Московский государственный технический университет им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

«Основные конструкции языка Python»

Выполнил:
студент группы ИУ5 - 32Б:
Васильев Н. Д.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.
Подпись и дата:

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 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.

Код программы:

Файл Lab1.py

```
import sys
import math


def get_coef_procedural(index, prompt):
    while True:
        try:
            if len(sys.argv) > index:
                coef_str = sys.argv[index]
                print(f"Коэффициент из командной строки: {coef_str}")
            else:
                print(prompt)
                coef_str = input()

            return float(coef_str)
        except ValueError:
            print("Ошибка: введите корректное действительное число")
            if len(sys.argv) > index:
                print("Некорректный параметр командной строки. Введите значение
```

```
заново.")


def calculate_discriminant(a, b, c):
    return b * b - 4 * a * c


def solve_quadratic(a, b, c):
    roots = []
    D = calculate_discriminant(a, b, c)

    if D == 0.0:
        roots.append(-b / (2.0 * a))
    elif D > 0.0:
        sqD = math.sqrt(D)
        roots.append((-b + sqD) / (2.0 * a))
        roots.append((-b - sqD) / (2.0 * a))

    return roots


def solve_biquadratic_procedural(a, b, c):
    result = []

    # Особый случай: a = 0
    if a == 0:
        if b == 0:
            if c == 0:
                return ["бесконечно много решений"]
            else:
                return []
        else:
            # Решаем b*x^2 + c = 0
            if -c / b >= 0:
                root = math.sqrt(-c / b)
                result.extend([root, -root])
            return result

    # Решаем относительно y = x^2
    intermediate_roots = solve_quadratic(a, b, c)

    # Для каждого корня y находим корни x
    for y in intermediate_roots:
        if y > 0:
            root_x = math.sqrt(y)
            result.extend([root_x, -root_x])
        elif y == 0:
            result.append(0.0)
```

```
    # Удаляем дубликаты и сортируем
    unique_roots = []
    for root in result:
        if root not in unique_roots:
            unique_roots.append(root)

    unique_roots.sort()
    return unique_roots


def display_roots_procedural(roots):
    if not roots:
        print('Действительных корней нет')
    elif roots[0] == "бесконечно много решений":
        print('Уравнение имеет бесконечно много решений')
    else:
        print(f'Найдено корней: {len(roots)}')
        for i, root in enumerate(roots, 1):
            print(f'Корень {i}: {root:.6f}')


def main_procedural():
    print("=== РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ (процедурный стиль) ===")
    print("Уравнение вида: A*x^4 + B*x^2 + C = 0")

    a = get_coef_procedural(1, 'Введите коэффициент A:')
    b = get_coef_procedural(2, 'Введите коэффициент B:')
    c = get_coef_procedural(3, 'Введите коэффициент C:')

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

    roots = solve_biquadratic_procedural(a, b, c)
    display_roots_procedural(roots)


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

Файл Lab1-1.py

```
import sys
import math


class CoefficientReader:
    def __init__(self):
        self.coefficients = []

    def read_coefficient(self, index, prompt):
        while True:
            try:
                if len(sys.argv) > index:
                    coef_str = sys.argv[index]
                    print(f"Коэффициент из командной строки: {coef_str}")
                else:
                    print(prompt)
                    coef_str = input()

                coefficient = float(coef_str)
                self.coefficients.append(coefficient)
                return coefficient
            except ValueError:
                print("Ошибка: введите корректное действительное число")
                if len(sys.argv) > index:
                    print("Некорректный параметр командной строки. Введите значение
заново.")


class QuadraticSolver:
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c
        self.discriminant = self._calculate_discriminant()

    def _calculate_discriminant(self):
        return self.b * self.b - 4 * self.a * self.c

    def solve(self):
        roots = []

        if self.discriminant == 0.0:
            roots.append(-self.b / (2.0 * self.a))
        elif self.discriminant > 0.0:
            sqD = math.sqrt(self.discriminant)
            roots.append((-self.b + sqD) / (2.0 * self.a))
            roots.append((-self.b - sqD) / (2.0 * self.a))

        return roots


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

    def _solve(self):
```

```
        if self.a == 0:
            self._handle_special_case()
            return

        # Решаем относительно y = x^2
        quadratic_solver = QuadraticSolver(self.a, self.b, self.c)
        intermediate_roots = quadratic_solver.solve()

