

**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

Байдаков Владислав

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
```

```
class Solution:
```

```
    def __init__(self):
```

```
        pass
```

```
class OneRoot(Solution):
```

```
    def __init__(self, x):
```

```
        self.root = x
```

```
    def RootPrint(self):
```

```
        print(f'one root {self.root}')
```

```
class NoRoots(Solution):
```

```
    def RootPrint(self):
```

```
        print('no roots')
```

```
class TwoRoots(Solution):
```

```
    def __init__(self, x1, x2):
```

```
        self.root1 = x1
```

```
        self.root2 = x2
```

```
    def RootPrint(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():

    a = []

    for i in stdin:

        a.append(int(i))

        if len(a)>=3:

            break

    ans = Solve(a[0],a[1],a[2])

    ans.RootPrint()

if __name__ == "__main__":

    main()

```

Анализ результатов

```

• kusneid@dsfont ~/M/y/n/PCPL (main)> python lab1/lab1.py
1
-18
81
one root 9.0
• kusneid@dsfont ~/M/y/n/PCPL (main)> python lab1/lab1.py
1
-8
20
no roots
• kusneid@dsfont ~/M/y/n/PCPL (main)> python lab1/lab1.py
256
-32
1
one root 0.0625
• kusneid@dsfont ~/M/y/n/PCPL (main)> python lab1/lab1.py
5
-2
1
no roots
• kusneid@dsfont ~/M/y/n/PCPL (main)> python lab1/lab1.py
1
-9
0
two roots 9.0; 0.0
• kusneid@dsfont ~/M/y/n/PCPL (main)> python lab1/lab1.py
1
10
9
two roots -1.0; -9.0

```