

Муниципальное общеобразовательное учреждение
«Средняя общеобразовательная школа имени Героя Советского Союза С.М. Иванова»
р.п. Турки Турковского района Саратовской области

<p>Протокол Педагогического совета № <u>1</u> от « <u>28</u> » августа <u>2023</u>г.</p>	<p>УТВЕРЖДАЮ Директор _____ Лопаткина Л.В. Приказ № <u>298</u> от « <u>01</u> » <u>сентября</u> <u>2023</u> г.</p>
--	--

**Дополнительная общеобразовательная
(общеразвивающая) программа
«Основы программирования (PYTHON.)»
(технической направленности)**

**Возраст обучающихся: 14-17 лет
Срок реализации: 1 год
Автор- составитель:
Чучков Алексей Сергеевич
педагог дополнительного образования**

р.п. Турки 2023-2024 г.г.

СОДЕРЖАНИЕ

Раздел №1 «Комплекс основных характеристик программы»

- 1.1. Пояснительная записка..... 3
- 1.2. Содержание программы.....5

Раздел №2 «Комплекс организационно-педагогических условий»

- 2.1. Календарный учебный график (КУГ)11
- 2.2. Условия реализации программы 11
- 2.3. Форма аттестации..... 12
- 2.4. Оценочные материалы 12
- 2.5. Методические материалы 13
- 2.6. Список литературы14
- Приложения.....15

РАЗДЕЛ №1 «Комплекс основных характеристик программы»

1.1. Пояснительная записка

Дополнительная общеобразовательная (общеразвивающая) программа **«Основы программирования»** имеет **техническую направленность**.

Дополнительность программы заключается в том, что она отвечает интересам детей, способствует углублению знаний, умений и навыков, приобретаемых в общеобразовательной школе на уроках информатики.

Актуальность программы.

С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать.

Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

Отличительные особенности программы.

Основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект в результате освоения программы. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

Новизна программы

Новизна Программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard,

используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

Педагогическая целесообразность Программы

Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и сложные программы.

Адресат программы.

Программа ориентирована на детей, склонных связать свою будущую жизнь с профессией программиста.

Набор производится по желанию учащихся и их родителей.

Возрастная категория обучающихся по программе от 14 до 17 лет.

Срок реализации программы составляет 1 год. Общее количество часов в год составляет 144 часа.

Форма обучения – очная форма реализации программы.

Особенности организации образовательного процесса - групповые – для всей группы при изучении общих практических и теоретических вопросов. Наполняемость группы до 15 человек. Состав группы постоянный.

В ходе реализации программы применяется дифференцированный, индивидуальный подход к каждому обучающемуся.

Занятия проводятся 2 раза в неделю по 2 часа (2 раза по 45 минут с перерывом 15 минут).

Уровень реализуемой программы – базовый.

Цель Программы: Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи Программы

Обучающие:

- обучить языку программирования Python и созданию программ на его основе;
- научить создавать прикладное программное обеспечение;
- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

Развивающие:

- развивать память и внимание, познавательную и творческую активность;
- развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
- развивать логическое мышление.

Воспитательные:

- прививать интерес к активному творческому самовыражению, культуре труда;
- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма.

1.2. Содержание программы.

Учебный (тематический) план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Введение в Python	14	6	8	
1.1.	Язык Python. Среда программирования. Особенности ввода-вывода	4	2	2	Практическое задание
1.2.	Типы данных, операции. Оператор присваивания	4	2	2	Выполнение упражнений
1.3.	Числа. Стандартные операции	6	2	4	Практическое задание
2.	Алгоритмические конструкции	22	6	16	
2.1.	Условный оператор	2	2	-	Опрос

2.2.	Цикл while. Теория чисел	10	2	8	Практическое задание
2.3.	Цикл for	10	2	8	Практическое задание
3.	Строки	20	8	12	
3.1.	Литералы строк	4	2	2	Практическое задание
3.2.	Срезы строк	4	2	2	Практическое задание
3.3.	Методы строк	12	4	8	Практическое задание
4.	Функции	20	8	12	
4.1.	Парадигма и преимущества структурного программирования	2	2	-	Опрос
4.2.	Граф и стек вызовов функций. Области видимости переменных	4	2	2	Практическое задание
4.3.	Прямая рекурсия	10	2	8	Практическое задание
4.4.	Косвенная рекурсия	4	2	2	Решение задач повышенной трудности
5.	Списки и кортежи	20	8	12	
5.1.	Списки и кортежи в Python. Сходства и различия	2	2	-	Опрос
5.2.	Операции со списками	10	2	8	Практическое задание
5.3.	Срезы списков	4	2	2	Практическое задание
5.4.	Матрицы. Операции над матрицами.	4	2	2	Решение задач повышенной трудности
6	Словари и множества	8	4	4	
6.1.	Словари	4	2	2	Практическое задание
6.2.	Множества	4	2	2	Практическое задание
7.	Объектно-ориентированное программирование	38	4	34	
7.1.	Классы в Python	2	2	-	Опрос
7.2.	Разработка собственного класса	6	2	4	Практическое задание

