

Задачи по программированию

Часть 2

Задача 1. В отдельном файле создать модуль *arithmetic.py*, в котором определены функции сложения, вычитания, умножения и деления двух действительных чисел. Далее, требуется импортировать модуль *arithmetic.py* в программу *main.py*, которая вычисляет и выводит на экран значение выражения $\left(\frac{3}{14} + \frac{2}{21}\right) \cdot \left(\frac{5}{13} - \frac{1}{39}\right)$, не используя стандартные операторы "+", "-", "*", и "/", а только функции данного модуля.

Задача 2. Дано описание родительского класса **Figure**, как показано ниже:

```
class Figure():
    name = ''
    def __init__(self, name):
        self.name = name
```

Требуется определить два дочерних класса **Circle** (окружность) и **Square** (квадрат), дополнительно имеющие атрибуты радиус окружности и длина стороны квадрата, соответственно, а также конструкторы, которые инициализируют все атрибуты этих классов, включая атрибуты родительского класса с помощью функции *super()*.

Далее требуется определить функцию *compare_figures*, которая на входе принимает два объекта из указанных выше дочерних классов (**Circle** и/или **Square**), и выводит на экран какой из объектов больше по площади.

Написать программу *main.py*, в которой реализованы данные требования и продемонстрирована работа функции *compare_figures*.

Например:

для объекта *A* класса **Circle** с радиусом 1 и объекта *B* класса **Square** со стороной 2, результат работы функции *compare_figures(A, B)* должен быть следующим:

Второй объект имеет большую площадь

для объекта *A* класса **Circle** с радиусом 2 и объекта *B* класса **Circle** с радиусом 1, результат работы функции *compare_figures(A, B)* должен быть следующим:

Первый объект имеет большую площадь

Задача 3. Дан текстовый файл *matrix.txt*, который содержит значения некоторой матрицы *A* размером **m** x **n** (**m, n** > 2) в формате:

```
a11 a12 a13 ... a1n
a21 a22 a23 ... a2n
...
am1 am2 am3 ... amn
```

Написать программу **main.py**, которая считывает файл **matrix.txt**, и создает **m** текстовых файлов **sum_row_i.txt**, где $1 \leq i \leq m$, а сам файл содержит сумму элементов **i**-ой строки матрицы A.

Задача 4. Написать программу **main.py**, которая на входе принимает значение целого числа **n** ($2 \leq n$) и выводит на экран равнобедренный треугольник, основание которого состоит из **n** звездочек и **n-1** пробелов, высота треугольника равна $\text{floor}(n/2) + 1$ строк, а ребра закрашены по периметру звездочками. Примеры треугольников для $n = 7$ и $n = 8$ показаны ниже.

```
      *
     * *
    *   *
   *     *
  *       *
 * * * * *
n = 7

      *
     * *
    *   *
   *     *
  *       *
 * * * * *
n = 8
```

Задача 5. Дан текстовый файл **code.txt**, который содержит правило отображения символов в целые числа в следующем формате:

```
a 16
b 10
c 18
d 24
...
- 27
! 8
? 26
```

Замечание: используйте число **0** для отображения пробела.

Требуется определить две функции **encoder** и **decoder**, где **encoder** – это функция (шифрователь), преобразующая текст в последовательность чисел, а **decoder** – это функция (дешифрователь), преобразующая последовательность чисел в текст по правилам отображения, указанным в файле **code.txt**.

Напишите программу **crypto.py**, в которой необходимо определить и продемонстрировать работу функций **encoder** и **decoder**, на примере зашифрованного текстового сообщения, хранящегося в виде одной строки в файле **message.txt**. Вывести на экран оригинальное сообщение, дешифрованное и повторно зашифрованные сообщения. Используйте шаблон файла **crypto.py**.