

Муниципальное образовательное учреждение
дополнительного образования
«Центр дополнительного образования детей»

Принята на заседании
методического совета
« 14 » 08 2017 г.
Протокол № 1

УТВЕРЖДАЮ
Директор МАОУДО «ЦДОД»
И.Ф. Сорочан
Приказ № 245 от "14" 08 2017 г.



Дополнительная общеобразовательная программа

«Юный программист»

Возраст учащихся: 9 – 13 лет

Срок реализации программы: 3 года

Автор/составитель:

Соловьева Елена Анатольевна,
педагог дополнительного
образования МАОУДО «ЦДОД»

Полярный

2017

1. Пояснительная записка.

Дополнительная общеобразовательная программа «Юный программист» (далее программа) имеет техническую направленность, она предназначена для получения учащимися дополнительных знаний в области программирования в игровой, увлекательной форме, используя языки программирования: Scratch, КуМир и Python.

Нормативные правовые акты и государственные программные документы в соответствии с которыми разработана дополнительная общеобразовательная программа:

1. федеральный закон Российской Федерации от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;

2. приказ Министерства образования и науки Российской Федерации от 29 августа 2013 года № 1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

3. письмо Министерства образования и науки Российской Федерации от 18 ноября 2015 года № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ»;

4. распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;

5. постановление Главного государственного санитарного врача Российской Федерации от 4 июля 2014 года № 41 г. Москва «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».

Тип программы - авторская.

Актуальность программы.

Большинство школьных программ по информатике в основном охватывают пользовательский уровень обучения работы на компьютере. Школьников учат работать с текстовыми документами, элементарной графикой, создавать презентации и так далее. Темам «алгоритмы» и «программирование» отводится небольшое количество часов, изучаются они поздно, в старших классах. Отсутствуют такие необходимые для развития логического мышления предметы, как алгоритмика, логика, которые были, когда учились более старшие поколения. Это замедляет формирование алгоритмического мышления и не способствует развитию интереса учащихся

к программированию. Парадокс ещё заключается в том, что наибольший бал в заданиях Единого Государственного Экзамена и во всех значимых олимпиадах, приносят как раз задания на программирование.

Обучение по данной программе создает благоприятные условия для более раннего «погружения» учащихся в мир логики, и позволяет сформировать у детей стойкий интерес к получению и усовершенствованию знаний в области программирования и IT - технологий, что очень актуально в современном образовательном процессе, также выявить способных к программированию детей и оказать помощь в профессиональном самоопределении.

Новизна программы.

Новизна представленной программы заключается, **во-первых**, в том, что она является первой ступенью непрерывного курса обучения детей программированию, дизайну и сайтостроению со ступени начального до старшего звена в «Школе программирования и дизайна».

В основе организации целостного педагогического процесса – известные принципы преемственности, последовательности и систематичности, но, в данном случае они предполагают такую организацию педагогического процесса, при которой существует взаимосвязь между образовательными программами, при которой последующий курс является более высокой ступенькой ранее проводимой работы, закрепляет и развивает достигнутое, поднимает обучающегося на более высокий уровень развития. Видение перспективы своего творческого развития способствует росту познавательного интереса.

«Школа программирования и дизайна» – это серия дополнительных общеобразовательных программ для детей 9 - 17 лет, основные направления которых - программирование, дизайн и сайтостроение.

Обучение построено по ступенчатому принципу: отучившись по программе «Юный программист» ребёнок может продолжить обучение на более старших ступенях: «Компьютерная графика и дизайн» и «Web – дизайн и программирование». В зависимости от подготовки ребенка возможен переход сразу на какую-либо ступень.

Во-вторых, в сочетании известных педагогических форм, методов, приёмов организации образовательного процесса с инновационными (мультимедийные учебные пособия, флеш-анимации, видеоролики, веб-

кейсы и другие интернет – технологии), что стимулирует развитие интереса обучающихся. Созданные проекты размещаются на Международном портале Scratch, одаренные учащиеся работают над индивидуальными проектами, с которыми участвуют в конкурсах по программированию.

Отличительные особенности программы.

Отличительными особенностями программы «Юный программист» являются:

1. Проектный подход.

Использование метода проектов позволяет обеспечить условия для развития у учащихся навыков самостоятельной постановки задач и выбора оптимального варианта их решения, самостоятельного достижения цели, анализа полученных результатов с точки зрения решения поставленной задачи.

2. Межпредметность.

В программе прослеживается тесная взаимосвязь с математикой, физикой, географией, русским языком, музыкой и другими предметами школьного цикла. Знания, полученные на других предметах, логичным образом могут быть использованы при разработке проектов.

3. Пропедевтика.

Отличительной особенностью программы является ранний возраст обучающихся программированию. Через разработку проектов учащиеся получают знания, обозначенные в программах более старших классов. Так, например, осваиваются основные алгоритмические конструкции (информатика), действия с десятичными дробями, отрицательными числами, понятие координатной плоскости (математика).

4. Коммуникация.

Огромным достоинством программы является возможность обучению навыкам работы в парах, группе, командах, создание коллективных проектов, чего практически невозможно достичь при изучении традиционных языков Бейсик и Паскаль. Обязательное условие — публичная презентация и защита проектов. Возможность увидеть результаты своего труда в Интернет также стимулирует интерес детей к обучению.

Обучение программированию по программе «Юный программист» осуществляется в свободно распространяемых средах Scratch, КуМир и Python. Для начала обучения детей программированию выбрана среда Scratch, так как данный язык очень прост для ребенка. Программы в Scratch не пишут, а собирают из готовых блоков, то есть снимается проблема работы с текстом программы, не нужно заучивать наизусть названия команд и уметь писать их без ошибок. Основной задачей становится разработка алгоритма и его воплощение в виде компьютерной игры, мультфильма, видеоролика. Освоив язык программирования Scratch ребятам будет легче изучать другие языки.

Вторым языком программирования выбран КуМир. В данной среде при кодировании алгоритма уже требуется изучить синтаксис языка, но все конструкции команды и переменные мы можем задавать простыми русскими словами, что облегчает написание программы для детей. Также плюсами КуМира являются: автоматическое выравнивание, показывающее структуру программы; осуществление постоянного контроля за правильностью, вводимой программы, на полях сообщается о всех обнаруженных ошибках; наглядная визуализация процесса выполнения программы. Среда КуМир обладает расширенным набором Исполнителей (Черепашка, Водолей, Робот, Чертежник) с разнообразными системами команд, несмотря на кажущуюся простоту в ней можно решать вполне достойные задачи. Таким образом, мы продолжаем формировать правильное представление об основных конструкциях языка (цикл, условие, переменная, подпрограмма) и закрепляем и развиваем навыки, полученные на первом этапе в Scratch. Приобретенные навыки программирования в среде КуМир могут быть в дальнейшем использованы при подготовке как к ОГЭ в 9 классе, так и ЕГЭ в 11 классе.

Третьим языком программирования выбран Python, так как это уже настоящий язык, а не просто учебный. Он применяется тысячами профессионалов по всему миру, используется для создания одних из самых мощных приложений на свете, например, Google Maps, YouTube и Gmail. Но тем не менее Python хорошо подходит и для изучения программирования. Это мощный современный язык, обладающий богатыми возможностями, и в тоже время высокой дружелюбностью к начинающим - прост, понятен, красив. По данным сайта tiobe.com, который ежемесячно оценивает популярность языков программирования, на основе подсчета результатов поисковых запросов, содержащих название языка и рассчитывает индекс ТЮБЕ, на данный момент Python располагается на 5 месте среди языков

программирования высокого уровня. В дальнейшем ребята без труда перейдут к освоению других популярных на сегодняшний день языков программирования: Java, C, C++, C#, JavaScript.

Педагогическая целесообразность программы заключается в том, что она является целостной и непрерывной в течение всего процесса обучения. Позволяет формировать у обучающихся не только логическое мышление, но и раскрывать в себе творческие возможности, развивать навыки работы с мультимедиа и самореализоваться в современном мире. Несмотря на очевидные преимущества обучения основам программирования в средах Scratch, КуМир, Python, в настоящее время они еще мало освоены, очень мало методических разработок и учебников по их освоению .

Цель программы:

создать комфортные условия для формирования и развития у обучающихся алгоритмического мышления в процессе изучения основ программирования в средах Scratch, КуМир и Python.

Задачи программы:

Образовательные:

- ✓ обучение основным базовым алгоритмическим конструкциям;
- ✓ освоение основных этапов решения задачи;
- ✓ обучение навыкам разработки, тестирования и отладки несложных программ;
- ✓ формирование навыков решения олимпиадных задач;
- ✓ знакомство с понятием проекта, его структуры, дизайна и разработки.