7.3.	Разработка и программирование собственного проекта	30	-	30	-
8.	Заключительное занятие	2	-	2	
8.1.	Подведение итогов. Индивидуальный проект	2		2	Защита проекта
Всего		144	44	100	

Содержание учебного (тематического) плана

1. Введение в Python

Основные понятия: трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная, динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты операций, литералы чисел, операция присваивания, PEP 8.

1.1. Язык Python. Среда программирования. Особенности ввода-вывода

Теория. Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим работы программы.

Практика. Установка языка программирования Python 3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.

1.2. Типы данных, операции. Оператор присваивания

Теория. Ввод и вывод числовой информации.

Практика. Тренировочное задание на ввод и вывод числовой информации.

1.3. Числа. Стандартные операции

Теория. Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8).

Практика. Решение простых задач в интерактивном режиме.

2. Алгоритмические инструкции

Основные понятия: логический тип данных, логические операции (and, or, not, A), условный оператор, условное и альтернативное исполнение алгоритма, операторы сравнения, вложенность операторов, оператор цикла, переменная-флаг, генерация псевдослучайной последовательности, инструкции break, continue и pass.

2.1. Условный оператор

Теория. Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.

2.2. Цикл while

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики.

2.3. Цикл for

Теория. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

3. Строки

Основные понятия: символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные escape- последовательности, положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

3.1. Литералы строк

Теория. Понятие «литералы строк».

Практика. Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

3.2. Срезы строк

Теория. Форматирование строки.

Практика. Преобразование строки. Решение упражнений.

3.3. Методы строк

Теория. Методы работы со строкой.

Практика. Применение методов строки. Решение задач.

4. Функции

Основные понятия: подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции.

4.1. Парадигма и преимущества структурного программирования

Теория. Обзор парадигм программирования. Особенности применения языков программирования.

4.2. Граф и стек вызовов функций. Области видимости переменных

Теория. Применение стек и граф вызовов функций.

Практика. Выполнение тренировочных упражнений.

4.3. Прямая рекурсия

Теория. Понятие прямой рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

4.4. Косвенная рекурсия

Теория. Понятие косвенной рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

Практические занятия к темам 4.2., 4.3., 4.4.: Нахождение суммы чисел. Вычисление степени. Числа Фибоначчи. Ханойские башни
Использование библиотеки математических функций. Решение задач повышенной трудности.

5. Списки и кортежи

Основные понятия: список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировки списка, случайное перемешивание.

5.1. Списки и кортежи в Python. Сходства и различия

Теория. Представление списка и кортежа в памяти компьютера, сходства и различия.

5.2. Операции со списками

Теория. Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка.

5.3. Срезы списков

Теория. Методы работы со списком и кортежем. Методы сортировки списка.

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

5.4. Матрицы. Операции над матрицами

Теория. Вычисление сложности алгоритма. Многомерные списки.

Практика. Решение задач повышенной трудности.

6. Словари и множества

Основные понятия: *словарь, множество, ключ, кодирование.*

6.1. Словари

Теория. Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

Практика. Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.

6.2. Множества

Теория. Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

Практика. Решение задач повышенной трудности.

7. Объектно-ориентированное программирование (ООП)

Основные понятия: ООП, класс, метод INIT, экземпляр, наследование, полиморфизм, исключения, виджет, интерфейс, событие, техническое задание, проект, проектная деятельность, виды проектов.

7.1. Классы в Python

Теория. Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «интерфейс», «событие».

7.2. Разработка собственного класса

Теория. Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.

Практика. Создание собственного класса.

7.3. Разработка и программирование собственного проекта

Практика. Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и презентации проекта.

8. Подведение итогов

8.1. Защита индивидуального проекта.

Организационно-педагогические условия реализации программы.

