МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РЕСПУБЛИКИ БАШКОРТОСТАН

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ

ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ

ЦЕНТР ВЫЯВЛЕНИЯ, ПОДДЕРЖКИ И РАЗВИТИЯ

СПОСОБНОСТЕЙ И ТАЛАНТОВ У ДЕТЕЙ И МОЛОДЁЖИ

РЕСПУБЛИКИ БАШКОРТОСТАН «АВРОРА»

|  |  |
| --- | --- |
| «РАССМОТРЕНО»На заседании экспертного советаГАОУ ДО ЦРТ «Аврора» пр. №\_\_\_\_\_\_от\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | «УТВЕРЖДАЮ» Директор ГАОУ ДО ЦРТ «Аврора»\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.М.Сайгафаровприказ №\_\_\_\_\_\_\_\_от\_\_\_\_\_\_\_\_\_\_  |

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ**

**ПРОГРАММА**

 «Информатика»

(онлайн видеокурсы)

Место реализации: ГАОУ ДО ЦРТ «Аврора»

Срок реализации: 15 день/ 15 час

Возраст обучающихся: 14 – 16 лет

 Автор программы

Полупанов Д.В. – канд.

тех. наук, доцент,

преподаватель ГАУОДО Центра

развития талантов «АВРОРА»

Уфа – 2020

Содержание

1.[Пояснительная записка 3](#_Toc51669595)

[2. Учебно-тематический план видеокурса 4](#_Toc51669596)

[3. Содержание программы 5](#_Toc51669597)

[4. Методическое обеспечение 8](#_Toc51669598)

[5. Диагностика результативности 9](#_Toc51669599)

[Список использованной литературы 31](#_Toc51669600)

# Пояснительная записка

Образовательная программа дополнительного образования по информатике относится к программам социально-педагогической направленности. Она даёт возможность в пределах процесса обучения информатике способствовать адаптации учащихся в современном обществе, расширению кругозора, пополнения знаний в сфере личных интересов. В связи с этим составление образовательной программы онлайн в ГАОУДО «Центр Развития Талантов «Аврора» по предмету «информатика» является достаточно актуальным.

Онлайн видеокурс по предмету «физика» представляет собой серию видеоуроков длительностью не более 20 минут каждый. После каждого урока даются вопросы на составление и анализ программ для уточнения понимания содержания видеоурока. Всего на курсе 15 онлайн видеоуроков, охватывающих различные аспекты программирования на языке программирования высокого уровня Python 3.

**Цель программы:**

Создание условий для профориентации и развития общего кругозора учащихся. Цели видеокурса по **информатике** следующие:

* формирование у обучающихся информационной культуры;
* формирование умения логически мыслить;
* усвоение обучающимися знаний о методах программирования на языке высокого уровня Python 3, получение навыков разработки эффективных алгоритмов решения прикладных задач;
* развитие познавательных интересов и способностей обучающихся.

Эти цели достигаются благодаря решению следующих задач:

* знакомство обучающихся с основными алгоритмическими структурами;
* формирование у обучающихся умений программировать на языке высокого уровня Python 3;
* овладение учащимися умениями использовать дополнительные источники информации, в частности, всемирной сети Интернет.

 Объём программы: 15 видеоуроков

Планируемые результаты обучения:

В результате освоения программы слушатель получает первоначальные сведения о языке программирования высокого уровня Python 3.

В результате освоения программы слушатель должен

ЗНАТЬ:

* основные алгоритмические структуры;
* основные типы данных и операторы языка программирования Python 3.

УМЕТЬ:

* использовать базовые и составные типы данных Python3 и методы их обработки;
* правильно оформлять код программы согласно протоколу PEP8;
* находить приёмы, обеспечивающие оптимальность разработки программы.

ВЛАДЕТЬ:

* основными методами и приёмами программирования на языке Python 3;
* навыками применения основных модулей и библиотек Python 3.

# 2. Учебно-тематический план видеокурса

|  |  |  |  |
| --- | --- | --- | --- |
| № | Раздел модуля | Темы на уроке | теория |
| 1. | **Знакомство со средой.**  | Знакомство со средой. Основы переменных и ввода-вывода, первые программы |  |
| 2. | **Типы данных и операторы.**  | Появляются числа. Различные типы данных. Арифметические операции. Операции сравнения |  |
| 3. | **Условный оператор.**  | Условный оператор. Блоки кода. Сложные условия. Операции над строками |  |
| 4. | **Знакомство с циклом while.**  | Цикл while –цикл с условием. Поиск максимума и минимума |  |
| 5. | **Знакомство с циклом for.**  | Цикл for – цикл с параметром. Логический (булев) тип данных. Операторы break и continue. Вложенные циклы |  |
| 6. | **Множества.**  | Коллекции в Python. Объекты типа set. Операции над множествами. Сравнение множеств |  |
| 7. | **Обработка строк.**  | Строка как коллекция. Индексация в строках. Хранение текстов в памяти компьютера. Срезы строки |  |
| 8. | **Обработка списков.**  | Создание. Индексация. Перебор элементов. Срезы. Список и массив |  |
| 9. | **Кортежи. Преобразование коллекций.**  | Кортеж – ещё одна коллекция. Присваивание кортежей. Методы сортировки. Преобразования между коллекциями |  |
| 10. | **Методы split и join. Списочные выражения.**  | Популярные методы строк. Правила генерации списков. Считывание значений, введённых одной строкой |  |
| 11. | **Методы списков и строк.**  | Основные методы списков и строк. Цепочки вызовов. Двумерные вложенные списки. Обработка матриц |  |
| 12. | **Словари.**  | Создание словаря, работа с элементами, ключи и значения, методы работы со словарями |  |
| 13. | Функции в Python. | Функции. Локальные и глобальные переменные. Рекурсии |  |
| 14. | Модули и библиотеки Python | Модули math и random. Создание собственных модулей |  |
| 15. | Введение в объектно-ориентированное программирование в Python | Парадигма объектно-ориентированного программирования. Классы и объекты. Полиморфизм, инкапсуляция, наследование |  |

# 3. Содержание программы

**Тема 1. Знакомство со средой.** Знакомство со средой. Основы переменных и ввода-вывода, первые программы. 2 часа

**Тема 2. Типы данных и операторы**. Появляются числа. Различные типы данных. Арифметические операции. Операции сравнения. 2 часа

**Тема 3. Условный оператор**. Условный оператор. Блоки кода. Сложные условия. Операции над строками. 2 часа

**Тема 4. Знакомство с циклом while.** Цикл while. Поиск максимума и минимума. 2 часа

**Тема 5. Знакомство с циклом for**. Цикл for – цикл с параметром. Логический (булев) тип данных. Операторы break и continue. 2 часа

