реализовываться с использованием электронного обучения (ЭО) и дистанционных образовательных технологий (ДОТ).

1.2. Цель и задачи программы

Целью изучения программы «Информатика» является получение теоретических и практических знаний, умений и навыков в области современной информатики; формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира.

Задачи программы:

Обучающие:

- формирование информационной и алгоритмической культуры;

- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей;
- овладение важнейшими общеучебными умениями и универсальными учебными действиями (формулировать цели деятельности, планировать ее, находить и обрабатывать необходимую информацию из различных источников, включая Интернет и др.).

Воспитывающие:

- создание условий для развития интеллектуальных и творческих способностей учащихся, необходимых для успешной социализации и самореализации личности.

Развивающие:

- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм.

1.3. Содержание программы

1.3.1. Учебно-тематический план

№ п/п	Наименование раздела (темы)	Количество часов			Формы аттестации / контроля
		Всего	Теория	Практика	
1	Знакомство со средой программирования Scratch	3	1,5	2	тест
1.1	Исполнитель Scratch, цвет и размер пера.	1	0,5	0,5	практическое задание
1.2	Основные инструменты встроенного графического редактора программной среды SCRATCH.	2	1	1	практическое задание
2	Алгоритм. Линейный алгоритм. Создание блок-схемы. Основные графические примитивы векторного редактора LibreOffice.Draw.	4	2	2	тест
2.1	Линейный алгоритм. Рисование линий исполнителем Scratch.	1	0,5	0,5	практическое задание
2.2	Линейный алгоритм. Исполнитель Scratch рисует	1	0,5	0,5	практическое задание

	квадраты и прямоугольники линейно.				
2.3	Конечный цикл. Scratch рисует квадраты, линии.	1	0,5	0,5	практическое задание
2.4	Конечный цикл. Scratch рисует несколько линий и фигур. Копирование фрагментов программы.	1	0,5	0,5	практическое задание
3	Циклический алгоритм. Цикл в цикле.	4	2	2	самоконтроль
3.1	Цикл в цикле. Повторение пунктирной линии с поворотом. Блок-схема цикла.	1	0,5	0,5	практическое задание
3.2	Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов.	2	1	1	самоконтроль
3.3	Сцена как исполнитель. Создаем модель таймера.	1	0,5	0,5	практическое задание
4	Одинаковые программы для нескольких исполнителей.	23	9	14	самоконтроль
4.1	Параллельное выполнение действий несколькими исполнителями	1	0,5	0,5	практическое задание
4.2	Разбиение программы на части для параллельного выполнения исполнителями. Таймер.	2	1	1	практическое задание
4.3	Два исполнителя со своими программами. Мини-проект «Часы с кукушкой».	1	0,5	0,5	практическое задание
4.4	Алгоритмы с ветвлением. Условие ЕСЛИ.	1	0,5	0,5	самоконтроль
4.5	Цикл с условием. Мини-проект «Шарики в лабиринте»	1	0,5	0,5	практическое задание
4.6	Цикл с условием. Исполнитель определяет цвет. Сенсор «касается цвета»	1	0,5	0,5	практическое задание
4.7	Оператор случайных чисел.	1	0	1	тест
4.8	Перемещение исполнителей между слоями.	1	0,5	0,5	практическое задание
4.9	Действия исполнителей в разных слоях.	1	0,5	0,5	практическое задание
4.10	Взаимодействие исполнителей.	1	0,5	0,5	самоконтроль
4.11	Последовательное выполнение команд исполнителями.	1	0,5	0,5	практическое задание
4.12	Программирование клавиш. Мини-проект «Лабиринт»	1	0,5	0,5	практическое задание
4.13	Управление событиями.	1	0	1	тест
4.14	Координатная плоскость.	1	0,5	0,5	практическое задание

	Геометрические фигуры.				
4.15	Координатная плоскость. Переменные.	1	0,5	0,5	самостоятельная работа
4.16	Создание списков.	1	0,5	0,5	практическое задание
4.17	Использование подпрограмм.	1	0,5	0,5	практическое задание
4.18	Сообщество Scratch.	1	0,5	0,5	самоконтроль
4.19	Отладка программ с ошибками.	1	0,5	0,5	практическое задание
4.20	Итоговый проект.	3	0	3	защита проекта

1.3.2. Содержание учебно-тематического плана

Раздел 1. Знакомство со средой программирования Scratch (3 ч.)