Занятия проводятся в следующих формах: лекции, семинары, практические задания.

Дидактический материал, необходимый для проведения занятий:

- краткие конспекты материалов для лекций;
- распечатки заданий для практикумов;
- презентационные материалы для объяснения;
- карточки с индивидуальными заданиями.

Техническое оснащение занятий:

- компьютер для демонстрации презентаций;
- проектор;
- рабочие компьютеры учащихся для работы с доступом в Интернет;
- принтер для распечатки заданий.

Ожидаемые результаты и способы определения их результативности

По итогам реализации программы, учащиеся

будут знать:

- принципы программирования на языке Python;
- основы дизайнерского оформления созданных программ.

По итогам реализации программы, учащиеся

будут уметь:

- производить чтение и запись программ на языке Python;
- запускать и отлаживать программу.

Формы определения результативности обучения

Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий.

Формы аттестации и оценочные материалы

В процессе реализации программы предусмотрены следующие виды контроля:

- *входной контроль проводится* с целью определения уровня знаний учащихся (Приложение № 1);

- *промежуточный контроль* проводится регулярно на занятиях с целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;

- *итоговый контроль* – защита проекта.

Формы подведения итогов реализации программы

Результаты обучения по программе выявляются по итогам проведения олимпиад, соревнований по программированию, защиты проекта.

РАЗДЕЛ №2 «Комплекс организационно-педагогических условий»

2.1. Календарный учебный график

Место проведения: МОУ Елаурская СШ им. А.П. Дмитриева

Время проведения занятий:

Изменения расписания занятий:

№ п/п	Тема занятия	Кол-во часов	Форма занятия	Формы контроля	Дата Планируемая (число, месяц)	Дата Фактическая (число, месяц)	Причины изменения даты
1	Вводное занятие. Инструктаж по ТБ. Язык программирования Python. Достоинства и недостатки.	1	Теоретическое	Педагогическое наблюдение, беседа			
2	Области применения. Интерактивный режим работы программы.	1	Комбинированное	Педагогическое наблюдение, беседа			
3	Установка языка программирования Python 3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.	1	Практическое	Практическая работа			
4	Особенности ввода-вывода. Входной контроль.	1	Комбинированное	Практическая работа Тест			
5-6	Ввод и вывод числовой информации.	2	Теоретическое	Педагогическое наблюдение, беседа			
7-8	Тренировочное задание на ввод и вывод числовой информации.	2	Практическое	Практическая работа			
9-10	Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8).	2	Теоретическое	Педагогическое наблюдение, беседа			
11-14	Решение простых задач в интерактивном режиме.	4	Практическое	Практическая работа			
15-16	Условная и циклическая инструкции. Каскадность и вложенность	2	Теоретическое	Опрос			

	алгоритмических инструкций.						
17-18	Цикл while. Теория чисел. Фильтрация потока чисел.	2	Теоретическое	Педагогическое наблюдение, беседа			
19-20	Решение задач на анализ чисел потока и целочисленной арифметики (НОД и НОК)	2	Практическое	Практическая работа			
21-22	Решение задач на анализ чисел потока и целочисленной арифметики (проверка числа на простоту)	2	Практическое	Практическая работа			
23-24	Решение задач на анализ чисел потока и целочисленной арифметики (Алгоритм Евклида)	2	Практическое	Практическая работа			
25-26	Решение задач на анализ чисел потока и целочисленной арифметики (Нахождение максимума и минимума)	2	Практическое	Практическая работа			
27-28	Цикл for. Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.	2	Теоретическое	Педагогическое наблюдение, беседа			
29-30	Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (НОД и НОК).	2	Практическое	Практическая работа			
31-32	Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (Проверка числа на простоту).	2	Практическое	Практическая работа			
33-34	Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (Алгоритм Евклида).	2	Практическое	Практическая работа			
35-36	Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности (Нахождение максимума и минимума).	2	Практическое	Практическая работа			
37-	Понятие «литералы строк».	2	Теоретическое	Педагогическое			