Развивающие:

- ✓ развивать познавательный интерес обучающихся;
- ✓ развивать творческое воображение, математическое и образное мышление учащихся;
- ✓ развивать умение работы с компьютерными программами;
- ✓ развивать навыки планирования проекта, умение работать в группе.

Воспитательные:

- ✓ воспитывать интерес к занятиям информатикой;
- ✓ воспитывать культуру общения между учащимися;

- ✓ воспитывать культуру безопасного труда при работе за компьютером;
- ✓ воспитывать культуру работы в глобальной сети.
- ✓ стимулирование интереса и склонности к выбору будущей профессии в сфере информатики и математики, а также в смежных областях.

Сроки реализации программы.

Содержание программы рассчитано на 3 года обучения.

1 год обучения – 108 часов (3 часа в неделю в течение учебного года).

2 год обучения – 108 часов (3 часа в неделю в течение учебного года).

3 год обучения – 108 часов (3 часа в неделю в течение учебного года).

Характеристика обучающихся.

Дополнительная общеобразовательная программа «Юный программист» предназначена для обучения детей в возрасте 9-13 лет. Учащиеся набираются по желанию. Наполняемость групп: число учащихся в группах - 12 человек.

Формы и режим занятий.

Основными формами организации образовательного процесса являются:

- ✓ лекции, в том числе с использованием наглядных средств обучения;
- ✓ беседы;
- ✓ викторины;
- ✓ дискуссии;
- ✓ практические занятия на компьютере;
- ✓ конкурсы;
- ✓ игры;
- ✓ самостоятельная работа учащихся;
- ✓ соревнования;
- ✓ защита проектов.

Программа реализуется в течение всего календарного года, включая каникулярное время.

Число занятий – 2 раза в неделю - 3 академических часа:

1 раз - 1 академический час;

2 раз - 2 академических часа.

Работа с использованием компьютера, с перерывами на теорию, лекции, обсуждение проектов, эвристические беседы, дискуссии, а также перерывами между академическими часами 10 минут.

Ожидаемые результаты по итогам обучения.

**По окончании изучения теории учащиеся 1 года обучения должны
знать:**

- ✓ правила безопасной работы;
- ✓ что такое исполнитель, среда исполнителя, команды исполнителя;
- ✓ что такое алгоритм, скрипт, программа;
- ✓ типы алгоритмов (линейный, разветвляющийся, циклический);
- ✓ интерфейс программы Scratch;
- ✓ логические операции;
- ✓ координаты на плоскости;
- ✓ понятие проект, его структура и реализация в среде Scratch.

**По окончании освоения практической части учащиеся 1 года
обучения должны уметь:**

- ✓ составлять алгоритмы любого типа;
- ✓ оформлять алгоритмы в изучаемой среде программирования;
- ✓ тестировать и отлаживать созданную программу;
- ✓ создавать личные проекты в среде Scratch;
- ✓ использовать сеть для обмена материалами работы.

**По окончании изучения теории учащиеся 2 года обучения должны
знать:**

- ✓ общий подход к решению задач по программированию, разнообразие способов решения задач;
- ✓ основные виды алгоритмов;
- ✓ основы алгоритмического языка КуМир;
- ✓ графические возможности КуМир;
- ✓ понятие «хороший стиль программирования».

**По окончании освоения практической части учащиеся 2 года
обучения должны уметь:**

- ✓ писать простейшие программы на алгоритмическом языке.
- ✓ создавать программы, рисующие несложные картинки.
- ✓ конструировать многомодульные программы.
- ✓ создавать читаемые программы (т.е. использовать «хороший стиль программирования»).

**По окончании изучения теории учащиеся 3 года обучения должны
знать:**

- ✓ основы языка программирования Python;
- ✓ структуру программы на языке Python;
- ✓ типы данных в Python, понятия - переменная, список, кортеж, словарь, множество;
- ✓ циклы и ветвления;
- ✓ функции и модули в Python;
- ✓ зачем нужны классы и объекты в Python.

**По окончании освоения практической части учащиеся 3 года
обучения должны уметь:**

- ✓ писать самостоятельно несложные программы на языке программирования Python;
- ✓ подключать стандартные модули в свои программы;
- ✓ работать с переменными, списками, кортежами, словарями, множествами, конструкциями циклов и ветвлений;
- ✓ создавать графический интерфейс пользователя (GUI) с помощью модулей.

Формы промежуточной и итоговой аттестации.

Промежуточная и итоговая аттестация проводится в следующих формах подведения итогов:

- ✓ опрос;
- ✓ зачёт;
- ✓ тестирование;
- ✓ самостоятельная работа;
- ✓ открытое занятие для родителей;
- ✓ участие в тематических конкурсах, играх, соревнованиях.

Таблицы (КИМ), в которых фиксируются результаты аттестации по полугодиям в приложении.

Способы определения результативности.

С целью диагностики успешности освоения программы, достижений обучающихся и коррекции образовательного процесса применяются следующие методы:

- педагогическое наблюдение за обучающимися при выполнении заданий, в ходе работы над проектом;
- педагогический анализ выполнения творческих работ, результатов тестирования, анкетирования, участия в тематических играх, конкурсах, соревнованиях.
- педагогический мониторинг (диагностика развития личностных качеств обучающихся, диагностика освоения ЗУНов).

Реализуется программа в ЦДОД с 2015 года. Результативность программы подтверждена победами учащихся в конкурсах, олимпиадах и конференциях разного уровня.

2. Учебно-тематический план (1 год обучения).

Программа включает следующие основные темы:

№	Наименование раздела	Количество часов		
		Теория	Практика	Всего часов
1. Введение в мир программирования.				
1.	Введение в мир программирования.	1	1	2
2.	Думай как компьютер.	1	1	2
3.	Что такое компьютерная программа.	1		1
Итого		3	2	5
2. Основные приемы программирования и создания проектов в среде Scratch.				
4.	Знакомство со средой Scratch.	1	1	2
5.	Спрайты. Библиотеки костюмов и сцен.	1	1	2
6.	Графический редактор Scratch.	1	2	3
7.	Учимся управлять спрайтами.	1	1	2
8.	Создание спецэффектов - команды «Внешность».	1	2	3
9.	Навигация в среде Scratch.	1	2	3
10.	Команды группы «Перо».	1	2	3
11.	Команды управления. Простые циклы.	1	2	3
12.	Вложенные циклы.	1	2	3
13.	Команды группы - «Звук».	1	2	3
14.	Звуковые события. Распознавание движения.	1	1	2
15.	Обмен сообщениями.	1	2	3
16.	Блоки Scratch: обзор и типы данных.	1	1	2
17.	Математические операторы.	1	2	3
18.	Проект «Генератор кубиков».		2	2
19.	Считывание и распознавание.	1	2	3
20.	Истина или ложь.	1	2	3
21.	Принятие решений и ветвление.	1	2	3
22.	Проект «Простой Paint».	1	2	3
23.	Игра «Лабиринт».		2	2
24.	Как не заикнется.	1	2	3
25.	Команды группы - «Данные». Переменные.	1	2	3

26.	Строки и списки.	1	2	3
27.	Создание блоков.	1	2	3
Итого		22	43	65
3. Разработка групповых проектов в среде Scratch.				
28.	Создание мультфильмов в среде Scratch.		3	3
29.	Создание комиксов в среде Scratch.		3	3
30.	Создание интерактивных поздравительных открыток в среде Scratch.		3	3
31.	Создание презентаций в среде Scratch.		3	3
32.	Создание компьютерных игр в среде Scratch.		2	2
33.	Создание интерактивных игр.		3	3
34.	Создание интерактивных тестов.		3	3
35.	Создание музыкального клипа в среде Scratch.		3	3
36.	Создание учащимися разных проектов.		3	3
Итого			26	26
4. Выполнение творческой работы - создание личного проекта в среде Scratch.				
37.	Разработка личного проекта.	1		1
38.	Дизайн проекта.		3	3
39.	Программируем проект.		3	3
40.	Тестирование и отладка проекта.		2	2
41.	Защита проекта.		3	3
Итого		1	11	12
Итого		26	82	108

Учебно-тематический план (2 год обучения).