**Тема 6. Множества.** Коллекции в Python. Объекты типа set. Операции над множествами. Сравнение множеств. 2 часа

**Тема 7. Обработка строк.** Строка как коллекция. Индексация в строках. Хранение текстов в памяти компьютера. Срезы строки. 2 часа

**Тема 8. Обработка списков.** Создание. Индексация. Перебор элементов. Срезы. Список и массив. 2 часа

**Тема 9. Кортежи. Преобразование коллекций.** Кортеж – ещё одна коллекция. Присваивание кортежей. Методы сортировки. Преобразования между коллекциями. 2 часа

**Тема 10. Методы split и join. Списочные выражения.** Популярные методы строк. Правила генерации списков. Считывание значений, введённых одной строкой. 2 часа

**Тема 11. Методы списков и строк.** Основные методы списков и строк. Цепочки вызовов.Двумерные вложенные списки. Обработка матриц

**Тема 12. Словари.** Создание словаря, работа с элементами, ключи и значения, методы работы со словарями. 2 часа

Тема 13. Функции в Python. Функции. Локальные и глобальные переменные. Рекурсии. 2 часа

Тема 14. Модули и библиотеки Python. Модули math и random. Создание собственных модулей. 2 часа

Тема 15. Введение в объектно-ориентированное программирование в Python. Парадигма объектно-ориентированного программирования. Классы и объекты. Полиморфизм, инкапсуляция, наследование. 2 часа

#  4. Методическое обеспечение

 Выбор методов и форм для реализации настоящей программы обучения определяется:

 - Наличием специальной методической литературы по информатике, экспериментальным задачам по физике, педагогике и психологии.

 - Возможностью повышения профессионального мастерства: участие в методических объединениях, семинарах, конкурсах; прохождение курсов повышения квалификации.

 - Разработка собственных методических пособий (сборник задач, упражнений), дидактического и раздаточного материала.

 Методическое обеспечение по «информатике» включает в себя дидактические принципы, методы, техническое оснащение, организационные формы работы, формы подведения итогов.

 Дидактические принципы:

 Принцип наглядности, поскольку психофизическое развитие обучающихся 15 – 16 лет характеризуется конкретно-образным мышлением. Следовательно обучающиеся способны полностью усвоить материал при осуществлении практической деятельности с применением практических упражнений, учебно-наглядных пособий. Большое внимание также уделяется принципам доступности и посильности в обучение, связи теории с практикой, прочности овладения знаниями и умениями. Каждое занятие включает в себя элементы теории, практики, демонстрацию выполненного задания. Наиболее предпочитаемые формы организации занятий – групповые и индивидуальные.

 Большое воспитательное значение имеет подведение итогов работы, анализ, оценка. Метод проектов позволяет обучающимся системно овладеть организацией практической цепочки – от идеи через цели и задачи, «мозговой штурм» до реализации проекта и его защиты. Проектная деятельность основана на развитии самостоятельности детей, гибкой организации процесса обучения. В результате проектной деятельности полнее обеспечиваются современные требования к развитию личности обучающегося, учитываются его индивидуальные интересы и способности, выполняются и осваиваются конкретные поисковые действия.

 Обучение основам научно-исследовательской деятельности обучающимися: составление плана работы, поиск, выдвижение проблемы и подготовка проекта к защите, – все это предусмотрено конечным результатом после окончания обучения.

 Формы подведения итогов:

 - Участие в городских, окружных, региональных, всероссийских, международных олимпиадах и конкурсах, выставках и фестиваля.

 Методы организации учебно-познавательной деятельности:

 словесные, наглядные и практические, репродуктивные и проблемно-поисковые, индуктивные и дедуктивные методы обучения.

 Методы стимулирования и мотивации учебно-познавательной деятельности: познавательные игры, учебные дискуссии и др.

 Методы контроля обучения:

 Устный (предварительный, текущий, тематический, итоговый), практические задания и самоконтроль обучающихся.

 Дидактическое обеспечение программы:

 - Учебно-методическая литература

 - Презентации

 - Цифровые образовательные ресурсы;

 -Развивающие и диагностические приемы: эвристические вопросы, тесты, упражнения, творческие задания, игры, ребусы, конкурсы, сценарии.

 Материально-техническое обеспечение:

 Лаборатория инф соответствует нормам САН ПИН 2.2.2/2,41340-03, оборудована компьютерами, столами, стульями, общим освещением, классной доской, интерактивной доской, таблицами, мультимедийным оборудованием (проектор, экран, аудиоустройства);

#

# 5. Диагностика результативности

Урок 1.

1) Первое задание будет очень простым. Напишите программу, которая на первую строчку выводит «Здравствуйте!», а на вторую «Очень приятно с Вами познакомиться!».

2) Напишите программу, которая сначала считывает три фразы по очереди, а потом воспроизводит их в обратной последовательности, каждую на отдельной строчке

3) Напишите программу, которая считывает с клавиатуры последовательно три строки: название фильма, название кинотеатра и время, после чего выводит на экран «Билет на " [название фильма] " в " [название кинотеатра] " на [время] забронирован.» .

Пример

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| Тайна дома с часамиРодина13:45 | Билет на " Тайна дома с часами " в " Родина " на 13:45 забронирован. |

4) Напишите программу, которая якобы составляет гороскоп. Программа последовательно считывает с клавиатуры: имя, фамилию, любимое животное, знак зодиака, после этого выводит:

Индивидуальный гороскоп для пользователя [имя] [фамилия]

Кем вы были в прошлой жизни: [любимое животное]

Ваш знак зодиака - [знак зодиака] , поэтому вы - тонко чувствующая натура.

Пример

| **Ввод** | **Вывод** |
| --- | --- |
| МарияСтепановакошкаблизнецы | Индивидуальный гороскоп для пользователя Мария СтепановаКем вы были в прошлой жизни: кошкаВаш знак зодиака - близнецы, поэтому вы - тонко чувствующая натура. |

Слова про тонко чувствующую натуру выводятся абсолютно всегда, независимо от того, что именно вводил пользователь. В один и тот же фиксированный текст подставляются те слова, которые вводил пользователь.

Урок 2.

Задание на дом

1) Предположим, что нужно сверстать визитную карточку. Сколько оставить места для букв имени? Делать мы это будем по следующему правилу: нужно взять длинное имя, которое только можно придумать, удвоить количество букв и добавить места ещё для трёх букв на всякий случай.

Напишите программу, которая производит вычисления согласно этому правилу. Вводится одна строка — самое длинное имя, которое только можно придумать. Выводится одно целое число — количество букв, на которое нужно оставить место на визитной карточке.

