

Государственное бюджетное общеобразовательное учреждение Самарской области средняя общеобразовательная школа пос. Кировский муниципального района Красноармейский Самарской области

Проверено
И. О. зам. директора по УВР

Королева А. О.

«29» августа 2023 г.

Утверждаю
Директор ГБОУ СОШ пос. Кировский

Рыженкова
О.В.

«29» августа 2023 г.

Дополнительная общеобразовательная общеразвивающая программа

техническая направленность

«Основы программирования на языке Python»

(начальный уровень)

Возраст учащихся: 12-14 лет

Срок реализации программы– 1 год

Составитель:

учитель информатики Балашов
Василий Владимирович.

Краткая аннотация

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Основы программирования на языке Python» (далее – Программа) включает в себя 3 тематических модуля. Программа имеет общекультурный характер и направлена на овладение начальными знаниями в области программирования на языке Python. Изучая программу, учащиеся смогут ознакомиться с основными понятиями, при освоении которых они научатся программировать (данными, переменными, циклами, функциями и графикой).

Данная программа разработана с учётом интересов конкретной целевой аудитории, обучающихся, и дает возможность участия в творческих конкурсах технической направленности.

Пояснительная записка

Направленность дополнительной общеразвивающей программы «Основы программирования на языке Python» техническая.

Актуальность дополнительной общеразвивающей образовательной программы обусловлена тем, что она нацелена на решение задач, определенных в Стратегии развития воспитания в Российской Федерации на период до 2025 года от 29 мая 2015 г. № 996-р г., направленных на формирование гармоничной личности, ответственного человека. В соответствии с Целевой моделью развития региональных систем дополнительного образования детей, утвержденной приказом Минпросвещения России от 03.09.2019 г. № 467 программа направлена на формирование эффективной системы выявления, поддержки и развития способностей и талантов обучающихся. В связи с этим изучение основ программирования на языке Python позволяет сформировать у детей стойкий интерес к программированию, отвечает всем современным требованиям

объектно-ориентированного программирования. Среда Python позволяет сформировать навыки программирования. На занятиях у обучающихся формируются важные для практико-ориентированной деятельности умения, связанные с представлением, анализом и интерпретацией данных.

Так же данная программа направлена на овладение знаниями в области компьютерной трехмерной графики конструирования и технологий на основе методов активизации творческого воображения, и тем самым способствует развитию конструкторских, изобретательских, научно-технических компетентностей и нацеливает детей на осознанный выбор необходимых обществу профессий, как инженер-конструктор, инженер-технолог, проектировщик, дизайнер и т.д.

Содержание заданий программы позволяет развивать и организационные умения: планировать этапы предстоящей работы, определять последовательность действий, осуществлять контроль и оценку их правильности, поиск путей преодоления ошибок.

Новизна данной программы состоит в том, что занятия по «Основы программирования на языке Python» помогают приобрести глубокие знания в области технических наук, ценные практические умения и навыки, воспитывают трудолюбие, дисциплинированность, культуру труда, умение работать в коллективе. Знания, полученные при изучении программы «Основы программирования на языке Python», учащиеся могут применить для разработки игр, а также решения задач по различным предметам. Данная образовательная программа практически значима для современного школьника, т.к. дает возможность увидеть практическое назначение алгоритмов и программ, что будет способствовать развитию интереса к профессиям, связанным с программированием. А так же новизна данной дополнительной образовательной программы заключается в том, что по форме организации образовательного процесса она является модульной.

Отличительной особенностью данной программы является применение конвергентного подхода, позволяющего выстраивать обучение,

включающее в себя элементы нескольких направленностей, которые выражаются в использовании в программе при освоении материала таких дисциплин, как компьютерная графика, использование графического интерфейса Pygame, изучение Python-кода и применение кода в игре Minecraft.

При этом создаётся комфортная учебно-воспитательная среда, в которой происходит наиболее полная самореализация ребёнка.

