

Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа №3»
Режевского муниципального округа

«РАССМОТРЕНО»
на заседании педагогического совета
протокол №1
от 26.08.2025 г.



Дополнительная общеобразовательная общеразвивающая
программа

**«ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ
PYTHON»**

Направленность программы: техническая

Возраст обучающихся: 14-18 лет

Срок реализации: 1 год

Составитель:

Собянин Олег Алексеевич,
учитель информатики

г. Реж, 2025

Оглавление

«Программирование на языке Python»	
1.1. Пояснительная записка.....	3
Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» имеет техническую направленность.	3
1.2. Цели и задачи дополнительной общеразвивающей программы «Программирование на языке Python»	4
1.3. Содержание дополнительной общеобразовательной общеразвивающей программы.....	5
Учебный план	5
Содержание учебного плана	7
1.4. Планируемые результаты освоения дополнительной	11
общеразвивающей программы «Программирование на языке Python»	11
II Комплекс организационно-педагогических условий реализации дополнительной общеразвивающей программы «Программирование на языке Python»	12
2.1. Календарный учебный график на 2025-2026 учебный год «Программирование на языке Python» техническая направленность	12
2.2. Условия реализации дополнительной общеразвивающей.....	13
программы «Программирование на языке Python»	13
2.3 . Формы аттестации/контроля.....	13
2.4. Оценочные материалы.....	14
2.5. Методические материалы.....	14
Список литературы	17
Приложение 1	18
Здоровьесберегающие технологии.....	18
Приложение 2	19
Входящий контроль	19
Приложение 3.....	22
Итоговый контроль	22

1.1. Пояснительная записка

Направленность программы

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» имеет техническую направленность.

Актуальность программы обусловлена широким распространением информационно-коммуникационных технологий в обществе и необходимостью обеспечивать связанную с этим инфраструктуру специалистами. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём на уровне начинающего программиста.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке программирования. Для обучения в рамках программы выбран язык Python, который является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

Синтаксис языка достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на алгоритмических аспектах программирования.

Отличительная особенность программы, новизна: Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

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.

Адресат программы:

Программа адресована детям 8-11 класса

Содержание программы разработано с учетом принятия всех желающих, не имеющих противопоказаний по здоровью.

Дополнительная общеразвивающая программа «Программирование на языке Python» рассчитана на детей 14–18 лет, проявляющих интерес к информационно-коммуникационным технологиям. Учащиеся объединения являются разными по возрасту и социальному статусу. Объединение комплектуется на основании заявлений законных представителей учащихся (самих учащихся с 14 лет). Группы формируются из школьников разного возраста на добровольной внеконкурсной основе.

Программа составлена с учётом индивидуальных и возрастных особенностей учащихся. Психолого-педагогические особенности учащихся определяют и методы индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута

Группы постоянного состава. Набор обучающихся свободный по 15 человек в группу.

Объём и срок освоения программы.

Объём программы – 75 часов (2,5 часа в неделю).

Программа рассчитана на 1 год обучения.

Форма обучения.

Очная

Особенности организации образовательного процесса.

Занятия проводятся со всем составом группы. Группа формируется из обучающихся 14-18летнего возраста. Состав группы постоянный.

Формы реализации образовательной программы: традиционная.

Режим занятий.

Продолжительность одного академического часа - 45 мин. Общее количество часов в неделю – 2,5 часа. Занятие проводится 1 раз в неделю по 2,5 часа.

1.2. Цели и задачи дополнительной общеразвивающей программы «Программирование на языке Python»

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:

Образовательные (предметные):

- Познакомить с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных.
- Научить составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.
- Научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач.
- Научить организовывать данные для эффективной алгоритмической обработки.
- Научить разрабатывать алгоритмы и реализовывать их на языке программирования Python.
- Научить осуществлять отладку и тестирование программы.

Личностные:

- повышение общекультурного уровня;
- вооружение правильным методологическим подходом к познавательной и практической деятельности;
- выделение и раскрытие роли информационных технологий и компьютеров в развитии современного общества;
- привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- воспитание стремления к овладению техникой исследования;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

Метапредметные:

владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;

- планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;
- прогнозирование – предвосхищение результата.