        # Для каждого корня y находим корни x
        for y in intermediate_roots:
            if y > 0:
                root_x = math.sqrt(y)
                self._add_unique_roots([root_x, -root_x])
            elif y == 0:
                self._add_unique_roots([0.0])

    def _handle_special_case(self):
        if self.b == 0:
            if self.c == 0:
                self.roots = ["бесконечно много решений"]
            else:
                self.roots = []
        else:
            # Решаем b*x^2 + c = 0
            if -self.c / self.b >= 0:
                root = math.sqrt(-self.c / self.b)
                self.roots = [root, -root]

    def _add_unique_roots(self, new_roots):
        for root in new_roots:
            if root not in self.roots:
                self.roots.append(root)
        self.roots.sort()

    def get_roots(self):
        return self.roots

    def get_roots_count(self):
        if not self.roots or self.roots[0] == "бесконечно много решений":
            return 0
        return len(self.roots)

    def display_equation(self):
        return f"{self.a}*x^4 + {self.b}*x^2 + {self.c} = 0"

    def display_solution(self):
        if not self.roots:
            return "Действительных корней нет"
        elif self.roots[0] == "бесконечно много решений":
            return "Уравнение имеет бесконечно много решений"
        else:
            result = f"Найдено корней: {len(self.roots)}\n"
            for i, root in enumerate(self.roots, 1):
                result += f"Корень {i}: {root:.6f}\n"
            return result.strip()


class BiquadraticSolverApp:
    def __init__(self):
        self.coefficient_reader = CoefficientReader()
        self.equation = None

    def run(self):
```

```
        print("=== РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ (ООП стиль) ===")
        print("Уравнение вида: A*x^4 + B*x^2 + C = 0")

        a = self.coefficient_reader.read_coefficient(1, 'Введите коэффициент A:')
        b = self.coefficient_reader.read_coefficient(2, 'Введите коэффициент B:')
        c = self.coefficient_reader.read_coefficient(3, 'Введите коэффициент C:')

        # Создаем и решаем уравнение
        self.equation = BiquadraticEquation(a, b, c)

        print(f"\nУравнение: {self.equation.display_equation()}")
        print(self.equation.display_solution())


def main_oop():
    app = BiquadraticSolverApp()
    app.run()


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

Скриншоты работы:

```
(.venv) PS C:\Users\Николай\Desktop\Python_Course_3Sem\Lab1-3Sem> python Lab1.py
=== РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ (процедурный стиль) ===
Уравнение вида: A*x^4 + B*x^2 + C = 0
Введите коэффициент A:
0
Введите коэффициент B:
0
Введите коэффициент C:
0

Уравнение: 0.0*x^4 + 0.0*x^2 + 0.0 = 0
Уравнение имеет бесконечно много решений
(.venv) PS C:\Users\Николай\Desktop\Python_Course_3Sem\Lab1-3Sem> python Lab1.py
=== РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ (процедурный стиль) ===
Уравнение вида: A*x^4 + B*x^2 + C = 0
Введите коэффициент A:
1
Введите коэффициент B:
3
Введите коэффициент C:
2

Уравнение: 1.0*x^4 + 3.0*x^2 + 2.0 = 0
Действительных корней нет
(.venv) PS C:\Users\Николай\Desktop\Python_Course_3Sem\Lab1-3Sem> python Lab1.py
=== РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ (процедурный стиль) ===
```

```
Действительных корней нет
(.venv) PS C:\Users\Николай\Desktop\Python_Course_3Sem\Lab1-3Sem> python Lab1.py
=== РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ (процедурный стиль) ===
Уравнение вида: A*x^4 + B*x^2 + C = 0
Введите коэффициент A:
4
Введите коэффициент B:
-5
Введите коэффициент C:
1

Уравнение: 4.0*x^4 + -5.0*x^2 + 1.0 = 0
Найдено корней: 4
Корень 1: -1.000000
Корень 2: -0.500000
Корень 3: 0.500000
Корень 4: 1.000000
(.venv) PS C:\Users\Николай\Desktop\Python_Course_3Sem\Lab1-3Sem> python Lab1-1.py
=== РЕШЕНИЕ БИКВАДРАТНОГО УРАВНЕНИЯ (ООП стиль) ===
Уравнение вида: A*x^4 + B*x^2 + C = 0
Введите коэффициент A:
4
Введите коэффициент B:
-5
Введите коэффициент C:
1

Уравнение: 4.0*x^4 + -5.0*x^2 + 1.0 = 0

Уравнение вида: A*x^4 + B*x^2 + C = 0
Введите коэффициент A:
4
Введите коэффициент B:
-5
Введите коэффициент C:
1

Уравнение: 4.0*x^4 + -5.0*x^2 + 1.0 = 0
Найдено корней: 4
Корень 1: -1.000000
Корень 2: -0.500000
Корень 3: 0.500000
Корень 4: 1.000000
(.venv) PS C:\Users\Николай\Desktop\Python_Course_3Sem\Lab1-3Sem>
```