Педагогическая целесообразность. Сфера человеческой деятельности в технологическом плане в настоящее время очень быстро меняется, на смену существующим технологиям достаточно быстро приходят новые, которые специалисту вновь приходится осваивать. В связи с этим целесообразно в рамках дополнительного образования организовать для обучающихся 5 - 7 классов изучение популярного языка программирования Python, в рамках которой происходит обучение программированию и информационным технологиям. В ходе изучения языка программирования Python, ребята научатся писать программы, взаимодействующие с популярной игрой Minecraft. Создавая игры, обучающиеся в полной мере, могут раскрыть свои творческие таланты,

Программа разработана с учетом разноуровневого принципа, а именно учитываются:

- возрастные особенности детей;
- физиологические особенности детей;
- интересы детей;
- мотивированность;
- степень владения компьютером.

Цель программы — формирование и развитие у обучающихся интеллектуальных и практических компетенций в области программирования.

Цели и задачи каждого модуля формулируются отдельно.

Задачи программы:

Обучающие:

- познакомить обучающихся с основными инструментами программного обеспечения для программирования;
- сформировать у обучающихся базовые представления о языках программирования, алгоритме, исполнителе, способах записи алгоритма;
- способствовать формированию у обучающихся информационной и функциональной компетентности, развитие алгоритмического мышления
- изучить основные базовые алгоритмические конструкции;
- познакомить с понятием переменной и команды присваивания;
- сформировать навыки разработки программ;
- изучить основные этапы решения задачи;
- сформировать навыки разработки, тестирования и отладки несложных программ;
- сформировать представление о профессии «программист»;
- способствовать развитию критического, системного, алгоритмического мышления.

Развивающие:

- развивать инженерное, логическое и творческое мышление;
- развитие пространственного мышления;
- способствовать развитию критического, системного, алгоритмического и творческого мышления;
- развивать внимание, память, наблюдательность, познавательный интерес;
- развивать умение работать с компьютерными программами и дополнительными источниками информации;
- развивать навыки планирования проекта, умение работать в группе.

Воспитательные:

- воспитать положительное отношение к информатике и ИКТ;

- воспитание самостоятельности и формировать умение работать в паре, малой группе, коллективе;
- формировать умение демонстрировать результаты своей работы.

Возраст детей, участвующих в реализации программы: 12 – 14 лет.

Высокая способность детей в этот возрастной период быстро овладевать теми или иными видами деятельности определяет большие потенциальные возможности разностороннего развития. Им нравится исследовать все, что незнакомо, они понимают законы последовательности и последствия, имеют хорошее чувство времени, пространства, расстояния.

Дети этого возраста более усидчивы, умеют определённое время концентрировать свое внимание на задачах. На занятиях по программе «Основы программирования на языке Python» подача нового материала сопряжена с созданием собственных игр на платформе Minecraft.

Сроки реализации: программа рассчитана на 1 год, объем – 108 часов

Формы обучения:

- занятие;
- лекция;
- практическая работа;
- защита проекта.

Формы организации деятельности: групповая.

Режим занятий: 2 раза в неделю по 1,5 академических часа. Одно занятие длится 40 минут.

Наполняемость учебных групп: составляет 13-16 человек.

Планируемые результаты

Личностные:

- критическое отношение к информации и избирательность её восприятия;

- осмысление мотивов своих действий при выполнении заданий жизненными ситуациями;
- получение опыта социально-значимой деятельности на уровне класса, школы.

Метапредметные:

Познавательные:

- умение определять последовательность действий;
- умение строить логические цепочки рассуждений.

Регулятивные:

- ставить учебные цели,
- умение принимать и сохранять учебную цель и задачи;
- планировать свои действия в соответствии с поставленной задачей и условиями её решения, в том числе, во внутреннем плане;
- осуществлять итоговый и пошаговый контроль, сличая результат с эталоном;
- вносить коррективы в действия в случае расхождения результата решения задачи и ранее поставленной целью.

Коммуникативные:

- умение объяснить свой выбор

Предметные результаты.

Модульный принцип построения программы предполагает описание предметных результатов в каждом конкретном модуле.

Учебный план

№ модуля	Название модуля	Количество часов		
		всего	теория	практика
1	Основные понятия языка Python	37	13	24
2	Графический интерфейс	29	10	19
3	Программируем с Minecraft	42	10	32
	Всего	108	33	75

Критерии оценки знаний, умений и навыков при освоении программы

Для того чтобы оценить усвоение программы, в течение года используются следующие методы диагностики: собеседование, наблюдение, выполнение отдельных творческих заданий, участие в конкурсах, викторинах.