Пример

| **Ввод** | **Вывод** |
| --- | --- |
| Дмитрий | 17 |

2) Напишите программу, которая определяет, сколько нужно отправить людей на строительные работы. Вводятся три числа – длина рва, которую необходимо выкопать, число дней, которое должно быть затрачена на работу и число метров, которое один работник может вырыть за один рабочий день. Выводится одно число – необходимое число работников

Пример

| **Ввод** | **Вывод** |
| --- | --- |
| 160033 | 178 |

 3) Напишите программу так, выводящую количество минут в не високосном году. Программа должна выводить одно целое число. Для задания числа дней в году, часов в сутках и минут в часах удобно задавать свои переменные.

4) Напишите программу, которая выводит на экран решение уравнения x + b = a, где x — неизвестное, a и b — числа, которые вводятся с клавиатуры. Выводится одно число – решение уравнения.

5) Вводится двузначное число. Нужно вывести в первую строку первую цифру данного числа, а во вторую строку – вторую цифру данного числа. Можно пользоваться только целочисленными переменными.

6) Определите стоимость телеграммы, если стоимость отправки одного символа (включая знаки препинания) составляет 40 копеек. На ввод подается строка, содержащая сообщение. Выводится строка - стоимость отправки сообщения.

Пример

| **Ввод** | **Вывод** |
| --- | --- |
| Привет! Как дела? | 6 р. 80 коп. |

**Урок 3.**

1. Какой результат получится после выполнения следующего кода:

**True** + 2

2. С клавиатуры вводятся координаты клетки шахматного поля, от 1 до 8, вставьте пропущенное место в программу, чтобы она выводила YES, если клетка белая, и NO, если клетка черная

x = int(input(**'Введите координату x: '**))
y = int(input(**'Введите координату y: '**))
**if** ....................................:
 print(**'YES'**)
**else**:
 print(**'NO'**)

3. Выпишите все варианты начальных значений для переменных word1 и word2, при которых программа выведет «Привет!»

word1 = input()
word2 = input()
**if** word1 == **'друг' and** \
 (word2 == **'брат' or** word2 == **'приятель'**):
 print(**'Привет!'**)
**else**:
 print(**'Здравствуйте!'**)

Урок 4.

Задания для проверки усвоения.

**Задание 1.** С клавиатуры вводится строка, содержащая три цифры. Определите, можно ли построить треугольник, стороны которого равны данным цифрам. Если нет, то выведите слово «нельзя». Иначе выведите тип треугольника

**Ограничение времени:** 1 секунда

**Ограничение памяти:** 64Mb

**Формат ввода**: строка

**Формат вывода:** строка, результат решения

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 534 | прямоугольный |

**Задание 2.** С клавиатуры вводится целое число N. Определите:

* S – сумму цифр данного числа;
* S\_odd – сумму нечетных цифр данного числа;
* S\_even – сумму четных цифр данного числа;
* Число M, цифры которого расположены в обратном порядке, по отношению к числу N (если N = 123, то M = 321);
* Число L, каждая цифра которого равна разности между 10 и соответствующей цифрой числа N, нуль не меняется (если N = 406, то L = 604);

**Ограничение времени:** 1 секунда

**Ограничение памяти:** 64Mb

**Формат ввода:** число N.

**Формат вывода**: числа S, S\_odd, S\_even, M, L, каждое с новой строки

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 1234 | 104643219876 |

**Примечание. Переменную типа «строка» использовать запрещено!!!**

**Задание** 3. Заполните пропущенные места в программе, вычисляющий количество цифр введенного натурального числа

n = int(input())
length = ...
**while** n > 0:
 .............
 .............
print(length)

**Задание** 4. Что должно стоять на месте пропуска, чтобы программа последовательно выводила четные числа?

z = 10
**while** z:
 z -= 1
 **if** z % 2 != 0:
 ...........
 print(z, end=**" "**)

**Задание** 5. Запишите пример последовательности чисел, которую надо задать в программу, чтобы она выполнила цикл 5 раз, а на выходе напечатала «Количество искомых чисел: 3»

count = 0
number = int(input())
**while** number != 0:
 **if** number % 3 == 0 **and** number % 2 == 1:
 count += 1
 number = int(input())
print(**'Количество искомых чисел:'**, count)

…………………………………………………………………………………………
…………………………………………………………………………………………

Урок 5.

**Задание 1.** Напишите программу, принимающую на вход целое положительное число N и выводящую на экран последовательность от 1 до N «ёлочкой».

**Ограничение времени:** 1 секунда

**Ограничение памяти:** 64Mb

**Формат ввода.** Строка, содержащая число N.

**Формат вывода.** Ёлочка из нескольких строк.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 14 | 12 34 5 67 8 9 1011 12 13 14 |

**Задание 2.** Определите первые n членов последовательности Фибоначчи, начиная с третьего, число n задается пользователем. Определяется последовательность так: первый и второй члены последовательности равны 1, а каждый последующий — сумме двух предыдущих

**Ограничение времени:** 1 секунда

**Ограничение памяти:** 64Mb

**Формат ввода:** число *n*.

**Формат вывода**: числа Фибоначчи, каждое с новой строки

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 5 | 235 |

**Задание** 3. Заполните пропущенные места в программе, чтобы она последовательно вывела 0, 2, 4, 6, 8

**for** x **in** range(..., ..., ...):
 print(x)

**Задание** 4. Напишите, что будет выведено в результате выполнения программ

|  |
| --- |
| m = 0**for** i **in** range(10): **if** i % 2 != 1: **continue** m += iprint(m)……………………………………………………………………………………… |
| m = **''for** i **in** range(3, 10): **if** i % 3 == 0: **continue if** i % 8 == 0: **break** m += str(i)print(m)……………………………………………………………………………………… |
| k = 0**for** i **in** range(6): p = i \* 3 **while** p > 0: k += 1 p //= 10print(k)……………………………………………………………………………………… |

Урок 6.

**Задание 1.**

Чему будет равна переменная s после операции присваивания?

s = {**'a'**, **'b'**, **'c'**} & {**'b'**, **'c'**, **'d'**}
………………………………………………………………………………………
s = {**'a'**, **'c'**, **'d'**} | {**'b'**, **'d'**, **'e'**}
………………………………………………………………………………………
s = {**'a'**, **'b'**, **'c'**, **'d'**} ^ {**'d'**, **'e'**}
………………………………………………………………………………………

**Задание 2.** Требуется определить, сколько мужчин-однофамильцев работает в организации. Имеется список фамилий, на основании которого нужно вычислить количество фамилий, которые совпадают с другими.