Тема 1.1 Исполнитель Scratch, цвет и размер пера.(1ч.)

Теория: Исполнитель Scratch, цвет Понятие ИСПОЛНИТЕЛЯ. Команды управления пером: «Опустить перо», «Поднять перо», «Очистить», «Установить цвет пера», «Установить размер пера.

Практика: Кейс 1. Рисуем разноцветные лужи и облака для прогулки кота и размер пера.

Форма контроля: Практическое задание

Тема 1. 2.Основные инструменты встроенного графического редактора программной среды SCRATCH.(2 ч.)

Теория: Инструменты растрового графического редактора — кисточка, ластик, заливка, квадрат, круг, линия. Копирование, поворот, горизонтальное отражение, вертикальное отражение во встроенном редакторе программной среды Scratch. Командный блок внешность (фиолетовый) — команды начальной установки эффекта цвет «Установить эффект цвет в значение 0 и команда начальной установки размера «Установить размер 100%». Команды: «Изменить цвет эффект на _», «Изменить размер на _» Командный блок управления пером (зелёный) — команда «Печать» для копирования графического изображения исполнителя в нужном месте экрана. Эффекты, которые могут быть применены к графическим изображениям действующего исполнителя. В графическом редакторе можно создавать сложные

графические изображения, почти не прикладывая собственных усилий, а лишь правильно применяя встроенные возможности программной среды.

Практика: Кейс 2. Свободное рисование. Придумай, чем можно дополнить проект с прошлого урока (Нарисовать дерево? Воздушный шар? Дом?) и реализуй это.

Форма контроля: Практическое задание

Раздел 2. Алгоритм. Линейный алгоритм. Создание блок-схемы. (4 ч.)

Тема 2.1 Линейный алгоритм. Рисование линий исполнителем Scratch.

Теория: Решение поставленной задачи в виде последовательного выполнения команд, графики. Последовательное выполнение команд. Изменение параметров пера.

Практика: Кейс 3. Что бывает полосатое?

Придумайте линейный алгоритм для создания полосатого объекта (заранее придумайте, что это будет – шарф, зебра, забор?) Создайте блок-схему этого алгоритма и реализуйте алгоритм в среде Scratch.

Форма контроля: Практическое задание

Тема 2.2 Линейный алгоритм. Исполнитель Scratch рисует квадраты и прямоугольники линейно.

Теория: Создание алгоритма для рисования исполнителем квадрата путем последовательного выполнения команд. Рисование линейного алгоритма, состоящего из двух колонок блоков команд.

Практика: Использование векторного редактора офисного пакета LibreOffice в качестве инструмента для создания блок-схем. Выбор нужного значения из предлагаемого списка вариантов. Отладка программы для получения верного результата. Команда «повернуть в направление». Пошаговое выполнение

Форма контроля: Практическое задание

Тема 2.3. Конечный цикл. Scratch рисует квадраты, линии. Сохранение готовых программ для дальнейшего использования.

Теория: Команда открыть из пункта меню File; Команда сохранить как... из пункта меню File; Циклический алгоритм; Блок-схема циклического алгоритма; Команды: «повернуться на 90° по часовой стрелке», «повернуться на 90° против часовой стрелки», «повторить _». Упражнение. Сравнение двух алгоритмов, рисующих квадрат. Идти 100 шагов Повторить 4 Повернуться на 90° Идти 100 шагов Идти 100 шагов Повернуться на 90° Повернуться на 90° Идти 100 шагов Повернуться на 90° Идти 100 шагов Повернуться на 90° Рекомендуется объяснить, что команды, обеспечивающие повторяющееся выполнение одной или нескольких команд называются циклическими, показать, что использование команды цикла всегда сокращает количество команд в программе, а значит делает её более красивой. В более короткой программе легче обнаружить ошибку или неточность.

Практика: Рисование блок-схемы циклического алгоритма. Использование векторного редактора офисного пакета LibreOffice в качестве инструмента для создания блок-схем; Использование команд поворота на прямой угол (90°) по часовой и против часовой стрелки; Использование циклического алгоритма для рисования исполнителем квадрата; Оптимизация линейного алгоритма за счёт использования циклической конструкции в программной среде Scratch.