**1.3. Содержание дополнительной общеобразовательной общеразвивающей
программы
Учебный план**

№ п/п	Тема занятий	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
	Инструктаж по технике безопасности в компьютерном классе	0,5	0,5		Опрос
1.	Раздел 1. Знакомство с языком Python	4,5	2	2,5	Тестирование, решение практических задач
1.1	Занятие 1. Общие сведения о языке Практическая работа: Установка программы Python	2	1	1	
1.2.	Занятие 2. Режимы работы Практическая работа: Режимы работы с Python	2,5	1	1,5	
2.	Раздел 2. Переменные и выражения	10	4	6	Тестирование, решение практических задач
2.1	Занятие 3. Переменные Практическая работа: Работа со справочной системой	2,5	1	1,5	
2.2	Занятие 4. Выражения Практическая работа: Выражения	2,5	1	1,5	
2.3	Занятие 5. Ввод и вывод	2,5	1	1,5	
2.4	Занятие 6. Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на элементарные действия с числами	2,5	1	1,5	
3	Раздел 3. Условные предложения	7,5	3	4,5	Тестирование, Решение практических задач
3.1	Занятие 7. Логические выражения и операторы. Практическая работа: Логические выражения	2,5	1	1,5	
3.2	Занятие 8. Условный оператор Практическая работа: «Условный оператор»	2,5	1	1,5	
3.3	Занятие 9. Множественное ветвление Практическая работа: Множественное ветвление	2,5	1	1,5	

	Практическая работа: «Условные операторы»				
4	Раздел 4. Циклы	10	4	6	Тестирование, решение практических задач, творческая работа
4.1.	Занятие 10. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	2,5	1	1,5	
4.2.	Занятие 11. Оператор цикла for Практическая работа Решение задачи с циклом for.	2,5	1	1,5	
4.3.	Занятие 12. Вложенные циклы Практическая работа: Реализация циклических алгоритмов	2,5	1	1,5	
4.4.	Занятие 13. Случайные числа Практическая работа: Случайные числа	2,5	1	1,5	
5	Раздел 5. Функции	10	4	6	Тестирование, решение практических задач
5.1.	Занятие 14. Создание функций Практическая работа Создание функций	2,5	1	1,5	
5.2.	Занятие 15. Локальные переменные Практическая работа Локальные переменные	2,5	1	1,5	
5.3.	Занятие 16. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	2,5	1	1,5	
5.4	Занятие 17. Рекурсивные функции Практическая работа 5.4. Рекурсивные функции	2,5	1	1,5	
6	Раздел 6. Строки - последовательности символов	7,5	3	4,5	Решение практических задач
6.1.	Занятие 18. Строки Практическая работа: Строки	2,5	1	1,5	
6.2.	Занятие 19. Срезы строк	2,5	1	1,5	
6.3	Занятие 20. Примеры решения задач со строками Практическая работа:	2,5	1	1,5	

	Решение задач со строками.				
7	Раздел 7. Сложные типы данных	17,5	7	10,5	Тестирование, Решение практических задач
7.1.	Занятие 21. Списки Практическая работа: Списки	2,5	1	1,5	
7.2	Занятие 22. Срезы списков	2,5	1	1,5	
7.3	Занятие 23. Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	2,5	1	1,5	
7.4	Занятие 24. Матрицы	2,5	1	1,5	
7.5	Занятие 25. Кортежи	2,5	1	1,5	
7.6	Занятие 26. Введение в словари	2,5	1	1,5	
7.7	Занятие 27. Множества в языке Python	2,5	1	1,5	
8	Раздел 8. Стил программирования и отладка программ	7,5	2	5,5	Решение практических задач, презентация проекта, рефлексия.
8.1	Занятие 28. Стил программирования	2,5	1	1,5	
8.2	Занятие 29. Отладка программ	2,5	1	1,5	
8.3	Занятие 30. Зачет по курсу «Программирование на языке Python»	2,5		2,5	
	ВСЕГО	75	29,5	45,5	

Содержание учебного плана

Вводное занятие (0,5 часа).