**Ограничение времени:**  1 секунда

**Ограничение памяти:** 64Mb

**Формат ввода**. В первой строке указывается количество мужчин - сотрудников организации (N). Затем идут N строк с фамилиями этих сотрудников в произвольном порядке.

**Формат вывода**. Целое число - количество однофамильцев в организации.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 6СмирновАхметовМатвеевАхметовСмирновАхметов | 5 |

**Задание 3**

Каждый ученик в классе изучает либо английский, либо немецкий, либо оба этих языка. У классного руководителя есть списки учеников, изучающих английский и немецкий языки. Помогите ему выяснить, сколько учеников в классе изучают только один язык.

**Формат ввода**

В первых двух строках указывается количество учеников, изучающих английский и немецкий языки (M и N). Затем идут M+N строк с фамилиями учеников в произвольном порядке. Гарантируется, что среди учеников нет однофамильцев.

**Формат вывода**

Количество учеников, которые изучают только один язык. Если таких не окажется, в строке вывода нужно написать NO.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 32СмирновАхметовБулатовСмирновМихайлов | 3 |

**Задание 4**

Школьница получила в конце учебного года список литературы на лето. Теперь надо выяснить, какие книги из этого списка у неё есть, а каких нет. К счастью, на компьютере есть текстовый документ, в котором записаны все книги из домашней библиотеки в случайном порядке. Определите, какие книги из списка на лето есть у девочки, а каких нет.

## Формат ввода

В первой строке записано число M — число книг в домашней библиотеке. Во второй строке записано число N - число книг в списке на лето. В домашней библиотеке и списке книг есть хотя бы по одной книге (M ≥ 1 и N ≥ 1). Далее идут M строчек с названиями книг из домашней библиотеки и N строчек названий из списка на лето. Гарантируется, что все слова в названиях книг разделены одним пробелом, а после последнего слова сразу идёт перевод строки (т. е. нет «невидимых» пробелов).

## Формат вывода

Выходные данные: N строчек, в каждой из которых написано слово YES, если книга найдена в библиотеке, и NO, если нет.

### Пример

| **Ввод** | **Вывод** |
| --- | --- |
| 42Алиса в ЗазеркальеАлиса в стране чудесПриключения Робинзона КрузоОстров сокровищПриключения Робинзона КрузоДети капитана Гранта | YESNO |

**Задание 5**

В холодильнике есть некоторые продукты, из которых можно приготовить блюда к столу. В кулинарной книге для каждого рецепта записаны необходимые ингредиенты. Определите, какие блюда на основе этих рецептов можно приготовить. Считайте, что если продукт есть в холодильнике, то его достаточно для приготовления любого блюда по любому рецепту.

## Формат ввода

В первой строчке указано число продуктов в холодильнике M. Далее идут M строчек с названиями продуктов. После этого — строчка с числом рецептов N. Далее — N блоков, описывающие каждый из рецептов. Блок начинается со строчки с названием рецепта, затем, на следующей строке - количество ингредиентов в нём. Далее идут названия ингредиентов. Названия каждого ингредиента и рецепта состоят из одного слова.

## Формат вывода

Названия рецептов, которые можно приготовить из продуктов в холодильнике, в порядке их появления во входном файле.

### Пример

| **Ввод** | **Вывод** |
| --- | --- |
| 2хлебмасло1бутерброд2маслохлеб | бутерброд |

Урок 7.

операторов?

s = **'ПРОГРАММИСТ'**print(s[2:6]) ………………………………………
print(s[1:8:3]) …………………………………
print(s[7:]) …………………………………………
print(s[:-2]) ………………………………………
print(s[::-1]) ……………………………………

**Задание 2**

Восстановите значения строковой переменной s1, если значение s2 равно крат, переменная s равна сумме s1 и s2, значение s[::2] равно дмрт и значение s[1::2] равно ока

**Задание 3**

Напишите программу, которая будет зашифровывать послание с помощью шифра Цезаря с заданным шагом шифрования. Суть шифра заключается в следующем. Сначала выбирается шаг шифрования (число), а затем все буквы послания заменяются на буквы, отстоящие от них в алфавите на шаг шифрования. Например, при шаге шифрования 3 (таким чаще всего пользовался Цезарь), буква А заменяется на букву Г, буква Б – на букву Д.

Обратите внимание, что алфавит «зациклен», то есть при сдвиге буквы Я на шаг 3 получится буква В.

## Формат ввода

Две строки. Первая содержит шаг шифрования, вторая – послание.

## Формат вывода

Строка с зашифрованным посланием.

### Пример 1

| **Ввод** | **Вывод** |
| --- | --- |
| 3АБВ | ГДЕ |

### Пример 2

| **Ввод** | **Вывод** |
| --- | --- |
| 5На дворе трава, на траве дрова! | Те йзухк чхезе, те чхезк йхузе! |

## Примечания

Символы русского алфавита расположены в стандартной для Python таблице кодировки подряд, то есть номера, выдаваемые функцией ord(symbol), идут подряд. Буква «ё» идёт в таблице кодировки отдельно от основного алфавита. При решении задачи считайте, что буквы «ё» в русском алфавите нет.

**Задание 4**

Пользователь вводит слова. Начиная со второго введённого слова, программа проверяет, совпадает ли первая буква нововведённого слова с последней буквой предыдущего. Если да, то программа работает дальше (считывает очередное слово). Если нет — выводит последнее на этот момент введённое слово и завершает работу.

**Формат ввода**

Вводятся несколько строк подряд.

**Формат вывода**

Выводится одна строка.

**Пример**

| Ввод | Вывод |
| --- | --- |
| новгороддублинновгороддублинтула | тула |

**Задание 5**

Напишите программу, обрабатывающую один раунд игры «Быки и коровы». Пользователь вводит две строки. Гарантируется, что это две строки одинаковой длины и что все символы в каждой из них разные. Необходимо вывести отдельно количество быков — символов, которые есть в обеих строках и стоят на одном и том же месте, и количество коров — символов, которые есть в обеих строках, но на разных местах.

**Формат ввода**

Две строки.

**Формат вывода**

Два целых числа через пробел — количество быков и коров.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| питонпилот | 3 1 |

Задание 6

Напишите программу, которая проверяет, подходит ли строка для того, чтобы быть именем пользователя, если в имени разрешается использовать только латинские буквы, цифры и символ «\_».

**Формат ввода**

Вводится одна строка — предполагаемое имя пользователя.

**Формат вывода**

Выводится одна строка: если введённая строка состоит только из маленьких латинских букв, цифр и символа «\_», то выводится «OK» (латиницей); иначе выводится: «Неверный символ: », после чего следует первый не подходящий символ из введённой строки.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| m&m's | Неверный символ: & |

Урок 8.

