

Практическое занятие № 4

Тема: составление программ циклической структуры в IDE PyCharm Community. .

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ циклической структуры в IDE PyCharm Community.

Постановка задачи №1.

Дано вещественное число X и целое число $N (> 0)$. Найти

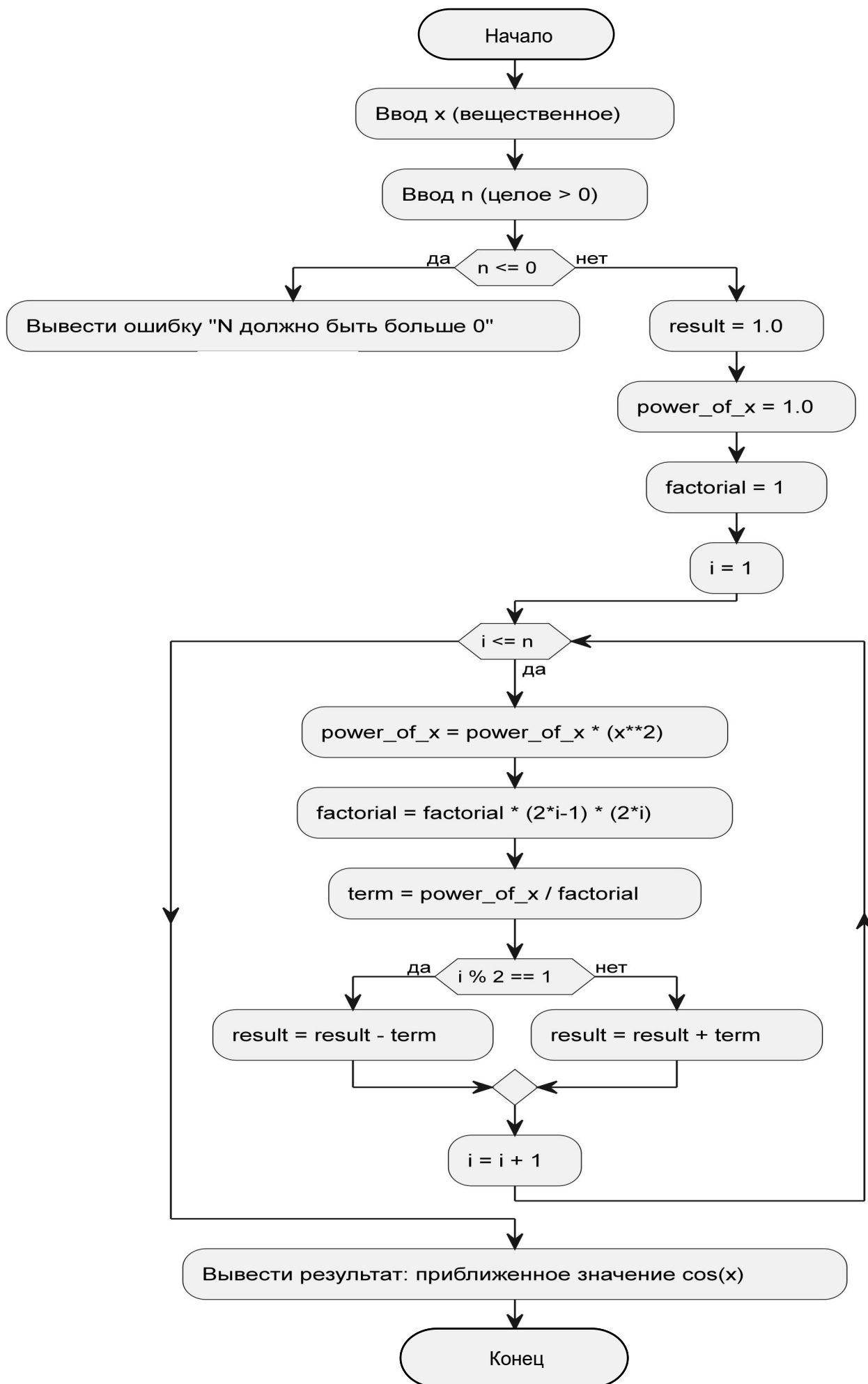
значение выражения $1 - X^2/(2!) + X^4/(4!) - \dots + (-1)^N X^{2N}/(2N!)$

($N! = 1 \cdot 2 \cdot \dots \cdot N$). Полученное число является

приближенным значением функции \cos в точке X

Тип алгоритма: циклический .

Блок-схема алгоритма:



Текст программы:

```
try:
    # Ввод данных
    x = float(input("Введите вещественное число X: "))
    n = int(input("Введите целое число N (> 0): "))

    # Проверка, что N больше 0
    if n <= 0:
        raise ValueError("N должно быть больше 0.")

    # Начальные значения
    result = 1.0 # Начинаем с 1 (первый член ряда)
    power_of_x = 1.0 # x^0 для первого члена
    factorial = 1 # Факториал для 0! (начальное значение)
    i = 1 # Инициализация счетчика

    # Вычисление ряда
    while i <= n:
        # Обновляем степень x^(2*i)
        power_of_x *= x ** 2 # x^(2*i)

        # Обновляем факториал (2*i)!
        factorial *= (2 * i - 1) * (2 * i)

        # Вычисляем текущий член ряда
        term = power_of_x / factorial

        # Чередуем знак
        if i % 2 == 1:
            result -= term # Для нечетных i вычитаем
        else:
            result += term # Для четных i добавляем

        i += 1 # Увеличиваем i на 1 для следующей итерации

    # Вывод результата
    print(f"Приближенное значение cos({x}) с использованием N={n}: {result}")

except ValueError as e:
    print(f"Ошибка ввода: {e}")
```

Протокол работы программы:

