

**Московский государственный технический
Университет им Н.Э.Баумана**

Факультет «Информатика и системы управление»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»
Отчет по лабораторной работе №1

Выполнил:

Студент группы ИУ5-34Б

Малышко А. В.

Подпись и дата:

Проверил:

Преподаватель каф. ИУ5

Нардид А. Н.

Подпись и дата:

Москва 2024 г.

Постановка задачи

Разработать программу для решения [биквадратного уравнения](#).

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ([вариант задания параметров приведен в конце файла с примером кода](#)). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки](#).
4. Если коэффициент А, В, С введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.
5. Дополнительное задание 1 (*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

Текст программы

```
import sys
import math

class SquareRoots:

    def __init__(self):
        '''
        Конструктор класса
        '''
        # Объявление коэффициентов
        self.coef_A = 0.0
        self.coef_B = 0.0
        self.coef_C = 0.0
        # Количество корней
        self.num_roots = 0
```

```

# Список корней
self.roots_list = []

def get_coef(self, index, prompt):
    """
    Читаем коэффициент из командной строки или вводим с клавиатуры
    Args:
        index (int): Номер параметра в командной строке
        prompt (str): Приглашение для ввода коэффициента
    Returns:
        float: Коэффициент квадратного уравнения
    """
    while True:
        try:
            # Пробуем прочитать коэффициент из командной строки
            coef_str = sys.argv[index]
        except:
            # Вводим с клавиатуры
            print(prompt)
            coef_str = input()
        # Переводим строку в действительное число
        try:
            # Пытаемся преобразовать строку в число
            coef = float(coef_str)
            return coef
        except ValueError:
            print(f"Некорректное значение: '{coef_str}'. Пожалуйста, введите действительное число.")

def get_coefs(self):
    """
    Чтение трех коэффициентов
    """
    self.coef_A = self.get_coef(1, 'Введите коэффициент A:')
    self.coef_B = self.get_coef(2, 'Введите коэффициент B:')
    self.coef_C = self.get_coef(3, 'Введите коэффициент C:')

def calculate_roots(self):
    """
    Вычисление корней квадратного уравнения
    """
    a = self.coef_A
    b = self.coef_B
    c = self.coef_C
    # Вычисление дискриминанта и корней
    D = b*b - 4*a*c
    if D == 0.0:
        root = -b / (2.0*a)
        #self.num_roots = 0
        #self.roots_list.append(root)
        if root < 0:
            self.num_roots = 0
        elif (root == 0):
            self.num_roots = 1
            self.roots_list.append(root)
        elif (root > 0):
            root1 = root**0.5
            root2 = -(root**0.5)
            self.num_roots = 2
            self.roots_list.append(root1)
            self.roots_list.append(root2)

    elif D > 0.0:
        sqD = math.sqrt(D)

```

```

root1 = (-b + sqD) / (2.0*a)
root2 = (-b - sqD) / (2.0*a)
#self.num_roots = 2
#self.roots_list.append(root1)
#self.roots_list.append(root2)
D1=max(root1,root2)
D2=min(root1,root2)
if(D1 < 0 and D2<0):
    self.num_roots=0
elif((D1==0 and D2<0) or (D1==0 and D2==0)):
    self.num_roots=1
    self.roots_list.append(0)
elif(D1>0 and D2<0):
    self.num_roots=2
    root3 = D1**0.5
    root4 = -(D1 ** 0.5)
    self.roots_list.append(root3)
    self.roots_list.append(root4)
elif(D1>0 and D2==0):
    self.num_roots = 3
    root3 = D1 ** 0.5
    root4 = -(D1 ** 0.5)
    self.roots_list.append(root3)
    self.roots_list.append(root4)
    self.roots_list.append(0)
elif(D1>0 and D2>0):
    self.num_roots = 4
    root3 = D1 ** 0.5
    root4 = -(D1 ** 0.5)
    root5 = D2 ** 0.5
    root6 = -(D2 ** 0.5)
    self.roots_list.append(root3)
    self.roots_list.append(root4)
    self.roots_list.append(root5)
    self.roots_list.append(root6)

def print_roots(self):
    # Проверка отсутствия ошибок при вычислении корней
    if self.num_roots != len(self.roots_list):
        print(('Ошибка. Уравнение содержит {} действительных корней, ' + \
            'но было вычислено {} корней.').format(self.num_roots,
len(self.roots_list)))
    else:
        if self.num_roots == 0:
            print('Нет корней')
        elif self.num_roots == 1:
            print('Один корень: {}'.format(self.roots_list[0]))
        elif self.num_roots == 2:
            print('Два корня: {} и {}'.format(self.roots_list[0], \
                self.roots_list[1]))
        elif self.num_roots == 3:
            print('Три корня: {} и {} и {}'.format(self.roots_list[0], \
                self.roots_list[1],self.roots_list[2]))
        elif self.num_roots == 4:
            print('Четыре корня: {} и {} и {} и {}'.format(self.roots_list[0], \
                self.roots_list[1], \
                self.roots_list[2],\
                self.roots_list[3]))

def main():
    '''
    Основная функция

```

```

'''
# Создание объекта класса
r = SquareRoots()
# Последовательный вызов необходимых методов
r.get_coefs()
r.calculate_roots()
r.print_roots()

# Если сценарий запущен из командной строки
if __name__ == "__main__":
    main()

```

Экранные формы с примерами выполнения программы

```

Введите коэффициент A:
1
Введите коэффициент B:
a
Некорректное значение: 'a'. Пожалуйста, введите действительное число.
Введите коэффициент B:
1
Введите коэффициент C:
0
Один корень: 0

Process finished with exit code 0
|

```

```

Введите коэффициент A:
1
Введите коэффициент B:
-7
Введите коэффициент C:
12
Четыре корня: 2.0 и -2.0 и 1.7320508075688772 и -1.7320508075688772

Process finished with exit code 0
|

```

```
Введите коэффициент A:
afa
Некорректное значение: 'afa'. Пожалуйста, введите действительное число.
Введите коэффициент A:
asf
Некорректное значение: 'asf'. Пожалуйста, введите действительное число.
Введите коэффициент A:
1
Введите коэффициент B:
1
Введите коэффициент C:
1
Нет корней

Process finished with exit code 0
|
```

```
Введите коэффициент A:
1
Введите коэффициент B:
-1
Введите коэффициент C:
0
Три корня: 1.0 и -1.0 и 0

Process finished with exit code 0
|
```

```
Введите коэффициент A:
1
Введите коэффициент B:
-2
Введите коэффициент C:
1
Два корня: 1.0 и -1.0

Process finished with exit code 0
|
```