38				наблюдение, беседа			
39-40	Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.	2	Практическое	Практическая работа			
41-42	Форматирование строки.	2	Теоретическое	Педагогическое наблюдение, беседа			
43-44	Преобразование строки. Решение упражнений.	2	Практическое	Практическая работа			
45-46	Методы работы со строкой.	2	Теоретическое	Педагогическое наблюдение, беседа			
47-48	Методы работы со строкой. Решение задач.	2	Практическое	Практическая работа			
49-50	Применение методов строки.	2	Теоретическое	Педагогическое наблюдение, беседа			
51-56	Применение методов строки. Решение задач.	6	Практическое	Практическая работа			
57-58	Обзор парадигм программирования. Особенности применения языков программирования.	2	Теоретическое	Опрос			
59-60	Применение стек и граф вызовов функций.	2	Теоретическое	Педагогическое наблюдение, беседа			
61-62	Выполнение тренировочных упражнений.	2	Практическое	Практическая работа			
63-64	Понятие прямой рекурсии и ее применение.	2	Теоретическое	Педагогическое наблюдение, беседа			
65-72	Выполнение тренировочных упражнений.	8	Практическое	Практическая работа			
73-74	Понятие косвенной рекурсии и ее применение.	2	Теоретическое	Педагогическое наблюдение, беседа			

75-76	Выполнение тренировочных упражнений.	2	Практическое	Практическая работа			
77-78	Представление списка и кортежа в памяти компьютера, сходства и различия.	2	Теоретическое	Опрос			
79-80	Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).	2	Теоретическое	Педагогическое наблюдение, беседа			
81-88	Решение задач на ввод-вывод элементов одномерного и многомерного списка.	8	Практическое	Практическая работа			
89-90	Методы работы со списком и кортежем. Методы сортировки списка	2	Теоретическое	Педагогическое наблюдение, беседа			
91-92	Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.	2	Практическое	Практическая работа			
93-94	Вычисление сложности алгоритма. Многомерные списки.	2	Теоретическое	Педагогическое наблюдение, беседа			
95-96	Решение задач повышенной трудности.	2	Практическое	Практическая работа			
97-98	Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.	2	Теоретическое	Педагогическое наблюдение, беседа			
99-100	Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.	2	Практическое	Практическая работа			
101-102	Понятие множества. Создание множеств. Множество, преимущества и недостатки,	2	Теоретическое	Педагогическое наблюдение, беседа			

	методы работы с множеством.						
103-104	Решение задач повышенной трудности.	2	Практическое	Практическая работа			
105-106	Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «событие».	2	Теоретическое	Опрос			
107-108	Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.	2	Теоретическое	Педагогическое наблюдение, беседа			
109-112	Создание собственного класса.	4	Практическое	Практическая работа			
113-142	Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и презентации проекта.	30	Практическое	Практическая работа			
143-144	Защита индивидуального проекта.	2	Практическое	Практическая работа			
	Итого	144					

2.2. Условия реализации программы

Материально-техническое обеспечение:

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары;

Информационное обеспечение:

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 Гб DDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19’;

Программное обеспечение:

- Операционная система Windows 7 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JatBrainsPyCharm;
- FoxitReader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернет серфинга.
-

2.3. Формы аттестации (контроля)

Входящий контроль осуществляется при комплектовании группы в начале учебного года. *Цель* – определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися.

Форма контроля: тестирование.

Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

Промежуточный контроль осуществляется в конце I полугодия учебного года.

Форма контроля: тест, решение практических задач средствами языка программирования.

Итоговый контроль осуществляется в конце учебного года.

Форма контроля: защита творческого проекта.

Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

2.4. Оценочные материалы

Входящий контроль: *Тестирование (Приложение 1)*

Промежуточный контроль: *Тестирование, решение задач (Приложение 2)*

Итоговый контроль: *учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.*

2.5. Методическое обеспечение

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации на период до 2020 года.

Методы обучения:

1. **Словесный:** объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.

2. **Наглядный:** применение демонстрационного материала, наглядных пособий, презентаций по теме.

3. **Практический:** индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.

4. **Интерактивный:** создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Технологии:

1. **Технология проблемного диалога.** Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.

2. **Технология коллективного взаимообучения** («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. **Игровая технология.** Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.

4. **Элементы здоровьесберегающих технологий** являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. **Проектная технология** предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования,

школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

2.6. Список литературы

Нормативно-правовые акты и документы

1. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ
2. Приказ Министерства образования и науки РФ от 29.08.2013 №1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Письмо Министерства образования Российской Федерации от 18 июня 2003 №28-02-484/16 «Требования к содержанию и оформлению образовательных программ дополнительного образования детей»,
4. Постановление Главного государственного санитарного врача Российской Федерации от 4.07.2014 г. «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
5. Стратегия инновационного развития Российской Федерации на период до 2020 года. Распоряжение правительства Российской Федерации от 08.12.2011 № 2227-р.
6. Концепция развития дополнительного образования детей. Распоряжение Правительства Российской Федерации от 24 апреля 2015 года № 729-р.
7. «Стратегия развития воспитания в Российской Федерации на период до 2025 года». Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р.