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. Практика:

Опрос. Тренинг на командообразование.

Раздел 1. Знакомство с языком Python (4,5 часов)

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python Практическая работа 1.2. Режимы работы с Python Тест № 1. Знакомство с языком Python

Учащиеся должны знать / понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Учащиеся должны уметь:

- выполнить установку программы;

- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Раздел 2. Переменные и выражения (10 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами Тест № 2. Выражения и операции.

Учащиеся должны знать / понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Раздел 3. Условные предложения (7,5 часов)

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа 3.4. "Условные операторы"

Самостоятельная работа № 1. Решение задач по теме "Условные операторы". Зачетная работа № 1. "Составление программ с ветвлением".

Тест № 3. "Условные операторы".

Учащиеся должны знать / понимать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы (10 часов)

Теория: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи" Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом. Самостоятельная работа № 2 "Составление программ с циклом"

Тест № 4. Циклы

Творческая работа № 1. "Циклы"

Учащиеся должны знать / понимать:

- 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.

Учащиеся должны уметь:

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

Раздел 5. Функции (10 часов)

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция `lambda`. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи. Практическая работа

5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций Практическая работа

5.4. Рекурсивные функции

Самостоятельная работа № 3 по теме "Функции" Тест № 5. Функции

Учащиеся должны знать / понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Учащиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Раздел 6. Строки - последовательности символов (7,5 часов)

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор `in`. Модуль `string`. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Учащиеся должны знать / понимать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

Учащиеся должны уметь:

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

Раздел 7. Сложные типы данных (17,5 часов)

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками. Тест № 7. Списки

Учащиеся должны знать / понимать:

- 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.

Учащиеся должны уметь:

- 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.

Раздел 8. Стиль программирования и отладка программ (7,5 часов)

Стиль программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

Учащиеся должны знать / понимать:

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

1.4. Планируемые результаты освоения дополнительной общеразвивающей программы «Программирование на языке Python»

Реализация программы предполагает достижение обучающимися следующих личностных, метапредметных и предметных результатов:

Предметные:

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать
- изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- формирование умения соблюдать нормы информационной этики и права.

Метапредметные:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно- исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно- исследовательской деятельности.

Личностные:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно- исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

II Комплекс организационно-педагогических условий реализации дополнительной общеразвивающей программы «Программирование на языке Python»

2.1. Календарный учебный график на 2025-2026 учебный год «Программирование на языке Python» техническая направленность

Календарный учебный график творческого объединения составлен на основе Годового календарного графика учреждения и является документом, регламентирующим организацию образовательной деятельности в учреждении (Приложение).

Начало учебного года 01.09.2025 г. Окончание учебного года 23.05.2026 г.
Продолжительность учебного года 34 недели.

Год обучения (уровень)	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год обучения			34	34	75 часов, 2,5 часа в неделю	1 раз в нед. по 2,5 часа

2.2. Условия реализации дополнительной общеразвивающей программы «Программирование на языке Python»

Материально-техническое обеспечение:

Помещение: кабинет для занятий, расположенный по адресу: Свердловская область, г. Реж, МБОУ СОШ №3. Кабинет, соответствующий санитарно-гигиеническим требованиям.

Для реализации образовательной программы используются:

1. столы для компьютера;
2. компьютерные стулья;
3. шкафы для дидактических материалов, пособий;
4. специальная и научно-популярная литература для педагога и учащихся;
5. канцтовары;

Информационно-методическое обеспечение:

1. персональный компьютер (12 шт.);
2. мультимедийный проектор;
3. видеоматериалы разной тематики по программе;
4. оргтехника;
5. выход в сеть Internet;

Аппаратное обеспечение:

1. Процессор не ниже Core2 Duo;
2. Объем оперативной памяти не ниже 4 ГбDDR3;
3. Дисковое пространство на менее 128 Гб;
4. Монитор диагональю на мене 19';

Программное обеспечение:

1. Операционная система Windows 7 Профессиональная или выше;
2. Интерпретатор Python версии 3.7 и выше;
3. IDE JatBrains PyCharm;
4. Foxit Reader или другой просмотрщик PDF файлов;
5. WinRAR;
6. Пакет офисных программ;
7. Adobe Photoshop или другой растровый графический редактор;
8. Любой браузер для интернет серфинга.

