

Муниципальное бюджетное образовательное учреждение  
Средняя общеобразовательная школа №3

Принята на  
педагогическом совете  
протокол № 1 от 30.08.2023 г.



УТВЕРЖДАЮ  
Директор МБОУ СШ №3

*Т.Е.Хлобыстова* Т.Е.Хлобыстова

Приказ № 239 от 30.08.2023 г.

**Программа курса внеурочной деятельности  
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ  
«Основы программирования на языке Python»**

Возраст обучающихся: 12-15 лет  
Срок реализации: 9 месяцев

Автор-составитель:  
Васенин Антон Дмитриевич,  
учитель информатики  
первой квалификационной категории

## 1. НОРМАТИВНО-ПРАВОВАЯ БАЗА

Программы реализуется в соответствии со следующими нормативно-правовыми документами:

1. Законом РФ «Об образовании в Российской Федерации» (№ 273-ФЗ от 29.12.2012)
2. Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (Приказ Минобрнауки РФ от 29.08.2013 г. Закон РФ «Об образовании в Российской Федерации» (№ 273-ФЗ от 29.12.2012) № 1008);
3. Постановлением Главного государственного санитарного врача РФ от 4 июля 2014 г. № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей»;
4. Концепцией развития дополнительного образования детей (утв. распоряжением Правительства РФ от 31 марта 2022 г. № 678-р);
5. Письмом Минобрнауки РФ от 18.11.2015 № 09-3242 «О направлении рекомендаций» (вместе Методические рекомендации по проектированию дополнительных общеразвивающих программ);
6. Письмом Минобрнауки РФ от 14.12.2015 г. № 09-3564 «О внеурочной деятельности и реализации дополнительных общеобразовательных программ» (вместе с «Методическими рекомендациями по организации внеурочной деятельности и реализации дополнительных общеобразовательных программ»);
7. Положением о порядке организации и осуществления образовательной деятельности по дополнительным образовательным программам, утверждённого Приказом управления образования Администрации города Глазова от 31.08.2015 г. №193-ОД;
8. Уставом Муниципального бюджетного общеобразовательного учреждения «Средняя общеобразовательная школа №3»;
9. Концепцией и программой развития образования Муниципального бюджетного общеобразовательного учреждения «Средняя общеобразовательная школа №3» на 2020-2025 г.г.;
10. Положением об организации дополнительного образования детей в МБОУ «СШ №3».

## **2. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

### **2.1. Направленность программы**

Программа внеурочной деятельности «Основы программирования на языке Python» относится к технической направленности.

### **2.2. Актуальность программы**

Актуальность программы обусловлена быстрым внедрением компьютерной техники в повседневную жизнь, переходом к новым технологиям обработки информации. Изучая программирование, обучающиеся лучше понимают возможности и границы применения компьютеров. К ним приходит осознание того, что компьютер является инструментом, управляемым людьми. Не все обучающиеся станут профессиональными программистами, но все выиграют от того, что постигли природу программирования и научились создавать собственные программы.

### **2.3. Отличительные особенности программы**

Отличительная особенность программы «Основы программирования на языке Python» в том, что она является практико-ориентированной. Освоение подростками IT-навыков происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и уверенно овладевать IT-технологиями, что поможет им самоопределиться и выстроить траекторию личностного роста в современном информационном обществе. Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python.

### **2.4. Адресат программы**

Обучение по программе ведется в разновозрастной группе, которая комплектуется из обучающихся 12-15 лет, имеющих начальные умения и навыки работы в графических, текстовых редакторах, редакторах презентаций, с браузером. Рекомендуемое количество обучающихся в группе – 15 человек.

### **2.5. Объем программы**

Программа рассчитана на 102 часа.

### **2.6. Формы обучения и виды занятий по программе**

Обучение предполагает посещение учащимися теоретических занятий и самостоятельном выполнении практических заданий.

### **2.7. Срок реализации программы**

Программа реализуется в течение 9 месяцев.

### **2.8. Режим занятий**

Занятия проходят 3 раза в неделю

### 3. ЦЕЛЬ И ЗАДАЧИ РЕАЛИЗАЦИИ ПРОГРАММЫ

**Цель:** ознакомить обучающихся с возможностями, синтаксисом, технологией языка Python и обучить методами программирования для решения прикладных математических и информационных задач.

**Задачи:**

- сформировать представление об основах программирования в среде Python;
- сформировать навыки грамотной работы в системе программирования Python;
- сформировать навыки грамотной работы в IDE PyCharm;
- ознакомить с базовыми понятиями теории алгоритмов при решении математических задач;
- обучить методам решения задач, реализуемым на языке Python;
- сформировать практические навыки решения прикладных задач;
- сформировать практические навыки разработки Telegram ботов;
- сформировать навыки поиска информации, работы с технической литературой;
- развить навыки самостоятельного и творческого подхода к решению задач с помощью средств современной вычислительной техники;
- развить алгоритмическое и логическое мышление учащихся;
- развить интеллектуальные и практические умения, самостоятельно приобретать и применять на практике полученные знания;
- содействовать воспитанию устойчивого интереса к изучению программирования;
- содействовать воспитанию информационной культуры.

## 4. СОДЕРЖАНИЕ ПРОГРАММЫ

### **Раздел 1. Введение в язык программирования Python.**

Тема 1.1. Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж.

*Теория.* Структура образовательной программы, её цель и задачи, содержание обучения. Основные правила и требования техники безопасности при работе за компьютером. Понятие алгоритма и программы.

*Практика.* Первичная диагностика. Тестирование. Установка, работа в среде программирования Python.

Тема 1.2. Элементы языка. Структура программы. Операции и переменные. Типы данных.

*Теория.* Знакомство с основными типами переменных, синтаксисом языка программирования, основными процедурами ввода исходных данных и вывода результатов

*Практика.* Отработка процедур ввода исходных данных и вывода результатов. Тестирование.

Тема 1.3. Ввод и вывод данных на языке Python. Ввод и отладка программ в среде.

*Практика.* Практикум по созданию элементарных программ вводавывода данных, работа со средой, отладка программ. Анализ возможных синтаксических ошибок.