Литература для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

Литература для родителей и учащихся

1. Сайт / справочные материалы[Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
3. Сайт / Адаптивный тренажер Python[Электронный ресурс] – Режим доступа: <https://stepik.org/course/431> , свободный.
4. Сайт / среда разработки для языка Python[Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net, свободный.

Входящий контроль

Низкий уровень: 0-70% выполненных заданий;

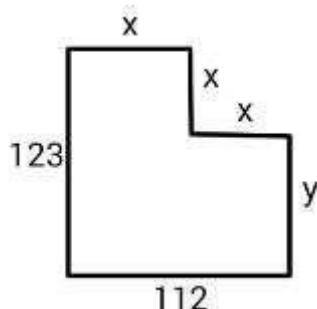
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

Для учащихся – 14-17 лет

Фамилия, Имя

1. Из прямоугольника вырезали квадрат со стороной x , получилась фигура как на рисунке. Чему равна сумма цифр y ?



2. В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой.

Каково максимальное количество слов в этом языке?

3. Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым — 47 кусков, а если по зеленым — 31 кусок.

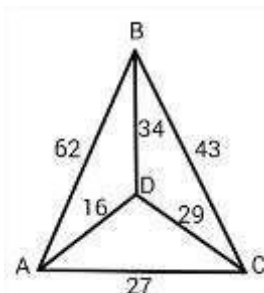
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

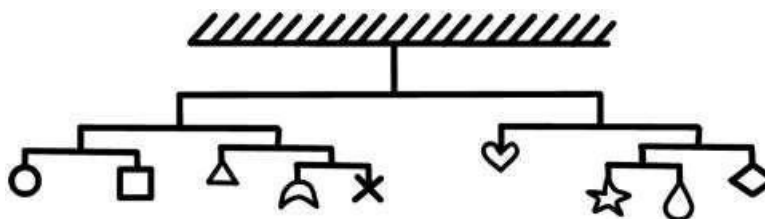
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

4. На рисунке показано расположение городов А, В, С и D и расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу.

Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



5. Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке. Оказалось, что все её части находятся в равновесии. Сколько весит ромб?



6. Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

7. Сравните пары слов. Сколько среди них полностью идентичных?

O/Sanmarco

O/Samnarco

Ф.Wagonerrte

Ф.Wagonertre

A.S.Schmetterling

A.S.Schnetterling

N.V.Murfreesboroque

N.V.Munfreesboroque

P.S.Splendoursec

P.S.Sqlendoursec

Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра – воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

8. Вам предложены несколько высказываний и следствие из них (выделено жирным).

Согласны ли Вы с этим следствием?

1. Все клёны — растения.

2. Некоторые растения быстро желтеют.

Значит, некоторые клёны быстро желтеют.

<input type="checkbox"/>	Да
<input type="checkbox"/>	Нет

9. Гусеница прогрызает яблоко диаметром 6 сантиметров насквозь за 16 секунд, вылезая снаружи полностью.

Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути.

Какова длина гусеницы в сантиметрах?

Итоговый контроль
Диагностическая карта

№/п	Имя	Входящая Диагностика	Промежуточ- ная диагностика	Итоговая диагностика
1				
2				
3+				
	Итого в %	Н С В		Н С В

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).

Низкий уровень

Учащиеся должны знать/понимать:

- технику безопасного поведения во время занятий;
- правила поведения в общественных местах,
- понятие программы Python; общую структуру программы;
- основные типы данных;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- основные циклы с условием;
- основные правила записи циклов условием;
- формат записи цикла с параметром;
- понятие функции;
- основные принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- операции со строками;
- способ описания списка;
- способ описания кортежа;
- способ описания словаря;
- основные операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога;
- работать в паре, малой группе;
- выполнить установку программы под руководством педагога;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе под руководством педагога;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием под руководством педагога;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога;
- создавать и использовать основные функции;
- описывать и соединять строки;
- находить подстроку в строке с помощью педагога;
- находить количество слов в строке;
- вводить и выводить элементы списка под руководством педагога;
- приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога;
- описывать множества под руководством педагога;
- определять принадлежность элемента множеству по наводящим вопросам;
- определять вид ошибок и находить ошибки в программе под руководством педагога;
- составлять элементарные алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

Средний уровень

Учащиеся должны знать/ понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности;
- основные приемы взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;

- основные способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- основные шаги работы над проектом, его презентации.