Программа включает следующие основные темы:

№	Наименование раздела	Количество часов		
		Теория	Практика	Всего часов
1. Закрепление пройденного в среде Scratch.				
1.	Устройство компьютера. Правила техники безопасности.	1	2	3
2.	Путешествие в страну «Программию» с котенком Скретчем.	1	2	3
3.	Компьютерная графика.	1	2	3
4.	Волшебство программирования.		3	3
5.	Атака вируса.		3	3
6.	Зоопарк исполнителей.		3	3
7.	Последние испытания.		3	3
Итого		3	18	21
2. Создание новых проектов в среде Scratch.				
8.	Создание музыки в среде Scratch.		3	3
9.	Создание мультфильмов в среде Scratch.		3	3
10.	Создание комиксов в среде Scratch.		3	3
11.	Создание поздравительных открыток в среде Scratch.		3	3
12.	Создание интерактивных презентаций в среде Scratch.		3	3
13.	Создание компьютерных игр в среде Scratch.		3	3
14.	Создание интерактивных тестов.		3	3
Итого			21	21
3. Основные приемы программирования и создания проектов в среде КуМир.				
15.	Алгоритмы и исполнители.	1	2	3
16.	Знакомство с системой КуМир.	2	1	3
17.	Среда КуМир. Исполнитель Черепаха.	1	2	3
18.	Среда КуМир. Исполнитель Кузнечик.	1	2	3
19.	Среда КуМир. Исполнитель	1	2	3

	Водолей.			
20.	Среда КуМир. Исполнитель Робот.	1	2	3
21.	Вспомогательные алгоритмы.	1	2	3
22.	Проектирование сверху вниз и снизу вверх.	1	2	3
23.	Метод последовательного уточнения.	1	2	3
24.	Цикл пока.	1	2	3
25.	Команда ветвления.	1	2	3
26.	Циклы и условия.	1	2	3
27.	Тестирование алгоритмов.	1	2	3
28.	Исполнитель Чертежник.	1	2	3
29.	Рисуем с Чертежником.	1	2	3
30.	Математические операции и функции в среде КуМир.	1	2	3
31.	Основные этапы разработки проекта.	1	2	3
Итого		18	33	51
4. Создание личного проекта.				
32.	Разработка личного проекта.		3	3
33.	Дизайн проекта.		3	3
34.	Программируем проект.		3	3
35.	Тестирование и отладка проекта.		3	3
36.	Защита проекта.		3	3
Итого			15	15
Итого		21	87	108

Учебно-тематический план (3 год обучения).

Программа включает следующие основные темы:

№	Наименование раздела	Количество часов		
		Теория	Практика	Всего часов
1. Язык Python. Основы.				
1.	Введение в язык программирования Python.	2	1	3
2.	Ввод и вывод данных программам на языке Python.	1	2	3
3.	Переменные и типы данных в Python.	2	1	3
4.	Математика в Python.	1	2	3
5.	Подключение модуля - turtle: «Черепашья» графика.	1	2	3
6.	Цикл с параметром в Python.	2	1	3
7.	Продолжаем работу с циклом for.	1	2	3
8.	Вложенные циклы в Python.	1	2	3
9.	Цикл с предусловием в Python.	1	2	3
10.	Цикл с постусловием в Python.	1	2	3
11.	Преобразование одной инструкции цикла в другую.	1	2	3
12.	Условия (что если?).	1	2	3
13.	Сложные условия. Каскадные ветвления.	2	1	3
14.	Переменные без значения — None и комментарии в Python.	2	1	3
15.	Встроенные структуры данных: списки, кортежи, словари и множества.	1	2	3
16.	Списки мощнее строк.	1	2	3
17.	Кортежи в Python.	1	2	3
18.	Двоичный поиск и сортировка.	1	2	3
19.	Словари в Python — не для поиска слов. Множества.	2	1	3
20.	Функции в Python.	1	2	3
21.	Расширяем знания о функциях.		3	3
22.	Объекты и классы.	1	2	3
23.	Модули Python.	1	2	3

24.	Еще больше стандартных модулей Python.	1	2	3
25.	Графический интерфейс пользователя (GUI).	2	1	3
26.	Создание простой анимации с модулем tkinter.		3	3
Итого		31	47	78
2. Программирование игр: коддинг для развлечения.				
27.	Первая игра: «Пинг-понг» с графическим интерфейсом.		3	3
28.	Продолжаем работать над игрой «Пинг-понг».		3	3
29.	Доделываем «Пинг-понг».		3	3
30.	Разработка игр с модулем Pygame.	1	2	3
31.	Разработка игры «Стреляющий корабль».		3	3
32.	Продолжаем работать над игрой «Стреляющий корабль».		3	3
33.	Учим наш корабль стрелять.		3	3
34.	Добавляем врагов в игру «Стреляющий корабль».		3	3
35.	Учим врагов перемещаться в игре.		3	
36.	Завершение игры «Стреляющий корабль».		3	
Итого		1	29	30
Итого		31	77	108

3. Содержание дополнительной общеобразовательной программы для детей (1 год обучения).

Тема 1: Введение в мир программирования (2 часа).

Теория - 1 час

Знакомство. Цель, задачи программы. План работы на учебный год. Режим занятий. Первичный инструктаж по ОТ, ПБ, ГО и ЧС. Первичный инструктаж по темам: «Правила поведения в МАОУДО «ЦДОД», «Охрана жизни и здоровья учащихся на учебных занятиях». Демонстрация примеров проектов, сделанных в среде Scratch обучающимися объединения «Юный программист».

Практика – 1 час

Анкетирование с целью выявления интересов и ожиданий. Входная диагностика. Собеседование.

Тема 2: Думай как компьютер (2 часа).

Теория - 1 час

Понятие исполнителя, алгоритма. Основные характеристики исполнителя. Способы управления исполнителем. Просмотр видеоролика «Алгоритм». Способы записи алгоритма.

Практика – 1 час

Тренировочные и творческие задания по алгоритмам, описанным различными способами. Игра «Перевозчик». Самостоятельная работа.

Тема 3: Что такое компьютерная программа (1 час).

Теория - 1 час

Компьютер как универсальный исполнитель. Понятие программы, назначение, и использование. Как стать программистом. Просмотр видеоролика «Программа, программист, программирование». Виды программ, отличие программы от скрипта.

Тема 4: Знакомство со средой Scratch (2 часа).

Теория - 1 час

Что такое Scratch. Установка и запуск Scratch. Основные элементы интерфейса программы Scratch. Меню и инструменты. Основные характеристики исполнителя, их особенности в среде исполнителя Scratch. Цветные блоки. Назначение блоков. Скрипты, порядок выполнения скриптов.

Практика – 1 час

Установка Scratch. на свой компьютер. Создание, сохранение и открытие проектов в среде Scratch. Настройка среды. Знакомство с проектами сообщества Scratch.

Тема 5: Спрайты. Библиотеки костюмов и сцен (2 часа).

Теория - 1 час

Спрайты. Информация спрайта. Что могут спрайты. Операции над спрайтами. Костюмы. Выбор костюмов. Создание декораций - фонов. Изменение внешнего вида спрайта и фона. Импорт изображений. Экспорт спрайтов и их использование в проектах.

Практика – 1 час

Творческие задания для детей - создание исполнителей и фонов в проекте: «Круглые рисунки». Создание костюмов. Пользуемся помощью интернета.

Тема 6: Графический редактор Scratch (3 часа).

Теория - 1 час

Растровая и векторная графика. Особенности работы с растровым и векторным режимами графического редактора Scratch. Общие инструменты рисования. Обозначение центра изображения. Установка прозрачных цветов. Инструменты растрового и векторного режимов рисования. Слои изображения. Группировка фигур.

Практика – 2 часа

Рисуем в Scratch в растровом режиме: домик, кувшинку, кораблик, рыбку, робота. Рисуем в векторном режиме: машинку, монстра. Сохраняем спрайты для их использования в проектах.

Тема 7: Учимся управлять спрайтами (2 часа).

Теория - 1 час

Основные команды движения и события. Коричневые блоки - команды группы «События» - «Когда щёлкнут по флажку» и «Когда клавиша на клавиатуре нажата». Синие блоки - команды группы «Движение». Знакомство с отрицательными числами. Движение задом наперед.

Практика – 1 час

Создание алгоритма первого проекта в среде Scratch. Анимация «Научим кота бегать», запуская скрипт разными событиями: клик мышкой, нажатие разных клавиш. Задание: «Автомобиль с пятью скоростями». Выполнение дополнительных заданий.

Тема 8: Создание спецэффектов - команды «Внешность» (3 часа)

Теория - 1 час

Фиолетовые блоки - «Внешность» - блоки управления внешностью для спрайтов и сцены. Что такое анимация. Анимирование костюма. Эффекты изображений: учим спрайты исчезать и появляться - телепортация с помощью эффектов, изменять цвет и т. д. Размер и видимость. Диалоги - спрайты умеют думать и говорить. Слои.