По завершению учебного плана каждого модуля оценивание знаний проводится посредством викторины, интеллектуальной игры или интерактивного занятия.

Применяется 3-х балльная система оценки знаний, умений и навыков обучающихся (выделяется три уровня: ниже среднего, средний, выше среднего). Итоговая оценка результативности освоения программы проводится путём вычисления среднего показателя, основываясь на суммарной составляющей по итогам освоения 3-х модулей.

Уровень освоения программы ниже среднего – ребёнок овладел менее чем 50% предусмотренных знаний, умений и навыков, учащиеся не знают значительной части материала, допускают существенные ошибки, с большими затруднениями выполняют практические задания.

Средний уровень освоения программы – объём усвоенных знаний, приобретённых умений и навыков составляет 50-70%; работает с учебным материалом с помощью педагога; в основном, выполняет задания на основе образца, удовлетворительно владеет теоретической информацией по темам курса, умеет пользоваться литературой.

Уровень освоения программы выше среднего – учащийся овладел на 70-100% предусмотренным программой учебным планом; работает с учебными материалами самостоятельно, не испытывает особых трудностей; выполняет практические задания с элементами творчества; свободно владеет теоретической информацией по курсу, умеет анализировать литературные источники, применять полученную информацию на практике.

Формы контроля качества образовательного процесса:

- собеседование,
- наблюдение,
- интерактивное занятие;
- выполнение творческих заданий,
- участие в конкурсах, викторинах в течение года.

Модуль «Основные понятия языка Python»

Цель: создание условий для формирования интереса к языку программирования

Python.

Задачи:

Обучающие:

- сформировать представление о работе в среде IDLE;
- обучить основным командам на языке Python;
- сформировать представление о типах данных и операциям совершающихся над ними;
- сформировать навыки создания простых игр на основе изученных функций;
- обучить основным алгоритмическим конструкциям: линейной, разветвляющейся, циклической;

Развивающие:

- развитие творческого мышления при создании программ;
- развитие навыков применения компьютерных технологий в различных областях компьютерного моделирования;

Воспитательные:

- воспитать умение слушать и выполнять инструкции;
- осознавать ценность знаний по программированию;

- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками и взрослыми в процессе образовательной деятельности.

Предметные ожидаемые результаты

Обучающийся должен знать:

- синтаксис языка Python;
- основные функции Python, типы данных и операции над ними;
- построение сложных условий (and, or, not);
- строки и операции с ними.

Обучающийся должен уметь:

- запускать IDLE, создавать новый файл и сохранять программу;
- писать программу с учётом рассмотренных функций;
- использовать точечную запись для обращения к функциям.

Обучающийся должен приобрести навык:

- написания небольшой рабочей программы на языке Python;
- написания программы используя, операторы сравнения, сложные условия и циклы.

Учебно-тематический план

№ п/п	Наименование темы	Количество часов			Формы аттестации/контроля
		всего	теория	практика	
1	Инструктаж. Введение в язык программирования Python	3	1	2	Компьютерный практикум
2	Типы данных и операции над ними	3	1	2	Решение математических задач
3	Условные операторы	4	2	2	Компьютерный практикум
4	Циклы while и for	3	1	2	Самоконтроль
5	Сложные условия: and, or, not	2	1	1	Самоанализ качества

					выполнения практической работы. Наблюдение педагога
6	Списки, кортежи и словари	5	2	3	Самоанализ качества выполнения практической работы. Наблюдение педагога
7	Циклические алгоритмы	3	1	2	Самоанализ качества выполнения практической работы. Наблюдение педагога
8	Вложенные условные операции и циклы	3	1	2	Самоанализ качества выполнения практической работы. Наблюдение педагога
9	Функции	4	1	3	Самоанализ качества выполнения практической работы. Наблюдение педагога
10	Массивы	4	1	3	Самоанализ качества выполнения практической работы. Наблюдение педагога
11	Строки и операции с ними	3	1	2	Самоанализ качества выполнения практической работы. Наблюдение педагога
	Итого	37	13	24	

Содержание программы

Тема 1. Инструктаж. Введение в язык программирования Python.

