



Отчет

Лабораторная работа № 1

ИСПОЛНИТЕЛЬ:

ФИО: Митрохина А.А.

Группа: ИБМЗ-34Б

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

Тема: Изучение основных конструкций языка Python на примере решения биквадратного уравнения

Цель работы:

Изучение и практическое применение основных конструкций языка Python, включая:

Ввод/вывод данных.

Условные конструкции (if/else).

Циклы (while/for).

Обработку исключений (try/except) для обеспечения корректности ввода.

Работу с аргументами командной строки (sys или argparse).

Реализацию программы в двух парадигмах: **процедурной** и **объектно-ориентированной (ООП)**.

Задание:

Разработать консольное приложение на языке Python для решения биквадратного уравнения вида $Ax^4 + Bx^2 + C = 0$.

Требования:

Обеспечить ввод коэффициентов A, B, C как через **параметры командной строки**, так и с **клавиатуры**.

Реализовать **проверку корректности ввода**: коэффициент должен быть действительным числом. Повторять ввод до получения корректного значения.

Найти все **действительные корни** уравнения в зависимости от дискриминанта.

(Дополнительно) Разработать две версии программы: **процедурную** и **ООП**.

1. Программа в Процедурной Парадигме

1.1. Текст программы

```
# biquad_proc.py
import sys
import math
```

```
def get_float_input(prompt):
    """Надежный ввод действительного числа с клавиатуры."""
    while True:
        try:
            # Использование float() для проверки корректности
            value = float(input(prompt))
            return value
        except ValueError:
            print(" Некорректный ввод. Пожалуйста, введите действительное число.")
```

```
def get_coefficients():
    """Получение коэффициентов A, B, C из командной строки или с клавиатуры."""
```

```
    # 1. Попытка получить из командной строки
    if len(sys.argv) == 4:
        try:
            A = float(sys.argv[1])
            B = float(sys.argv[2])
            C = float(sys.argv[3])
            print(f" Коэффициенты взяты из командной строки: A={A}, B={B}, C={C}")
```

```
        # Проверка, что A не ноль, если взяли из командной строки
        if A == 0:
            print(" Коэффициент A не может быть равен 0 для биквадратного уравнения.")
            return get_coefficients_from_keyboard()
```

```

        return A, B, C
    except ValueError:
        print(" Ошибка: Аргументы командной строки некорректны (не числа).")
        # Переход к вводу с клавиатуры
        return get_coefficients_from_keyboard()

```

```

# 2. Ввод с клавиатуры
else:
    return get_coefficients_from_keyboard()

```

```

def get_coefficients_from_keyboard():
    """Ввод коэффициентов с клавиатуры с проверкой  $A \neq 0$ . """
    print("Ввод коэффициентов с клавиатуры:")

```

```

    # Ввод A с дополнительной проверкой  $A \neq 0$ 
    while True:
        A = get_float_input("Введите коэффициент A ( $A \neq 0$ ): ")
        if A != 0:
            break
        print(" A не может быть равно 0 для биквадратного уравнения.")

```

```

    B = get_float_input("Введите коэффициент B: ")
    C = get_float_input("Введите коэффициент C: ")
    return A, B, C

```

```

def solve_biquadratic(A, B, C):
    """Основная логика решения биквадратного уравнения. """

```

```

    print(f"\nРешение уравнения: {A}x^4 + {B}x^2 + {C} = 0")

```

```

    # Решаем квадратное уравнение  $Ay^2 + By + C = 0$ , где  $y = x^2$ 

```

```

    # 1. Вычисление дискриминанта
    D = B ** 2 - 4 * A * C
    print(f'Дискриминант (D) для  $y = x^2$ :  $D = B^2 - 4AC = {D:.4f}$ ')

```

```

    if D < 0:
        print(" Нет действительных корней ( $D < 0$ ).")
        return []

```

```

    # 2. Вычисление корней  $y$  ( $y = x^2$ )
    sqrt_D = math.sqrt(D)