Учащиеся должны уметь:

- уважительно относиться к преподавателям и сверстникам;
- применять некоторые приемы логического (абстрактное) мышления;
- концентрировать внимание на одном или двух объектах;
- понимать причины успеха/неуспеха с помощью анализа педагога;
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;

- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать и сохранять цели и задачи учебной деятельности в ходе занятия;
- планировать свою деятельность с помощью взрослого;
- сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи;
- понимать причины успеха/неуспеха учебной деятельности под руководством взрослого;
- делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога;
- выражать творческие идеи, разработать творческий проект на основе образца;
- конструктивно взаимодействовать в составе группы в ходе работы над проектом.

Высокий уровень

Учащиеся должны знать/понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий;
- приемы конструктивного взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- Общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;

- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- правила и этапы работы над проектом;
- приемы успешной презентации проекта.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;

- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать учебную задачу, анализировать достижение результата;
- делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений;
- понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога;
- планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи;
- создать проектировочную команду и организовать ее деятельность;
- разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;
- самостоятельно подготовить презентацию и защитить проект.

<p>вопросы для промежуточного контроля по усвоению материала <i>(Промежуточный контроль проводится в последнюю неделю декабря)</i></p>
<p>Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления: $11111 \cdot 11111111$ — произведение чисел 11111 (5 единиц) и 11111111 (7 единиц)</p>
<p>Запишите число $1.2345e3$ в виде десятичной дроби.</p>
<p>Составьте и запишите выражение для вычисления: 2014.0^{14} (Возвестив 14 степень) Обратите внимание на запись числа: это вещественное число.</p>
<p>Приведите к целому типу число 2.99</p>
<p>Расставьте скобки в выражении $a \text{ and } b \text{ or not } a \text{ and not } b$ в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).</p>
<p>Найдите результат выражения для заданных значений a и b</p>

<p>Учитывайте регистр символов при ответе.</p> <p>a = True b = False a and b or not a and not b</p>
<p>Отметьте выражения, значения которых равны True:</p> <ul style="list-style-type: none">• "239" < "30" and 239 < 30• "239" < "30" and 239 > 30• "239" > "30" and 239 < 30• "239" > "30" and 239 > 30
<p>Укажите результат выражения: "123" + "42"</p>
<p>Какое значение будет у переменной i после выполнения фрагмента программы?</p> <pre>i = 0 while i <= 10: i = i + 1 if i > 7: i = i + 2</pre>
<p>Сколько итераций цикла будет выполнено в этом фрагменте программы?</p> <pre>i = 0 while i <= 10: i = i + 1 if i > 7: i = i + 2</pre>
<p>Сколько всего знаков * будет выведено после исполнения фрагмента программы:</p> <pre>i = 0 while i < 5: print('*') if i % 2 == 0: print('**') if i > 2: print('***') i = i + 1</pre>
<p>Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:</p> <pre>i = 0 s = 0 while i < 10: i = i + 1 s = s + i if s > 15: break</pre>

$i = i + 1$

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

```
 $i = 0$   
 $s = 0$   
while  $i < 10$ :  
   $i = i + 1$   
   $s = s + i$   
if  $s > 15$ :  
  continue  
 $i = i + 1$ 
```

Задачи для промежуточного контроля по усвоению материала

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.

Поддерживаемые операции: +, -, /, *, mod, pow, div, где mod — это взятие остатка от деления, pow — возведение в степень, div — целочисленное деление.

Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!".

Обратите внимание, что на вход программе приходят вещественные числа.

Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты.

Для числа π в стране Малевии используют значение 3.14.

Формат ввода, который используют Малевийцы:

Треугольник

a

b

c

где a , b и c — длины сторон треугольника

прямоугольник

a

b

где a и b — длины сторон прямоугольника

круг

r

где r — радиус окружности

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.

Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

На вход программе подаётся строка из шести цифр. (Пример:123321)

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и сразу после этого выводит сумму квадратов всех считанных чисел.

Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, после этого считывание продолжать не нужно.

В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число n — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если $n = 7$, то программа должна вывести 1 2 2 3 3 3 4.

Выведите таблицу размером $n \times n$, заполненную числами от **1** до n^2 по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере (здесь $n=5$)