

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

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

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1
«Решение биквадратного уравнения»

Выполнил:
студент группы ИУ5-35Б
Ходырев Роман
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий Евгеньевич
Подпись и дата:

Постановка задачи

Научиться решать биквадратные уравнения на языке Python. Привести два решения. Первый основан на процедурном подходе, второй - на объектно-ориентированном подходе.

Разработать возможность ввода коэффициентов биквадратного уравнения через командную строку. Если коэффициенты не были переданы на вход программе в качестве аргументов командной строки, их следует вводить через консоль.

Обработать возможные виды исключений: ввод вместо числа строки, недостаточный ввод коэффициентов, деление на 0, ввод первого коэффициента равным нулю, что сводит биквадратное уравнение к линейному и др.

Разработка алгоритма

Описание алгоритма

- 1)Запускаем программу через командную строку и передаем ей коэффициенты биквадратного уравнения
- 2)Обрабатываем коэффициенты или вызываем исключения
- 3)Сохраняем вычисленные корни биквадратного уравнения в списке
- 4) Выводим их на экран

Входные данные

- int a, b, c – коэффициенты биквадратного уравнения

Результат:

- list results – список из корней биквадратного уравнения

Текст программы

main.py – решение процедурным подходом

```
import sys
import math
import time

KOEFS = {
    1: 'a',
    2: 'b',
    3: 'c'
}

#Получаем коэффициенты из командной строки или считываем их из консоли
def get_koef(ind, prompt):
    try:
        coef = int(sys.argv[ind])
    except:
        print(f"Введите коэффициент {prompt.upper()}: ", end="")
        coef = ""
        while type(coef) != int:
            coef = input()
            try:
                coef = int(coef)
            except:
                print("Неверный ввод! Повторите попытку: ", end="")
    return coef
```

```

#Получаем корни биквадратного уравнения
def get_roots(a,b,c):
    roots = []
    D = float(b**2-4*a*c)
    if D > 0.0:
        rt_1 = (-b+math.sqrt(D))/(2*a)
        rt_2 = (-b-math.sqrt(D))/(2*a)
        roots.extend([rt_1, rt_2])
    elif D==0.0:
        rt_1 = (-b)/(2*a)
        roots.append(rt_1)
    return roots

#Если A=0, то решаем линейное уравнение
def unbiquadarate(b,c):
    roots = [(-c)/b]
    return roots

def main():
    a,b,c = [get_koef(i, KOEFS[i]) for i in range(1,4)]
    if a!=0:
        roots = get_roots(a,b,c)
    elif b!=0:
        roots = unbiquadarate(b,c)
    elif c!=0:
        roots = []
    else:
        print("x - любое")
        time.sleep(10)
        return

    #Обработка результата
    print(f"Введенное уравнение: {a}x^2+{b}x+{c}=0")
    if len(roots)!=0:
        if len(roots) == 2:
            print("Два действительных корня")
            print(f"Первый корень: {roots[0]}")
            print(f"Второй корень: {roots[1]}")
        else:
            print(f"Один действительный корень: {roots[0]}")
    else:
        print("Нет корней")

    time.sleep(10)

if __name__ == "__main__":
    main()