Теория: Ведение в программу. Правила техники безопасности при работе. Правила поведения в учреждении, на занятиях, в кабинете и др. Знакомство с языком программирования Python.

Практика: запуск IDLE, создание программы, её сохранение и запуск.

Тема 2. Типы данных и операции над ними.

Теория: рассмотрение основных типов данных в Python. Арифметические выражения и операции. Стандартные функции.

Практика: Применение полученных знаний на практике: написание программы для вычисления площади треугольника, если известны его длина основания и высота. Самостоятельное написание программы для вычисления значения заданного выражения.

Тема 3. Условные операторы.

Теория: объяснение, что такое ветвление, условные операторы. Оператор if, else. Роль сдвигов операторов относительно левой границы (отступы). Дополнительные блоки elif. Операторы сравнения.

Практика: 1) Написание программы, разрешающей доступ только для тех, чей возраст старше 21 года. 2) Написание программы, которая запрашивает возраст претендента и выдает ответ: «подходит» он или «не подходит» по заданному признаку. 3) Написание программы по сохранению файла с использованием условных операторов и операторов сравнения.

Тема 4. Циклы while и for.

Теория: рассмотрение двух видов команд для записи циклов: while и for. Повторение некоторых частей кода с помощью циклов. Отличительная особенность цикла с предусловием.

Практика: 1) Написание программы для вывода на экран всех целых чисел от 1 до 5. 2) Определение количества цифр в десятичной записи целого положительного числа. 3) Выведем степени числа два от 2^1 до 2^{10} ($k =$ степени двойки).

Тема 5. Сложные условия: and, or, not.

Теория: Правильное выстраивание условий. Сложные условия, состоящие из нескольких составных условий, соединённых логическими операторами (связками) «и» (and), «или» (or), «не» (not). Логическое сложение, умножение, отрицание.

Практика: 1) С помощью оператора and составить два сложных логических выражения, одно из которых дает истину, другое – ложь. 2) Аналогично выполните задачу 1, но уже с оператором or.

Тема 6. Списки, кортежи и словари.

Теория: Однообразные данные организовывать в списки, кортежи и словари. Операции со списками. Правило создания кортежа. Уникальный ключ в словарях.

Практика: Создание словаря «Рыбы», где необходимо элементы разделить на 3 вида: «речные», «озерные» и «морские рыбы». Вывести на экран сначала только ключи, а потом элементы словаря.

Тема 7. Циклические алгоритмы.

Теория: Три алгоритмические конструкции. Повторение, что такое цикл. Рассмотрение применения цикла for более подробно. Аргументы для функции range. Сравнение циклов for и while.

Практика: написание игры, в которой пользователь должен угадать число, загаданное компьютером.

Тема 8. Вложенные условные операции и циклы.

Теория: Алгоритм программы, который предполагает выбор больше, чем из двух путей, например, из трех, четырех или даже больше.
«Каскадное» ветвление. Конструкция «цикл в цикле» или «вложенный цикл».

Практика: 1) Вывести на экран таблицу умножения. Для этого во внешнем цикле надо перебрать числа от 1 до 9. Для каждого из них нужно перебрать во внутреннем цикле числа от 1 до 9. 2) Составить программу нахождения всех простых чисел в интервале от 2 до 100.

Тема 9. Функции.

Теория: Повторение встроенных в интерпретатор, изученных ранее, функций. Правила создания собственных функций, как вспомогательных алгоритмов. Функции и их аргументы. Глобальные и локальные переменные. Возврат значений из функций.

Практика: Составить функцию, которая вычисляет сумму всех цифр числа (н-р, для числа 147 нужно сложить цифры: $1+4+7=12$). Сложение цифр начинаем с последней, в данном примере это цифра 7.

Тема 10. Массивы.

Теория: Обработка большого количества данных. Одномерные массивы. Использование списков при работе с массивами. Разные способы составления списков. Перебор элементов списка.

Практика: написать программу, которая находит количество учеников, рост которых больше 120 см, но меньше 150 см.

Тема 11. Строки и операции с ними.

