

08 - 13

**Муниципальное казенное общеобразовательное учреждение
«Архангельская средняя школа имени А. А. Кудрявцева»**

Рассмотрено
на заседании педагогического совета.
Протокол № 11 от 30.08.2024.

Утверждено
Приказом по школе
№ 105-о от 30.08.2024.

**Рабочая программа внеурочной деятельности
«Визуальное программирование»**

2024 – 2025 учебный год

Уровень реализации: основное общее образование

Класс: 5

Направление: общеинтеллектуальное (дополнительное изучение учебных предметов)

Срок реализации: 1 год

Составитель: Купцов Е.А., учитель информатики

с. Архангельское

2023 год

Оглавление

1.	Пояснительная записка	4
2.	Планируемые результаты освоения учебного предмета	5
3.	Содержание курса	8
4.	Тематическое планирование	9

Пояснительная записка

Рабочая программа внеурочной деятельности «Визуальное программирование» (5 класс) разработана на основе требований к результатам освоения основной образовательной программы основного общего образования Муниципального казенного общеобразовательного учреждения «Архангельская средняя школа имени А. А. Кудрявцева», Примерной основной образовательной программы основного общего образования (одобрена решением федерального учебно-методического объединения по общему образованию, протокол от 18 марта 2022 г. № 1/22) с учётом Примерной рабочей программы основного общего образования по информатике (5-6 класс) (одобрена решением Федерального учебно-методического объединения по общему образованию, протокол 3/21 от 27.09.2021), Рабочей программы воспитания МКОУ «АСШ им. А. А. Кудрявцева».

Рабочая программа ориентирована на литературу:
Сорокина Т.Е. МОДУЛЬ «ПРОПЕДЕВТИКА ПРОГРАММИРОВАНИЯ СО SCRATCH»
Видео-, аудиоматериалы:

Видеоуроки по Scratch <http://www.youtube.com/watch?v=vd20J2r5wUQ>

Цифровые ресурсы:

1. Курс «Введение в Scratch»

http://window.edu.ru/resource/056/78056/files/scratch_lessons.pdf

2. <https://scratch.mit.edu/>

3. <https://ru.libreoffice.org/>

4. Полное руководство по Google Docs: все, о чем вы не знали, но боялись спросить
<https://texterra.ru/blog/polnoerukovodstvopo-google-docs.html>

Цель и задачи обучения

Целью изучения предмета «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.

Для достижения поставленной цели необходимо решение следующих **задач**:

- создание условий для развития интеллектуальных и творческих способностей учащихся, необходимых для успешной социализации и самореализации личности;
- формирование информационной и алгоритмической культуры;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм;
- 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.).

Планируемые результаты освоения учебного предмета.

Личностными результатами, формируемыми при изучении предмета информатика, являются:

- формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в

мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов;

- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творческой и других видов деятельности.

Метапредметные результаты изучения предмета «Информатика»:

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности её решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать

конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение;

- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

Предметные результаты изучения предмета «Информатика»:

- формирование представления об основных изучаемых понятиях курса;
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для решения конкретной задачи;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей, с использованием соответствующих программных средств обработки данных;
- формирование представления о том, что значит “программировать” на примере языка Scratch, формирование умения составлять сценарии проектов среды Scratch;
- знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- формирование умения тестировать и оптимизировать алгоритмы исполнителей;
- формирование умения создавать и редактировать документы в текстовом процессоре;
- формирование умения размещать документы в облачном хранилище. организовывать коллективную работу с документами, настраивать права доступа к документам;
- формирование умения формализации и структурирования информации,
- использование готовых прикладных компьютерных программ по выбранной специализации;
- развитие умений применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимости справочных материалов, компьютера;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.

Содержание курса

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. Работа с текстовым процессором LibreOffice.org Writer (34 часа)

Загрузка и установка LibreOffice. Интерфейс редактора. Стандартные действия. Форматирование документа: шрифты, стили, размер шрифта. Работа с цветом. Сложное форматирование.

Использование списков. Колонтитулы. Изображения в текстовых документах. Графика в текстовых документах. Таблицы в документах. Работа с Google-docs.