Практика – 2 часа

Задания: «Осьминог», «Танцующая балерина», «Бегущий человек», «Оживляем динозавра», «Программируем страшилки» - спрайты неожиданно появляются и исчезают, как привидения.

Тема 9: Навигация в среде Scratch (3 часа).

Теория - 1 час

Навигация . Координатная плоскость. Что такое система координат. Знакомимся с координатой X. Знакомимся с координатой Y. Перемещение объектов по горизонтали. Перемещение объектов по вертикали. Что такое угол поворота. Крутой поворот. Вращение спрайтов. Блоки перемещения по координатной плоскости.

Практика – 2 часа.

Ориентируемся по координатам. Задания: расшифруй пословицы, определи координаты точек, нарисуй рисунок по координатам. Рисование по координатам в среде Scratch. Задания с использованием команд движения: «Движение царпки по периметру сцены», «Движение по диагонали на нажатие клавиш».

Тема 10: Команды группы «перо» (3 часа).

Теория - 1 час

Темно-зеленые блоки - группа «Перо». Спрайты умеют рисовать. Перьевая черепаха. Исследуем печать.

Практика – 2 часа.

Игры с пером. Рисуем разные фигуры, буквы, лесенки, штрихпунктирные линии, спирали, цветные пятна, узоры из фигур. Задания: «Конфетти», «След от полёта», «Паук плетёт паутину».

Тема 11: Команды управления. Простые циклы (3 часа).

Теория - 1 час

Желтые блоки - команды группы «Управление». Паузы. Простые циклы. Цикл «Повторить». Запись цикла со счетчиком в виде блок – схемы. Понятия: «Заголовок цикла», «Тело цикла». Циклы и эффект цвета. Циклы и эффект призрака. Вращение. Бесконечный цикл. Автоматическая печать.

Практика – 2 часа.

Учим спрайты совершать повторяющиеся действия. Создаём «Вечный двигатель». Усовершенствуем ранее созданные программы в Scratch. Создаём новые: «Гонки», «Полет самолета», «Разноцветный экран».

Тема 12: Вложенные циклы (3 часа).

Теория - 1 час

Вложенные циклы. Работа с алгоритмами. Советы и секреты.

Практика – 2 часа.

Задания: «Вращающиеся квадраты», «Танцующий динозавр», «Снежинка», «Цветок», «Разные орнаменты», «Создаем рисунки из многоугольников».

Тема 13: Команды группы - «Звук»(3 часа).

Теория - 1 час

Добавление звуков к спрайтам. Проигрывание звука. Запись звука. Как проигрывать аудиофайлы. Переворачиваем звуки. Игра на барабанах и другие звуки. Создание своей музыки. Полный звукоряд. Устанавливаем темп и такт. Контроль громкости звука. Ноты. Как рисуют ноты.

Практика – 2 часа.

Задания: «Организуем ансамбль», «Танцы на сцене», «Фейерверки». Создаем песни: «Маленькой ёлочке», «Чижик-пыжик», «Гуси» и другие.

Тема 14: Звуковые события. Распознавание движения (2 часа).

Теория - 1 час

Коричневые блоки - команды группы «События» - «Когда громкость больше параметра», «Когда движение видео больше параметра», «Когда фон меняется».

Практика – 1 час.

Задания: «Учим кота реагировать на звук, на изменение фона и на распознавание движения».

Тема 15: Обмен сообщениями (3 часа).

Теория - 1 час

Как осуществить связь между спрайтами в виде беседы. Коричневые блоки - команды группы «События» - «Когда я получу сообщение», «Передать сообщение», «Передать сообщение и ждать». Создаем процедуры (подпрограммы) при помощи передачи сообщений.

Практика – 2 часа.

Задания: «Рисование квадратов случайных цветов в ответ на передачу сообщения», «Бесконечное движение между фонами», «Лампа», «Презентация». Создаем проекты: «Дюймовочка», или «Путешествие в страну безопасного интернета» или на свою тему.

Тема 16: Блоки Scratch: обзор и типы данных. (2 часа)

Теория - 1 час.

Командные блоки. Блоки-функции (репортеры). Блоки-триггеры. Контрольные блоки. Что означает форма блока. Разновидности данных в Scratch. Числовые данные. Строковые данные.

Практика – 1 час.

Проект «Кот и птичка в клетке». Создание героев и декораций проекта. Программирование проекта. Отладка программы.

Тема 17: Математические операторы (3 часа).

Теория - 1 час.

Светло-зелёные блоки - группа «Операторы». Арифметические операторы. Случайные числа. Математические функции.

Практика – 2 часа.

Решаем задачи, используя операторы. Задания: «Простая арифметика со Скретчем», «Делаем кота большим и маленьким», «Рисующий волшебник», «Барабан», «Расширяющаяся спираль».

Тема 18: Проект «Генератор кубиков»(2 часа).

Практика – 2 часа.

Проект со случайными числами «Генератор кубиков». Рисование спрайтов и фона проекта. Программирование проекта.

Тема 19: Считывание и распознавание (3 часа).

Теория - 1 час.

Голубые блоки - команды группы «Сенсоры». Сенсоры восприятия внутренних и внешних событий: управление с клавиатуры, столкновения спрайтов, касания, сенсор общения с человеком, для управления временем в проектах и т. д.

Практика – 2 часа.

Использование блоков группы «Сенсоры». Создаём игры: «Кошки-мышки», «Игра с пончиком».

Тема 20: Истина или ложь (3 часа).

Теория - 1 час

Операторы сравнения. Логические операторы: блоки «и», «или», «не». Объединение вопросов. Соблюдение условий. Использование логических операторов для проверки областей числовых значений.

Практика – 2 часа.

Самостоятельная работа: понятия «Истина» и «Ложь». Проекты с использованием блоков «и», «или», «не»: «Сравнение десятичных дробей», «Угадай мои координаты».

Тема 21: Принятие решений и ветвление (3 часа).

Теория - 1 час

Использование блока «если». Структура блока «если». Использование переменных как флагов. Использование блока «если/иначе». Структура блока «если/иначе». Вложенные блоки «если» и «если/иначе».

Практика – 2 часа.

Работа с алгоритмами с ветвлением. Проекты: «Осторожно лужи», «Управляемый робот», «Тренажер памяти».

Тема 22: Проект «Простой Paint»(2 часа).

Практика – 2 часа.

Рисуем интерфейс программы для рисования: фон, цветные карандаши и ластик. Создаём скрипты, проверяем работу программы.

Тема 23: Игра «Лабиринт»(2 часа).

Практика – 2 часа.

Создание игры - «Лабиринт» - три варианта.

Тема 24: Как не заикнется (3 часа).

Теория - 1 час

Сложные условия. Блок «повторять, пока не выполнится условие».. Создание блока «всегда если». Блок «ожидание». Стоп-команды: блок «стоп».

Практика – 2 часа.

Создание проектов: «Часы», «Стрельба по воробьям».

Тема 25: Команды группы - «Данные». Переменные (3 часа).

Теория - 1 час

Оранжевые блоки - группа команд «Данные». Понятие - переменная. Создание, переименование и удаление переменной. Использование переменных. Присвоение и изменение значений. Сравнение переменных. Константа. Датчики.

Практика – 2 часа.

Создание проектов:«С переменной время», «С переменной счёт», «С переменной прыжок», «Идеальный вес», «Оптические иллюзии».

Тема 26: Строки и списки (3 часа).

Теория - 1 час

Работа со словами. Строковые операторы. Подсчет специальных символов в строке. Создание списков. Команды управления списками. Доступ к элементам списка. Динамические списки.

Практика – 2 часа.

Игры со списками: «Угадай слово», «Замена букв».

Тема 27: Создание блоков (3 часа).

Теория - 1 час

Создание своего блока. Подпрограмма (процедура). Блоки с параметрами. Переменные в подпрограммах. Вложенные процедуры.

Практика – 2 часа.

Создание игр и тестов: «Звёздное небо», «Психологический тест».

Тема 28: Создание мультфильмов в среде Scratch (3 часа).

Практика – 3 часа.

Мультфильмы по сказкам «Колобок», «Репка».

Тема 29: Создание комиксов в среде Scratch (3 часа).

Практика – 3 часа.

Создание комиксов «Охота за приведениями», «Мой город».

Тема 30: Создание интерактивных поздравительных открыток в среде Scratch (3 часа).

Практика – 3 часа.

Создание открыток «С днём рождения», «С Новым годом».

Тема 31: Создание презентаций в среде Scratch (3 часа).