Форма контроля: Практическое задание.

Тема 2.4. Конечный цикл. Scratch рисует несколько линий и фигур.

Копирование фрагментов программы.

Теория: Использование операции цикла для решения учебных задач. Применение поворота на прямой угол (90°) при создании геометрических фигур и перемещении исполнителя. Оптимизация линейного алгоритма за счёт использования циклической конструкции в программной среде Scratch. Тело цикла. Конечный и бесконечный циклы. Блок-схема бесконечного цикла. Имя спрайта и костюма.

Практика: Изменение костюма исполнителя. Копирование фрагмента программы. Команды: «Следующий костюм», «Перейти к костюму».

Рекомендуется объяснить, что циклические конструкции могут применяться не только для рисования фигур. Например, в цикле можно менять внешность исполнителя.

Кейс 4. Создай картинку из квадратов. Предложить ребятам нарисовать свою картинку из квадратов, может быть, это будет лестница, может быть, ковер, может быть, что-то еще.

Форма контроля: Практическое задание.

Раздел 3. Циклический алгоритм. Цикл в цикле.(4 ч.)

Теория: Использование операции цикла в цикле для решения учебных задач. Использование поворота на прямой угол (90°) по часовой и против часовой стрелки. Оптимизация алгоритма за счёт использования конструкции «цикл в цикле». Блок-схема конструкции цикл в цикле. Например, расписание уроков составляется на неделю, в месяце в среднем четыре недели. Если расписание не меняется в течение учебного года, то получаем цикл из девяти месяцев по четыре недели с одинаковым расписанием.

Практика: Использование циклов «повторить». Нарисуйте квадраты, как показано на рисунке, используя конструкцию цикл в цикле.

Форма контроля: Самоконтроль

Тема 3.1. Цикл в цикле. Повторение пунктирной линии с поворотом. Блок-схема цикла.

Теория: Использование конструкции «цикл в цикле». Создание и реализация алгоритма рисования квадрата несплошными линиями; Оптимизация алгоритма за счёт использования конструкции «цикл в цикле» в программной среде Scratch; Использование операции копирования внешности исполнителя путём копирования костюма. Несплошные линии. Алгоритм рисования несплошных линий с использованием циклических конструкций. Рисование квадрата несплошными линиями, используя конструкцию «цикл в цикле». При перемещении исполнителя можно рисовать линии отличные от сплошной. Эти возможности определяются использованием команд из блока ПЕРО.

Практика: Кейс 5. Мой необычный дом.

Форма контроля: Практическое задание

Тема 3.2. Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов.

Теория: Бесконечный цикл. Анимация исполнителя с помощью смены костюмов. Эффект «призрак».

Практика: Изменение размера исполнителя. Интерактивное взаимодействие с исполнителем с помощью клавиатуры. Команды «спросить» и «думать».

Форма контроля: Самоконтроль.

Тема 3.3. Сцена как исполнитель.

Практика: Создаем модель таймера. Анимация сцены (фона). Синхронная анимация сцены и исполнителя.

Форма контроля: Практическое задание

Раздел 4. Одинаковые программы для нескольких исполнителей. (23 ч.)

Тема 4.1 Параллельное выполнение действий несколькими исполнителями.

Теория: Анимация с помощью вращения. Дублирование исполнителей. При копировании исполнителей копируются и их скрипты. Использование одинаковых программ, но разных костюмов у исполнителей.

Практика: Дублирование исполнителей. Синхронное выполнение скриптов. При параллельном выполнении скриптов результат работы всех исполнителей получается гораздо быстрее, чем в тех задачах, когда работал один исполнитель.

Форма контроля: Практическая работа.

Тема 4.2. Разбиение программы на части для параллельного выполнения исполнителями. Таймер.

Теория: Планирование действий разных исполнителей во времени для решения общей задачи. Сенсор «таймер» Сброс «таймера» Команда «ждать до ...»

Практика: Кейс 6. Мини-проект «Смена времени суток».

Создайте скрипт, в котором будет четыре сцены: «утро», «день», «вечер», «ночь». Используйте «таймер» для своевременного изменения сцен, своевременного запуска скриптов разных исполнителей.

Форма контроля: Практическая работа.

Тема 4.3. Два исполнителя со своими программами. Мини- проект «Часы с кукушкой».