**Задание 1**

Напишите программу, которая выводит букву с заданным номером из каждой введённой строки. Это может использоваться, например, для конструирования аббревиатур или чтения акростихов. Если некоторые строки слишком короткие, и в них нет символа с заданным номером, то такие строки при выводе нужно просто пропускать.

**Формат ввода**

На первой строке вводится натуральное число — количество строк.
Далее следуют сами строки.
В конце вводится натуральное число — номер буквы (нумерация начинается с единицы).

**Формат вывода**

Буква с соответствующим номером из каждой строки; все буквы выводятся на одной и той же строке подряд.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 5Перваяизтехнологий, которойобучаютначинающих программистов1 | Питон |

**Задание 2**

Напишите жёсткий фильтр поисковых запросов для интернета в школе, который пропускает лишь те запросы, которые есть в «белом списке».

## Формат ввода

На первой строке вводится количество пунктов «белого списка».
Затем — сами пункты «белого списка».
На отдельной строке — количество запросов, которые нужно проанализировать.
Затем — сами запросы.

## Формат вывода

Те запросы из введённых, которые есть в «белом списке».

### Пример 1

| **Ввод** | **Вывод** |
| --- | --- |
| 3учебникикотикипочта4сериалыкотикимемыучебники | котикиучебники |

### Пример 2

| **Ввод** | **Вывод** |
| --- | --- |
| 1учебники4учебникискачать бесплатно рефераткак обойти фильтр поисковых запросовучебники | учебникиучебники |

**Задание 3**

Напишите программу, которая считывает сначала количество наименований покупок, потом по очереди для каждой покупки её название и количество (целое число) на отдельных строках, затем выводит их же в обратном порядке, выводя количество через пробел от названия покупки.

**Формат ввода**

На первой строке — количество наименований покупок.
Затем список покупок, каждая позиция списка записывается на двух строках: наименование покупки и количество экземпляров.

**Формат вывода**

Тот же список покупок в обратном порядке, каждая позиция списка записывается на одной строке, количество через пробел от наименования.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 4магнитофон импортный1пиджак замшевый1кинокамера заграничная2портсигар золотой отечественный3 | портсигар золотой отечественный 3кинокамера заграничная 2пиджак замшевый 1магнитофон импортный 1 |

**Задание 4**

Напишите программу, которая считывает список чисел, а затем выводит сумму в диапазоне от заданного до заданного. **При этом в программе не должно использоваться обращение к элементам по индексу**.

**Формат ввода**

На первой строке вводится натуральное число N — количество чисел в списке.
Далее следуют N целых чисел из списка, каждое на отдельной строке.
Затем на отдельных строках следуют начальный (p) и конечный (q) номера элементов списка, который нужно просуммировать. Нумерация идёт с единицы. Гарантируется, что 1≤ p ≤ q ≤ N.

**Формат вывода**

Выводится сумма элементов списка с p-го включительно по q-й включительно.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 510022230044450024 | 966 |

Урок 9.

**Задание 1**

В переменных a, b, c, d, e содержатся числа 2, 5, 1, 4, 3 соответственно. В какой последовательности необходимо записать следующие строки, чтобы в переменных a, b, c, d, e оказались числа 5, 4, 3, 2, 1?

1: b, d = d, b

2: a, b = b, a

3: c, e = e, c

………………………………………………………………………………………

**Задание 2**

Отсортируйте введённые числа по убыванию с помощью метода сортировки выбором.

**Формат ввода**

На первой строке вводится натуральное число N — количество чисел.
Далее следуют N целых чисел, которые надо будет отсортировать, каждое на отдельной строке.

**Формат вывода**

Выводятся те же N чисел, но не в том порядке, в котором они вводились, а в порядке убывания.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 43424343292 | 2922434334 |

**Задание 3**

Отсортируйте введённые строки по возрастанию в алфавитном порядке. Под алфавитным порядком подразумевается, что каждая следующая строка должна быть больше предыдущей с точки зрения оператора сравнения «>» в Python; в частности, все строчные буквы идут после всех заглавных. Используйте метод пузырьковой сортировки.

**Формат ввода**

На первой строке вводится натуральное число N — количество строк.
Далее следуют N строк, которые надо будет отсортировать.

**Формат вывода**

Выводятся те же N строк, но не в том порядке, в котором они вводились, а в алфавитном порядке по возрастанию.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 4тричетырепятьшесть | пятьтричетырешесть |

**Задание 4**

 Несколько команд участвуют в турнире по игре «Что? Где? Когда?». Согласно регламенту турнира, в финал проходит половина команд (с округлением вверх), набравшие больше правильных ответов, чем остальные. Необходимо сформировать итоговый протокол.

**Ограничение времени:** 1 секунда

**Ограничение памяти:** 64Mb

**Формат ввода**. На первой строке указано натуральное число N — количество команд. Далее следует описание результатов N команд в произвольном порядке. Данные о каждой команде записываются на двух строках: на первой — название команды, на второй — количество правильных ответов. Гарантируется, что количество правильных ответов у всех команд разное.

**Формат вывода.** Нужно вывести на отдельных строках сначала в алфавитном порядке названия всех команд, вышедших в финал (т. е. команд из лучшей по количеству ответов половины), а затем названия всех остальных команд (также на отдельных строках в алфавитном порядке).

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 4Школьники22Тугодумы20Уфимские знатоки24Юные дарования23 | Уфимские знатокиЮные дарованияТугодумыШкольники |

Урок 10.

**Задание 1**

Необходимо составить список из квадратов нечётных натуральных чисел от 1 до 999 ([1, 9, 25, … , 998001]). Для этого используется списочное выражение следующего вида:

numbers = [………… **for** i **in** range(………, …………) **if** …………………………]

Заполните пропуски в выражении.

**Задание 2**

Напишите программу для построения горизонтальных столбчатых диаграмм с помощью символа, введенного с клавиатуры. Подсказка: выводя строку, не используйте для этого цикл. Желально написать программу в одну строку

**Формат ввода**

В первой сроке вводится символ, во второй строке вводится несколько натуральных чисел, разделённых пробелами.

**Формат вывода**

Для каждого введённого числа выводится строка, состоящая из символов, длина которой равна этому числу.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| \*3 7 1 10 8 | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

**Задание 3**

Используя списочные выражения и метод split, составьте список из введённых чисел, записанных на одной строке без указания заранее их количества; затем выведите их квадраты также на одной строке.

**Формат ввода**

Несколько натуральных чисел на одной строке.