```

```
y1 = (-B + sqrt_D) / (2 * A)
y2 = (-B - sqrt_D) / (2 * A)
```

```
print(f"Корни для  $y = x^2$ : y1 = {y1:.4f}, y2 = {y2:.4f}")
```

```
# 3. Вычисление действительных корней x ( $x = \pm\sqrt{y}$ )
real_roots = set() # Используем set для автоматического исключения
дубликатов
```

```
for y in [y1, y2]:
    if y >= 0:
        root1 = math.sqrt(y)
        root2 = -math.sqrt(y)
```

```
# Добавляем корни x
real_roots.add(root1)
real_roots.add(root2)
```

```
return sorted(list(real_roots))
```

```
# --- Основная программа ---
if __name__ == "__main__":
    A, B, C = get_coefficients()
```

```
roots = solve_biquadratic(A, B, C)
```

```
print("\n--- Результат ---")
if roots:
    print(f"Уравнение имеет {len(roots)} действительных корня:")
    for root in roots:
        print(f"     $x = {root:.4f}$ ")
else:
    print("Уравнение не имеет действительных корней.")
# biquad_proc.py
import sys
import math
```

```
def get_float_input(prompt):
    """Надежный ввод действительного числа с клавиатуры."""
    while True:
        try:
            # Использование float() для проверки корректности
            value = float(input(prompt))
```

```
        return value
    except ValueError:
        print(" Некорректный ввод. Пожалуйста, введите действительное  
число.")
```

```
def get_coefficients():
    """Получение коэффициентов A, B, C из командной строки или с  
клавиатуры."""
```

```
    # 1. Попытка получить из командной строки
    if len(sys.argv) == 4:
        try:
            A = float(sys.argv[1])
            B = float(sys.argv[2])
            C = float(sys.argv[3])
            print(f" Коэффициенты взяты из командной строки: A={A}, B={B},  
C={C}")
```

```
        # Проверка, что A не ноль, если взяли из командной строки
        if A == 0:
            print(" Коэффициент A не может быть равен 0 для биквадратного  
уравнения.")
            return get_coefficients_from_keyboard()
```

```
        return A, B, C
    except ValueError:
        print(" Ошибка: Аргументы командной строки некорректны (не  
числа).")
        # Переход к вводу с клавиатуры
        return get_coefficients_from_keyboard()
```

```
    # 2. Ввод с клавиатуры
    else:
        return get_coefficients_from_keyboard()
```

```
def get_coefficients_from_keyboard():
    """Ввод коэффициентов с клавиатуры с проверкой A != 0."""
    print("Ввод коэффициентов с клавиатуры:")
```

```
    # Ввод A с дополнительной проверкой A != 0
    while True:
        A = get_float_input("Введите коэффициент A (A ≠ 0): ")
        if A != 0:
```

```
        break
    print(" A не может быть равно 0 для биквадратного уравнения.")
```

```
B = get_float_input("Введите коэффициент B: ")
C = get_float_input("Введите коэффициент C: ")
return A, B, C
```

```
def solve_biquadratic(A, B, C):
    """Основная логика решения биквадратного уравнения."""
```

```
    print(f"\nРешение уравнения: {A}x^4 + {B}x^2 + {C} = 0")
```

```
    # Решаем квадратное уравнение  $Ay^2 + By + C = 0$ , где  $y = x^2$ 
```

```
    # 1. Вычисление дискриминанта
    D = B ** 2 - 4 * A * C
    print(f'Дискриминант (D) для  $y = x^2$ :  $D = B^2 - 4AC = {D:.4f}$ ')

```

```
    if D < 0:
        print(" Нет действительных корней ( $D < 0$ ).")
        return []

```

```
    # 2. Вычисление корней  $y$  ( $y = x^2$ )
    sqrt_D = math.sqrt(D)
    y1 = (-B + sqrt_D) / (2 * A)
    y2 = (-B - sqrt_D) / (2 * A)

```

```
    print(f'Корни для  $y = x^2$ :  $y1 = {y1:.4f}$ ,  $y2 = {y2:.4f}$ ')

```

```
    # 3. Вычисление действительных корней  $x$  ( $x = \pm\sqrt{y}$ )
    real_roots = set() # Используем set для автоматического исключения
    дубликатов

```

```
    for y in [y1, y2]:
        if y >= 0:
            root1 = math.sqrt(y)
            root2 = -math.sqrt(y)

```

```
        # Добавляем корни  $x$ 
        real_roots.add(root1)
        real_roots.add(root2)

```

```
    return sorted(list(real_roots))
```

```
# --- Основная программа ---
if __name__ == "__main__":
    A, B, C = get_coefficients()

    roots = solve_biquadratic(A, B, C)

    print("\n--- Результат ---")
    if roots:
        print(f" Уравнение имеет {len(roots)} действительных корней:")
        for root in roots:
            print(f"    x = {root:.4f}")
    else:
        print(" Уравнение не имеет действительных корней.")
```

1.2. Экранные формы с примерами выполнения

Пример 1: Ввод с клавиатуры (Корни существуют)

```
Ввод коэффициентов с клавиатуры:
Введите коэффициент A (A ≠ 0): 1
Введите коэффициент B: -5
Введите коэффициент C: 4

Решение уравнения: 1.0x^4 + -5.0x^2 + 4.0 = 0
Дискриминант (D) для y = x^2: D = B^2 - 4AC = 9.0000
Корни для y = x^2: y1 = 4.0000, y2 = 1.0000

--- Результат ---
Уравнение имеет 4 действительных корней:
    x = -2.0000
    x = -1.0000
    x = 1.0000
    x = 2.0000

Process finished with exit code 0
```