### **Раздел 2. Линейные алгоритмы и их реализация на языке Python.**

Тема 2.1. Линейные алгоритмы целочисленных данных и их реализация на Python.

*Теория.* Знакомство со структурой линейного алгоритма, правилами записи арифметических выражений. Различные типы данных, допустимые операции над ними и ресурсы оперативной памяти. Выполнение операции присваивания в оперативной памяти (далее – ОП) компьютера.

*Практика.* Создание блок-схемы линейного алгоритма. Запись арифметических выражений на языке программирования.

Тема 2.2. Решение задач по теме «Обработка целочисленных данных».

*Практика.* Практикум по разработке линейных алгоритмов. Ввод и отладка программ, реализующих линейный алгоритм обработки целых чисел. Анализ готовых линейных программ.

Тема 2.3. Линейные алгоритмы вещественных чисел и их реализация на Python.

*Теория.* Различные типы данных, допустимые операциях над вещественными числами и ресурсы оперативной памяти.

*Практика.* Разбор типичных задач с линейной структурой алгоритма.

Тема 2.4. Решение задач по теме «Обработка вещественных чисел».

*Практика.* Практикум по реализации линейных алгоритмов вещественных чисел. Ввод и отладка программ, реализующих линейный алгоритм обработки вещественных чисел.

Тема 2.5. Самостоятельная работа по темам раздела: «Линейные алгоритмы и их реализация в среде Python».

*Практика.* Самостоятельная работа по составлению линейного алгоритма, написанию программы, вводу и отладке программного кода, анализу результатов.

### **Раздел 3. Разветвляющиеся алгоритмы и их реализация на языке Python.**

Тема 3.1. Алгоритм «выбор», графическое изображение, полное и неполное ветвление.

*Теория.* Понятие алгоритма «выбор», графическое изображение.

*Практика.* Составление алгоритма «полное ветвление».

Тема 3.2. Реализация алгоритма «выбор» на Python. Примеры решения задач.

*Теория.* Типовые задачи, использующие алгоритм «выбор», изображение алгоритма в виде блок-схемы.

*Практика.* Ввод и отладка программ в среде Python.

Тема 3.3. Решение задач по теме «Полное и неполное ветвление».

*Практика.* Практикум по реализации алгоритмов «полное ветвление» и «неполное ветвление». Ввод и отладка программ в среде Python.

Тема 3.4. Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления.

*Теория.* Составление и анализ алгоритмов.

*Практика.* Ввод и отладка программ в среде Python.

Тема 3.5. Множественный выбор и его реализация с помощью вложенных ветвлений.

*Теория.* Понятие множественного выбора, изображение на блок-схеме.

*Практика.* Ввод и отладка программ в Python.

Тема 3.6. Решение задач по теме «Сложные условия. Каскадные ветвления».

*Практика.* Практикум по составлению алгоритмов для решения задач со сложными условиями и каскадными ветвлениями. Ввод и отладка программ в среде Python.

Тема 3.7. Самостоятельная работа по темам раздела «Разветвляющиеся алгоритмы и их реализация на языке Python».

*Практика.* Самостоятельная работа по составлению алгоритмов, написанию программного кода, вводу и отладке программ в среде Python. Анализ работы.

#### **Раздел 4. Циклические алгоритмы и их реализация на языке Python.**

Тема 4.1. Циклические алгоритмы. Цикл с предусловием. Инструкции управления циклом.

*Теория.* Понятие цикла с предусловием, графическое изображение.

*Практика.* Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 4.2. Решение задач по теме «Циклические алгоритмы. Цикл с предусловием».

*Практика.* Практикум по составлению блок-схем, написанию программного кода, вводу и отладке программ, реализующих цикл с предусловием в среде Python.

Тема 4.3. Циклические алгоритмы. Цикл с постусловием. Инструкции управления циклом.

*Теория.* Понятие цикла с постусловием, графическое изображение.

*Практика.* Ввод и отладка простейших программ, реализующих цикл с постусловием в среде Python.

Тема 4.4. Решение задач по теме «Циклические алгоритмы. Цикл с постусловием».

*Практика.* Практикум по решению типовых задач с помощью цикла с постусловием. Составление блок-схем, написание программного кода, ввод и отладка программ, реализующих цикл с постусловием в Python.

Тема 4.5. Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием».

*Практика.* Самостоятельная работа по решению типовых задач с помощью цикла с предусловием. Ввод и отладка программ, реализующих цикл с предусловием в среде Python. Анализ составленного алгоритма.

Тема 4.6. Циклические алгоритмы. Цикл с параметром. Инструкции управления циклом.

*Теория.* Понятие цикла с параметром, графическое изображение.

*Практика.* Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

Тема 4.7. Решение задач по теме: «Циклические алгоритмы. Цикл с параметром».

*Практика.* Практикум по разработке алгоритмов цикла с параметром, анализу логических ошибок. Ввод и отладка программ, реализующих цикл с параметром в среде Python.

Тема 4.8. Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком».

*Практика.* Самостоятельная работа по разработке блок-схем и отладке программ, реализующих циклы с параметром в среде Python. Анализ самостоятельной работы.

Тема 4.9. Сложные циклические алгоритмы. Вложенные циклы. Решение задач.

*Теория.* Понятие вложенного цикла, графическое изображение.

*Практика.* Решение задач по разработке и отладке программ, реализующих вложенные циклы в среде Python. Анализ разработанного алгоритма. Оценка эффективности разработанного алгоритма.

Тема 4.10. Самостоятельная работа по темам раздела «Циклические алгоритмы и их реализация на языке Python». Анализ самостоятельной работы.

*Практика.* Самостоятельная работа по разработке блок-схем и отладке программ, реализующих циклические алгоритмы в среде Python.

## **Раздел 5. Этапы решения задач на языке Python.**

Тема 5.1. Последовательное конструирование алгоритма.

*Теория.* Этапы и особенности решения задачи на компьютере. Этап создания алгоритма. Использование принципа последовательного конструирования алгоритма. Будет также рассмотрен принцип его работы.

*Практика.* Разработка алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.