Кадровое обеспечение программы: Педагог, работающий по данной программе, должен знать основы программирования. По данной образовательной программе работают педагоги с высшим образованием (НТГПИ, специальность: «Физика и информатика», квалификация: учитель информатики) Собянин Олег Алексеевич, с первой квалификационной категорией. Педагогический стаж работы 35 лет.

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

Диагностируются результаты различным образом и на различных этапах деятельности. Для определения уровня знаний и умений обучающихся проводятся: в начале года - входной контроль, в середине года - промежуточный контроль, в конце года — итоговый контроль. Основной способ оценки – самооценка выполнения программ на языке программирования. Эта оценка присутствует на всех этапах. Одной из форм диагностики результатов является зачет или контрольная работа.

Входящий контроль: *Тестирование (Приложение 2)*

Промежуточный контроль: *Тестирование, решение задач (Приложение 3)*

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

2.5. Методические материалы

Учебно-воспитательный процесс направлен на развитие природных задатков обучающихся, на реализацию их интересов и способностей. Каждое занятие обеспечивает развитие личности. При планировании и проведении занятий применяется личностноориентированная технология обучения, в центре внимания которой неповторимая личность, стремящаяся к реализации своих возможностей, а также системно-деятельностный метод обучения. Данная программа допускает творческий, импровизированный подход со стороны детей и педагога того, что касается возможной замены порядка раздела, введения дополнительного материала, методики проведения занятий. Руководствуясь данной программой, педагог имеет возможность увеличить или уменьшить объем и степень технической сложности материала в зависимости от состава группы и конкретных условий работы.

Методы обучения:

Выбор метода обучения зависит от содержания занятий, уровня подготовки и опыта обучающихся.

1. Традиционные методы развития интереса обучающихся к техническому творчеству можно объединить в группы, каждая из которых наиболее продуктивна на том или ином этапе работы. 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.

Педагогические технологии, применяемые в реализации программы:

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.

Формы организации учебного занятия: комбинированное занятие; беседа; игра; викторина, творческая встреча, соревнование, экскурсия и др.

Алгоритм учебного занятия.

I этап - организационный.

Задача: подготовка детей к работе на занятии.

Содержание этапа: организация начала занятия, создание психологического настроения на учебную деятельность и активизация внимания.

II этап - подготовительный (подготовка к восприятию нового содержания).

Задача: Организовать занимающихся, сообщение задач занятия, подготовка к основной части занятия, к предстоящей работе. Содержание этапа: сообщение темы, цели учебного занятия и мотивация учебной деятельности детей (пример, познавательная задача, проблемное задание детям).

III этап - основной.

В качестве основного этапа могут выступать следующие:

1. Усвоение новых знаний и способов действий.

Задача: обеспечение восприятия, осмысления и первичного запоминания связей и отношений в объекте изучения. Целесообразно при усвоении новых знаний использовать задания и вопросы, которые активизируют познавательную деятельность детей.

2. Первичная проверка понимания.

Задача: установление правильности и осознанности усвоения нового учебного материала, выявление неверных представлений, их коррекция. Применяют пробные

практические задания, которые сочетаются с объяснением соответствующих правил или обоснованием.

3. Закрепление знаний и способов действий, применяют творческие задания, выполняемые детьми самостоятельно.

4. Обобщение и систематизация знаний.

Задача: формирование целостного представления знаний по теме. Распространенными способами работы являются беседа и практические задания.

IV этап – контрольный.

Задача: выявление качества и уровня овладения знаниями, их коррекция.

Используются виды устного и письменного опроса, вопросы и задания различного уровня сложности (репродуктивного, творческого, поисково-исследовательского).

V этап - итоговый.

Задача: дать анализ и оценку успешности достижения цели и наметить перспективу последующей работы.