Практика – 3 часа.

Презентации: «Мои увлечения», «Моя семья».

Тема 32: Создание компьютерных игр в среде Scratch (2 часа).

Практика – 2 часа.

Проект «Убегая от преследователя», «Пора в космос».

Тема 33: Создание интерактивных игр (3 часа).

Практика – 3 часа.

Игра «Бешенные обезьяны», «Интерактивное общение».

Тема 34: Создание интерактивных тестов (3 часа).

Практика – 3 часа.

Тесты по школьным предметам: «Играем в города по географии», «Зоопарк по биологии».

Тема 35: Создание музыкального клипа в среде Scratch (3 часа).

Практика – 3 часа.

Создание клипов в среде Scratch: «Говорящий алфавит», «Как незнайка учил ноты».

Тема 36: Создание учащимися разных проектов (3 часа).

Практика – 3 часа.

Проекты «Театр в Scratch», «Квест в Scratch», «Перевертыши».

Тема 37: Разработка личного проекта (1 час)

Теория - 1 час

Понятие проект, его структура и реализация в среде Scratch. Этапы разработки проекта (постановка, алгоритмизация, кодирование, тестирование, отладка).

Тема 38:Дизайн проекта. (3 часа).

Практика – 3 часа.

Создание материалов для проекта.

Тема 39:Программируем проект. (3 часа).

Практика – 3 часа.

Разработка и создание программы своего проекта.

Тема 40:Тестирование и отладка проекта. (2 часа).

Практика – 2 часа.

Проверяем и исправляем ошибки.

Тема 41:Защита проекта (3 часа).

Практика – 3 часа.

Представляем свой проект. Публикуем проект в интернете.

Содержание дополнительной общеобразовательной программы для детей (2 год обучения).

Тема 1: Устройство компьютера. Правила техники безопасности.

Теория - 1 час

Цель, задачи программы, план работы на 2 год обучения. Режим занятий. Инструктаж «Правила поведения в МАОУДО «ЦДОД», «Охрана жизни и здоровья учащихся на учебных занятиях». Изучение темы «Устройство компьютера».

Практика – 2 часа

Анкетирование с целью выявления интересов и ожиданий. Входная диагностика. Участие в викторине «Что мы знаем о компьютерах».

Тема 2: Путешествие в страну «Программию» с котенком Скретчем.

Теория - 1 час

Повторение основ программирования в среде Scratch. Знакомство со страной «Программия».

Практика – 2 часа

Игра «Догонялки».

Тема 3: Компьютерная графика.

Теория - 1 час

Графическая информация. Растровая графика. Векторная графика. Цвет в компьютерной графике. Кодирование графики. Измерение информационного объема текста в стандартной кодировке.

Практика – 2 часа

Викторины: «Пиксельная головоломка», «Шифруем все».

Тема 4: Волшебство программирования.

Практика – 3 часа

Создание игры «Атака вируса».

Тема 5: Погоня за золотом.

Практика – 3 часа

Создание игры «Погоня за золотом».

Тема 6: Зоопарк исполнителей.

Практика – 3 часа

Создание игр «Лабиринт», «Черепашка».

Тема 7: Последние испытания.

Практика – 3 часа

Создание интерактивного квеста: «Найди 20 предметов за 1 минуту».

Тема 8: Создание музыки в среде Scratch.

Практика – 3 часа

Игры «Домисолька», «Послушай мелодию и воспроизведи её».

Тема 9: Создание мультфильмов в среде Scratch.

Практика – 3 часа

Создание своего мультфильма на свободную тему.

Тема 10: Создание комиксов в среде Scratch.

Практика – 3 часа

Создание своего комикса на свободную тему.

Тема 11: Создание поздравительных открыток в среде Scratch.

Практика – 3 часа

Создание своей открытки на свободную тему.

Тема 12: Создание интерактивных презентаций в среде Scratch.

Практика – 3 часа

Создание своего мультфильма на свободную тему.

Тема 13: Создание компьютерных игр в среде Scratch.

Практика – 3 часа

Создание своей компьютерной игры на свободную тему.

Тема 14: Создание интерактивных тестов.

Практика – 3 часа

Создание своего интерактивного теста на свободную тему.

Тема 15: Алгоритмы и исполнители.

Теория - 1 час

Жизнь как алгоритм. Алгоритм и его свойства. Формы записи алгоритмов. Блок-схемы.

Практика – 2 часа

Работа с блок-схемами. Выполнение вычислений по блок-схемам. Отгадывание чисел. Игра «Исполнитель и программист».

Тема 16: Знакомство с системой КуМир.

Теория - 2 час

Структура программы в среде КуМир. Отступы в языке КуМир. Исполнители в среде КуМир.

Практика – 1 часа

Создание, сохранение и открытие проектов в среде КуМир. Знакомство с исполнителями. Игра «Программа без ошибок».

Тема 17: Среда КуМир. Исполнитель Черепаха.

Теория - 1 час

Знакомство с исполнителем Черепаха и с его СКИ.

Практика – 2 часа

Осваивать среду исполнителя Черепаха, управлять движением исполнителя с помощью пульта. Составлять маршрут движения и записывать его на языке исполнителя. Использовать переменные при составлении программ.

Тема 18: Среда КуМир. Исполнитель Кузнечик.

Теория - 1 час

Исполнитель Кузнечик в среде КуМир.

Практика – 2 часа

Задания с исполнителем Кузнечик. Игры «Кузнечик на поле», «Кузнечик собирает буквы, что получится».

Тема 19: Среда КуМир. Исполнитель Водолей.

Теория - 1 час

Исполнитель Водолей в среде КуМир. Отладка программы на языке КуМир.

Практика – 2 часа

Программы для Водолея: «Сколько раз?», «Какие команды?».

Тема 20: Среда КуМир. Исполнитель Робот.

Теория - 1 час

Исполнитель Робот в среде КуМир. Циклы n раз в КуМир.

Практика – 2 часа

Задачи: «Робот иди (лабиринт в форме креста), (лабиринт 4x4)», «Прогулка робота», «Стертая команда».

Тема 21: Вспомогательные алгоритмы.

Теория - 1 час

Робот: линейные алгоритмы, простые циклы. Робот: вспомогательные алгоритмы.

Практика – 2 часа

Задачи на использование вспомогательных алгоритмов. Задача «Фигура из фрагментов».

Тема 22: Проектирование сверху вниз и снизу вверх.

Теория - 1 час

Проектирование программы «сверху» и «снизу».

Практика – 2 часа

Задачи «Эксперименты с программами», «Кто получит пятерку». Задачи для самостоятельного решения.

Тема 23: Метод последовательного уточнения.

Теория - 1 час

Метод последовательного уточнения. Команды-вопросы робота.

Практика – 2 часа

Задачи «Робот, что будет нарисовано?», «Эксперименты с программами», «Узоры из фрагментов», «Робот отвечает на вопросы». Задачи для самостоятельного решения.

Тема 24: Цикл пока.

Теория - 1 час

Циклы с условием в КуМир. Свойства цикла пока. Составные логические условия.

Практика – 2 часа

Задачи «Робот снова отвечает на вопросы», «В какой клетке окажется робот», «Закрасить клетки и вернуться», «Робот закрашивает прямоугольник», «Как избежать заикливания».

Тема 25: Команда ветвления.

Теория - 1 час

Ветвление в языке КуМир. Блок-схемы «Полная и краткая формы ветвления».

Практика – 2 часа

Примеры с ветвлением «Какая клетка будет закрашена», «Где окажется робот».

Тема 26: Циклы и условия.

Теория - 1 час

Циклы и условия в языке КуМир. Блок-схемы.

Практика – 2 часа

Тестирование. Задача «Робот размечает клетки коридора».

Тема 27: Тестирование алгоритмов.

Теория - 1 час

Тестирование алгоритмов, фрагментов алгоритма.

Практика – 2 часа

Задачи «Робот в лабиринте», «Робот должен попасть в угол», «Как погиб робот?»

Тема 28: Исполнитель Чертежник.

Теория - 1 час

Исполнитель Чертежник в среде КуМир. Прямоугольная система координат. Абсолютное и относительное смещение.

Практика – 2 часа

СКИ Чертежник. Задачи «Чертежник смещается в точки», «Чертежник рисует цифры».

Тема 29: Рисуем с Чертежником.

Теория - 1 час

Рисуем с Чертежником. Метод последовательного уточнения.

Практика – 2 часа

Задачи «Ломанная», «Фигура», «Симметричная фигура», «Улица из трех домиков»

Тема 30: Математические операции и функции в среде КуМир.