**Формат вывода**

Несколько чисел на одной строке.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 7 1 6 | 49 1 36 |

**Задание 4**

Напишите программу для генерации вертикальных столбчатых диаграмм в рамке с помощью символа, введенного с клавиатуры.

**Формат ввода**

В первой сроке вводится символ, во второй строке вводится несколько натуральных чисел, разделённых пробелами.

**Формат вывода**

Выводятся вертикальные столбцы из символов, высоты которых равны введённым числам, причём имеются дополнительные столбцы слева и справа от диаграммы и горизонтальная линия сверху, проведённая так, чтобы между ней и самым высоким из столбцов диаграммы был зазор ровно в один ряд.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| \*3 5 4 2 | \*\*\*\*\*\*\* \*\* \* \*\* \*\* \*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\* |

**Задание 5**

В водится несколько натуральных чисел, разделённых пробелами..

Нужно вывести в отдельных строках числа из введенной последовательности, которые удовлетворяют следующим условиям:

* числа должны быть больше квадрата первого числа и меньше квадрата последнего числа последовательности;
* числа должны быть четными.
* **Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 1 2 8 16 | 2816 |

Урок 11.

Используя списочное выражение и подходящую стандартную функцию, найдите длину самого длинного из введённых слов.

## Формат ввода

Одна строка: слова, разделённые пробелами.

## Формат вывода

Одно число: длина самого длинного из введённых слов.

## Пример

| **Ввод** | **Вывод** |
| --- | --- |
| окажем скорейшее вспомоществование | 17 |

**Задание 2**

Выясните, какой символ чаще всего встречается во введенной строке.

## Формат ввода

Вводится одна строка.

## Формат вывода

Выводится один символ в нижнем регистре — наиболее часто встречающийся во введённой строке, кроме пробела, без учёта регистра, если таких несколько — выводится первый по алфавиту.

## Пример

| **Ввод** | **Вывод** |
| --- | --- |
| Баобаб | б |

**Задание 3**

Напишите программу, которая определяет, в каких словах встречается введенная буква

## Формат ввода

На первой строке вводится один символ — строчная буква.
На второй строке вводится предложение.

## Формат вывода

Нужно вывести список слов (словом считается часть предложения, окружённая символами пустого пространства), в которых присутствует введённая буква в любом регистре, в том же порядке, в каком они встречаются в предложении.

## Пример

| Ввод | Вывод |
| --- | --- |
| mMary had a little lamb. | Marylamb. |

**Задание 4**

Треугольник Паскаля — это последовательность рядов чисел, которые строятся по следующему принципу: первый ряд состоит из одного числа 1, каждый следующий ряд состоит из попарных сумм соседних элементов предыдущего ряда, а также начальной и конечной единиц. Например, первые пять рядов таковы:

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

Скажем, ряд «**1 4 6 4 1**» получается из предыдущего ряда «*1 3 3 1*» так: в начале всегда **1**, *1+3*=**4**, *3+3*=**6**, *3+1*=**4**, в конце всегда **1**.

Числа, составляющие треугольник Паскаля, обладают многочисленными важными и интересными свойствами. В частности, они встречаются в биноме Ньютона — формуле для разложения суммы двух переменных на отдельные слагаемые:



Поэтому эти числа также называются биномиальными коэффициентами.

Напишите программу, которая выводит первые несколько рядов треугольника Паскаля.

## Формат ввода

Натуральное число n — количество рядов треугольника Паскаля, которые нужно вывести.

## Формат вывода

Элементы n первых рядов треугольника Паскаля, разделённые символами пустого пространства (пробелами, табуляциями, переносами строки); точный формат вывода неважен, но крайне желательно, чтобы одному ряду треугольника Паскаля соответствовала одна строка вывода.

## Пример

| **Ввод** | **Вывод** |
| --- | --- |
| 4 | 11 11 2 11 3 3 1 |

**Задание 5**

Напишите программу, которая составляет новое предложение по номерам слов из a в исходном. Предложение должно начинаться с большой буквы. Гарантируется, что в исходной строке только одна заглавная буква (в начале предложения).

## Формат ввода

Сначала строка, содержащая номера слов через пробел. Затем строка, содержащая сами слова, записанные через пробел.

## Формат вывода

Строка, составленная из введенных слов в порядке, заданном номерами. Строка должна начинаться с заглавной буквы и не содержать других заглавных букв, кроме первой.

### Пример 1

| **Ввод** | **Вывод** |
| --- | --- |
| 4 3 1Буря мглою небо кроет | Кроет небо буря |

### Пример 2

| **Ввод** | **Вывод** |
| --- | --- |
| 3 2 1Ты и я | Я и ты |

## Примечания

Нумерация слов начинается с единицы.

Урок 12.

Восстановите пропущенные места в программе, если значение переменной s1 равно апельсин, а значение переменной s2 равно спаниель

dct = {3: s1[...], ...: s1[1], 6: s1[...], ...: s1[3],
 8: s1[...], ...: s1[...], ...: s1[6], 4: s1[...]}
lst = [key **for** key **in** dct.keys()]
lst.sort()
s2 =**''**.join(dct[k] **for** k **in** lst)

**Задание 2**

Дан английский текст. Закодируйте его с помощью азбуки Морзе.

Каждая буква заменяется на последовательность точек и тире. В качестве тире используйте обычный дефис: «-», а в качестве точки — точку «.». Например, буква «g» превратится в трёхсимвольную строку «--.». Между закодированными буквами нужно ставится ровно один пробел. Например, слово «Help» превратится в «.... . .-.. .--.».

Обратите внимание, что строчные и заглавные буквы кодируются одинаково.

Для решения удобно использовать следующий словарь

morse = {'A': '.-', 'B': '-...', 'C': '-.-.',

 'D': '-..', 'E': '.', 'F': '..-.',

 'G': '--.', 'H': '....', 'I': '..',

 'J': '.---', 'K': '-.-', 'L': '.-..',

 'M': '--', 'N': '-.', 'O': '---',

 'P': '.--.', 'Q': '--.-', 'R': '.-.',

 'S': '...', 'T': '-', 'U': '..-',

 'V': '...-', 'W': '.--', 'X': '-..-',

 'Y': '-.--', 'Z': '--..', }

**Формат ввода**

Весь текст записан в единственной строке. Текст состоит из английских букв и пробелов, других символов в тексте нет. В тексте не может быть двух или более пробелов подряд.

Длина текста не превышает 100 000 символов, количество слов в тексте не превышает 1000.

**Формат вывода**

Выведите каждое слово исходного текста, закодированное азбукой Морзе, на отдельной строке. Количество строк в ответе должно совпадать с количеством слов в исходном тексте.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| Help me SOS | .... . .-.. .--.-- .... --- ... |

**Задание 3**

Дана строчка русского текста, состоящая из слов и пробелов. Словом считается последовательность русских букв, слова разделены одним или большим числом пробелов.