Тема 5.2. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма.

*Практика.* Практикум по последовательному конструированию алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.

## **Раздел 6. Вспомогательные алгоритмы и их реализация на языке Python.**

Тема 6.1. Вспомогательные алгоритмы и подпрограммы. Функции.

*Теория.* Понятие вспомогательного алгоритма. Формат записи вспомогательного алгоритма в виде функции. Типовые задачи.

*Практика.* Ввод и отладка программ с использованием функции в среде Python.

Тема 6.2. Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.

*Практика.* Практикум по программированию решения отдельных подзадач с помощью отдельных функций, которые потом при необходимости вызываются в различных местах программы. Польза функций при решении задач. Важные принципы в программировании – модульность и повторное использование кода. Их польза при разработке сложных программ.

Тема 6.3. Вспомогательные алгоритмы и подпрограммы. Рекурсия *Теория.* Понятие рекурсии, в чём её сложность. Рекурсивные алгоритмы и их особенности: когда нужно использовать рекурсию в программировании, а когда лучше обойтись без этого.

*Практика.* Ввод и отладка программ с использованием рекурсии в среде Python.

Тема 6.4. Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии.

*Практика.* Практикум по решению типовых задач с вводом и отладкой программ с использованием рекурсии в среде Python.

## **Раздел 7. Разработка Telegram ботов.**

### Тема 7.1. Регистрация Telegram бота.

*Теория.* Регистрация бота, путем взаимодействия в Отцом Ботов @BotFather.

### Тема 7.2. Установка библиотек в IDE PyCharm для разработки бота.

*Теория.* Установка и настройка Telegram Bot API.

### Тема 7.3. Разработка Telegram бота ToDo.

*Теория.* Рассмотрение необходимых алгоритмов для написания бота ToDo и их реализация на языке программирования Python.

*Практика.* Написание Telegram бота ToDo.

### Тема 7.4. Разработка Telegram бота обучающимися на свободную тему.

*Практика.* Разработка Telegram бота.

### Тема 7.5. Презентация и защита разработанного Telegram бота.

## **4.3. Планируемые результаты**

По итогам обучения обучающиеся будут **знать**:

- технику безопасности и требования, предъявляемые к организации рабочего места;
- термины: «данные», «кодирование», «алгоритм», «программа», понимание различий между употреблением этих терминов в обыденной речи и в информатике;
- методы представления и алгоритмы обработки данных, программную реализацию алгоритмов;
- основы алгоритмической культуры, нормы информационной этики;
- математические и компьютерные модели, их использование;
- виды программного обеспечения, решаемые с его помощью задачи.

По итогам обучения обучающиеся будут **уметь**:

- составлять несложные программы;
- работать с основными видами программных систем и интернет сервисов;
- определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирая основания и критерии для классификации;
- устанавливать причинно-следственные связи, делать логическое умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- составлять алгоритмы управления исполнителями и записывать их на выбранном алгоритмическом языке;
- использовать логические значения, операции и выражения с ними;
- выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов, простых и табличных величин;
- записывать алгоритмические структуры на языке программирования Python;
- решать простые, сложные и нестандартные задачи.

### 5.Календарный учебный график:

№ п/п	Тема занятия	Место проведения
1.	Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж.	Кабинет 31
2.	Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж.	Кабинет 31
3.	Элементы языка. Структура программы. Операции и переменные. Типы данных.	Кабинет 31
4.	Элементы языка. Структура программы. Операции и переменные. Типы данных.	Кабинет 31
5.	Элементы языка. Структура программы. Операции и переменные. Типы данных.	Кабинет 31
6.	Элементы языка. Структура программы. Операции и переменные. Типы данных.	Кабинет 31
7.	Ввод и вывод данных на языке Python. Ввод и отладка программ.	Кабинет 31
8.	Ввод и вывод данных на языке Python. Ввод и отладка программ.	Кабинет 31
9.	Ввод и вывод данных на языке Python. Ввод и отладка программ.	Кабинет 31
10.	Ввод и вывод данных на языке Python. Ввод и отладка программ.	Кабинет 31
11.	Линейные алгоритмы целочисленных данных и их реализация на Python.	Кабинет 31
12.	Линейные алгоритмы целочисленных данных и их реализация на Python.	Кабинет 31
13.	Решение задач по теме «Обработка целочисленных данных».	Кабинет 31
14.	Решение задач по теме «Обработка целочисленных данных».	Кабинет 31

15.	Линейные алгоритмы вещественных чисел и их реализация на Python.	Кабинет 31
16.	Линейные алгоритмы вещественных чисел и их реализация на Python.	Кабинет 31
17.	Решение задач по теме «Обработка вещественных чисел».	Кабинет 31
18.	Решение задач по теме «Обработка вещественных чисел».	Кабинет 31
19.	Самостоятельная работа по темам раздела «Линейные алгоритмы и их реализация в среде Python». Анализ сам.раб.	Кабинет 31
20.	Самостоятельная работа по темам раздела «Линейные алгоритмы и их реализация в среде Python». Анализ сам.раб.	Кабинет 31
21.	Алгоритм «выбор», графическое изображение, полное и неполное ветвление.	Кабинет 31
22.	Алгоритм «выбор», графическое изображение, полное и неполное ветвление.	Кабинет 31
23.	Реализация алгоритма «выбор» на Python. Примеры решения задач.	Кабинет 31
24.	Реализация алгоритма «выбор» на Python. Примеры решения задач.	Кабинет 31
25.	Решение задач по теме «Полное и неполное ветвление».	Кабинет 31
26.	Решение задач по теме «Полное и неполное ветвление».	Кабинет 31
27.	Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления.	Кабинет 31
28.	Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления.	Кабинет 31
29.	Множественный выбор и его реализация с помощью вложенных ветвлений.	Кабинет 31
30.	Множественный выбор и его реализация с помощью вложенных ветвлений.	Кабинет 31