Содержание этапа: педагог сообщает, как работали обучающиеся на занятии, что нового узнали, какими умениями и навыками овладели.

VI этап - рефлексивный.

Задача: мобилизация детей на самооценку. Может оцениваться работоспособность, психологическое состояние, результативность работы, содержание и полезность учебной работы.

Изложенные этапы могут по-разному комбинироваться, какие-либо из них могут не иметь места в зависимости от педагогических целей и учитывая формы занятия.

Дидактические материалы:

- таблицы;
- схемы;
- памятки;
- видеозаписи, мультимедийные материалы;
- упражнения.

Список литературы

Нормативные документы

Программа соответствует действующим нормативным документам:

1. Федеральный Закон от 29.12.2012г. № 273-ФЗ «Об образовании в Российской Федерации» (далее – ФЗ);
2. Федеральный закон от 14.07.2022 г. № 295-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации» по вопросам воспитания обучающихся»;
3. Федеральный закон РФ от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребенка в Российской Федерации» (в редакции 2013 г.);
4. Постановление Главного государственного санитарного врача РФ от 28.09.2020 N 28 "Об утверждении санитарных правил СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи";
5. Концепция развития дополнительного образования детей до 2030 года (распоряжение Правительства РФ от 31.03.2022. № 678-р);
6. Приказ Министерства просвещения Российской Федерации от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (далее – Порядок).

Методические рекомендации:

- Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (разработанные Минобрнауки России совместно с ГАОУ ВО «Московский государственный педагогический университет», ФГАУ «Федеральный институт развития образования», АНО ДПО «Открытое образование», 2015г.) (Письмо Министерства образования и науки РФ от 18.11.2015 № 09-3242);
- Примерная программа воспитания. Утверждена на заседании Федерального учебнометодического объединения по общему образованию 2.06.2020 г. (<http://form.instrao.ru>);
- Методические рекомендации по разработке программ воспитания.

Литература для обучающихся:

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.
1. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net , свободный.

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.

Приложение 1

Здоровьесберегающие технологии

На занятиях нужно систематически проводить гимнастику для глаз, физкультминутки, выполнять упражнения для пальцев, рук, плеч. Преподаватель должен постоянно напоминать обучающимся о правильной посадке, следить за положением рук, спины, ног.

Примерный комплекс упражнений для глаз:

1. Закрывать глаза, сильно напрягая глазные мышцы, на счет 1 - 4, затем открыть глаза, расслабить мышцы глаз, посмотреть вдаль на счет 1-6. Повторить 4-5 раз.
2. Посмотреть на переносицу и задержать взгляд на счет 1 - 4. До усталости глаза не доводить. Затем посмотреть вдаль на счет 1 - 6. Повторить 4 - 5 раз.
3. Не поворачивая головы, посмотреть направо и зафиксировать взгляд на счет 1 - 4, затем посмотреть вдаль прямо на счет 1 - 6. Аналогичным образом проводятся упражнения с фиксацией взгляда влево, вверх и вниз. Повторить 3 - 4 раза.
4. Перевести взгляд быстро по диагонали: направо вверх - налево вниз, потом прямо вдаль на счет 1 - 6; затем налево вверх - направо вниз и прямо вдаль на счет 1-6. Повторить 4 -5 раз.

Создание комфортного психологического климата, учет индивидуальных особенностей обучающихся, их темперамента, умение работать с различными группами обучающихся, использовать дифференцированный подход в обучении.

Необходимо создавать благоприятный эмоциональный климат на занятиях. Чувство успеха при выполнении заданий, очень положительно влияет на здоровье обучающихся. Проявление доверительного подхода к обучающимся, наличие на занятиях эмоциональной разрядки, создание ситуации успеха – оптимальные приемы создания положительных эмоций на занятии.

И последним, но немаловажным компонентом здоровьесберегающих технологий является формирование, пропаганда здорового образа жизни. Если показывать обучающимся, как соотносится изучаемый материал с повседневной жизнью, приучать их постоянно заботиться о своем здоровье, систематически закреплять их на занятиях, это должно стать для них совершенно естественным и при работе в домашних условиях.