Теория - 1 час

Математические операции в среде КуМир. Примеры случайных событий. Работать с функциями случайных чисел в языке КуМир. Правила записи математических выражений.

Практика – 2 часа

Проект- игра «Угадай число»

Тема 31: Основные этапы разработки проекта.

Теория - 1 час

Основные этапы разработки проекта. План работы над проектом.

Практика – 2 часа

Составляем план работы над групповым проектом.

Тема 32: Разработка личного проекта.

Практика – 3 часа

Составляем план работы над личным проектом. Постановка задачи. Выбор темы. Продумываем дизайн проекта.

Тема 33: Дизайн проекта.

Практика – 3 часа

Создание материалов для дизайна проекта в среде КуМир.

Тема 34: Программируем проект.

Практика – 3 часа

Разработка и создание программы своего проекта в среде КуМир.

Тема 35: Тестирование и отладка проекта.

Практика – 3 часа

Проверяем и исправляем ошибки. Групповая проверка созданного проекта. Устранение ошибок.

Тема 36: Защита проекта.

Практика – 3 часа

Представляем свой проект, демонстрируем перед классом. Публикуем проект в интернете.

Содержание дополнительной общеобразовательной программы для детей (3 год обучения).

Тема 1: Введение в язык программирования Python (3 часа).

Теория - 2 часа

Цель, задачи программы, план работы на 2 год обучения. Режим занятий. Инструктаж «Правила поведения в МАОУДО «ЦДОД», «Охрана жизни и здоровья учащихся на учебных занятиях». История и особенности Python. Универсальность. Простота и минимализм. Кроссплатформенность и интеграция. Версии языка Python. Интерпретация и компиляция. Интегрированные среды разработки (IDE). Домашняя страница Python (<http://python.org/>). Назначение окон среды программирования. Среда программирования WING IDE.

Практика – 1 час

Скачивание Python. Установка Python. Первый запуск среды разработки. Тестирование Python. Написание первой программы. Сохранение и запуск Python-программ. Скачивание и установка среды разработки WING IDE.

Тема 2: Ввод и вывод данных программам на языке Python (3 часа).

Теория - 1 час

Структура программы на языке Python. Ключевые слова и идентификаторы в Python. От ввода до вывода. Аналоги элементов языка Scratch в Python. Считывание данных с клавиатуры: функция input(). Вывод данных на экран: функция print(). Дружественность интерфейса. Разделители. Форматирование вывода.

Практика – 2 часа

Задания: «На ввод и вывод данных», «На использование разных разделителей». Создание ошибки. Учимся понимать подсветку синтаксиса. Ищем информацию в сети Интернет по ошибке. Тестирование по теме.

Тема 3: Переменные и типы данных в Python (3 часа).

Теория - 2 часа

Переменные как ярлыки для данных. Инструкция присваивания. Выбор имен и использование переменных. Обмен переменных значениями в Python. Основные типы данных в Python: целочисленный тип (int), вещественные числа (float), символьные строки (str). Типизация в языке Python. Функции str (), int() и float (). Вспоминаем, что такое линейный алгоритм. Блок-схема линейного алгоритма.

Практика – 1 час

Задания на использование переменных: «Написать программу смены значений двух переменных местами», «Персональный привет», «Python заполняет экран информацией о пользователе», «Чепуха», «На преобразование типов данных.»

Тема 4: Математика в Python (3 часа).

Теория - 1 час

Запись арифметических выражений в языке Python. Операции обработки числовых данных. Операнды. Приоритет выполнения арифметических операций. Множественное присваивание. Основные операции с вещественными числами. Основные операции со строками. Конкатенация и репликация строк. Переменные внутри строк. Функция len().

Практика – 2 часа

Задания: «Программа вывода суммы, разности, произведения двух чисел, введенных с клавиатуры», «На порядок действий», «На действия со скобками», «Сложные вычисления», «Калькулятор стоимости пиццы».

Тема 5: Подключение модуля - turtle: «черепашья» графика (3 часа).

Теория - 1 час

Что такое модуль в Python. Импортинг модулей. Рисование с Python. Создание холста. Перемещение черепашки. Функции модуля turtle.

Практика – 2 часа

Задания: «Программа, в которой на экране получается изображение с заданными параметрами: прямоугольника, правильного шестиугольника, равностороннего треугольника, отрисовка окружностей, звезд»; «Рисуем дом, крышу, окно, трубу и солнце».

Тема 6: Цикл с параметром в Python (3 часа).

Теория - 2 часа

Циклические алгоритмы. Виды циклов. Безусловно повторяющиеся действия. Блок-схема цикла с параметром. Запись цикла с параметром на языке Python: цикл for и функция range(), переменные цикла.

Практика – 1 час

Оптимизируем задания с модулем turtle, используя цикл for.

Тема 7: Продолжаем работу с циклом for (3 часа).

Теория - 1 час

Разбираемся в работе цикла `for`, знакомясь с программами: «Слово по буквам», «Считалка». Составляем блок-схемы программ. Вносим изменения, чтобы улучшить их работу. Алгоритм отрисовки квадратной спирали.

Практика – 2 часа

Задания: «Гипнотическая квадратная спираль», «Винтовая лестница», «Спиралевидные окружности», «Добавляем цвет», «Сколько сторон?», «Мячик из цветных резинок», «Цветная текстовая спираль», «Машина».

Тема 8: Вложенные циклы в Python (3 часа).

Теория - 1 час

Сложные циклические алгоритмы. Цикл внутри цикла. Разбор программы: «Спираль из спиралей». Составляем блок-схему программы.

Практика – 2 часа

Оптимизируем задания прошлых занятий, используя вложенные циклы. Задание: «Спиралевидные розетки», «Спираль спиралей из имен членов вашей семьи».

Тема 9: Цикл с предусловием в Python (3 часа).

Теория - 1 час

Блок-схема цикла с предусловием. Запись цикла с предусловием на языке Python: цикл `while`. Особенности работы цикла с предусловием. Бесконечный цикл. Как остановить работу бесконечного цикла.

Практика – 2 часа

Задания: «Python решает домашнее задание по математике», «Почемучка, или разговор с малышом».

Тема 10: Цикл с постусловием в Python (3 часа).

Теория - 1 час

Блок-схема цикла с постусловием. Запись цикла с постусловием на языке Python. Составляем алгоритм мытья посуды.

Практика – 2 час

Задания на использование цикла с предусловием и постусловием. Тестирование по пройденным темам.

Тема 11: Преобразование одной инструкции цикла в другую (3 часа).

Теория - 1 час

Как заменить одну инструкцию цикла (for или while) на другую. Схема, иллюстрирующая возможность замены. Опасность заикливания программы. Предотвращение заикливания.

Практика – 2 часа

Задания на разработку программ по теме.

Тема 12: Условия (что если?) (3 часа).

Теория - 1 час

Ветвление. Разветвляющиеся алгоритмы. Логические высказывания. Переменные типа bool. Условия и сравнение значений. Запись инструкции ветвления: условная конструкция if-then-else. Полная и сокращённая формы инструкции ветвления.

Практика – 2 часа

Задания: «Примеры: True или False», «Спираль рисуется, если пользователь ответит на вопрос да, иначе не рисуется», «Получение прав на вождение».

Тема 13: Сложные условия. Каскадные ветвления (3 часа).

Теория - 2 часа

Логические высказывания: простые и сложные. Объединение условий. Логические операции: конъюнкция (and), дизъюнкция (or), инверсия (логическое отрицание not). Каскадные ветвления. Запись каскадного ветвления: конструкция elif (сокращение от else-if).

Практика – 1 час

Примеры использования «и» и «или». Задания: «Принадлежит ли точка заданному прямоугольнику» - составляем алгоритм и программу, «Какая отметка», «Выбираем одежду, ориентируясь на погоду»

Тема 14: Переменные без значения — None и комментарии в Python (3 часа).

Теория - 2 часа

Пустое значение None. Проверка переменной на значение None с помощью конструкции if. Добавление комментариев. Однострочные комментарии. Комментарии в конце строки. Многострочные комментарии. Тройные кавычки. Стиль комментариев. Комментарии в книге. Превращение инструкции в комментарий.

Практика – 1 час

Примеры программ с переменными без значения. Эксперименты с комментариями.

Тема 15: Встроенные структуры данных: списки, кортежи, словари и множества (3 часа).

Теория - 1 час

Почему они являются «встроенными». Что такое список. Определение кортежа. Когда использовать кортежи, а когда - списки. Что такое словарь. Сравнение словарей и списков. Что такое множество.

