

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

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

**Отчет по лабораторной работе №1  
«Основные конструкции языка Python»  
по дисциплине  
«Парадигмы и конструкции языков программирования»**

Выполнил:  
студент группы ИУ5-33Б:  
Буйдина К.А.  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.  
Подпись и дата:

Москва, 2023 г.

## Описание задания:

Разработать программу для решения биквадратного уравнения.

- Программа должна быть разработана в виде консольного приложения на языке Python.
- Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
- Коэффициенты A, B, C 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.
- Если коэффициент A, B, C 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.
- Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
- Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

## Текст программы:

### 1. main\_oop

```
import math
import sys

class Sq_Roots:
    """Класс коэффициентов"""
    def __init__(self):
        self.A = 0.0
        self.B = 0.0
        self.C = 0.0
        self.root_list = set()

    def check_root(self, ind):
        """Проверка коэфов"""
        try:
            global coef
            try:
                coef = float(sys.argv[ind])
            except ValueError:
                print("Ошибка. Попробуйте еще раз")
        except:
            print("Введите коэффициент {} :".format(ind) )
            try:
                coef = input()
            # Проверка A на ноль (иначе впоследствии ошибка - деление на
            ноль)
```

```

        if float(coef) == 0.0 and ind == 1:
            print("Коэффициент 1 равен 0. Так не пойдет")
            self.check_root(1)
        except ValueError:
            self.check_root(ind)
    return float(coef)

def get_roots(self):
    """Получение значений коэфов"""
    self.A = self.check_root(1)
    self.B = self.check_root(2)
    self.C = self.check_root(3)

def calculation(self):
    """Подсчет корней"""
    #Дискриминант
    all_roots = set()
    D = self.B**2 - 4*self.A*self.C
    if (D>=0):
        self.root_list.add( (-self.B + math.sqrt(D)) / (2*self.A) )
        self.root_list.add( (-self.B - math.sqrt(D)) / (2*self.A) )
    for r in self.root_list:
        if r >= 0:
            all_roots.add(math.sqrt(r))
            all_roots.add(- math.sqrt(r))
    self.root_list = all_roots

def print_ans(self):
    """Вывод ответов"""
    if len(self.root_list)==0:
        print("Нет корней, дискриминант меньше нуля :(")
        return
    print("Корни:")
    for e in self.root_list:
        print(e, " ")

def main():
    """Основная функция"""
    sr = Sq_Roots()
    sr.get_roots()
    sr.calculation()
    sr.print_ans()

if __name__ == "__main__":
    main()

```

## 2. main\_proc

```

import math
import sys

def check_root(ind):
    """Проверка коэфов"""
    try:
        global coef
        try:
            coef = float(sys.argv[ind])
        except ValueError:
            print("Ошибка. Попробуйте еще раз")
    except:
        print("Введите коэффициент {} :".format(ind) )
        try:
            coef = input()

```

```

        # Проверка A на ноль (иначе впоследствии ошибка - деление на ноль)
        if float(coef) == 0.0 and ind == 1:
            print("Коэффициент 1 равен 0. Так не пойдет")
            check_root(1)
        except ValueError:
            check_root(ind)
    return float(coef)

def get_roots(coef_list):
    """Получение значений коэфов -> возвращает список коэфов"""
    for i in range(0,3):
        coef_list[i] = check_root(i+1)
    return coef_list

def calculation(coef_list):
    """Подсчет корней -> возвращает множество корней"""
    A = coef_list[0]
    B = coef_list[1]
    C = coef_list[2]
    #Дискриминант
    D = B**2 - 4*A*C
    #Корни
    root_list = set()
    all_roots = set()
    if (D >= 0.0):
        root_list.add( (-B + math.sqrt(D)) / (2.0 * A) )
        root_list.add( (-B - math.sqrt(D)) / (2.0 * A) )
    for r in root_list:
        if r >= 0:
            all_roots.add(math.sqrt(r))
            all_roots.add(- math.sqrt(r))
    return all_roots

def print_ans(root_list):
    """Вывод ответов"""
    if len(root_list) == 0:
        print("Нет корней, дискриминант меньше нуля :(")
        return
    print("Корни:")
    for e in root_list:
        print(e, " ")

def all_process(coef_list):
    """Запускает последовательно все функции"""
    print_ans(calculation(get_roots(coef_list)))

def main():
    """Основная функция"""
    coef_list = [0.0]*3
    all_process(coef_list)

if __name__ == "__main__":
    main()

```

Экранные формы с примерами выполнения программы:

1. Пример преподавателя

```
Введите коэффициент 1 :  
1  
Введите коэффициент 2 :  
0  
Введите коэффициент 3 :  
-4  
Корни:  
2.0  
-2.0  
Process finished with exit code 0
```

## 2. Пример с отсутствием корней

```
Введите коэффициент 1 :  
4  
Введите коэффициент 2 :  
0  
Введите коэффициент 3 :  
10  
Нет корней, дискриминант меньше нуля :(  
Process finished with exit code 0
```

## 3. Пример, при котором первый коэффициент вводится равным 0 (при дальнейшей работе программы это привело бы к делению на 0)

```
Введите коэффициент 1 :  
0  
Коэффициент 1 равен 0. Введите отличное от 0 число.  
Введите коэффициент 1 :  
3  
Введите коэффициент 2 :  
16  
Введите коэффициент 3 :  
0  
Корни:  
0.0  
-5.333333333333333  
Process finished with exit code 0
```

## 4. Пример ввода иных символов

Введите коэффициент 1 :

bebebe

Введите коэффициент 1 :

1

Введите коэффициент 2 :

-5

Введите коэффициент 3 :

6

Корни:

1.4142135623730951

-1.7320508075688772

1.7320508075688772

-1.4142135623730951

Process finished with exit code 0

Введите коэффициент 1 :

:(

Введите коэффициент 1 :

1

Введите коэффициент 2 :

-17

Введите коэффициент 3 :

16

Корни:

1.0

4.0

-4.0

-1.0

Process finished with exit code 0