31.	Решение задач по теме «Сложные условия. Каскадные ветвления».	Кабинет 31
32.	Решение задач по теме «Сложные условия. Каскадные ветвления».	Кабинет 31
33.	Самостоятельная работа по темам раздела: «Разветвляющиеся алгоритмы и их реализация на языке Python». Анализ сам.раб.	Кабинет 31
34.	Самостоятельная работа по темам раздела: «Разветвляющиеся алгоритмы и их реализация на языке Python». Анализ сам.раб.	Кабинет 31
35.	Циклические алгоритмы. Цикл с предусловием. Инструкции управления циклом.	Кабинет 31
36.	Циклические алгоритмы. Цикл с предусловием. Инструкции управления циклом.	Кабинет 31
37.	Решение задач по теме «Циклические алгоритмы. Цикл с предусловием».	Кабинет 31
38.	Решение задач по теме «Циклические алгоритмы. Цикл с предусловием».	Кабинет 31
39.	Решение задач по теме «Циклические алгоритмы. Цикл с предусловием».	Кабинет 31
40.	Циклические алгоритмы. Цикл с постусловием. Инструкции.	Кабинет 31
41.	Циклические алгоритмы. Цикл с постусловием. Инструкции.	Кабинет 31
42.	Решение задач по теме «Циклические алгоритмы. Цикл с постусловием».	Кабинет 31
43.	Решение задач по теме «Циклические алгоритмы. Цикл с постусловием».	Кабинет 31
44.	Решение задач по теме «Циклические алгоритмы. Цикл с постусловием».	Кабинет 31
45.	Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием». Анализ сам.раб.	Кабинет 31
46.	Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием». Анализ сам.раб.	Кабинет 31

47.	Циклические алгоритмы. Цикл с параметром. Инструкции управления циклом.	Кабинет 31
48.	Циклические алгоритмы. Цикл с параметром. Инструкции управления циклом.	Кабинет 31
49.	Решение задач по теме «Циклические алгоритмы. Цикл с параметром».	Кабинет 31
50.	Решение задач по теме «Циклические алгоритмы. Цикл с параметром».	Кабинет 31
51.	Решение задач по теме «Циклические алгоритмы. Цикл с параметром».	Кабинет 31
52.	Решение задач по теме «Циклические алгоритмы. Цикл с параметром».	Кабинет 31
53.	Решение задач по теме «Циклические алгоритмы. Цикл с параметром».	Кабинет 31
54.	Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком». Анализ сам.раб.	Кабинет 31
55.	Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком». Анализ сам.раб.	Кабинет 31
56.	Сложные циклические алгоритмы. Вложенные циклы.	Кабинет 31
57.	Сложные циклические алгоритмы. Вложенные циклы.	Кабинет 31
58.	Сложные циклические алгоритмы. Вложенные циклы.	Кабинет 31
59.	Самостоятельная работа по разделу: «Циклические алгоритмы и их реализация на Python». Анализ сам.раб.	Кабинет 31
60.	Самостоятельная работа по разделу: «Циклические алгоритмы и их реализация на Python». Анализ сам.раб.	Кабинет 31
61.	Последовательное конструирование алгоритма.	Кабинет 31
62.	Последовательное конструирование алгоритма.	Кабинет 31

63.	Последовательное конструирование алгоритма.	Кабинет 31
64.	Этапы решения задачи на компьютере. Последовательное конструирование алгоритма.	Кабинет 31
65.	Этапы решения задачи на компьютере. Последовательное конструирование алгоритма.	Кабинет 31
66.	Этапы решения задачи на компьютере. Последовательное конструирование алгоритма.	Кабинет 31
67.	Вспомогательные алгоритмы и подпрограммы. Функции.	Кабинет 31
68.	Вспомогательные алгоритмы и подпрограммы. Функции.	Кабинет 31
69.	Вспомогательные алгоритмы и подпрограммы. Функции.	Кабинет 31
70.	Вспомогательные алгоритмы и подпрограммы. Функции.	Кабинет 31
71.	Вспомогательные алгоритмы и подпрограммы. Функции.	Кабинет 31
72.	Вспомогательные алгоритмы и подпрограммы. Функции.	Кабинет 31
73.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.	Кабинет 31
74.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.	Кабинет 31
75.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.	Кабинет 31
76.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.	Кабинет 31
77.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.	Кабинет 31
78.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.	Кабинет 31
79.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.	Кабинет 31

80.	Вспомогательные алгоритмы и подпрограммы. Рекурсия.	Кабинет 31
81.	Вспомогательные алгоритмы и подпрограммы. Рекурсия.	Кабинет 31
82.	Вспомогательные алгоритмы и подпрограммы. Рекурсия.	Кабинет 31
83.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии.	Кабинет 31
84.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии.	Кабинет 31
85.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии.	Кабинет 31
86.	Регистрация Telegram бота	Кабинет 31
87.	Установка библиотек в IDE PyCharm для разработки Telegram ботов	Кабинет 31
88.	Установка библиотек в IDE PyCharm для разработки Telegram ботов	Кабинет 31
89.	Разработка Telegram бота ToDo	Кабинет 31
90.	Разработка Telegram бота ToDo	Кабинет 31
91.	Разработка Telegram бота ToDo	Кабинет 31
92.	Разработка Telegram бота ToDo	Кабинет 31
93.	Разработка Telegram бота ToDo	Кабинет 31
94.	Разработка Telegram бота ToDo	Кабинет 31
95.	Разработка Telegram бота обучающимися на свободную тему	Кабинет 31
96.	Разработка Telegram бота обучающимися на свободную тему	Кабинет 31

97.	Разработка Telegram бота обучающимися на свободную тему	Кабинет 31
98.	Разработка Telegram бота обучающимися на свободную тему	Кабинет 31
99.	Разработка Telegram бота обучающимися на свободную тему	Кабинет 31
100.	Разработка Telegram бота обучающимися на свободную тему	Кабинет 31
101.	Презентация и защита разработанного Telegram бота	Кабинет 31
102.	Презентация и защита разработанного Telegram бота	Кабинет 31

## 6. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

### 6.1. Материально-техническое обеспечение

**Кабинет информатики**, соответствующий санитарно-гигиеническим нормам освещения и температурного режима (18-21 градус Цельсия; влажность воздуха должна быть в пределах 40-60 %), противопожарным требованиям, оснащенный мебелью, соответствующей возрастным особенностям детей возраста 12-15 лет (ученические одноместные парты с компьютерным стулом, подставкой для ног), 15 ноутбуков, стол учительский с тумбами; шкафы для хранения учебников, дидактических материалов, пособий, телевизор.