Тематическое планирование

№ урока	Тема занятия	Виды деятельности.	Кол-во часов
Модуль 1. Программирование в Scratch			
1	Знакомство со средой программирования Scratch	Свободное программное обеспечение. Авторы программной среды Scratch. Параметры для скачивания и установки программной среды на домашний компьютер. Основные элементы пользовательского интерфейса программной среды Scratch. Внешний вид рабочего окна. Блочная структура систематизации информации. Функциональные блоки. Блоки команд, состояний, программ, запуска, действий и исполнителей. Установка русского языка для Scratch. Создание и сохранение документа. Понятия спрайта, сцены, скрипта. Очистка экрана. Библиотека персонажей. Сцена и разнообразие сцен, исходя из библиотеки данных. Систематизация данных библиотек персонажей и сцен. Иерархия в организации хранения костюмов персонажа и фонов для сцен. Импорт костюма, импорт фона. Кейс 1. Научи kota бегать и мяукать.	1
2	Исполнитель Scratch, цвет и размер пера.	Понятие ИСПОЛНИТЕЛЯ. Команды управления пером: «Опустить перо», «Поднять перо», «Очистить», «Установить цвет пера», «Установить размер пера». Кейс 2. Рисуем разноцветные лужи и облака для прогулки kota.	1
3	Основные инструменты встроенного графического редактора программной среды SCRATCH.	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. Кейс 3. Свободное рисование. Придумай, чем можно дополнить проект с прошлого	1

		урока (Нарисовать дерево? Воздушный шар? Дом?) и реализуй это.	
4	Алгоритм. Линейный алгоритм. Создание блок-схемы. Основные графические примитивы векторного редактора LibreOffice.Draw.	Создание собственных изображений в других программах (например, LibreOfficeDraw) и импортирование их в программную среду Scratch. Знакомство с основными графическими примитивами векторного редактора LibreOfficeDraw. Возможность создания геометрических фигур без внутренней заливки, но с текстовым блоком внутри. Стрелки, их направление. Алгоритм, блок-схема как способ записи. Кейс 4. Запиши мой алгоритм! Ребята разбиваются на пары, в паре формулируют друг другу какую-то повседневную задачу, для решения которой нужен линейный алгоритм, придумывают этот алгоритм и рисуют блок-схему для задачи, заданной им напарником.	1
5	Линейный алгоритм. Рисование линий исполнителем Scratch.	Решение поставленной задачи в виде последовательного выполнения команд. Создание блок-схемы линейного алгоритма средствами редактора векторной графики. Последовательное выполнение команд. Изменение параметров пера. Кейс 5. 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.	1
6	Линейный алгоритм. Исполнитель Scratch рисует квадраты и прямоугольники линейно.	Создание алгоритма для рисования исполнителем квадрата путем последовательного выполнения команд. Рисование линейного алгоритма, состоящего из двух колонок блоков команд. 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 программы для её отладки. Центр костюма исполнителя Scratch.	1
7	Конечный цикл. Scratch рисует квадраты, линии.	Сохранение готовых программ для дальнейшего использования. 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	1

		<p>программной среде Scratch; 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°.</p> <p>Повернуться на 90°. Идти 100 шагов. Повернуться на 90°. Идти 100 шагов. Повернуться на 90°.</p> <p>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.</p>	
8	<p>Конечный цикл. Scratch рисует несколько линий и фигур. Копирование фрагментов программы.</p>	<p>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. 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.</p> <p>Кейс 6. Создай картинку из квадратов. Предложите ребятам нарисовать свою картинку из квадратов, может быть, это будет лестница, может быть, ковер, может быть, что-то еще.</p>	1
9	<p>Циклический алгоритм. Цикл в цикле.</p>	<p>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.</p> <p>Упражнение. Использование циклов «повторить _» Нарисуйте квадраты, как показано на рисунке, используя конструкцию цикл в цикле.</p>	1
10	<p>Цикл в цикле. Повторение пунктирной линии с поворотом. Блок-схема</p>	<p>Создание и реализация алгоритма рисования квадрата несплошными линиями; Оптимизация алгоритма за счёт использования конструкции «цикл в цикле» в</p>	1

	цикла. Использование конструкции «цикл в цикле».	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. Кейс 7. Мой необычный дом. Предложите ребятам нарисовать дом и что-то рядом с ним, используя линейные и циклические алгоритмы, а также смену костюмов Исполнителя. Может быть, над домом будут звезды разного размера, а около дома будет пунктирная разноцветная дорога, или деревья разного размера.	
11	Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов. Бесконечный цикл. Анимация исполнителя с помощью смены костюмов. Эффект «призрак». Изменение размера исполнителя.	Интерактивное взаимодействие с исполнителем с помощью клавиатуры. Команды «спросить» и «думать».	1
12	Сцена как исполнитель.	Создаем модель таймера. Анимация сцены (фона). Синхронная анимация сцены и исполнителя.	1
13	Одинаковые программы для нескольких исполнителей.	Анимация с помощью вращения. Дублирование исполнителей. При копировании исполнителей копируются и их скрипты. Использование одинаковых программ, но разных костюмов у исполнителей.	1
14	Параллельное выполнение действий несколькими исполнителями. Дублирование исполнителей. Синхронное выполнение скриптов.	При параллельном выполнении скриптов результат работы всех исполнителей получается гораздо быстрее, чем в тех задачах, когда работал один исполнитель.	1
15	Разбиение программы на части для параллельного выполнения исполнителями. Таймер. Планирование действий разных исполнителей во	Сенсор «таймер» Сброс «таймера» Команда «ждать до ...» Кейс 8. Мини-проект «Смена времени суток». Создайте скрипт, в котором будет четыре сцены: «утро», «день», «вечер», «ночь». Используйте «таймер»	1