Теория: Координаты в Scratch. Команда «идти в $x: _ y: _$ » (переместиться в точку с координатами (x,y)) Команда «плыть $_$ секунд в точку $x: _ y: _$ » (плавно двигаться в точку с координатами (x,y)) Команды «спрятаться», «показаться».

Практика: Кейс 7. Мини-проект «Часы с кукушкой».

Создайте сцену в виде часов, и три исполнителя: часовую стрелку, минутную стрелку и кукушку. Стрелки должны вращаться на циферблате. При наступлении каждого часа (т.е. в $_$ часов : 00 минут) должна появляться кукушка.

Форма контроля: Практическая работа

Тема 4.4. Алгоритмы с ветвлением. Условие ЕСЛИ.

Теория: Блок-схема алгоритма с ветвлением. Полная форма ветвления.

Практика: Команда «если $_$ или» Неполная форма ветвления. Команда «если $_$ » Сенсор «касается» (касания края, или другого спрайта, или указателя мыши).

Форма контроля: Самоконтроль

Тема 4.5. Цикл с условием.

Теория: Отскок шариков от стен

Практика: Мини-проект «Шарики в лабиринте» Сенсор «касается цвета» Программируем отскок шариков от стен. Создать сцену в виде бильярдного стола, на котором три шарика будут двигаться с разными скоростями и отталкиваться от бортиков.

Форма контроля: Практическое задание

Тема 4.6 . Цикл с условием.

Теория: Исполнитель определяет цвет. Сенсор «касается цвета».

Практика: Программируем поведение исполнителя в зависимости от цвета фона.

Форма контроля: Практическое задание.

Тема 4.7. Оператор случайных чисел.

Практика: Команда «Выдать случайное число от _ до _». Случайные перемещения исполнителя в координатной плоскости. Случайное количество шагов. Случайные координаты. Поворот на случайный угол.

Форма контроля: Самоконтроль

Тема 4.8. Перемещение исполнителей между слоями.

Теория: Команда «Перейти в верхний слой» Команда «Перейти назад на слои» Эффект «Призрак».

Практика: Отработка команд

Форма контроля: Практическое задание

Тема 4.9 Действия исполнителей в разных слоях.

Теория: Исполнители разных слов.

Практика: Кейс 8. Мини-проект «Дорога».

Создайте сцену, на которой изображена дорога с двумя полосами, по обочинам дороги расположите кусты и деревья. По дороге должны двигаться автомобили. Правильно расположите автомобили, деревья и кусты по слоям.

Форма контроля: Практическое задание

Тема 4.10 Взаимодействие исполнителей.

Теория: Команда «Касается».

Практика: Отработка навыков: Исполнители касаются друг друга. Поведение исполнителей при столкновении.

Форма контроля: практическое задание

Тема 4.11. Последовательное выполнение команд исполнителями.

Теория: Команда «Передать» Команда «Когда я получу».

Связи между программами разных исполнителей.

Практика: Отработка полученных знаний

Форма контроля: Практическое задание

Тема 4.12. Программирование клавиш. Мини-проект «Лабиринт».

Теория: Взаимодействие пользователя с программой. Клавиши управления перемещением исполнителя Команда «Изменить значение x на _» Команда «Изменить значение y на _»

Практика: Кейс 8. Игра «Лабиринт»

Форма контроля: Практическое задание

Тема 4.13. Управление событиями.

Практика: Передача сообщений между исполнителями и фоном. Запуск программ после получения сообщения.

Форма контроля: Тест

Тема 4.14. Координатная плоскость. Геометрические фигуры.

Рисование геометрических фигур на координатной плоскости.

Теория: Последовательное выполнение команд одним исполнителем. Параллельное выполнение команд несколькими исполнителями.

Практика: Кейс 8. Исследование времени выполнения программ.

Создайте одного исполнителя, рисующего квадраты, как показано на рисунке. Затем создайте четырех исполнителей, которые будут параллельно рисовать по три квадрата. Сравните время работы в этих двух случаях.

Форма контроля: Практическое задание

Тема 4.15 Координатная плоскость. Переменные.

Теория: Блок «Переменные»

Практика: Рисование геометрических фигур на координатной плоскости с использованием переменных. Построение перпендикуляров к координатным осям.

Форма контроля: Самоконтроль

Тема 4.16. Создание списков.