**Технические ресурсы:** 15 ноутбуков ученических, 1 ноутбук учительский, телевизор.

### 6.2. Информационное обеспечение

Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с., ил.

Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 857 с., ил.

### 6.3. Кадровое обеспечение

В реализации программы занят один педагог. Он имеет высшее педагогическое образование, 1 квалификационную категорию. Стаж работы 5 лет. Образование соответствует направленности программы.

### 6.4. Формы контроля

В процессе реализации программы используются следующие формы контроля:

- Беседа
- Текущий контроль
- Практикум
- Самостоятельная работа
- Промежуточный контроль

### 6.5. Оценочные материалы

#### Самостоятельная работа №1

Какая команда используется для вывода (печати) данных?	Выберите один вариант из списка  printf()  cout()  print()  Console.WriteLine()
Выберите верные строки кода	print('I'm 16 and I'm from Northern Ireland.')  print("I'm a math teacher and a programmer!")

	<pre>print('Просто текст...')  print('Поэма "Мёртвые души" одна из самых интересных')  print("3.1415")  print()</pre>
Выберите верную строку кода	<pre>print('Python', 'is the best', '!!!')  print("10", '100', '1000')  print("раз", "два", "три")  print("Python", , "is the best")</pre>
<p>Что выведет следующий код <code>print('1', '2', '3', '4', sep='*')</code>?</p> <p>Выберите один вариант из списка</p>	<pre>1234  1 2 3 4  1*2*3*4  24</pre>
Выберите верные строки кода	<pre>print("The world's a little blurry", "Or maybe it's my eyes", end='!!!', sep=' :) ')  print("Honey, what's your hurry", end='?')  print("Remember not to get too close to stars", "They're never gonna give you love like ours", sepp=" ")  print("Told you not to worry", "But maybe that's a lie", sep=' :) ')  print("Remember not to get too close to stars", end="")</pre>
<p>Какая команда используется для считывания данных с клавиатуры?</p> <p>Выберите один вариант из списка</p>	<pre>scanf()  cin  Console.ReadLine()  input()</pre>
Какая из указанных строк считывает целое число в переменную n?	<p>Выберите один вариант из списка</p> <pre>n = input()  n = str(input())  n = int(input())</pre>

	<pre>n = integer(input()) n = number(input())</pre>
<p>Выберите верные утверждения</p>	<p>Имя переменной может начинаться с символа подчёркивания (<u>_</u>)</p> <p>Имя переменной не может оканчиваться цифрой</p> <p>Имя переменной не может начинаться с цифры</p> <p>Имя переменной не может совпадать с ключевым (зарезервированным) словом</p>
<p>Какое число выведет следующий код?</p> <pre>s = 13 k = -5 d = s + 2 s = d k = 2 * s print(s + k + d)</pre>	
<p>Какое число выведет следующий код?</p> <pre>a = 17 // (23 % 7) b = 34 % a * 5 - 29 % 4 * 3 print(a * b)</pre>	
<p>Звездный прямоугольник  Напишите программу, которая выводит прямоугольник, по периметру состоящий из звездочек (*).</p> <p>Примечание. Высота и ширина прямоугольника равны 4 и 17 звёздочкам соответственно.</p> <pre>***** *       * *       * *****</pre>	
<p>Напишите программу, которая считывает два целых числа a и b и выводит на экран квадрат суммы и сумму квадратов этих чисел</p>	

Номер теста	Входные данные	Выходные данные
1	3 2	Квадрат суммы 3 и 2 равен 25 Сумма квадратов 3 и 2 равна 13
2	-5 1	Квадрат суммы -5 и 1 равен 16 Сумма квадратов -5 и 1 равна 26
3	17 0	Квадрат суммы 17 и 0 равен 289 Сумма квадратов 17 и 0 равна 289
4	0 25	Квадрат суммы 0 и 25 равен 625 Сумма квадратов 0 и 25 равна 625
5	10 10	Квадрат суммы 10 и 10 равен 400 Сумма квадратов 10 и 10 равна 200

Как известно, целые числа в языке Python не имеют ограничений, которые встречаются в других языках программирования. Напишите программу, которая считывает четыре целых положительных числа  $a$ ,  $b$ ,  $c$  и  $d$  и выводит на экран значение выражения

$$a^b + c^d.$$

Номер теста	Входные данные	Выходные данные
1	9 29 7 27	4710194409608608369201743232
2	2 3 2 5	40
3	1 1 1 1	2
4	7 8 9 10	3492549202

Напишите программу, которая считывает целое положительное число  $n$ ,  $n \in [1;9]$  и выводит значение числа  $n + nn + nnn$

	Номер теста	Входные данные	Выходные данные
	1	1	123
	2	2	246
	3	3	369
	4	9	1107
	5	5	615

## Самостоятельная работа №2

### Начало столетия

Напишите программу, которая определяет, оканчивается ли год с данным номером на два нуля. Если год оканчивается, то выведите «YES», иначе выведите «NO».

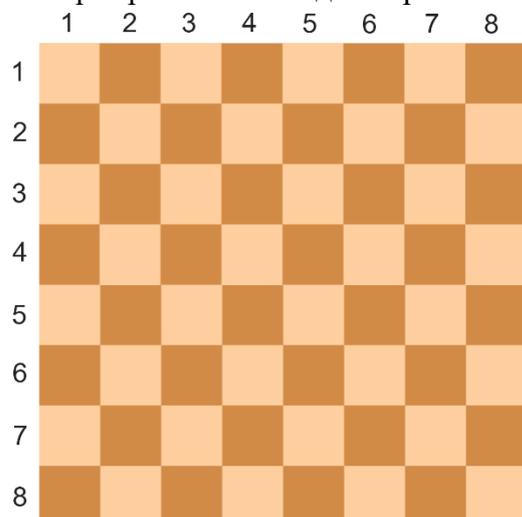
Тестовые данные □

Номер теста	Входные данные	Выходные данные
1	2000	YES
2	1999	NO
3	3000	YES
4	2100	YES
5	1900	YES

### Шахматная доска

Заданы две клетки шахматной доски. Напишите программу, которая определяет имеют ли указанные клетки один цвет или нет. Если они покрашены в один цвет, то выведите слово «YES», а если в разные цвета — то «NO».

