

**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 ИУ5-31Б

Шибанова Варвара
Подпись и дата:

Проверил:
преподаватель каф. ИУ5

Подпись и дата:

Москва, 2024 г

Цель лабораторной работы: изучение основных конструкций языка Python.

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 А,В,С, вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки (вариант задания параметров приведен в конце файла с примером кода (https://github.com/ugapanyuk/BKIT_2022/blob/main/code/lab1_code)). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки. (<https://realpython.com/python-command-line-arguments/#the-command-line-interface>)
4. Если коэффициент А, В, С 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.
5. Дополнительное задание 1 (*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.

Текст программы:

(с применением процедурной парадигмы)

```
import sys
import math

def get_coef(index, prompt):
    try:
        coef_str = sys.argv[index]
    except:
        print(prompt)
        coef_str = input()
    coef = float(coef_str)
    return coef

def solve(a, b, c):
    result = []
    D = b**2-4*a*c

    if D<0:
```

```

        print('нет решений')
        return []
    elif D==0:
        root = -b/2*a
        result.append(root)
    elif D>0:
        root1 = (-b-math.sqrt(D))/2*a
        root2 = (-b+math.sqrt(D))/2*a
        result.append(root1)
        result.append(root2)
    return result

def main():
    a = get_coef(1, 'A')
    b = get_coef(2, 'B')
    c = get_coef(3, 'C')

    roots = solve(a,b,c)

    len_roots = len(roots)

    if len_roots == 0:
        print("/")
    elif len_roots ==1:
        print('Один корень: x = {}'.format(roots[0]))

    elif len_roots == 2:
        print('Два корня: x1 = {}, x2 = {}'.format(roots[0], roots[1]))

if __name__ == '__main__':
    main()

```

(с применением объектно-ориентированной парадигмы)

```

import sys
import math

class Biquadratic:
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c
        self.roots = []

    def solve(self):
        discriminant = self.b**2 - 4 * self.a * self.c

        if discriminant < 0:
            print("Уравнение не имеет действительных корней.")
        elif discriminant == 0:

```

```

        y = -self.b / (2 * self.a)
        if y >= 0:
            self.roots = [(-y)**0.5, (y)**0.5]
        else:
            print("Уравнение не имеет действительных корней.")
    elif discriminant > 0:
        y1 = (-self.b + discriminant**0.5) / (2 * self.a)
        y2 = (-self.b - discriminant**0.5) / (2 * self.a)

        if y1 >= 0:
            self.roots.append((-y1)**0.5)
            self.roots.append((y1)**0.5)
        if y2 >= 0:
            self.roots.append((-y2)**0.5)
            self.roots.append((y2)**0.5)

    def print_roots(self):
        if self.roots:
            print("Действительные корни уравнения:")
            for root in self.roots:
                print(root)

def get_coefficient(name):
    while True:
        try:
            value = float(input(f"Введите коэффициент {name}: "))
            return value
        except ValueError:
            print("Некорректный ввод. Пожалуйста, введите число.")

if len(sys.argv) > 1:
    try:
        a = float(sys.argv[1])
        b = float(sys.argv[2])
        c = float(sys.argv[3])
    except (ValueError, IndexError):
        print("Некорректные параметры командной строки.")
        a = get_coefficient("A")
        b = get_coefficient("B")
        c = get_coefficient("C")
    else:
        a = get_coefficient("A")
        b = get_coefficient("B")
        c = get_coefficient("C")

equation = Biquadratic(a, b, c)
equation.solve()
equation.print_roots()

```

Вывод результатов:

```
PS C:\Users\VARVARA\Desktop\Парадигмы в конструкции языков программирования\Labs.py> python lab1_1.py
A
11
B
6
C
7
нет решений
/
PS C:\Users\VARVARA\Desktop\Парадигмы в конструкции языков программирования\Labs.py>

PS C:\Users\VARVARA\Desktop\Парадигмы в конструкции языков программирования\Labs.py> python lab1_2.py
Введите коэффициент A: 1
Введите коэффициент B: -5
Введите коэффициент C: 6
Действительные корни уравнения:
(1.0605752387249068e-16+1.7320508075688772j)
1.7320508075688772
(8.659560562354934e-17+1.4142135623730951j)
1.4142135623730951
PS C:\Users\VARVARA\Desktop\Парадигмы в конструкции языков программирования\Labs.py>
```