

**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» Отчёт по лабораторной
работе №1
«Основные конструкции языка
Python»

Выполнил:
студент группы ИУ5-35Б
Александров Георгий

Проверил: преподаватель каф. ИУ5
Нардид Анатолий Николаевич

Москва, 2025 г.

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++).

Текст программы

Процедурная реализация

оор.py

```
from abc import ABC, abstractmethod
```

```
class Figure(ABC):
    @abstractmethod
    def area(self):
        pass

    @classmethod
    def get_figure_type(cls):
        return cls.figure_type
```

Объектно-ориентированная реализация

procedural.py

```
import sys
```

```
import math
```

```
def get_valid_float(prompt):
    while True:
        try:
            value = float(input(prompt))
            return value
        except ValueError:
            print("Ошибка! Введите число.")

def solve_biquadratic(a, b, c):
    if a == 0:
        print("Коэффициент А не может быть равен 0!")
        return []

    print(f"\nУравнение: {a}*x^4 + {b}*x^2 + {c} = 0")

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

    roots = []

    if D < 0:
        print("Дискриминант отрицательный. Действительных корней нет.")
    elif D == 0:
        t = -b / (2*a)
```

```

if t > 0:
    x1 = math.sqrt(t)
    x2 = -math.sqrt(t)
    roots = [x1, x2]
    print(f"Корни уравнения: x1 = {x1:.4f}, x2 = {x2:.4f}")
elif t == 0:
    roots = [0]
    print("Корень уравнения: x = 0")
else:
    print("Нет действительных корней.")

```

```

else:

```

```

t1 = (-b + math.sqrt(D)) / (2*a)
t2 = (-b - math.sqrt(D)) / (2*a)

```

```

if t1 > 0:
    x1 = math.sqrt(t1)
    x2 = -math.sqrt(t1)
    roots.extend([x1, x2])
elif t1 == 0:
    roots.append(0)

```

```

if t2 > 0:
    x3 = math.sqrt(t2)
    x4 = -math.sqrt(t2)
    if x3 not in roots and -x3 not in roots:
        roots.extend([x3, x4])
elif t2 == 0 and 0 not in roots:
    roots.append(0)

```

```

if roots:
    roots = list(set(roots))
    roots.sort()
    print(f"Найдено {len(roots)} действительных корней:")
    for i, root in enumerate(roots, 1):
        print(f" x {i} = {root:.4f}")
else:
    print("Действительных корней нет.")

```

```

return roots

```

```

def main():
    print("=" * 60)
    print("РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ: A*x^4 + B*x^2 + C = 0")
    print("=" * 60)

```

```
a = b = c = None
```

```
if len(sys.argv) >= 4:
    try:
        a = float(sys.argv[1])
        b = float(sys.argv[2])
        c = float(sys.argv[3])
        print(f"Коэффициенты из командной строки: A={a}, B={b}, C={c}")
    except ValueError:
        print("Ошибка в аргументах. Будет выполнен ввод с клавиатуры.")
        a = b = c = None
```

```
if a is None:
    print("\nВведите коэффициенты биквадратного уравнения:")
    a = get_valid_float("Коэффициент A (не равен 0): ")
    while a == 0:
        print("Коэффициент A не может быть равен 0!")
        a = get_valid_float("Коэффициент A (не равен 0): ")
```

```
if b is None:
    b = get_valid_float("Коэффициент B: ")
```

```
if c is None:
    c = get_valid_float("Коэффициент C: ")
```

```
solve_biquadratic(a, b, c)
```

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

Листинг кода

Процедурная реализация

```
PS C:\Users\MSI-Crosshair16\Documents\лабы\lab_python_fp> & C:/Users/MSI-Crosshair16/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/MSI-Crosshair16/Documents/лабы/lab_python_fp/biquadratic_procedural.py
=====
РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ: A*x^4 + B*x^2 + C = 0
=====

Введите коэффициенты биквадратного уравнения:
Коэффициент A (не равен 0): 1
Коэффициент B: 1
Коэффициент C: 1

Уравнение: 1.0*x^4 + 1.0*x^2 + 1.0 = 0
Дискриминант D = -3.0
Дискриминант отрицательный. Действительных корней нет.
○ PS C:\Users\MSI-Crosshair16\Documents\лабы\lab_python_fp>
```

Объектно-ориентированная реализация

```
PS C:\Users\MSI-Crosshair16\Documents\лабы\lab_python_fp> & C:/Users/MSI-Crosshair16/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/MSI-Crosshair16/Documents/лабы/lab_python_fp/biquadratic_procedural.py
```

```
=====
РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ:  $A \cdot x^4 + B \cdot x^2 + C = 0$ 
=====
```

Введите коэффициенты биквадратного уравнения:

Коэффициент A (не равен 0): 1

Коэффициент B: 33

Коэффициент C: 5

Уравнение: $1.0 \cdot x^4 + 33.0 \cdot x^2 + 5.0 = 0$

Дискриминант D = 1069.0

Действительных корней нет.

```
PS C:\Users\MSI-Crosshair16\Documents\лабы\lab_python_fp>
```