Теория: Основной тип данных для работы с текстом в языке Python. Определение строки. Рассмотрение строк, как упорядоченную последовательность элементов, с которыми можно работать так же, как и с элементами в списках. Свойства строки.

Практика: С клавиатуры вводится строка, содержащая имя, отчество и фамилия человека полностью. Каждые два слова разделены одним пробелом, в начале строки пробелов нет. В результате обработки должна получиться новая строка, содержащая фамилию и инициалы.

Модуль «Графический интерфейс»

Цель: формирование умений работать с графикой turtle, создавать орнаменты с помощью строк кода.

Задачи:

Обучающие:

- дать представление об окружности, углах, толщине линий;
- обучить основным приемам работы с графическим интерфейсом;
- сформировать представление о логике программного кода;
- обучить написанию приложения, принимающего щелчки мышью и рисующее на экране;
- сформировать умение тестировать, отлаживать программы;
- сформировать умение понимать систему координат x и y используемой среды Turtle;
- сформировать умение создавать и редактировать программы, в которых используются вложенные циклы.

Развивающие:

- развить умение моделировать действия, процессы, явления;

- развить умение использовать координаты экрана x и y для отрисовки орнаментов;
- сформировать умение организовывать процесс передачи сообщений между объектами.

Воспитательные:

- воспитать чувство видения цвета, стиля, при создании различных изображений.

Предметные ожидаемые результаты

Обучающийся должен знать:

- как подключить модуль turtle и как использовать библиотеку инструментов данного модуля;
- каким образом каждая строка кода влияет на траекторию;
- как работать с переменными.

Обучающийся должен уметь:

- выбирать градус поворота для изменения фигур;
- владеть работой со списком;
- использовать переменные для сохранения числовых значений и строк;
- изменять цвет фона, параметр экрана;
- разбивать процесс решения задачи на этапы.

Обучающийся должен приобрести навык:

- рисования простой графики с помощью библиотеки Turtle;
- создания цикла, который повторяет набор инструкций.

- работы с цветом, путём изменения необходимых параметров;
- изменять, сохранять и запускать программы в IDLE.

Учебно-тематический план

№ п/п	Наименование темы	Количество часов			Формы аттестации/контроля
		всего	теория	практика	
1	Модуль Turtle	2	1	1	наблюдение, самоконтроль практическая работа
2	Изменение угла	2	1	1	практическая работа самоконтроль
3	Рисование окружности	2	1	1	практическая работа самоконтроль
4	Использование цвета	1	0	1	Наблюдение, практическая работа
5	Список цветов	1	0	1	наблюдение, групповой, практическая работа
6	Изменение цветов фона	1	0	1	наблюдение, самоконтроль, практическая работа
7	Добавление переменной	2	1	1	наблюдение, самоконтроль, практическая работа
8	Использование текста в рисовании спирали	2	1	1	наблюдение, самоконтроль, демонстрация работы
9	Орнаменты с разным количеством сторон. Списки.	3	1	2	наблюдение, практическая работа
10	Использование цикла for для создания розетки	3	1	2	практическая работа, самоконтроль
11	Цветные случайные спирали	3	1	2	практическая работа, самоконтроль
12	Функции. Рисование смайлика.	4	1	3	практическая работа, самоконтроль
13	Создаём графический редактор. TurtleDraw	3	1	2	наблюдение, групповой, демонстрация работы
	Итого	29	10	19	

Содержание программы

Тема 1. Модуль Turtle.

Теория: Подключение модуля Turtle (черепашка). Разбор кода, инструктирующий черепашку перемещаться заданными траекториями. Отслеживание движения «черепашки», чтобы понять, каким образом каждая строка кода влияет на траекторию движения «черепашки».

Практика: Написание программы квадратной спирали.

Тема 2. Изменение угла

Теория: Редактирование программы. Разбор, какая фигура может нарисоваться, если линию рисовать, поворачивая её на определённый градус. Вводи понятия градус, угол.

Практика: Изменение кода программы квадратной спирали и превращение её в винтовую лестницу.

Тема 3. Рисование окружности

Теория: Возможности библиотеки Turtle. Использование команды для рисования окружности circle.

Практика: Изменяем код программы и рисуем четыре спиралевидных окружности.

