

Защищено:  
Гапанюк Ю.Е.

"\_\_" \_\_\_\_\_ 2025 г.

Демонстрация:  
Константинов А.А.

"\_\_" \_\_\_\_\_ 2025 г.

**Отчет по лабораторной работе № 1 по курсу  
Парадигмы и конструкции языков программирования**

**Тема работы: " Основные конструкции языка Python"**

5

(количество листов)

студент группы ИУ5Ц-53Б

Константинов А.А.

\_\_\_\_\_  
(подпись)

"\_\_" \_\_\_\_\_ 2025 г.

## 1. Описание задания

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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

## 2. Листинг программы

### Часть 1

#### Процедурная версия

```
import sys
import math

def input_coefficient(name, args, index):
    """Получает коэффициент (из командной строки или с клавиатуры) с проверкой
    корректности."""
    while True:
        try:
            value = args[index] if len(args) > index else input(f"Введите
коэффициент {name}: ")
            return float(value)
        except ValueError:
            print(f"Ошибка: коэффициент {name} должен быть числом. Повторите ввод.")

def solve_biquadratic(a, b, c):
    """Решает биквадратное уравнение  $Ax^4 + Bx^2 + C = 0$ """
    print(f"\nРешаем уравнение: ({a})x4 + ({b})x2 + ({c}) = 0")

    if a == 0:
        print("Ошибка: коэффициент A не может быть равен 0 (это не биквадратное
уравнение).")
        return []

    D = b**2 - 4*a*c
    print(f"Дискриминант D = {D:.4f}")

    if D < 0:
        print("Действительных корней нет (дискриминант отрицателен).")
        return []

    y1 = (-b + math.sqrt(D)) / (2*a)
    y2 = (-b - math.sqrt(D)) / (2*a)
    print(f"Промежуточные значения: y1 = {y1:.4f}, y2 = {y2:.4f}")

    roots = []
    for y in [y1, y2]:
        if y > 0:
            roots.extend([math.sqrt(y), -math.sqrt(y)])
        elif y == 0:
            roots.append(0.0)

    if roots:
        roots = sorted(roots)
        print("Действительные корни уравнения:", ", ".join(f"{r:.4f}" for r in
roots))
    else:
        print("Действительных корней нет (y < 0).")

    return roots
```

```
def main():
    args = sys.argv[1:]
    a = input_coefficient("A", args, 0)
    b = input_coefficient("B", args, 1)
    c = input_coefficient("C", args, 2)

    solve_biquadratic(a, b, c)

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

## Часть 2

### Объектно-ориентированная версия

# Объектно-ориентированная версия решения биквадратного уравнения

```
import sys
import math
```

```
class BiquadraticEquation:
    """Класс для решения биквадратного уравнения  $Ax^4 + Bx^2 + C = 0$ """

    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c

    def discriminant(self):
        """Вычисляет дискриминант  $D = B^2 - 4AC$ """
        return self.b**2 - 4*self.a*self.c

    def solve(self):
        """Решает уравнение и возвращает действительные корни"""
        print(f"\nРешаем уравнение: ({self.a})x^4 + ({self.b})x^2 + ({self.c}) = 0")

        if self.a == 0:
            print("Ошибка: коэффициент A не может быть равен 0 (это не биквадратное уравнение).")
            return []

        D = self.discriminant()
        print(f"Дискриминант D = {D:.4f}")

        if D < 0:
            print("Действительных корней нет (дискриминант отрицателен).")
            return []

        y1 = (-self.b + math.sqrt(D)) / (2*self.a)
        y2 = (-self.b - math.sqrt(D)) / (2*self.a)
```

```

print(f"Промежуточные значения:  $y_1 = \{y1:.4f\}$ ,  $y_2 = \{y2:.4f\}$ ")

roots = []
for y in [y1, y2]:
    if y > 0:
        roots.extend([math.sqrt(y), -math.sqrt(y)])
    elif y == 0:
        roots.append(0.0)

    if roots:
        roots = sorted(roots)
        print("Действительные корни уравнения:", ", ".join(f"{r:.4f}" for r in
roots))
    else:
        print("Действительных корней нет ( $y < 0$ ).")

return roots

def get_coefficient(name, args, index):
    """Получает коэффициент из командной строки или вводом с клавиатуры"""
    while True:
        try:
            value = args[index] if len(args) > index else input(f"Введите
коэффициент {name}: ")
            return float(value)
        except ValueError:
            print(f"Ошибка: коэффициент {name} должен быть числом. Повторите ввод.")

def main():
    args = sys.argv[1:]
    a = get_coefficient("A", args, 0)
    b = get_coefficient("B", args, 1)
    c = get_coefficient("C", args, 2)

    equation = BiquadraticEquation(a, b, c)
    equation.solve()

if __name__ == "__main__":
    main()

```

### 3. Результаты работы программы

#### Часть 1 (Процедурная версия)

```
PS C:\Users\anton\AppData\Local\Programs\Microsoft VS Code> & C:\Users\anton\AppData\Local\Programs\Python\Python314\python.exe "c:/Users/anton/OneDrive/Рабочий стол/Учеба 3курс/ПиКЯП/ЛР1/LR1_Konst (1часть).py"
```

Введите коэффициент A: 1

Введите коэффициент B: -6

Введите коэффициент C: 10

Решаем уравнение:  $(1.0)x^4 + (-6.0)x^2 + (10.0) = 0$

Дискриминант D = -4.0000

Действительных корней нет (дискриминант отрицателен).

```
PS C:\Users\anton\AppData\Local\Programs\Microsoft VS Code> S
```

#### Часть 2 (Объектно-ориентированная версия)

```
PS C:\Users\anton\AppData\Local\Programs\Microsoft VS Code> & C:\Users\anton/OneDrive/Рабочий стол/Учеба 3курс/ПиКЯП/ЛР1/LR1_Konst (2часть).py"
```

Введите коэффициент A: 7

Введите коэффициент B: 4

Введите коэффициент C: -7

Решаем уравнение:  $(7.0)x^4 + (4.0)x^2 + (-7.0) = 0$

Дискриминант D = 212.0000

Промежуточные значения:  $y_1 = 0.7543$ ,  $y_2 = -1.3257$

Действительные корни уравнения: -0.8685, 0.8685

```
PS C:\Users\anton\AppData\Local\Programs\Microsoft VS Code> █
```