

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по лабораторной работе № 1  
“Основные конструкции языка Python”

**Выполнил:**  
Студент группы ИУ5-36Б  
Пингин Е.В.

*Пингин*

**Преподаватель:**  
Нардид. А.Н.

Москва 2025

## 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.

### main.py

```
import sys
import math

class EquationSolver:
    def __init__(self):
        self.a = self.getCoef(1)
        self.b = self.getCoef(2)
        self.c = self.getCoef(3)
        self.roots = set()

    def getCoef(self, pos):
        try:
            coef = float(sys.argv[pos])
        except Exception:
            flag = True
            while flag:
                print(f'Введите коэффициент {chr(ord("A") + pos - 1)}: ', end=' ')
                try:
                    coef = float(input())
                except ValueError:
                    print('Неверный коэффициент! Попробуйте ещё раз.')
                    continue
                flag = False
        return coef

    def calculateRoots(self):
```

```

if self.a == self.b == self.c == 0:
    return

if self.a == 0:
    if self.b == 0:
        return

    if self.c * self.b > 0:
        return

    self.roots.add(math.sqrt(abs(self.c / self.b)))
    self.roots.add(-math.sqrt(abs(self.c / self.b)))
    return

D = self.b * self.b - 4 * self.a * self.c
if D >= 0:
    t1, t2 = (-self.b + math.sqrt(D)) / 2 / self.a,
              (-self.b - math.sqrt(D)) / 2 / self.a

    if t1 >= 0:
        self.roots.add(math.sqrt(t1))
        self.roots.add(-math.sqrt(t1))

    if t2 >= 0:
        self.roots.add(math.sqrt(t2))
        self.roots.add(-math.sqrt(t2))

def printRoots(self):
    if len(self.roots) == 0:
        if self.a == self.b == self.c == 0:
            print('Бесконечное количество решений!')
            return

        print('Нет действительных корней!')
        return

    for i, x in enumerate(sorted(self.roots)):
        print(f'x{i} = {x}')

```

  
  

```

if __name__ == '__main__':
    eqsolver = EquationSolver()
    eqsolver.calculateRoots()
    eqsolver.printRoots()

```

- (venv) `egorp@LAPTOP-JS7VVS09:~/python/Labs3Sem/Lab1$` python main.py 1 -2 1  
x0 = -1.0  
x1 = 1.0
- (venv) `egorp@LAPTOP-JS7VVS09:~/python/Labs3Sem/Lab1$` python main.py  
Введите коэффициент A: asd  
Неверный коэффициент! Попробуйте ещё раз.  
Введите коэффициент A: 1  
Введите коэффициент B: -5  
Введите коэффициент C: 4  
x0 = -2.0  
x1 = -1.0  
x2 = 1.0  
x3 = 2.0
- (venv) `egorp@LAPTOP-JS7VVS09:~/python/Labs3Sem/Lab1$` python main.py 0 0 0  
Бесконечное количество решений!
- (venv) `egorp@LAPTOP-JS7VVS09:~/python/Labs3Sem/Lab1$`
- (venv) `egorp@LAPTOP-JS7VVS09:~/python/Labs3Sem/Lab1$` python main.py 0 0 1  
Нет действительных корней!