Практика – 2 часа

Пишем короткие программы, чтобы получить предварительное представление о списках, кортежах и словарях Python. Тестовые задания по теме.

Тема 16: Списки мощнее строк (3 часа).

Теория - 1 час

Список – динамическая структура. Как список выглядит в коде. Содержимое списков. Создание списка. Методы работы со списками. Добавление элементов в список. Что это за точка. Удаление элементов из списка. Списки в списках. Получение части списка с помощью среза. Копирование списка. Списковая арифметика - конкатенация и репликация списков. Использование циклов for со списками. Трюк с групповым присваиванием.

Практика – 2 часа

Задания на выполнение операций со списками: «Создать список гостей, добавить гостей, сократить список, используя разные методы», «Программа, определяющая список уникальных гласных, найденных в слове».

Тема 17: Кортежи в Python (3 часа).

Теория - 1 час

Кортежи неизменяемы. Запись элементов в кортеже. Перебор всех значений в кортеже. Замена кортежа.

Практика – 2 час

Задания на работу с кортежами: «Меню «шведского стола»», «Предметы одежды».

Тема 18: Двоичный поиск и сортировка (3 часа).

Теория - 1 час

Сортировка. Порядок сортировки. Квадратичные сортировки в Python - принцип «Разделяй и властвуй». Сортировка выбором. Сортировка методом пузырька. Сортировка вставками. Сортировка подсчетом. Сравнение скоростей алгоритмов.

Практика – 2 час

Задания на использование сортировок.

Тема 19: Словари в Python — не для поиска слов. Множества (3 часа).

Теория - 2 часа

Словари (ассоциативные массивы) в Python. Словари хранят пары ключ/значение. Словари должны быть в легко читаемом виде. Как определяются словари в коде. Добавление или удаление значений в словари. Множества не допускают повторений. Обозначение множеств в коде. Создаем множества эффективно. Использование методов множеств.

Практика – 1 час

Задания: «Программа, подсчета частоты появления согласных, найденных в слове», «Переводчик на русский».

Тема 20: Функции в Python (3 часа).

Теория - 1 час

Применение функций. Подпрограмма. Определение функции. Вызов функции. Аргументы и параметры. Переменные и область видимости. Локальные и глобальные переменные в Python. Повторное использование кода. Хранение функций в модулях. Импортирование всех функций модуля. Стилевое оформление функций.

Практика – 2 часа

Задания: «Инструкция», , «Принимай - возвращай», «День рождения».

Тема 21: Расширяем знания о функциях (3 часа).

Практика – 3 часа

Задания на создание и вызов функций: «Смайлики в случайных участках экрана», «Отраженные смайлики», «Функция рисования квадрата», «Рисуем заполненные квадраты», «Рисуем закрашенные звезды».

Тема 22: Объекты и классы (3 часа).

Теория - 1 час

Концепция объектно-ориентированного программирования. Объекты. Разделяем объекты на классы. Потомки и предки. Создаем объекты для

классов. Используем функции для задания характеристик класса. Зачем нужны классы и объекты.

Практика – 2 часа

Задания: «Объекты и классы в картинках», «Танец жирафа», «Рисунок черепашек».

Тема 23: Модули Python (3 часа).

Теория - 1 час

Python-модуль. Что такое модуль. Зачем нужны модули. Корзины блоков. Создание модуля. Применение модуля. Стандартные модули Python: copy, keyword, random. Функции copy, deepcopy, keyword, randint, choice и shuffle.

Практика – 2 часа

Задания: «Создаем свой модуль», «Создание копий с помощью стандартного модуля copy», «Что выведет на экран этот код?», «Ключевые слова и модуль keyword», «Отгадай число».

Тема 24: Еще больше стандартных модулей Python (3 часа).

Теория - 1 час

Управление оболочкой с помощью модуля sys. Работа со временем и модуль time. Преобразование дат с помощью функции asctime. Модуль pickle и сохранение информации.

Практика – 2 часа

Задания: «Выход из оболочки с помощью функции exit», «Ввод данных и объект stdin», «Вывод данных и объект stdout», «Получаем дату и время с помощью функции localtime», «Погружение программы в сон с помощью функции sleep», «Запись и загрузка».

Тема 25: Графический интерфейс пользователя (GUI) (3 часа).

Теория - 2 часа

Что такое GUI. Элементы управления: кнопки, метки и поля ввода. Модуль tkinter. Как создать кнопку. Как создать холст для рисования. Как рисовать линии, прямоугольники, дуги, многоугольники. Как добавить цвет.

Практика – 1 час

Задания: «Простейший GUI», «Выводим изображения с помощью tkinter».

Тема 26: Создание простой анимации с модулем tkinter (3 часа).

Практика – 3 часа

Задания: «Заполняем экран прямоугольниками», «Движущаяся фотография - вправо, вниз, влево и вверх».

Тема 27: Первая игра: «Пинг-понг» с графическим интерфейсом (3 часа).

Практика – 3 часа

Создаем игровой холст. Создаем класс для мяча. Рисуем на холсте мяч. Добавляем движение: перемещение мяча, отскоки мяча. Меняем первоначальное направление движения мяча.

Тема 28: Продолжаем работать над игрой «Пинг-понг» (3 часа).

Практика – 3 часа

Создаем ракетки и класс для ракеток. Управление ракетками нажатиями на клавиши-стрелки «влево» и «вправо». Проверка на столкновение мяча с ракетками и границами игрового холста. Анимлируем ракетки.

Тема 29: Додельываем «Пинг-понг» (3 часа).

Практика – 3 часа

Добавляем возможность проигрыша, задержку перед стартом игры. Делаем, чтобы при проигрыше на экране появлялось сообщение «Конец игры». Добавляем в игру подсчет очков.

Тема 30: Разработка игр с модулем Pygame (3 часа).

Теория - 1 час

Использование графического интерфейса пользователя Pygame. Что нового в Pygame. Как установить модуль Pygame. Структура игры или приложения, созданные с помощью модуля Pygame

Практика – 2 часа

Установка модуля Pygame. Практикуем с Pygame.

Тема 31: Разработка игры «Стреляющий корабль» (3 часа).

Практика – 3 часа

Планирование проекта. Создание пустого окна Pygame и обработка ввода. Выбираем цвет фона. Создание класса Settings для хранения всех настроек. Добавление изображения корабля. Создание класса Ship, для реализации большей части поведения корабля.

Тема 32: Продолжаем работать над игрой «Стреляющий корабль» (3 часа).

Практика – 3 часа

Вывод корабля на экран. Управление кораблем. Обработка нажатия клавиши. Непрерывное перемещение. Перемещение влево и вправо. Регулировка скорости корабля. Ограничение перемещений.

Тема 33: Учим наш корабль стрелять. (3 часа).

Практика – 3 часа

Стрельба. Добавление настроек. Группировка пуль. Обработка выстрелов. Удаление старых пуль. Ограничение количества пуль.

Тема 34: Добавляем врагов в игру «Стреляющий корабль» (3 часа).

Практика – 3 часа

Создание врага. Отображение врага на экране. Построение вражеской группы. (Вычисление количества врагов в одном ряду. Создание ряда врагов. Добавление рядов. Создание группы врагов.)

Тема 35: Учим врагов перемещаться в игре (3 часа).

Практика – 3 часа

Перемещение группы врагов. Проверка достижения края. Снижение группы и смена направления движения. Уничтожение врагов. Восстановление врагов. Ускорение пуль.

Тема 36: Завершение игры «Стреляющий корабль» (3 часа).

Практика – 3 часа

Обработка столкновений с кораблем. Достижение нижнего края экрана. Подсчет очков. Вывод счета. Конец игры. Создание кнопки для запуска новой игры.

Методическое обеспечение программы

В программе применяются следующие методы, позволяющие дать детям первоначальные основы программирования:

✓ Словесные методы позволяют быстро передавать детям информацию, ставить перед ними учебную задачу, указывать пути ее решения. Это лекции, беседы, дискуссии. Словесные методы и приемы сочетаются с наглядными, игровыми, практическими методами, делая последние более результативными.

✓ Наглядные методы: Наглядность оживляет процесс обучения, способствует побуждению у детей интереса к занятию. В качестве наглядности используются презентации, мультимедийные учебные пособия, флеш-анимации, видеоролики, веб-кейсы и другие интернет – технологии, демонстрационный раздаточный материал, карточки.