2. Программа в Объектно-Ориентированной Парадигме (ООП)

2.1. Текст программы (biquad_oop.py)

Данные и логика решения инкапсулированы в **класс** BiquadraticEquation.

```
# biquad_oop.py
import sys
import math

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

    def __init__(self, A, B, C):
        if A == 0:
            raise ValueError("Коэффициент A не может быть равен 0 для биквадратного уравнения.")
        self.A = A
        self.B = B
        self.C = C
        self.real_roots = []

    def calculate_discriminant(self):
        """Вычисляет дискриминант для  $Ay^2 + By + C = 0$ ."""
        return self.B ** 2 - 4 * self.A * self.C

    def solve(self):
        """Решает уравнение и находит действительные корни."""
        D = self.calculate_discriminant()
        print(f"Дискриминант (D) для  $y = x^2$ : D = {D:.4f}")

        if D < 0:
            return

        # Находим корни y ( $y = x^2$ )
        sqrt_D = math.sqrt(D)
        y1 = (-self.B + sqrt_D) / (2 * self.A)
```

```
y2 = (-self.B - sqrt_D) / (2 * self.A)
```

```
print(f"Корни для  $y = x^2$ : y1 = {y1:.4f}, y2 = {y2:.4f}")
```

```
# Находим действительные корни x ( $x = \pm\sqrt{y}$ )
found_roots = set()
for y in [y1, y2]:
    if y >= 0:
        root = math.sqrt(y)
        found_roots.add(root)
        found_roots.add(-root)
```

```
self.real_roots = sorted(list(found_roots))
```

```
def display_results(self):
    """Выводит результаты решения."""
    print(f"\nРешение уравнения: {self.A}x^4 + {self.B}x^2 + {self.C} = 0")
    print("\n--- Результат ---")
    if self.real_roots:
        print(f"Уравнение имеет {len(self.real_roots)} действительных корня:")
        for root in self.real_roots:
            print(f"    x = {root:.4f}")
    else:
        print("Уравнение не имеет действительных корней.")
```

```
# --- Вспомогательные функции для ввода ---
```

```
def get_float_input_robust(prompt):
    """Надежный ввод действительного числа."""
    while True:
        try:
            value = float(input(prompt))
            return value
        except ValueError:
            print("Некорректный ввод. Пожалуйста, введите действительное число.")
```

```
def get_params_and_run():
    """Получает параметры из командной строки или с клавиатуры и
    запускает решение."""
    A, B, C = None, None, None
```

```
# Попытка получить из командной строки
```

```

if len(sys.argv) == 4:
    try:
        A = float(sys.argv[1])
        B = float(sys.argv[2])
        C = float(sys.argv[3])
        print(f" Коэффициенты взяты из командной строки: A={A}, B={B}, C={C}")
    except ValueError:
        print(" Ошибка: Аргументы командной строки некорректны (не числа). Ввод с клавиатуры.")

```

```

# Если не удалось получить из командной строки, или были ошибки,
вводим с клавиатуры
if A is None or B is None or C is None:
    print("Ввод коэффициентов с клавиатуры:")

```

```

# Ввод A с дополнительной проверкой A != 0
while True:
    A = get_float_input_robust("Введите коэффициент A ( $A \neq 0$ ): ")
    if A != 0:
        break
    print(" A не может быть равно 0 для биквадратного уравнения.")

```

```

B = get_float_input_robust("Введите коэффициент B: ")
C = get_float_input_robust("Введите коэффициент C: ")

```

```

try:
    equation = BiquadraticEquation(A, B, C)
    equation.solve()
    equation.display_results()
except ValueError as e:
    print(f"\n--- Ошибка ---")
    print(f"Критическая ошибка: {e}")

```

```

# --- Основная программа ---
if __name__ == "__main__":
    get_params_and_run()

```

2.2. Экранные формы с примерами выполнения

```
Ввод коэффициентов с клавиатуры:
Введите коэффициент A (A ≠ 0): 1
Введите коэффициент B: -4
Введите коэффициент C: 4
Дискриминант (D) для  $y = x^2$ : D = 0.0000
Корни для  $y = x^2$ : y1 = 2.0000, y2 = 2.0000

Решение уравнения:  $1.0x^4 + -4.0x^2 + 4.0 = 0$ 

--- Результат ---
Уравнение имеет 2 действительных корня:
  x = -1.4142
  x = 1.4142

Process finished with exit code 0
```

Вывод

В ходе лабораторной работы были изучены и применены основные конструкции языка Python для решения биквадратного уравнения. Была реализована надежная обработка ввода коэффициентов с использованием циклов `while` и конструкции `try-except` для обеспечения корректности данных. Успешно выполнено дополнительное задание: программа была реализована в двух версиях, демонстрируя применение **процедурной** и **объектно-ориентированной** парадигм программирования. Обе версии корректно находят все действительные корни уравнения.