Тема 4. Использование цвета

Теория: Использование цветов из библиотеки. Разбираем, как закрасить линии созданных фигур определённым цветом. Изменение параметра цвета.

Практика: Закрашивание квадратной спирали выбранным цветом.

Тема 5. Список цветов

Теория: Обдумывание и составление алгоритма, который превращает созданный однотонный объект в четырёхцветный. Настройка цикла. Использование списка цветов.

Практика: Изменение программы рисования квадратной спирали, так, чтоб рисунок получился четырёхцветным.

Тема 6. Изменение цветов фона

Теория: Изменение цвета фона. Использование bgcolor.

Практика: Изменение кода программы, таким образом, чтоб изменились параметры экрана, а именно фон поменял свой цвет.

Тема 7. Добавление переменной

Теория: Переменные. Добавление переменных в программу. Разбираем, как будет влиять переменная на создание рисунка, если она будет отвечать за количество сторон фигуры.

Практика: Изменить код программы, путём добавления в неё переменную, отвечающую за количество сторон рисуемой фигуры. Изменить значение переменной несколько раз, сохранить, запустить программу и сравнить результат.

Тема 8. Использование текста в рисовании спирали

Теория: Функция для написания текста `textinput`. Объединение возможностей строк и цветной графики.

Практика: написать программу, где вместо рисований линий и окружностей, необходимо запросить имя пользователя, а затем нарисовать его на экране в виде цветной спирали.

Тема 9. Орнаменты с разным количеством сторон. Списки.

Теория: Списки. Сохраняем и используем список строк. Вызов функции `rencolor`. Использование функции `numinput`, позволяющей ввести число.

Практика: написать программу, которая будет запрашивать у пользователя количество сторон (от 1 до 8), а также предоставит пользователю опцию по умолчанию — 4, означающую, что если пользователь не введет число, то программа автоматически воспользуется параметром 4 для установки количества сторон.

Тема 10. Использование цикла `for` для создания розетки

Теория: Повторяющиеся фигуры. Цикл `for`. Определение повторяющихся шагов и количество повторов. Использование отступов, для каждой повторяемой в цикле команды.

Практика: написать программу для рисования орнамента розетка с четырьмя окружностями, используя цикл for.

Тема 11. Цветные случайные спирали

Теория: Модуль random. Понятие «генератор случайных чисел».

Практика: написать программу, которая заполнит экран спиралями случайных размеров и цветов.

Тема 12. Функции. Рисование смайлика.

Теория: Планирование алгоритма рисования смайлика на листке бумаги в клетку. Определение координат x и y. Вызов функции.

Практика: написание программы, которая вместо случайных спиралей будет рисовать смайликов.

Тема 13. Создаём графический редактор. TurtleDraw

Теория: Обработка событий. Программирование интерактивного взаимодействия с пользователем. Функция onclick.

Практика: создание графического редактора с помощью функции onclick.

Модуль «Программируем с Minecraft»

Цель: развитие индивидуальных творческих способностей и закрепление пройденного материала посредством создания программ в мире Minecraft.

Задачи:

Обучающие:

- сформировать представление о мире Minecraft, не только как об игре;
- сформировать представление о написании программ на языке Python взаимодействующие с Minecraft;
- повторить основные конструкции языка программирования Python;

- осуществлять перенос знаний, умений в новую ситуацию для решения неформализованных задач.

Развивающие:

- планировать, прогнозировать, корректировать свою деятельность;
- ставить цели, определять конечный результат деятельности;
- сформировать умение составлять план деятельности;

Воспитательные:

- воспитывать культуру безопасного труда при работе за компьютером;
- воспитывать чувство ответственности за результаты своего труда;
- воспитать интерес к программированию и получению новых знаний.

Предметные ожидаемые результаты

Обучающийся должен знать:

- базовые принципы программирования на языке Python;
- возможности создания программ в мире Minecraft.

Обучающийся должен уметь:

- планировать, прогнозировать, корректировать свою деятельность;
- устанавливать причинно-следственные связи.

Обучающийся должен приобрести навык:

- самостоятельной работы.