```

main_oop.py – решение объектно-ориентированным подходом

```
import sys
import math
import time

KOEFS = {
    1: 'a',
    2: 'b',
    3: 'c'
}

class Equation:
    #Конструктор
    def __init__(self):
        self.coef_A = 0.0
        self.coef_B = 0.0
        self.coef_C = 0.0

        self.roots = []

    #Получение коэффициентов
    def get_coef(self, ind, prompt):
        try:
            coef = int(sys.argv[ind])
        except:
            print(f"Введите коэффициент {prompt.upper()}: ", end="")
            coef = ""
            while type(coef) != int:
                coef = input()
                try:
                    coef = int(coef)
                except:
                    print("Неверный ввод! Повторите попытку: ", end="")
        return coef

    #Присвоение коэффициентов
    def get_coefs(self):
        self.coef_A, self.coef_B, self.coef_C = [self.get_coef(i, KOEFS[i]) for i
in range(1,4)]

    #Вычисление корней
    def calculate(self):
        a = self.coef_A
        b = self.coef_B
        c = self.coef_C

        if a != 0.0:
            D = b**2 - 4*a*c
            if D > 0.0:
                rt_1 = (-b + math.sqrt(D)) / (2*a)
                rt_2 = (-b - math.sqrt(D)) / (2*a)
```

```

        self.roots.extend([rt_1, rt_2])
    elif D==0.0:
        rt_1 = (-b)/(2*a)
        self.roots.append(rt_1)
    elif b!=0.0: #Проверка исключения на линейное уравнение
        self.roots.append(-c/b)

#Вывод корней
def print_roots(self):
    print(f"Введенное уравнение:
{self.coef_A}x^2+{self.coef_B}x+{self.coef_C}=0")
    if len(self.roots)!=0:
        if len(self.roots) == 2:
            print("Два действительных корня")
            print(f"Первый корень: {self.roots[0]}")
            print(f"Второй корень: {self.roots[1]}")
        else:
            print(f"Один действительный корень: {self.roots[0]}")
    elif self.coef_C!=0.0: #Если не равен, то ввели 5=0"
        print("Нет корней")
    else: #Если равен, то ввели 0=0"
        print("x - любое")

def main():
    r = Equation()
    r.get_coefs()
    r.calculate()
    r.print_roots()

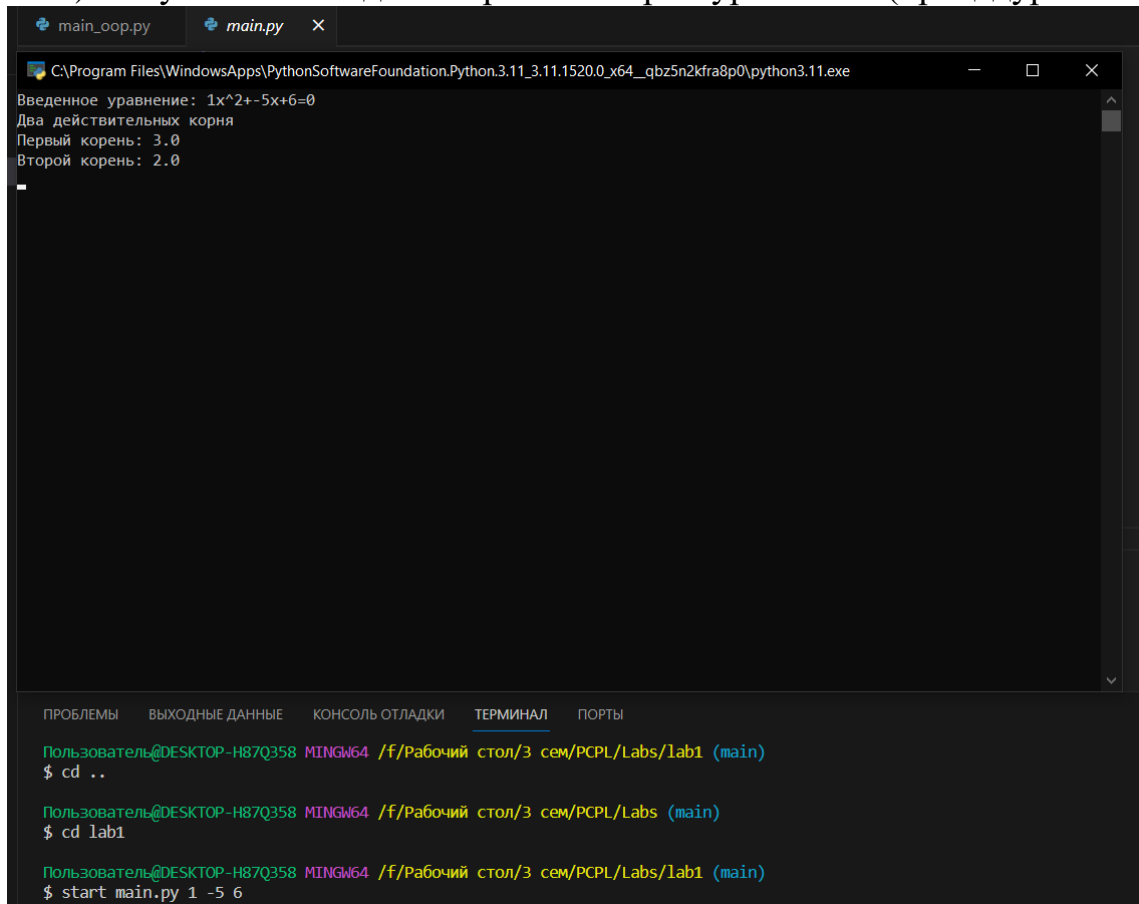
    time.sleep(10)

if __name__ == "__main__":
    main()

```

Анализ результатов

1) Запуск из командной строки и корни уравнения (процедурный подход)



The screenshot shows a Python IDE with two tabs: `main_oop.py` and `main.py`. The active tab `main.py` displays the following output:

```
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11_3.11.1520.0_x64__qbz5n2kfra8p0\python3.11.exe
Введенное уравнение: 1x^2+-5x+6=0
Два действительных корня
Первый корень: 3.0
Второй корень: 2.0
-
```

