

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе № 1
«Основные конструкции языка Python»
Вариант № 5

Выполнил:
студент группы РТ5-31Б
Деревянкина М. О.
Подпись и дата:

Проверил:
к.т.н., доц. каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

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

```
import argparse
import math
import sys

EPS = 1e-12

def is_finite_number(s: str) -> bool:
    try:
        x = float(s)
        return math.isfinite(x)
```

```

except Exception:
    return False

def prompt_float(name: str) -> float:
    """Цикл ввода до корректного значения."""
    while True:
        raw = input(f"Введите коэффициент {name}: ").strip()
        if is_finite_number(raw):
            return float(raw)
        print(f"Некорректный ввод для {name}. Повторите.")

def get_coeffs_from_args_or_prompt() -> tuple[float, float, float]:
    """Парсим -a/-b/-c или позиционные; если что-то некорректно – спрашиваем у
    пользователя."""
    p = argparse.ArgumentParser(
        description="Решение биквадратного уравнения a*x^4 + b*x^2 + c = 0."
    )
    p.add_argument("-a", type=str, help="коэффициент a")
    p.add_argument("-b", type=str, help="коэффициент b")
    p.add_argument("-c", type=str, help="коэффициент c")
    p.add_argument("pos", nargs="*", help="позиционные: a b c")
    args = p.parse_args()

    raws = {"A": None, "B": None, "C": None}

    if len(args.pos) >= 1:
        raws["A"] = args.pos[0]
    if len(args.pos) >= 2:
        raws["B"] = args.pos[1]
    if len(args.pos) >= 3:
        raws["C"] = args.pos[2]

    if raws["A"] is None and args.a is not None:
        raws["A"] = args.a
    if raws["B"] is None and args.b is not None:
        raws["B"] = args.b
    if raws["C"] is None and args.c is not None:
        raws["C"] = args.c

    coeffs = {}
    for name in ("A", "B", "C"):
        if raws[name] is not None and is_finite_number(raws[name]):
            coeffs[name] = float(raws[name])
        else:

            coeffs[name] = prompt_float(name)

    return coeffs["A"], coeffs["B"], coeffs["C"]

def solve_biquadratic(a: float, b: float, c: float):

```

```

"""
Возвращает (discriminant, roots, note).
discriminant: None, если уравнение не квадратно по y=x^2 (т.е. a=0)
roots: отсортированный список действительных корней x
note: текстовое пояснение для вырожденных случаев
"""

roots: list[float] = []

# Случай a ≈ 0: сводится к b*x^2 + c = 0
if abs(a) < EPS:
    if abs(b) < EPS:
        if abs(c) < EPS:
            return None, [], "Бесконечно много решений: любое x удовлетворяет"
            0 = 0."
        else:
            return None, [], "Решений нет: получаем противоречие c = 0 при c"
            ≠ 0."
    # b*x^2 + c = 0 → x^2 = -c/b
    y = -c / b
    if y < -EPS:
        return None, [], "Действительных корней нет: x^2 = отрицательное"
        число."
    if abs(y) < EPS:
        roots = [0.0]
    else:
        s = math.sqrt(y)
        roots = sorted([-s, s])
return None, roots, None

# Обычный случай: a*y^2 + b*y + c = 0, где y = x^2
D = b * b - 4 * a * c

if D < -EPS:
    return D, [], "Действительных корней нет: дискриминант < 0."
if abs(D) < EPS:
    y = -b / (2 * a)
    if y > EPS:
        s = math.sqrt(y)
        roots = sorted([-s, s])
    elif abs(y) <= EPS:
        roots = [0.0]
    else:
        roots = []
return D, roots, None

# D > 0
sqrtD = math.sqrt(D)
y1 = (-b + sqrtD) / (2 * a)
y2 = (-b - sqrtD) / (2 * a)
for y in (y1, y2):
    if y > EPS:

```

```

        s = math.sqrt(y)
        roots.extend([-s, s])
    elif abs(y) <= EPS:
        roots.append(0.0)
roots = sorted(set([+0.0 if abs(x) < EPS else x for x in roots]))
return D, roots, None

def main():
    a, b, c = get_coeffs_from_args_or_prompt()
    print(f"\nРешаем: {a} * x^4 + {b} * x^2 + {c} = 0")

    D, roots, note = solve_biquadratic(a, b, c)

    if D is not None:
        print(f"Дискриминант квадр. по y=x^2: D = {D}")
    if note:
        print(note)

    if roots:
        print("Действительные корни:")
        for i, r in enumerate(roots, 1):
            print(f" x{i} = {r}")
    else:
        print("Действительных корней нет.")

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        print("\nПрервано пользователем.")
        sys.exit(130)

```

Экранные формы с примерами выполнения программы

Ведите коэффициент А: 3
 Ведите коэффициент В: 4
 Ведите коэффициент С: 2

Решаем: $3.0 * x^4 + 4.0 * x^2 + 2.0 = 0$
 Дискриминант квадр. по $y=x^2$: $D = -8.0$
 Действительных корней нет: дискриминант < 0.
 Действительных корней нет.

PS D:\python>

Введите коэффициент A: -5

Введите коэффициент B: 4

Введите коэффициент C: 2

Решаем: $-5.0 * x^4 + 4.0 * x^2 + 2.0 = 0$

Дискриминант квадр. по $y=x^2$: D = 56.0

Действительные корни:

x1 = -1.0716022943960077

x2 = 1.0716022943960077