На вход программе подаётся четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки.



Тестовые данные

Номер теста	Входные данные	Выходные данные
1	1 1 2 6	YES
2	2 2 2 5	NO
3	2 2 2 4	YES
4	2 3 3 2	YES
5	2 3 7 8	YES

### Girls only

Футбольная команда набирает девочек от 10 до 15 лет включительно. Напишите программу, которая запрашивает возраст и пол претендента, используя обозначение пола буквы m (от male – мужчина) и f (от female – женщина) и определяет подходит ли претендент для вступления в команду или нет. Если претендент подходит, то выведите «YES», иначе выведите «NO».

Тестовые данные

Номер теста	Входные данные	Выходные данные
1	10 f	YES
2	11 m	NO
3	11 f	YES

Номер теста	Входные данные	Выходные данные
4	12 f	YES
5	13 f	YES

### Римские цифры

Напишите программу, которая считывает целое число и выводит соответствующую ему римскую цифру. Если число находится вне диапазона 1-10, то программа должна вывести текст «ошибка».

В таблице приведены римские цифры для чисел от 1 до 10.

Число	Римская цифра
1	I
2	II
3	III
4	IV
5	V
6	VI
7	VII
8	VIII
9	IX
10	X

Тестовые данные

Номер теста	Входные данные	Выходные данные
1	7	VII
2	12	ошибка
3	1	I
4	2	II

Номер теста	Входные данные	Выходные данные
5	3	III

### YES or NO вот в чем вопрос

Напишите программу, которая принимает на вход число и в зависимости от условий выводит текст «YES», либо «NO».

Условия:

- если число нечётное, то вывести «YES»;
- если число чётное в диапазоне от 2 до 5 (включительно), то вывести «NO»;
- если число чётное в диапазоне от 6 до 20 (включительно), то вывести «YES»;
- если число чётное и больше 20, то вывести «NO».

Тестовые данные □

Номер теста	Входные данные	Выходные данные
1	1	YES
2	2	NO
3	3	YES
4	101	YES
5	509	YES

### Ход слона 🐘

Даны две различные клетки шахматной доски. Напишите программу, которая определяет, может ли слон попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести «YES», если из первой клетки ходом слона можно попасть во вторую или «NO» в противном случае.

**Примечание.** Шахматный слон ходит по диагоналям.



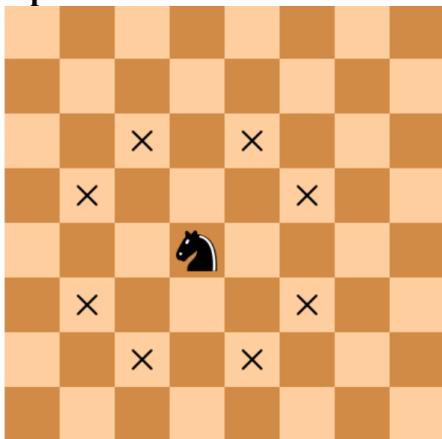
Тестовые данные □

Номер теста	Входные данные	Выходные данные
1	4 4 5 5	YES
2	4 4 5 4	NO
3	4 4 5 3	YES
4	4 4 4 5	NO
5	4 4 3 5	YES

### Ход коня

Даны две различные клетки шахматной доски. Напишите программу, которая определяет, может ли конь попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести «YES», если из первой клетки ходом коня можно попасть во вторую или «NO» в противном случае.

**Примечание.** Шахматный конь ходит буквой «Г».



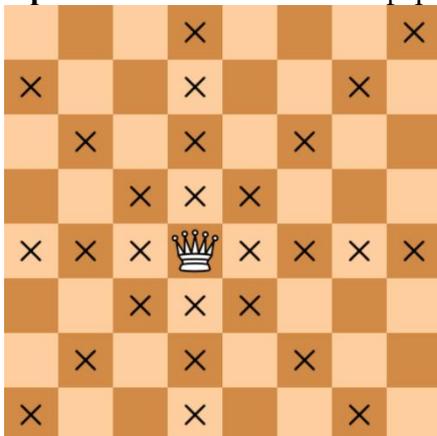
Тестовые данные □

Номер теста	Входные данные	Выходные данные
1	1 1 8 8	NO
2	2 4 3 2	YES
3	5 2 4 4	YES
4	5 5 3 7	NO
5	4 7 6 6	YES

### Ход ферзя

Даны две различные клетки шахматной доски. Напишите программу, которая определяет, может ли ферзь попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести «YES», если из первой клетки ходом ферзя можно попасть во вторую или «NO» в противном случае.

**Примечание.** Шахматный ферзь ходит по диагонали, горизонтали или вертикали.



Тестовые данные □

Номер теста	Входные данные	Выходные данные
1	1 1 2 2	YES
2	1 1 2 3	NO
3	5 6 3 3	NO
4	3 3 1 1	YES
5	6 5 2 5	YES

### Самостоятельная работа №3

Ревью кода - А

На обработку поступает последовательность из 4 целых чисел. Известно, что вводимые числа по абсолютной величине не превышают  $10^8$ .

Нужно написать программу, которая выводит на экран количество нечётных чисел в исходной последовательности и максимальное нечётное число.