Теория: Название списка. Элементы списка. Длина списка. Команда «Создать список». Выбор элемента списка.

Практика: Кейс 9. Мини-проект «Викторина».

Создайте проект «Викторина» в котором будет задано 10 вопросов. Правильный ответ надо выбрать из списка. В случае правильного ответа очки увеличиваются на единицу. В конце выставляется оценка: «отлично» за 9 или 10, «хорошо» за 7 или 8, «удовлетворительно» за 5 или 6 правильных ответов.

Форма контроля: Практическое задание

Тема 4.17. Использование подпрограмм.

Теория: Как сделать программу структурированной и более понятной.
Команда

Практика: «Передать и ждать» Команда «Играть звук» Команда «Ноту играть»

Форма контроля: Практическое задание

Тема 4.18 Сообщество Scratch. Scratch

Теория: Как зарегистрироваться на сайте. Создание личного кабинета.

Практика: Регистрация на сайте. Личный кабинет. Публикация проектов.

Форма контроля: Самоконтроль.

Тема 4.19. Отладка программ с ошибками.

Теория: Наиболее встречающиеся ошибки, и как их исправить.

Практика: Отработка навыков

Форма контроля: Практическое задание.

Тема 4.20. Итоговый проект

Практика: Учащиеся самостоятельно или в парах выполняют индивидуальный проект, согласованный с учителем. Как итог, можно провести конкурс проектов.

Форма контроля: Защита проекта.

1.4. Планируемые результаты

По окончании 1 года обучения учащийся будет знать:

- основы программной среды Scratch;

- виды алгоритмов и их структуры;
- разницу между программами и исполнителями;
- требования техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информационных и коммуникационных технологий.

Будет уметь:

- управлять одновременной работой нескольких исполнителей;
- уметь передавать сообщения между исполнителями;
- тестировать и оптимизировать алгоритмы исполнителей;
- составлять сценарии проектов среды Scratch;
- создавать и редактировать рисунки в графическом редакторе;
- использовать обширную библиотеку готовых сцен и исполнителей;
- изменять размер, костюм, прозрачность исполнителя;
- создавать линейные алгоритмы для исполнителя;
- создавать циклические алгоритмы;
- создавать ветвящиеся алгоритмы;
- управлять одновременной работой нескольких исполнителей;
- передавать сообщения между исполнителями;
- внедрять звуковые эффекты в алгоритмы исполнителей;
- создавать алгоритмы, которые будут выполняться одновременно (параллельно) несколькими исполнителями;
- тестировать и оптимизировать алгоритмы исполнителей.

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

2.1. Календарный учебный график

№	Месяц	Число/время	Форма занятия	Количество часов	Тема занятия	Место проведения	Форма контроля
1	09	10 17	Л/П	3	Знакомство со средой программирования Scratch	Коворкинг	тест
1.1	10	01	Л/П	1	Исполнитель Scratch, цвет и размер пера.	Коворкинг	практическое задание
1.2	10	08 15	Л/П	2	Основные инструменты встроенного графического редактора программной среды SCRATCH.	Коворкинг	практическое задание
2	10 11	22 5 12 19	Л/П	4	Алгоритм. Линейный алгоритм. Создание блок-схемы. Основные графические примитивы векторного редактора LibreOffice.Draw.	Коворкинг	тест
2.1	11	26	Л/П	1	Линейный алгоритм. Рисование линий исполнителем Scratch.	Коворкинг	практическое задание
2.2	12	3	Л/П	1	Линейный алгоритм. Исполнитель Scratch рисует квадраты и прямоугольники линейно.	Коворкинг	практическое задание
2.3	12	10	Л/П	1	Конечный цикл. Scratch рисует квадраты, линии.	Коворкинг	практическое задание
2.4	12	17	Л/П	1	Конечный цикл. Scratch рисует несколько линий и фигур. Копирование фрагментов программы.	Коворкинг	практическое задание
3	12	24	Л/П	4	Циклический алгоритм. Цикл в цикле.	Коворкинг	самоконтроль
3.1			Л/П	1	Цикл в цикле. Повторение пунктирной линии с поворотом. Блок-схема цикла.	Коворкинг	практическое задание
3.2			Л/П	2	Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов.	Коворкинг	самоконтроль