Для каждого слова этого текста узнайте порядковый номер его вхождения в текст именно в той форме, в которой указано слово. Для первого вхождения слова выведите «1», для второго вхождения того же слова выведите «2» и так далее.

**Формат ввода**

В единственной строчке записан русский текст. Длина текста не превышает 100 000 символов. Текст состоит только из русских букв и символов пробела.

**Формат вывода**

Для каждого слова исходного текста выведите одно целое число — номер вхождения этого слова в текст. Числа выведите через пробел. Количество чисел должно совпадать с количеством слов в исходном тексте.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| Раз раз раз как меня слышно Повторяю раз раз раз Повторяю | 1 1 2 1 1 1 1 3 4 5 2  |

**Задание 4**

Дан текст на русском языке. Посчитайте, сколько раз встречается в тексте каждое слово. Слово — это последовательность русских букв. Кроме слов, в тексте могут присутствовать пробелы и знаки препинания («,», «.», «!», «?», «:» и «;»). Других символов в тексте нет. Слова необходимо сравнивать с точностью до регистра. То есть «Привет» и «привет» — это одно и то же слово, а вот «рыба», «рыбы» и «рыбу» — три разных.

**Формат ввода**. В первой строчке задано одно целое число N (1 ≤ N ≤ 1000) — количество строк в тексте. В следующих N строках приведён сам текст. Длина текста не превышает 100 000 символов.

**Формат вывода.** Выведите слова исходного текста, упорядоченные по убыванию количества их попаданий в текст. В случае, если два слова встречаются в тексте одинаковое количество раз, выведите их в лексикографическом порядке.Каждое слово нужно вывести в отдельной строке. Первая буква слова должна быть заглавной (даже если она не была заглавной в исходном тексте), остальные — строчными.

**Пример**

| **Ввод** | **Вывод** |
| --- | --- |
| 2Ехал Грека через реку. Видит Грека в реке рак.Сунул Грека руку в реку, рак за руку Греку цап. | ГрекаВРакРекуРукуВидитГрекуЕхалЗаРекеСунулЦапЧерез |

Урок 13.

Во всех заданиях этого урока от Вас требуется только код функции. Вы можете (и должны) проверять работоспособность функции в теле основной программы, но прикладывать код применения функции не нужно

**Задание 1**

Напишите функцию print\_symbols(symbol, number), которая печатает символ symbol number раз, каждый в своем столбце. Данная функция ничего не должна возвращать

**Задание 2**

Напишите функцию my\_calc(s, a, b), которая в зависимости от значения symbol производит арифметические операции возведения в степень, деления на цело, вычисления остатка от деления или обыкновенного деления между a и b. Обратите внимание, что функция должна возвращать число. Если же значение symbol некорректно, то функция выдает сообщение об ошибке.

**Задание 3**

Напишите рекурсивную функцию sum\_number\_list(some\_list), принимающую в себя список целых чисел и возвращающую сумму его элементов

**Задание 4.** Напишите функцию parrot(phrase), которая будет симулировать речь попугая. Если в переменной phrase находится предложение, которое попугай слышал хотя бы один раз, то функция печатает эту фразу.

Пример проверки работы функции – тело основной программы:

| **Ввод** | **Вывод** |
| --- | --- |
| parrot("Привет!")parrot("Привет!")parrot("Как дела?")parrot("Что новенького в учебе?")parrot("Как поживает Ваша семья?")parrot("Как дела?") | Привет!Как дела? |

**Задание 5.** Дана строка, содержащая координату клетки на шахматном поле (например, “B1”). Напишите функцию possible\_turns(cell), которая будет по такой строке возвращать список координат (в такой же текстовой форме) всех клеток, на которые может попасть конь с указанной клетки за один ход. Не забудьте, что за пределы шахматного поля коню ходить нельзя.

Пример проверки работы функции – тело основной программы:

| **Ввод** | **Вывод** |
| --- | --- |
| print(possible\_turns("B1")) | ['A3', 'C3', 'D2'] |

**Примечание.** Полезно сделать некоторые вспомогательные функции, которые будут по шахматной клетке делать целочисленные координаты и наоборот, а также функцию, которая по кортежу целочисленных координат сообщает, находится ли это клетка внутри шахматного поля или снаружи.

Урок 14.

**Задание 1.**

Напишите, как вычислять площадь круга в случае различного способа вызова объекта pi из библиотеки math

**from** math **import** pi

r = float(input())

s = ..............

**from** math **import** \*

r = float(input())

s = ..............

**import** math

r = float(input())

s = ..............

**Задание 2.**

Из списка нужно случайным образом выбрать одного человека. Заполните недостающие места в программе, воспользовавшись наиболее подходящей функцией библиотеки random

**from** random **import** .....................

a = [**'Иван'**, **'Петр'**, **'Василий'**, **'Фёдор'**]
d = ....................................
print(d)

**Задание 3.**

Имеется список преподавателей, для каждого из них нужно выбрать посещающего занятия у коллеги. По правилам один преподаватель не может посещать занятия у нескольких коллег, очевидно, что нельзя посещать занятия у самого себя.

**Формат ввода.** Должность, фамилия и инициалы на каждой строке, например:

Профессор Волков И.Т.

Доцент Зайцев С.В.

Ст. преподаватель Лисицына З.Б.

**Формат вывода.** На каждой строчке должность, фамилия и инициалы преподавателя и должность, фамилия и инициалы посетившего его коллеги, разделённые дефисом, например:

Профессор Волков И.Т. - Ст. преподаватель Лисицына З.Б.

Ст. преподаватель Лисицына З.Б.- Доцент Зайцев С.В

Доцент Зайцев С.В - Профессор Волков И.Т.

**Задание 4.**

Требуется по запросу выдавать *n* различных паролей длиной *m* символов, состоящих из строчных и прописных латинских букв, и цифр, кроме тех, которые легко перепутать между собой: «l» (L маленькое), «I» (i большое), «1» (цифра), «o» и «O» (большая и маленькая буквы) и «0» (цифра).

Дополнительное условие: в каждом пароле обязательно должна присутствовать хотя бы одна цифра и как минимум по одной букве в верхнем и нижнем регистре.

Решение должно содержать **две функции**: вспомогательную generate\_password(m), возвращающую случайный пароль длиной m символов, и основную main(n, m), возвращающую список из n различных паролей, каждый длиной m символов.

Будем считать, что параметры *n* и *m* всегда таковы, что требуемые пароли возможно сгенерировать.

