

Постановка задачи

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](#)). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки](#).
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.
6. Дополнительное задание 2 (*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

Текст программы

```
import sys
import math

def get_coefficient(name: str) -> float:
    """Запрашивает у пользователя ввод
    коэффициента."""
    while True:
        try:
```

```
        return float(input(f"Введите коэффициент
{name}: "))
    except ValueError:
        print("Ошибка. Введите действительное
число.")
```

```
def read_coefficient(index: int, name: str) -> float:
    """Читает коэффициент из аргументов
командной строки или запрашивает у
пользователя."""
    try:
        return float(sys.argv[index])
    except (IndexError, ValueError):
        return get_coefficient(name)
```

```
def get_coefficients() -> tuple[float, float, float]:
    """Получает коэффициенты A, B и C."""
    a = read_coefficient(1, "A")
    b = read_coefficient(2, "B")
    c = read_coefficient(3, "C")
    return a, b, c
```

```
def calculate_roots(a: float, b: float, c: float) ->
list[float]:
    """Вычисляет корни квадратного уравнения."""
    if a == 0:
        if b == 0:
            return []
        return [-c / b]
```

```
discriminant = b * b - 4 * a * c
print(f"Дискриминант: {discriminant}")
```

```
roots = []
```

```
if discriminant > 0:
    sqrt_d = math.sqrt(discriminant)
    roots.append((-b + sqrt_d) / (2 * a))
    roots.append((-b - sqrt_d) / (2 * a))
```

```

elif discriminant == 0:
    roots.append(-b / (2 * a))

# Возвращаем корни с учетом извлечения
квадратного корня
result = []
for root in roots:
    if root > 0:
        result.extend([math.sqrt(root), -
math.sqrt(root)])
    elif root == 0:
        result.append(0)

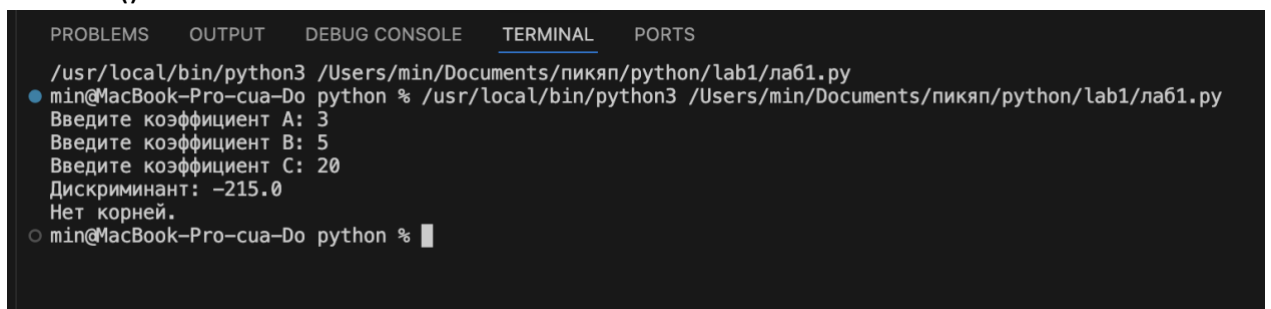
return sorted(result)

def display_roots(roots: list[float]):
    """Выводит корни на экран."""
    count = len(roots)
    if count == 0:
        print("Нет корней.")
    else:
        print(f"{count} {'корень' if count == 1 else
'корня'}: {' '.join(map(str, roots))}")

def main():
    """Главная функция программы."""
    a, b, c = get_coefficients()
    roots = calculate_roots(a, b, c)
    display_roots(roots)

if __name__ == "__main__":
    main()

```



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
/usr/local/bin/python3 /Users/min/Documents/пикяп/python/lab1/лаб1.py
min@MacBook-Pro-cua-Do python % /usr/local/bin/python3 /Users/min/Documents/пикяп/python/lab1/лаб1.py
Введите коэффициент A: 3
Введите коэффициент B: 5
Введите коэффициент C: 20
Дискриминант: -215.0
Нет корней.
min@MacBook-Pro-cua-Do python %

```