3.3			Л/П	1	Сцена как исполнитель. Создаем модель таймера.	Коворкинг	практическое задание
4			Л/П	23	Одинаковые программы для нескольких исполнителей.	Коворкинг	самоконтроль
4.1			Л/П	1	Параллельное выполнение действий несколькими исполнителями	Коворкинг	практическое задание
4.2			Л/П	2	Разбиение программы на части для параллельного выполнения исполнителями. Таймер.	Коворкинг	практическое задание
4.3			Л/П	1	Два исполнителя со своими программами. Мини-проект «Часы с кукушкой».	Коворкинг	практическое задание
4.4			Л/П	1	Алгоритмы с ветвлением. Условие ЕСЛИ.	Коворкинг	самоконтроль
4.5			Л/П	1	Цикл с условием. Мини-проект «Шарики в лабиринте»	Коворкинг	практическое задание
4.6			Л/П	1	Цикл с условием. Исполнитель определяет цвет. Сенсор «касается цвета»	Коворкинг	практическое задание
4.7			П	1	Оператор случайных чисел.	Коворкинг	тест
4.8			Л/П	1	Перемещение исполнителей между слоями.	Коворкинг	практическое задание
4.9			Л/П	1	Действия исполнителей в разных слоях.	Коворкинг	практическое задание
4.10			Л/П	1	Взаимодействие исполнителей.	Коворкинг	самоконтроль
4.11			Л/П	1	Последовательное выполнение команд исполнителями.	Коворкинг	практическое задание
4.12			Л/П	1	Программирование клавиш. Мини-проект «Лабиринт»	Коворкинг	практическое задание
4.13			П	1	Управление событиями.	Коворкинг	тест
4.14			Л/П	1	Координатная плоскость. Геометрические фигуры.	Коворкинг	практическое задание
4.15			Л/П	1	Координатная плоскость. Переменные.	Коворкинг	самостоятельная работа

4.16			Л/П	1	Создание списков.	Коворкинг	практическое задание
4.17			Л/П	1	Использование подпрограмм.	Коворкинг	практическое задание
4.18			Л/П	1	Сообщество Scratch.	Коворкинг	самоконтроль
4.19			Л/П	1	Отладка программ с ошибками.	Коворкинг	практическое задание
4.20			П	3	Итоговый проект.	Коворкинг	защита проекта

2.2. Условия реализации программы

1. Материально-техническое обеспечение:

- парта ученическая – 8 шт.;
- стул ученический – 15 шт.;
- доска аудиторная 3-х-элементная – 1 шт.;
- проектор мультимедийный – 1 шт.;
- компьютер портативный (ноутбук) с предустановленным программным обеспечением – 15 шт.;
- персональный компьютер для учителя с предустановленным программным обеспечением – 1 шт.;
- звуковые колонки – 1 шт.

2. Кадровое обеспечения:

К проведению занятий привлекаются специалисты, обладающие необходимыми компетенциями в области информатики.

2.3. Формы аттестации / контроля

Для системы оценки результативности обучения по программе используются разнообразные Результаты контроля учебной работы воспитанников служат основанием для внесения коррективов в

содержание и организацию процесса обучения, поощрения успешной работы лучших воспитанников, развития их творческих способностей, самостоятельности, инициативы в овладении знаниями, умениями и навыками.

Формы и виды контроля:

- вводный, организуется перед началом работы;
- текущий, проводится в ходе образовательного процесса;
- рубежный, проводится по завершению тем и определенных разделов;
- итоговый, проводится после завершения всей программы.

Контроль проводится в проектной форме.

Педагогом проводится диагностика освоения учащимися образовательной программы по направлениям деятельности в соответствии с целями и задачами каждого программы. По итогам изучения тем проводится защита проектов.

2.4. Методические материалы

Данная программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта основного общего образования и направлена на развитие творческих способностей детей через приобщение к миру телеискусства.

Методы обучения:

1. Наглядные: использование технических средств.
2. Словесные: беседа, инструктаж
3. Практические: практические занятия, анализ и решение.

2.5. Список литературы

1. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch. —

Оренбург: Оренб. гос. ин-т. менеджмента, 2009.