✓ Практические методы: Нельзя научить детей, только показывая и рассказывая, не предлагая самим детям каких - либо действий. Ребенок овладевает опытом только тогда, когда сам участвует в практической деятельности. практической работы учащихся за компьютером. При решении задач используется семинарское занятие

✓ Игровые методы: позволяют осуществлять учебные задачи в атмосфере легкости и заинтересованности, активности детей. Используются игровые задания, создание игр на компьютере, метод интерактивной игры, дидактические игры, и упражнения, выполнив которые ребенок легко может усвоить правила поведения, технику безопасности, гимнастику для глаз.

Формы обучения по программе:

✓ Фронтальная - подача учебного материала всему коллективу детей, на этих занятиях важен «эффект эмоционального воздействия и сопереживания», что приводит к повышению умственной активности, побуждает ребенка к самовыражению (интегрированные и итоговые занятия, интеллектуальные игры).

✓ Индивидуальная - используется при возникновении затруднения, не уменьшая активности детей и содействуя выработке навыков самостоятельной работы. В индивидуальных занятиях нуждаются дети с явно выраженными способностями к той или иной деятельности, дети с доминирующим познавательным интересом.

✓ Подгрупповая - предоставляется возможность самостоятельно построить свою деятельность на основе принципа индивидуализации и сознательности и активности, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности. Всё это способствует более быстрому выполнению задания.

4. Список используемых источников.

для педагога:

1. Патаракин Е. Д. Руководство для пользователя среды Scratch. Версия 0.2, 2007г.

2. Патаракин Е. Д. Учимся готовить в среде Скретч (Учебно-методическое пособие). М: Интуит.ру, 2008 г.

3. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch: учебно-методическое пособие — Оренбург, 2009 г.

4. Пашковская Ю. В. Творческие задания в среде Scratch: рабочая тетрадь для 5 - 6 классов/ Ю. В. Пашковская.- М.:БИНОМ. Лаборатория знаний, 2014. - 200 с. :ил.

5. Л. А. Анеликова, О. Б. Гусева Программирование на алгоритмическом языке КуМир. Под редакцией Кушниренко А. Г. - М.: СОЛОН-ПРЕСС, 2013. - 48 с.: ил. - (Элективный курс Профильное обучение).

6. Еремин Е. А. Газета « Информатика». Среда Scratch – первое знакомство. – М.: Первое сентября, 2008 – № 20 (573) – стр. 16–28.

7. У. Сэнд, К. Сэнд Hello World! Занимательное программирование. — СПб.: Питер, 2016. — 400 с.: ил. — (Серия «Вы и ваш ребенок»).

8. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих.: Пер. с англ.—М.: ООО “ИД. Вильямс”, 2016.—592с.: ил.—Парад. тит. англ.

9. Мэттиз Эрик Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»).

для ученика:

1. Торгашева Ю. В. Первая книга юного программиста. Учимся писать программы на Scratch. — СПб.: Питер, 2016. — 128 с.: ил. — (Серия «Вы и ваш ребенок»).

2. Программирование для детей/К. Вордерман, Дж. Вудкок, Ш. Макаманус и др. ; пер. с англ. С. Ломакина. - М. :Манн, Иванов и Фербер, 2015. - 224 с. :ил.

3. Пэйн, Брайсон. Python для детей и родителей / Брайсон Пэйн ; [пер. с англ. МА. Райтмана]. — Москва : Издательство «Э», 2017. — 352 с. : ил. — (Программирование для детей).

Приложения

Контрольно-измерительные материалы полугодического контроля по программе
"Юный программист" (1 полугодие, 1 год обучения)

Группа № _____

Педагог – _____

Таблица 1

№	Виды УУД	Параметры контроля	Методы контроля	Критерии контроля
1	Общеучебные	Правильная терминология при названии устройств компьютера	Наблюдение	Правильно или неправильно в разговоре с ребятами и педагогом называет устройства компьютера
2		Знание возможностей программы Scratch	Беседа	5 – понимает спектр возможностей программы и умеет их использовать; 4 – знает некоторые возможности программы и их использует; 3 – не знает возможности программы
3		Знание базовой конструкции цикла	Беседа	5 – знает, применяет 4 – знает, применяет с помощью педагога 3 – не знает
4		Знание и понимание алгоритма	Анализ творческой работы	5 – самостоятельно создает алгоритмы 4 – создает с помощью педагога или друга 3 – не умеет создавать
5		Умение применять команды движения	Анализ творческой работы	5 – самостоятельно применяет 4 – команды с помощью педагога или друга 3 – не умеет создавать

6	Коммуникативные	Умение работать в группе	Наблюдение	<p>5 –</p> <ul style="list-style-type: none"> ✓ активно участвовал в работе группы; ✓ проявлял лидерские качества; ✓ ответственно относится к порученной, лидером группы, работе; ✓ сделал всю работу добросовестно и в срок. <p>4 – участвовал в работе группы, но не сделал порученную работу в срок или сделал недостаточно качественно.</p> <p>3 – работал обособленно, не смог войти в контакт и не участвовал в работе группы</p>
7	Регулятивные	Умение контролировать соответствие результата поставленной цели	Наблюдение	<p>5 – самостоятельно сличает результат с эталоном;</p> <p>4 – проверяет соответствие результата эталону с помощью учителя;</p> <p>3 – не проверяет результат своей работы</p>
8		Умение последовательного изложения своей мысли, рассуждения	Анализ итоговой работы	<p>5 – самостоятельно выстраивает последовательный ответ;</p> <p>4 – с помощью учителя;</p> <p>3 – не умеет</p>
9		Способность к рефлексии	Наблюдение	<p>5 – умеет адекватно оценивать деятельность на занятии</p> <p>4 – оценивает деятельность на занятии с помощью педагога и товарищей</p> <p>3 – не умеет адекватно оценивать деятельность на занятии</p>

Контрольно-измерительные материалы итогового контроля по программе
«Юный программист» (2 год обучения)

Группа № _____

Педагог _____

Таблица 3

№	Параметры контроля	Методы контроля	Критерии контроля
1	Правильная терминология при названии устройств компьютера	Наблюдение	Правильно или неправильно в разговоре с ребятами и педагогом называет устройства компьютера
2	Знание понятия переменной, её использование в среде Скретч.	Анализ итоговой работы	5 - знает и применяет 3- не знает
3	Умение импортировать и экспортировать графику в среду Скретч.	Анализ итоговой работы	5 – умеет и применяет навыки 3 – не умеет
4	Умение работать с циклами, самостоятельно создавать блоки	Анализ итоговой работы	5 – самостоятельно создает блоки 4 – создает блоки с помощью педагога или друга 3 – не смог создать блок
5	Умение работать со звуком: Создание звукового ряда для проекта Сохранение звуковых файлов.	Анализ творческой работы	5 – самостоятельно создает звуковые файлы 4 – создает с помощью педагога или друга 3 – не умеет создавать звуковые файлы
6	Умение работать с проектами в среде КуМир	Анализ итоговой работы	5 – самостоятельно создает проекты 4 – создает с помощью педагога или друга 3 - не умеет создавать проекты
7	Умение тестировать и применять отладку проекта в среде КуМир	Анализ итоговой работы	5 – самостоятельно и целесообразно применяет тестирование и отладку проекта 4 – при помощи педагога 3 – не умеет тестировать

Контрольно-измерительные материалы итогового контроля по программе
«Юный программист» (3 год обучения)

Группа № _____

Педагог _____

Таблица 5

№	Параметры контроля	Методы контроля	Критерии контроля
1.	Умение осуществлять ввод и вывод данных в программах на Python.	Анализ итоговой работы	5 - умеет и применяет 3- не умеет
2.	Умение работать с циклами и ветвлениями в Python.	Анализ итоговой работы	5 – умеет и применяет навыки 3 – не умеет
3.	Умение создавать и использовать в программах переменные.	Анализ итоговой работы	5 – самостоятельно создает переменные 4 – создает переменные с помощью педагога или друга 3 – не смог создать переменных
4.	Умение работать со списками, кортежами, словарями и множествами.	Анализ творческой работы	5 – самостоятельно работает, понимает 3 – не умеет создавать
5.	Умение применять функции в программах на Python.	Анализ итоговой работы	5 – самостоятельно применяет 4 – применяет с помощью педагога или друга 3 - не умеет применять
6.	Умение работать с модулями в среде Python	Анализ итоговой работы	5 – самостоятельно работает 4 – при помощи педагога 3 – не умеет работать

Таблица 6

Фамилия Имя \ Параметры контроля	Умение осуществлять ввод и вывод данных в программах на Python.	Умение работать с циклами и ветвлениями в Python.	Умение создавать и использовать в программах переменные.	Умение работать со списками, кортежами, словарями и множествами.	Умение применять функции в программах на Python.	Умение работать с модулями в среде Python