Учебно-тематический план

№ п/п	Наименование темы	Количество часов			Формы аттестации/контроля
		всего	теория	практика	
1	Подготовительный этап. Настройка профиля игры.	2	1	1	практическая работа, наблюдение
2	Телепортация с помощью переменных.	4	1	3	практическая работа, самоконтроль
3	Математика, моментальное строительство и суперпрыжки.	5	1	4	практическая работа, самоконтроль

4	Общаемся с помощью строк	4	1	3	практическая работа, самоконтроль
5	Конструкция if, душ и потайная дверь	4	1	3	практическая работа, самоконтроль
6	Цикл while, дискотека и цветочный дождь	5	1	4	практическая работа, самоконтроль
7	Функции как источник больших возможностей	4	1	3	практическая работа, самоконтроль
8	Списки, словари и удары по блокам	5	1	4	практическая работа, самоконтроль
9	Циклы for и волшебство в Minecraft	4	1	3	практическая работа, самоконтроль
10	Объектно-ориентированное программирование	5	1	4	практическая работа, самоконтроль
	Итого	42	10	32	

Содержание программы

Тема 1. Подготовительный этап. Настройка профиля игры.

Теория: Что такое мир Minecraft. Настройка профиля игры — для того чтобы проверить, совместимы ли версии Spigot и Minecraft. Понятие «сервер».

Практика: запустить программу, настроить профиль по указанному алгоритму и играть в игру. Создание нового мира.

Тема 2. Телепортация с помощью переменных.

Теория: Что такое программа. Хранение информации в переменных. Синтаксис для переменных. Изменение значений переменных. Целые числа. Вещественные числа. Модуль time. Отладка.

Практика: 1) Телепортация игрока. 2) Перемещение в точности туда, куда надо. 3) Телепортационный тур. 4) Исправление неработающего телепортатора.

Тема 3. Математика, моментальное строительство и суперпрыжки. **Теория:** Выражения и команды.

Операции. Сложение. Вычитание. Математические операции и аргументы. Умножение. Деление. Возведение в степень. Полезные математические хитрости. Сокращенные

операции.

Случайные числа.

Практика: 1) Башенка из блоков. 2) Суперпрыжок 3) Измените блок под ногами игрока. 4) Быстрое строительство. 5) Потрясающие шпиды. 6) Суперпрыжок в неизвестность.

Тема 4. Общаемся с помощью строк.

Теория: Разбор, что такое строки. Функция print(). Функция input(). Склеивание строк. Преобразование числа в строку. Склеивание целых и вещественных чисел. Преобразование строки в целое число. Обработка исключений.

Практика: 1) Привет, мир Minecraft! 2) Отправка в чат сообщения. 3) Добавление перед сообщениями имён. 4) Позволение пользователю выбрать тип блока. 5) Допускаются только числа. 6) Отчёт о перемещениях.

Тема 5. Конструкция if, душ и потайная дверь

Теория: Конструкция IF. Конструкция else. Конструкция elif. Цепочки конструкций elif. Вложенные конструкции if. Проверка диапазона значений с помощью if. Логические операции и конструкции if.

Практика: 1) Делаем кратер. 2) Предотвратить разрушения или нет? 3) Подарок. 3) Телепортация в нужное место. 4) Потайная дверь. 5) Ограничить область телепортации. 6) Душ

Тема 6. Цикл while, дискотека и цветочный дождь

Теория: Простейшие циклы while. Управление циклами с помощью переменной count. Бесконечный цикл. Логические операции и цикл while. Проверка диапазона значений в условии while. Вложенные конструкции if и циклы while. Выход из цикла while с помощью break. Конструкция while – else.

Практика: 1) Телепортация в случайные места. 2) Водяное проклятие. 3) Цветочные следы. 4) Состязание ныряльщиков. 5) Построение танцпола. 6) Прикосновение Мидаса. 7) Постоянный чат на основе цикла. 8) «Горячо или холодно»

Тема 7. Функции как источник больших возможностей

Теория: Как создать собственные функции. Вызов функции. Функции принимают аргументы. Рефакторинг кода. Комментирование с помощью строк документации. Переносы строк в списке аргументов. Возвращение значений функции. If и while внутри функций. Глобальные и локальные переменные.

Практика: 1) Посадка леса. 2) Напоминалка типов блоков. 3) Цвет шерсти. 4) Создаём блоки повсюду. 5) Самодвижущийся блок.