Усталости и перегрузок на занятиях будет меньше, если будет чередоваться и вид деятельности обучающихся, и способы преподнесения информации (зрительная, слуховая) с физкультурными паузами. А создание благоприятной атмосферы в начале и в конце занятия, через улыбку, будет способствовать хорошему настроению обучающихся, как на самом занятии, так и после него.

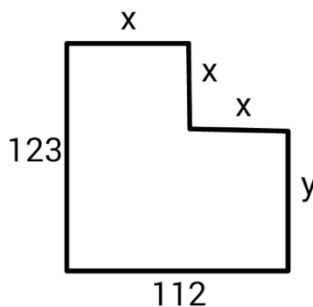
Приложение 2

Входящий контроль

Низкий уровень: 0-70% выполненных заданий;
 Средний уровень: 70-85% правильно выполненных заданий;
 Высокий уровень: 85-100% правильно выполненных заданий.
 Для учащихся – 15-18 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.

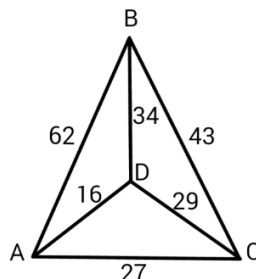
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

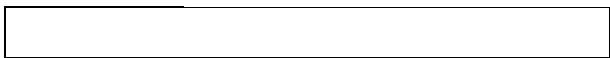
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

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?



--

	HeT
--	-----

--

Для учащихся 14-15 лет

Фамилия, Имя.	
---------------	--

1. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

--	--

2. Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?

--

3. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?

--

4. Как с помощью только одной палочки образовать на столе треугольник?

--

5. В каком месяце болтливая девочка говорит меньше всего?

--

6. Что становится на треть больше, если его поставить вверх ногами?

--

7. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?

--

8. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10.

Вопрос: Сколько кусков пиццы получилось у Юры.

--

9. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела.

Вопрос: Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

--

Приложение 3
Итоговый контроль
Диагностическая карта

№ / п	Имя	Входящая Диагностика	Промежуточная 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.

Вопросы для промежуточного контроля по усвоению материала <i>(Промежуточный контроль проводится в последнюю неделю декабря)</i>											
Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления: $11111 \cdot 1111111$ — произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)											
Запишите число 1.2345e3 в виде десятичной дроби.											
Составьте и запишите выражение для вычисления: 2014.0^{14}(Возвестив 14 степень) Обратите внимание на запись числа: это вещественное число.											
Приведите к целому типу число 2.99											
Расставьте скобки в выражении a and b or not a and not b в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).											
Найдите результат выражения для заданных значений <i>a</i> и <i>b</i> Учитывайте регистр символов при ответе. <table> <tr> <td>a</td><td>=</td><td>True</td></tr> <tr> <td>b</td><td>=</td><td>False</td></tr> <tr> <td colspan="3">a and b or not a and not b</td></tr> </table>			a	=	True	b	=	False	a and b or not a and not b		
a	=	True									
b	=	False									
a and b or not a and not b											
Отметьте выражения, значения которых равны True: • "239" < "30" and 239 < 30 • "239" < "30" and 239 > 30 • "239" > "30" and 239 < 30											

• "239" > "30" and 239 > 30
Укажите результат выражения: "123" + "42"
Какое значение будет у переменной i после выполнения фрагмента программы? <pre> i = 0 while i <= 10: i = i + 1 if i > 7: i = i + 2 </pre>
Сколько итераций цикла будет выполнено в этом фрагменте программы? <pre> i = 0 while i <= 10: i = i + 1 if i > 7: i = i + 2 </pre>
Сколько всего знаков * будет выведено после исполнения фрагмента программы: <pre> i = 0 while i < 5: print('*') if i % 2 == 0: print('**') if i > 2: print('***') i = i + 1 </pre>
Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы: <pre> i = 0 s = 0 while i < 10: i = i + 1 s = s + i if s > 15: break i = i + 1 </pre>
Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы: <pre> i = 0 s = 0 while i < 10: i = i + 1 s = s + i if s > 15: </pre>

```
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 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 $n=5$)