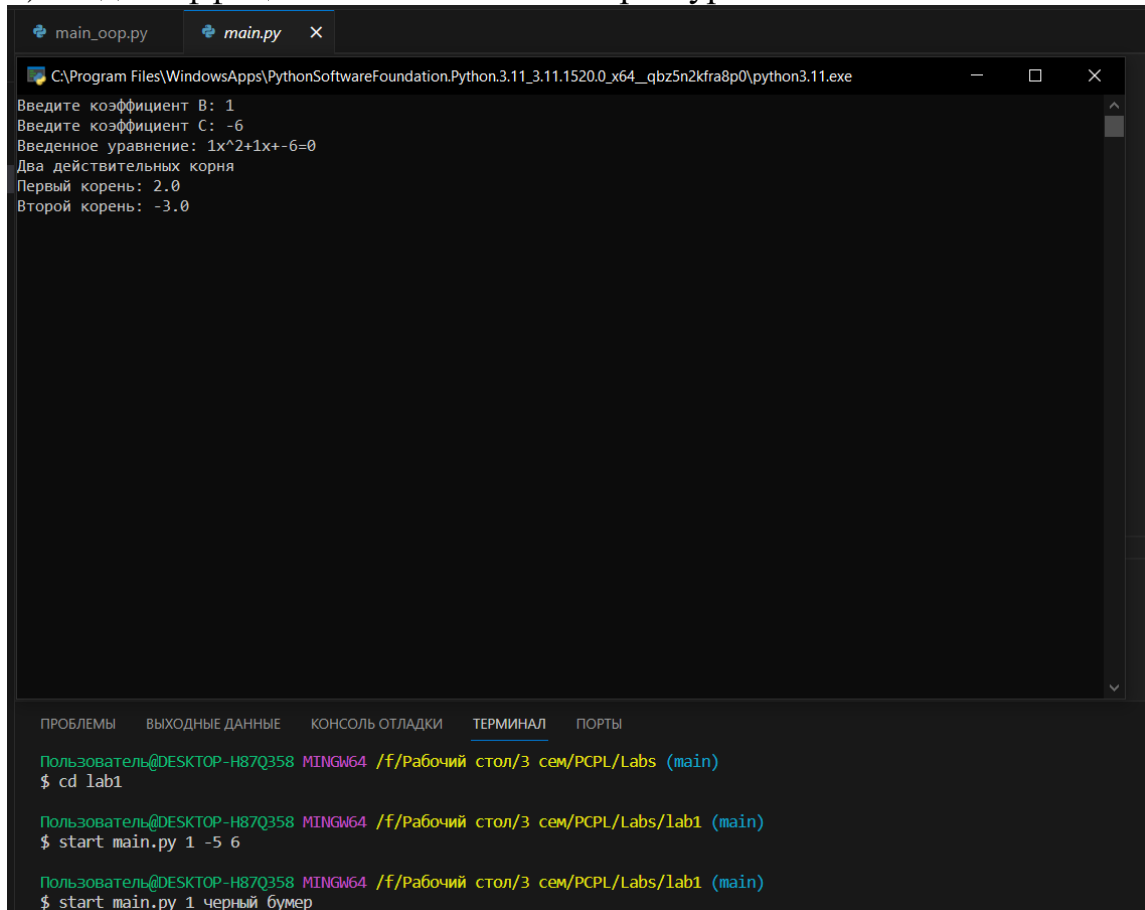
Below the code editor, the terminal window shows the following commands and output:

```
Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ cd ..

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs (main)
$ cd lab1

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main.py 1 -5 6
```

2) Ввод коэффициентов с консоли и корни уравнения



The screenshot shows a Python IDE with two tabs: `main_oop.py` and `main.py`. The active tab `main.py` displays the following output:

```
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11_3.11.1520.0_x64__qbz5n2kfra8p0\python3.11.exe
Введите коэффициент B: 1
Введите коэффициент C: -6
Введенное уравнение: 1x^2+1x+-6=0
Два действительных корня
Первый корень: 2.0
Второй корень: -3.0
```

Below the code editor, the terminal window shows the following commands and output:

```
Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs (main)
$ cd lab1

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main.py 1 -5 6

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main.py 1 черный бумер
```

3) Ввод из командной строки и корни уравнения (ООП)

```
main_oop.py X main.py
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11.1520.0_x64__qbz5n2kfra8p0\python3.11.exe
Введенное уравнение: 1x^2+-12x+41=0
Нет корней

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main.py 1 -5 6

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main.py 1 черный бумер

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main_oop.py 1 -12 41
```

4) Ввод с консоли и корни уравнения

```
main_oop.py X main.py
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11.1520.0_x64__qbz5n2kfra8p0\python3.11.exe
Введите коэффициент B: -7
Введите коэффициент C: 1
Введенное уравнение: 2x^2+-7x+1=0
Два действительных корня
Первый корень: 3.350781059358212
Второй корень: 0.14921894064178787

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main.py 1 черный бумер

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main_oop.py 1 -12 41

Пользователь@DESKTOP-H87Q358 MINGW64 /f/Рабочий стол/3 сем/PCPL/Labs/lab1 (main)
$ start main_oop.py 2 интер %%
```

Вывод

Я изучил основы языка Python, разработав решения биквадратного уравнения, пользуясь двумя подходами – процедурным и объектно-ориентированным