Если нечётных чисел нет, требуется на экран вывести «NO». Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

Примечание. Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

КОД С ОШИБКОЙ

```
n = 4
```

```
count = 0
```

```
maximum = 999
```

```
for i in range(1, n + 1):
```

```
    x = int(input())
```

```
    if x % 2 != 0:
```

```
        count += 1
```

```
        if x > maximum:
```

```
            maximum = i
```

```

        break
    if count > 0:
        print(count)
        print(maximum)
    else:
        print('NO')

n = 4
count = 0
maximum = -1000000000
for i in range(1, n + 1):
    x = int(input())
    if x % 2 != 0:
        count += 1
        if x > maximum:
            maximum = x
if count > 0:
    print(count)
    print(maximum)
else:
    print('NO')

```

#### Ревью кода - В

На обработку поступает натуральное число. Нужно написать программу, которая выводит на экран сумму чётных цифр этого числа или 0, если чётных цифр в записи нет. Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

Примечание. Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

#### КОД С ОШИБКАМИ

```

n = input()
s = 0
while n > 10:
    if n % 2 == 1:
        s = n % 10
        n //= 10
print(s)

n = int(input())
s = 0
k = 0
while n > 0:
    last = n % 10
    if last % 2 == 0:
        s += last
        k += 1
    n = n // 10
if k == 0:
    print('0')

```

```
else:  
    print(s)
```

### Ревью кода - С

На обработку поступает последовательность из 8 целых чисел. Известно, что вводимые числа по абсолютной величине не превышают  $10^{12}$ .

Нужно написать программу, которая выводит на экран количество делящихся нацело на 4 чисел в исходной последовательности и максимальное делящееся нацело на 4 число.

Если делящихся нацело на 4 чисел нет, требуется на экран вывести «NO». Программист торопился и написал программу неправильно.

Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк.

Примечание. Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

#### КОД С ОШИБКОЙ

```
n = 7  
count = 0  
maximum = 1000  
for i in range(1, n + 1):  
    x = int(input())  
    if x // 4 == 0:  
        count += 1  
        if x < maximum:  
            maximum = x  
if count > 0:  
    print(count)  
    print(maximum)  
else:  
    print('NO')
```

```
n = 8  
count = 0  
maximum = -10000000000000  
for i in range(1, n + 1):  
    x = int(input())  
    if x % 4 == 0:  
        count += 1  
        if x > maximum:  
            maximum = x  
if count > 0:  
    print(count)  
    print(maximum)  
else:  
    print('NO')
```

### Звездная рамка

На вход программе подается натуральное число  $n$ . Напишите программу, которая печатает звездную рамку размерами  $n \times 19$ .

Формат ВХОДНЫХ данных  
 На вход программе подаётся натуральное число  $n \in [3; 19]$  — высота звездной рамки.  
 Формат ВЫХОДНЫХ данных  
 Программа должна вывести звездную рамку размерами  $n \times 19$ .  
 Подсказка. Для печати звездной линии используйте умножение строки на число.  
 Тестовые данные

Номер теста	Входные данные	Выходные данные
1	3	<pre> ***** *                                     * ***** </pre>
2	4	<pre> ***** *                                     * *                                     * ***** </pre>

```

n = int(input())
for i in range (1, n + 1):
    if i == 1 or i == n:
        print(19*'*')
    else:
        print('*', '*', sep = 17*' ')

```

Третья цифра  
 Дано натуральное число  $n(n > 99)$ . Напишите программу, которая определяет его третью (с начала) цифру.  
 Тестовые данные

Номер теста	Входные данные	Выходные данные
1	100000000	0
2	3459087654	5
3	134	4
4	6765409	6
5	289	9

```

n = int(input())
k = 0
while n > 99:
    digit = n % 10
    k += 1
    n = n // 10
print(digit)

```

Все вместе Z

Дано натуральное число. Напишите программу, которая вычисляет:

- количество цифр «3» в нем;
- сколько раз в нем встречается последняя цифра;
- количество четных цифр;
- сумму его цифр, больших пяти;
- произведение цифр, больших семи (если цифр больших семи нет, то вывести 1, если такая цифра одна, то вывести ее);
- сколько раз в нем встречается цифры 0 и 5 (всего суммарно).

Тестовые данные □

Номер теста	Входные данные	Выходные данные
1	56639	1 1 2 21 9 1
2	56689932106	1 3 6 44 648 2
3	255	0 2 1 0 1 2

```
n = int(input())
colvo_cifr_3 = 0
colvo_last_digit = 0
colvo_chetnih = 0
summa_bolshe_5 = 0
proizv_bolshe_7 = 1
colvo_0_or_5 = 0
last_digit = n % 10
while n > 0:
    digit = n % 10
    if digit == 3:
        colvo_cifr_3 += 1
    if digit == last_digit:
        colvo_last_digit += 1
    if digit % 2 == 0:
        colvo_chetnih += 1
```

```
if digit > 5:
    summa_bolshe_5 += digit
if digit > 7:
    proizv_bolshe_7 *= digit
if digit == 0:
    colvo_0_or_5 += 1
if digit == 5:
    colvo_0_or_5 += 1
n = n // 10
print(colvo_cifr_3)
print(colvo_last_digit)
print(colvo_chetnih)
print(summa_bolshe_5)
print(proizv_bolshe_7)
print(colvo_0_or_5)
```

#### **Самостоятельная работа №4**

##### **№1**

Вставьте пропущенный фрагмент кода, так чтобы в результате выполнения программы была выведена длина строки s.

```
s = 'Python rocks!'
print()
```

```
s = 'Python rocks!'
print(len(s))
```

##### **№2**

Вставьте пропущенный фрагмент кода, так чтобы в результате выполнения программы был выведен четвертый символ строки s.

```
s = 'Python rocks!'
print()
```

```
s = 'Python rocks!'
print(s[3])
```

##### **№3**

Вставьте пропущенный фрагмент кода, так чтобы в результате выполнения программы были выведены символы строки s со 2 по 5 включительно.

```
s = 'Python rocks!'
print()
```

```
s = 'Python rocks!'
print(s[1:5])
```

##### **№4**

Вставьте пропущенный фрагмент кода, так чтобы в результате выполнения программы была выведена строка s без ведущих и замыкающих пробельных символов.

```
s = ' Python rocks! '
print()
```

```
s = ' Python rocks! '
print(s.strip())
```

### №5

Вставьте пропущенный фрагмент кода, так чтобы в результате выполнения программы была выведена строка *s* заглавными буквами (в верхнем регистре).

```
s = 'Python rocks!'
print()
```

```
s = 'Python rocks!'
print(s.upper())
```

### №6

Вставьте пропущенный фрагмент кода, так чтобы в результате выполнения программы была выведена строка *s* в которой символ «o» заменен на символ «@».

```
s = 'Python rocks!'
print()
```

```
s = 'Python rocks!'
print(s.replace('o','@'))
```

### №7

На вход программе подается строка текста. Напишите программу, которая удаляет из нее все символы с индексами кратными 3, то есть символы с индексами 0, 3, 6, ....