	времени для решения общей задачи.	для своевременного изменения сцен, своевременного запуска скриптов разных исполнителей.	
16	Два исполнителя со своими программами. Мини-проект «Часы с кукушкой». 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». Кейс 9. 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.	1
17	Алгоритмы с ветвлением. Условие ЕСЛИ.	Еще из сказок мы знаем, что на камне было написано: «Налево пойдешь – коня потеряешь, направо пойдешь – себя не найдешь, прямо пойдешь – в сказку попадешь» Блок-схема алгоритма с ветвлением. Полная форма ветвления. Команда «если _ или» Неполная форма ветвления. Команда «если _» Сенсор «касается» (касания края, или другого спрайта, или указателя мыши)	1
18	Цикл с условием. Мини-проект «Шарики в лабиринте»	Сенсор «касается цвета» Программируем отскок шариков от стен Кейс 10. Мини-проект «Шарики в лабиринте» Создайте сцену в виде бильярдного стола, на котором три шарика будут двигаться с разными скоростями и отталкиваться от бортиков.	1
19	Цикл с условием. Исполнитель определяет цвет. Сенсор «касается цвета»	Программируем поведение исполнителя в зависимости от цвета фона.	1
20	Оператор случайных чисел.	Команда «Выдать случайное число от _ до _». Случайные перемещения исполнителя в координатной плоскости. Случайное количество шагов. Случайные координаты. Поворот на случайный угол.	1
21	Перемещение исполнителей между слоями. Команда «Перейти в верхний слой»	Команда «Перейти назад на _ слоев» Эффект «Призрак»	1
22	Действия исполнителей в разных слоях.	Кейс 10. 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.	1
23	Взаимодействие исполнителей.	Исполнители касаются друг друга. Команда «Касается _» Поведение исполнителей при столкновении.	1

24	Последовательное выполнение команд исполнителями.	Команда «Передать» Команда «Когда я получу». Связи между программами разных исполнителей.	1
25	Программирование клавиш. Мини-проект «Лабиринт»	Взаимодействие пользователя с программой. Клавиши управления перемещением исполнителя Команда «Изменить значение x на _» Команда «Изменить значение y на _» Кейс 10. Игра «Лабиринт» Создайте сцену в виде лабиринта. С помощью клавиш со стрелками управляйте движением маленькой мышки. Мышка должна добраться до выхода, не задевая стенок лабиринта. В случае касания стенок мышка возвращается в исходную точку.	1
26	Управление событиями.	Передача сообщений между исполнителями и фоном. Запуск программ после получения сообщения.	1
27	Координатная плоскость. Геометрические фигуры.	Рисование геометрических фигур на координатной плоскости. Последовательное выполнение команд одним исполнителем. Параллельное выполнение команд несколькими исполнителями. Кейс 11. Исследование времени выполнения программ. Создайте одного исполнителя, рисующего квадраты, как показано на рисунке. Затем создайте четыре исполнителя, которые будут параллельно рисовать по три квадрата. Сравните время работы в этих двух случаях.	1
28	Координатная плоскость. Переменные.	Блок «Переменные». Рисование геометрических фигур на координатной плоскости с использованием переменных. Построение перпендикуляров к координатным осям.	1
29	Создание списков.	Название списка. Элементы списка. Длина списка. Команда «Создать список». Выбор элемента списка. Кейс 11. 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.	1
30	Использование подпрограмм.	Как сделать программу структурированной и более понятной. Команда «Передать _ и ждать» Команда «Играть звук _» Команда «Ноту _ играть _ тактов»	1
31	Сообщество Scratch. Scratch 2.0.	Регистрация на сайте. Личный кабинет. Публикация проектов.	1
32	Отладка программ с ошибками.	http://scratch.mit.edu/projects/10437040 http://scratch.mit.edu/projects/10437249 http://scratch.mit.edu/projects/10437366 http://scratch.mit.edu/projects/10437439 http://scratch.mit.edu/projects/10437476	1

33 - 34	Кейс 12. Итоговый проект.	<p>Учащиеся самостоятельно или в парах выполняют индивидуальный проект, согласованный с учителем. Как итог, можно провести конкурс проектов.</p> <p>Обучающийся научится:</p> <ul style="list-style-type: none"> • составлять сценарии проектов среды Scratch; • составлять алгоритмы, определять последовательность выполнения команд; • 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. 	2
------------	---------------------------	--	---