Тема 8. Списки, словари и удары по блокам

Теория: Правила работы со списками. Доступ к элементам списка. Изменение элементов списка. Изменение структуры списка. Работа со строками как со списками. Кортежи. Функции, возвращающие кортеж. Свойства списков. Словари.

Практика: 1) Высоко и низко. 2) Создание столбика – секундомера. 3) Скольжение. 4) Создание ударов по блокам. 5) Создание меча ночного видения. 6) Путеводитель. 7) Удары по блокам и таблица результатов.

Тема 9. Циклы for и волшебство в Minecraft

Теория: Простой цикл for. Функция range(). Другие функции для работы со списками. Перебор элементов словаря в цикле. Выход из цикла for-else с помощью break. Вложенные циклы for и многомерные списки. Два измерения. Доступ к элементам 2D-списка. Генерация 2D-списка с помощью циклов. Думаем в трех измерениях. Доступ к элементам 3D-списка.

Практика: 1) Создание волшебной палочки. 2) Создание волшебной лестницы. 3) Колонны. 4) Пирамида. 5) Таблица результатов. 6) Алмазоискатель. 7) Пиксель-арт. 8) Обветшалая стена. 9) Копирование конструкций.

Тема 10. Объектно-ориентированное программирование

Теория: Основы ООП. Создание класса. Создание объектов. Доступ к свойствам. Что такое методы. Методы, возвращающие значения. Создание нескольких объектов. Свойства класса. Наследование классов. Добавление новых методов в производный класс. Переопределение методов и свойств.

Практика: 1) Объекта - места. 2) Дом – призрак. 3) Замок – призрак. 4) Посёлок – призрак. 5) Гостиница – призрак. 6) Дерево – призрак.

Обеспечение программы

Методическое обеспечение

Основные принципы, положенные в основу программы:

- принцип доступности, учитывающий индивидуальные особенности каждого
- ребенка, создание благоприятных условий для их развития;
- принцип демократичности, предполагающий сотрудничество педагога и обучающегося;
- принцип системности и последовательности – знание в программе даются в определенной системе, накапливая запас знаний, дети могут применять их на практике.

Методы работы:

- *словесные методы:* беседа-объяснение – эти методы способствуют обогащению теоретических знаний детей, являются источником новой информации;
- *наглядные методы:* демонстрации, с использованием проектора, готовых программ, игр на языке Python. Наглядные методы дают возможность более детального обследования объектов, дополняют словесные методы, способствуют развитию мышления детей.
- *практические методы:* создание собственного продукта (игры). Данные методы позволяют воплотить теоретические знания на практике, способствуют развитию навыков и умений детей. Большое

значение приобретает выполнение правил культуры труда, экономного расходования материалов, бережного отношения к инструментам, приспособлениям и материалам.

Занятие состоит из следующих структурных компонентов:

1. Организационный момент, характеризующийся подготовкой учащихся к занятию;
2. Повторение материала, изученного на предыдущем занятии;
3. Постановка цели занятия перед учащимися;
4. Изложение нового материала;
5. Практическая работа;
6. Обобщение материала, изученного в ходе занятия;
7. Подведение итогов;
8. Приведение рабочего места в порядок.

Материально-техническое оснащение программы

Для проведения теоретических занятий необходимы:

- учебный кабинет;
- язык программирования Python;
- игра для программирования Minecraft;
- мультимедийный проектор;
- звуковое оборудование (колонки).

Список литературы:

1. Крэйг Ричардсон/ Програмуем с Minecraft. Создай свой мир с помощью Python / Крэйг Ричардсон; пер. с англ. Станислава Ломакина; [науч. ред. Г. Гаджиев]. — М.: Манн, Иванов и Фербер, 2017. — 368 с.: ил.
2. Пэйн Брайсон/Python для детей и родителей/Брайсон Пэйн; [пер. с англ. М.А. Райтмана]. — Москва: Издательство «Э», 2017. — 352 с.: ил.
3. Л.Самыкбаева, А.Беляев, А.Палитаев, И.Та Палитаев, И. Ташиев, С.Маматов – Фонд Сорос-Кыргызстан, 2019 – 84 с.