

**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
«Основные конструкции языка Python»

Выполнил:

студент группы ИУ5-34Б

Чернев Николай
Андреевич

Проверил:

преподаватель каф. ИУ5

Нардид Анатолий
Николаевич

Москва, 2024 г.

Постановка задачи

Задание.

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 A , B , C , вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. 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.
4. 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.

Текст программы

Main.py

```
from sys import stdin

from abc import ABC


class Solution(ABC):

    def print_ans(self):

        pass


class NoRoots(Solution):

    def print_ans(self):

        print('No roots')


class OneRoot(Solution):

    def __init__(self, x):

        self.root = x


    def print_ans(self):

        print(f'One root: {self.root}')


class TwoRoots(Solution):

    def __init__(self, x1, x2):

        self.root1 = x1

        self.root2 = x2


    def print_ans(self):

        print(f'Two roots: {self.root1}; {self.root2}')


def solve(a, b, c) -> Solution:

    d = b**2 - 4 * a * c

    if d < 0:

        ans = NoRoots()

    elif d == 0:

        ans = OneRoot(-b / 2 / a)

    else:

        ans = TwoRoots((-b + d**0.5) / 2 / a, (-b - d**0.5) / 2 / a)

    return ans
```

```
def main():
    coefs = []
    for i in stdin:
        coefs.append(int(i))
        if len(coefs) >= 3:
            break
    ans = solve(*coefs)
    ans.print_ans()

if __name__ == "__main__":
    main()
```

Анализ результатов

```
(.venv) (base) nachernev@MacBook-Air-Nickolay sem3labs % python lab1.py
10
-10
1
Two roots: 0.8872983346207416; 0.1127016653792583
(.venv) (base) nachernev@MacBook-Air-Nickolay sem3labs % python lab1.py
1
2
3
No roots
(.venv) (base) nachernev@MacBook-Air-Nickolay sem3labs % python lab1.py
1
-3
2
Two roots: 2.0; 1.0
(.venv) (base) nachernev@MacBook-Air-Nickolay sem3labs % python lab1.py
1
-2
1
One root: 1.0
```