Тестовые данные

Номер теста	Входные данные	Выходные данные
1	Python	yton
2	Hello	elo
3	qwer	we
4	p	

```
s = input()
ans = ""
for i in range (1, len(s)):
    if i % 3 != 0:
        ans = ans + s[i]
print(ans)
```

**№8**

На вход программе подается строка текста. Напишите программу, которая заменяет все вхождения цифры 1 на слово «one».

Тестовые данные

Номер теста	Входные данные	Выходные данные
1	1231	one23one
2	ab23456one	ab23456one

```
s = input()
print(s.replace('1','one'))
```

**№9**

На вход программе подается строка текста. Напишите программу, которая удаляет все вхождения символа «@».

Тестовые данные

Номер теста	Входные данные	Выходные данные
1	123@1@@34	123134
2	@@	
3	@	
4	@a@	a

```
s = input()
print(s.replace('@',''))
```

**№10**

На вход программе подается строка текста. Напишите программу, которая выводит индекс второго вхождения буквы «f». Если буква «f» встречается только один раз, выведите число -1, а если не встречается ни разу, выведите число -2.

Тестовые данные

Номер теста	Входные данные	Выходные данные
1	affective	2
2	python	-2
3	father	-1
4	ooooooooooooooooof	-1
5	ff	1
6	ffffffffffffffff	1

Номер теста	Входные данные	Выходные данные
7	fooooooooooooooof	15

```
s = input()
k = 0
for i in range(0, len(s)):
    if s[i] == 'f':
        k += 1
    if k == 2:
        print(i)
        break
if k == 1:
    print('-1')
if k == 0:
    print('-2')
```

### №11

На вход программе подается строка текста в которой буква «h» встречается как минимум два раза. Напишите программу, которая возвращает исходную строку и переворачивает последовательность символов, заключенную между первым и последним вхождением буквы «h».

Тестовые данные

Номер теста	Входные данные	Выходные данные
1	abch12345h	abch54321h
2	qwertyhasdfghzxcvb	qwertyhgfdсахzxcvb
3	In the hole in the ground there lived a hobbit	In th a devil ereht dnuorg eht ni eloh ehobbit
4	asdfghhxcvb	asdfghhxcvb
5	ahbhchdheha	ah ehdhchbha

```
s = input()
l = s.find('h')
r = s.rfind('h')
ans1 = s[0:s.find('h')+1]
ans2 = s[s.rfind('h'):]
ans3 = ""
for i in range(r-1,l,-1):
    ans3 += s[i]
print(ans1+ans3+ans2)
```

## 6.6. Методические материалы

### 6.6.1. Формы организации занятий

**Формы занятий:** комбинированные и интегрированные занятия, теоретические лекции, практические занятия.

### 6.6.2. Методы обучения

Формы и методы при проведении учебных занятий подбираются с учетом цели и задач, познавательных интересов и индивидуальных возможностей воспитанников, специфики содержания данной образовательной программы и возраста воспитанников:

- словесные методы (рассказ, объяснение, беседа, дискуссии), опрос (устный, письменный);
- наглядные методы (иллюстрация, демонстрация презентаций, схем, таблиц и т.д.);
- практические методы (задания, упражнения, тесты);
- исследовательский метод (предполагает самостоятельный поиск и пути решения поставленных задач);
- эвристический метод («нахожу», «открываю»).

## 7.Список литературы

### Дополнительная литература для учителя

1. *Лутц М.* Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
2. *Златопольский Д.М.* Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
3. *Лутц М.* Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
4. *Лутц М.* Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. *Гэддис Т.* Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
6. *Лучано Рамальо* Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.
7. *Свейгарт, Эл.* Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.
8. *Рейтц К., Шлюссер Т.* Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O'Reilly»).
9. *Любанович Билл* Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестселлеры O'Reilly»).
10. *Федоров, Д. Ю.* Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/437489> (дата обращения: 13.02.2020).
11. *Шелудько, В. М.* Основы программирования на языке высокого уровня Python: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 с. – ISBN 978-5-9275-2649-9. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www.iprbookshop.ru/87461.html> (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
12. *Шелудько, В. М.* Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017.

- 107 с. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www.iprbookshop.ru/87530.html> (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
13. Доусон М. Программируем на Python. – СПб.: Питер, 2014. – 416 с.
  14. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. – СПб.: БХВ-Петербург, 2012. – 704 с.
  15. Пилгрим Марк. Погружение в Python 3 (Dive into Python 3 на русском)
  16. Прохоренок Н.А. Самое необходимое. — СПб.: БХВ-Петербург, 2011. — 416 с.

#### **Дополнительная литература для обучающихся**

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
2. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
3. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
4. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
5. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
6. Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.
7. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.
8. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O'Reilly»).
9. Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестселлеры O'Reilly»).
10. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс). – ISBN 978-5-534-10971-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/437489> (дата обращения: 13.02.2020).
11. Шелудько, В. М. Основы программирования на языке высокого уровня Python: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 146 с. – ISBN 978-5-9275-2649-9. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www.iprbookshop.ru/87461.html> (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
12. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / В. М. Шелудько. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 с. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <http://www.iprbookshop.ru/87530.html> (дата обращения: 13.02.2020). – Режим доступа: для авторизир. пользователей
13. Доусон М. Программируем на Python. – СПб.: Питер, 2014. – 416 с.
14. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. – СПб.: БХВ-Петербург, 2012. – 704 с.
15. Пилгрим Марк. Погружение в Python 3 (Dive into Python 3 на русском)
16. Прохоренок Н.А. Самое необходимое. — СПб.: БХВ-Петербург, 2011. — 416 с.