**Формат ввода**. Пример вызова ваших функций для тестирования:

print("Случайный пароль из 7 символов:" , generate\_password(7))

print("10 случайных паролей длиной 15 символов:")

print(\*main(10, 15), sep="\n")

**Формат вывода.** Пример результатов:

Случайный пароль из 7 символов: V77LMgk

10 случайных паролей длиной 15 символов:

nsHVGJH4B9q3n7B

KgBmJKP9XLqHKcp

LuydsM9ZgBUJvVj

a3jxMpQjmtjs79g

nvU8SMLfGFq4erU

sfcmQmR6dLKQeNt

WNtFQd6Bd7wADNP

fDtMmhxA9eN4yJg

Uau2sRtWL5qH81R

G5hfFkHKxSBZxsu

Урок 15.

**Задание 1.** Напишите класс Balance для описания весов с двумя чашами. На левую и правую чашу объекта будут добавляться грузы с различным весом, ваша задача определить положение чаш.

Метод add\_right принимает целое число — вес, положенный на правую чашу весов, add\_left — на левую чашу. Метод result должен возвращать символ =, если вес на чашах одинаковый, R — если перевесила правая, L — если перевесила левая.

Пример проверки работы класса – тело основной программы:

| **Ввод** | **Вывод** |
| --- | --- |
| balance = Balance()balance.add\_right(10)balance.add\_left(9)balance.add\_left(2)print(balance.result()) | L |

**Задание 2.** Создайте класс BoundingRectangle, который обрабатывает точки на плоскости и строит по ним прямоугольник минимального размера, в который входят все эти точки. Если точка лежит на границе прямоугольника, считается, что она в него входит.

Нужно определить следующие методы:

add\_point(x, y) — добавить новую точку.

width() — ширина прямоугольника.

height() — высота прямоугольника.

bottom\_y() — Y-координата нижней границы прямоугольника.

top\_y() — Y-координата верхней границы прямоугольника.

left\_x() — X-координата левой границы прямоугольника.

right\_x() — X-координата правой границы прямоугольника.

Гарантируется, что хотя бы одна точка будет добавлена в экземпляр до вызова методов, возвращающих описание прямоугольника.

Пример проверки работы класса – тело основной программы:

| **Ввод** | **Вывод** |
| --- | --- |
| rect = BoundingRectangle()rect.add\_point(-11, -12)rect.add\_point(13, -14)rect.add\_point(-15, 10)print(rect.left\_x(), rect.right\_x())print(rect.bottom\_y(), rect.top\_y())print(rect.width(), rect.height())print()rect.add\_point(-21, -12)rect.add\_point(13, -14)rect.add\_point(-15, 36)print(rect.width(), rect.height())print(rect.left\_x(), rect.right\_x())print(rect.bottom\_y(), rect.top\_y())print()rect.add\_point(-21, 78)rect.add\_point(13, -14)rect.add\_point(-55, 36)print(rect.bottom\_y(), rect.top\_y())print(rect.width(), rect.height())print(rect.left\_x(), rect.right\_x())print() | -15 13-14 1028 2434 50-21 13-14 36-14 7868 92-55 13 |

**Задание 3.** Как известно, форматы записи дат в виде строки в России, впрочем, как и во всей Европе отличаются от принятых в США.

В США принят формат мм.дд.гггг, в Европе — дд.мм.гггг, где дд — день (дополняется нулём слева, если число меньше 10), мм — месяц (так же дополняется нулём слева), гггг — год.

Например, 10 апреля 2000 года будет записано в американском формате как 04.10.2000, а в европейском — как 10.04.2000. Все годы в задаче — четырёхзначные.

Реализуйте классы AmericanDate и EuropeanDate. При инициализации они должны принимать год, месяц и число (именно в этом порядке). Так же должны быть реализованы методы set\_year, set\_month, set\_day для изменения одной из компонентов даты, и get\_year, get\_month, get\_day для чтения компонентов даты. Метод format должен возвращать строковое представление (своё для каждого класса).

Гарантируется, что все даты в тестах корректны и существуют в календаре.

Пример проверки работы класса – тело основной программы:

| Ввод | Вывод |
| --- | --- |
| american = AmericanDate(2019, 5, 15)european = EuropeanDate(2019, 5, 15)print(american.format())print(european.format()) | 05.15.201915.05.2019 |

**Задание 4**. С помощью наследования постройте иерархию геометрических фигур, состоящую из следующих классов:

* Shape (Фигура)
* Polygon (Многоугольник)
* Triangle (Треугольник)
* IsoscelesTriangle (Равнобедренный треугольник)
* EquilateralTriangle (Равносторонний треугольник)
* Quadrilateral (Четырехугольник)
* Parallelogram (Параллелограмм)
* Rectangle (Прямоугольник)
* Square (Квадрат)

Иерархия должна быть построена в соответствии со следующим условием: класс B должен наследоваться от класса A, если B является частным случаем A.

Сами классы реализовывать не нужно.

# Список использованной литературы

* 1. Конвенция о правах ребенка (принята 20 ноября 1989 г. Резолюцией 44/25 Генеральной Ассамблеи ООН).
	2. Конституция Российской Федерации (01.07.2020).
	3. Закон РФ «Об образовании в Российской Федерации» от 29 декабря 2012 г. № 273-ФЗ.
	4. Санитарно-эпидемиологические требования к учреждениям дополнительного образования СанПиН 2.4.4.1251-03, утвержденные Главным государственным санитарным врачом РФ 1 апреля 2003 г., дата введения – 20 июня 2003 г.
	5. Концепция развития дополнительного образования детей (Распоряжение Правительства РФ от 4 сентября 2014 года № 1726-р).
	6. Постановление Главного государственного санитарного врача Российской Федерации от 4 июля 2014 г. № 41 г. Москва «Об утверждении СанПин 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
	7. Закон Республики Башкортостан «Об образовании в Республике Башкортостан» от 1 июля 2013 г. № 696-3.
	8. Васильев, А.Н. Python на примерах. Практический курс по программированию: учебное пособие / А.Н. Васильев. СПб.: Наука и Техника, 2017. — 432 с.
	9. Златопольский, Д.М. Основы программирования на языке Python: учебник / Д.М. Златопольский. М.: ДМК Пресс, 2017. — 284 с.
	10. Лучано, Р. Python. К вершинам мастерства М.: ДМК Пресс, 2016. — 768 с.
	11. Саммерфилд, М. Python на практике: учеб. Пособие. М.: ДМК Пресс, 2014. — 338 с.
	12. Хахаев, И.А. Практикум по алгоритмизации и программированию на Python: учебное пособие / И.А. Хахаев. М.: , 2016. — 178 с