2. «Пропедевтика идей параллельного программирования в средней школе при помощи среды Scratch», В.Г. Рындак, В.О. Джинжер, Л.В. Денисова;
3. «Раннее обучение программированию в среде Scratch», В.Г. Рындак, В.О. Джинжер, Л.В. Денисова;
4. Сорокина Т.Е. Поурочные разработки «Пропедевтика программирования со Scratch» для 5-го класса, 2015 г.
5. Творческие задания в среде Scratch: рабочая тетрадь для 5-6 классов/Ю.В. Пашковская. – М.: БИНОМ. Лаборатория знаний, 2014. – 200 с.:ил.

Интернет ресурсы:

1. <http://scratch.mit.edu> – официальный сайт Scratch
2. <http://letopisi.ru/index.php> /Скретч - Скретч в Летописи.ру
3. <http://setilab.ru/scratch/category/commun> - Учитесь со Scratch

Программирование в Scratch

Знакомство со средой программирования Scratch. Создание и сохранение документа. Понятия спрайта, сцены, скрипта. Библиотека персонажей. Исполнитель Scratch. Основные инструменты встроенного графического редактора программной среды SCRATCH. Линейный алгоритм. Создание блок-схемы. Рисование линий исполнителем Scratch. Конечный и бесконечный циклы. Цикл в цикле. Анимация исполнителя Scratch на основе готовых костюмов. Дублирование исполнителей. Алгоритмы с ветвлением. Цикл с условием. Перемещение исполнителей между слоями. Программирование клавиш. Управление событиями. Координатная плоскость. Создание списков. Использование подпрограмм. Отладка программ с ошибками.

ПРИЛОЖЕНИЕ 2

Требования к исследовательскому проекту по информатике

Исследование является одной из форм человеческой деятельности, в основе которой лежит интерес и любознательность, активное отношение к окружающему миру с целью его изменения и преобразования. Это относится и к научному исследованию, однако здесь имеется ряд характерных особенностей. Научное исследование – целенаправленное познание, результаты которого выступают в виде системы понятий, законов, теорий.

Если в научном исследовании главной целью является получение объективно нового результата, то целью учебной исследовательской деятельности выступает приобретение навыка исследования как универсального способа освоения действительности, развитие способности к исследовательскому типу мышления, активизация личностной позиции на основе приобретения новых знаний.

Учебная исследовательская деятельность связана с решением учащимися творческой, исследовательской задачи с заранее неизвестным решением и предполагающая наличие основных этапов, характерных для исследования в научной сфере. Под новыми знаниями нужно понимать самостоятельно получаемые знания, являющиеся новыми и личностно значимыми для конкретного учащегося.

В традициях науки принято, чтобы исследование проходило следующие основные этапы:

- постановка проблемы,
- изучение теории, посвященной данной проблематике,
- подбор методик исследования и практическое овладение ими,
- сбор собственного материала, его анализ и обобщение,
- научный комментарий,
- собственные выводы.

Текст ученической исследовательской работы должен отличаться своей логичностью. Для осуществления научной исследовательской деятельности необходимо знать и применять основные законы и правила логики.

Закон тождества – все понятия и суждения должны носить однозначный характер, исключая двусмысленность и неопределенность.

Закон противоречия – не могут быть одновременно истинными два высказывания, одно из которых что-то утверждает, а другое отрицает то же самое.

Закон достаточного основания – всякое суждение должно быть достаточно обоснованным, прежде чем быть принятым за истину.

Помимо собственно исследовательских работ результатом поисковой деятельности могут быть и другие типы ученических работ:

Реферативные – написанные на основе нескольких литературных источников, предполагающие сопоставление данных разных источников и на основе этого собственную трактовку поставленной проблемы.

Экспериментальные – написанные на основе выполнения эксперимента, описанного в науке и имеющего известный результат, но предполагают самостоятельную трактовку особенностей результата в зависимости от изменения исходных условий.

Описательные – направлены на наблюдение и качественное описание какого-либо явления. Могут иметь элемент научной новизны. Отсутствует корректная методика исследования.

Подготовка ученического исследования.

Исследовательская работа учащегося должна демонстрировать соответствующие знания и умения в области конкретных изучаемых предметов, умение работать с первоисточниками, использовать различные методы исследования, способность к систематизации и структурированию полученного материала, умение обобщать, делать выводы, сопоставлять различные умозаключения, давать оценку

событиям и явлениям. Ученическое исследование проводится по общей схеме научного исследования.