Введите вещественное число X: 3

Введите целое число N (> 0): 2

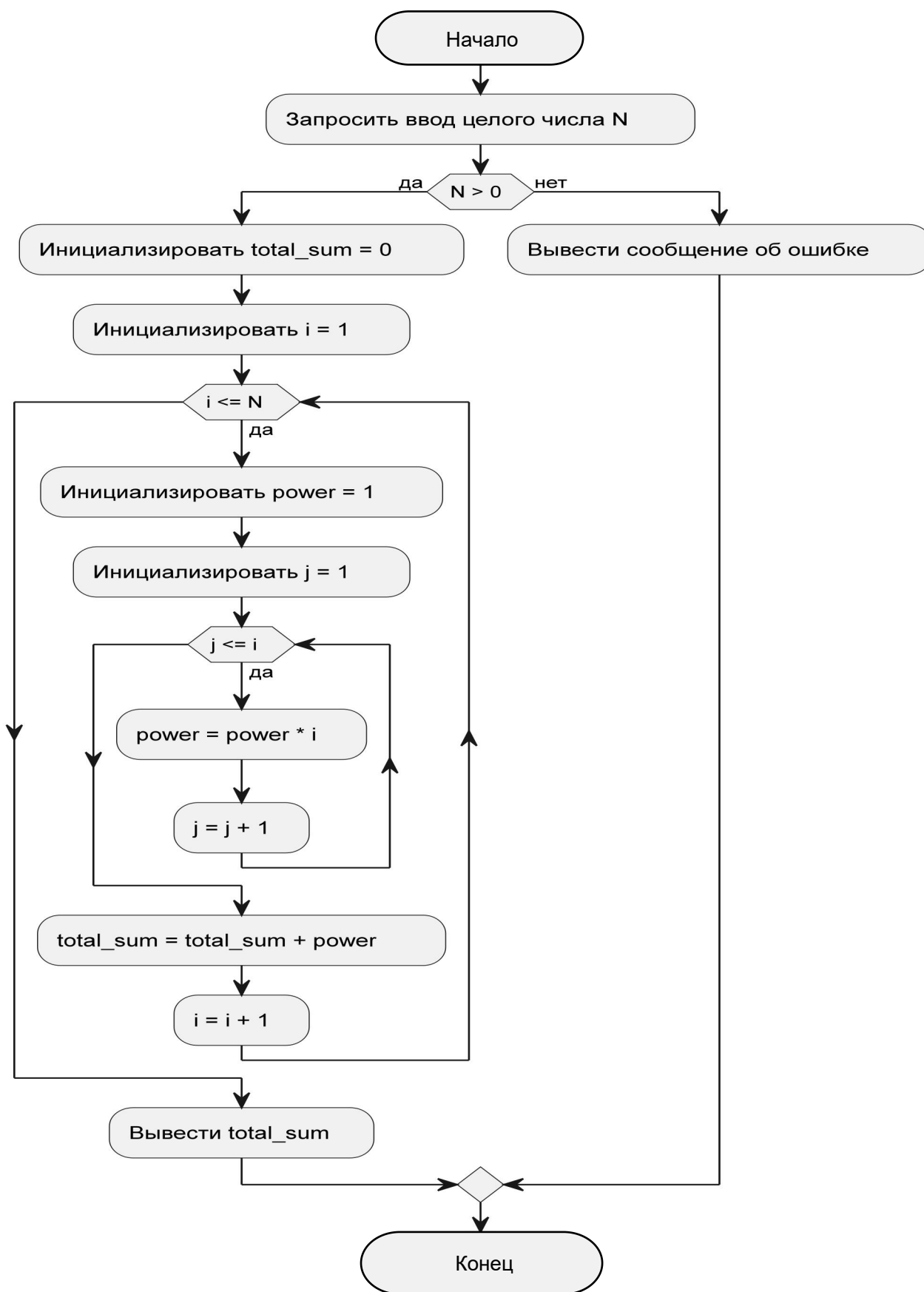
Приближенное значение $\cos(3.0)$ с использованием $N=2$: -0.125

Постановка задачи №2

Дано целое число N (> 0). Найти сумму $1^1 + 2^2 + \dots + N^N$

Тип алгоритма: циклический

Блок-схема алгоритма:



Текст программы:

```
try:
    # Запрашиваем у пользователя ввод целого числа N
    n = int(input("Введите целое число N (> 0): "))

    # Проверка, что N больше 0
    if n <= 0:
        print("Ошибка: N должно быть больше 0.")
    else:
        # Инициализация переменной для хранения суммы
        total_sum = 0
        i = 1 # Инициализация переменной i, с которой начнем вычисления

        # Цикл для вычисления суммы 1^1 + 2^2 + ... + N^N
        while i <= n:
            # Инициализация переменной power для хранения значения i^i
            power = 1
            j = 1 # Инициализация переменной j для подсчета степени

            # Цикл для вычисления i^i
            while j <= i:
                power *= i # Умножаем i на себя j раз
                j += 1 # Увеличиваем j на 1

            # Добавляем i^i к общей сумме
            total_sum += power
            i += 1 # Увеличиваем i на 1 для следующей итерации

        # Вывод результата
        print(f"Сумма 1^1 + 2^2 + ... + {n}^{n} для N={n} равна: {total_sum}")

# Обработка исключений на случай некорректного ввода
except ValueError:
    print("Ошибка: Пожалуйста, введите корректное целое число.")
```

Протокол работы программы:

Введите целое число N (> 0): 3

Сумма $1^1 + 2^2 + \dots + 3^3$ для N=3 равна: 32

Вывод: в процессе выполнения практического занятия выработал(а) навыки составления программ циклической структуры в IDE PyCharm Community. Были использованы языковые конструкции while, if.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.