

Практическое занятие №4

Тема: Составление программ ветвящейся структуры в IDE PyCharm Community.

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

Постановка задачи:

Дано вещественное число X и целое число $N (> 0)$. Найти значение выражения $X - X^3/(3!) + X^5/(5!) - \dots + (-1)^{N-X^2-N+1}/((2-N+1)!)$ ($N! = 12 \dots N$). Полученное число является приближенным значением функции \sin в точке X .

Тип алгоритма: Циклический

Текст программы:

```
#Дано вещественное число X и целое число N (> 0). Найти значение выражения X -
X3/(3!) + X5/(5!) - ... + (-1)N-X2-N+1/((2-N+1)!) (N! = 12 ...N).
# Полученное число является приближенным значением функции sin в точке X.

def mate_sin(x, n):
    sin = 0.0
    sign = 1 # Начальный знак для первого члена ряда
    factorial = 1 # Начальное значение факториала

    for i in range(n):
        if i > 0:
            factorial *= (2 * i) * (2 * i + 1) # Обновляем факториал для (2i + 1)
            sin += sign * (x ** (2 * i + 1)) / factorial
            sign *= -1 # Меняем знак на противоположный

    return sin

# Ввод данных
while True:
    x = input("Введите вещественное число X: ")
    n = input("Введите целое число N (> 0): ")

    try:
        x = float(x)
        n = int(n)

        if n <= 0:
            print("Ошибка ввода: N должно быть больше 0.")
            continue

        break
    except ValueError:
        print("Ошибка ввода: пожалуйста, введите корректные значения.")

# Вычисление и вывод результата
result = mate_sin(x, n)
print(f"Приближенное значение sin({x}) с помощью {n} членов ряда: {result}")
```

Протокол программ:

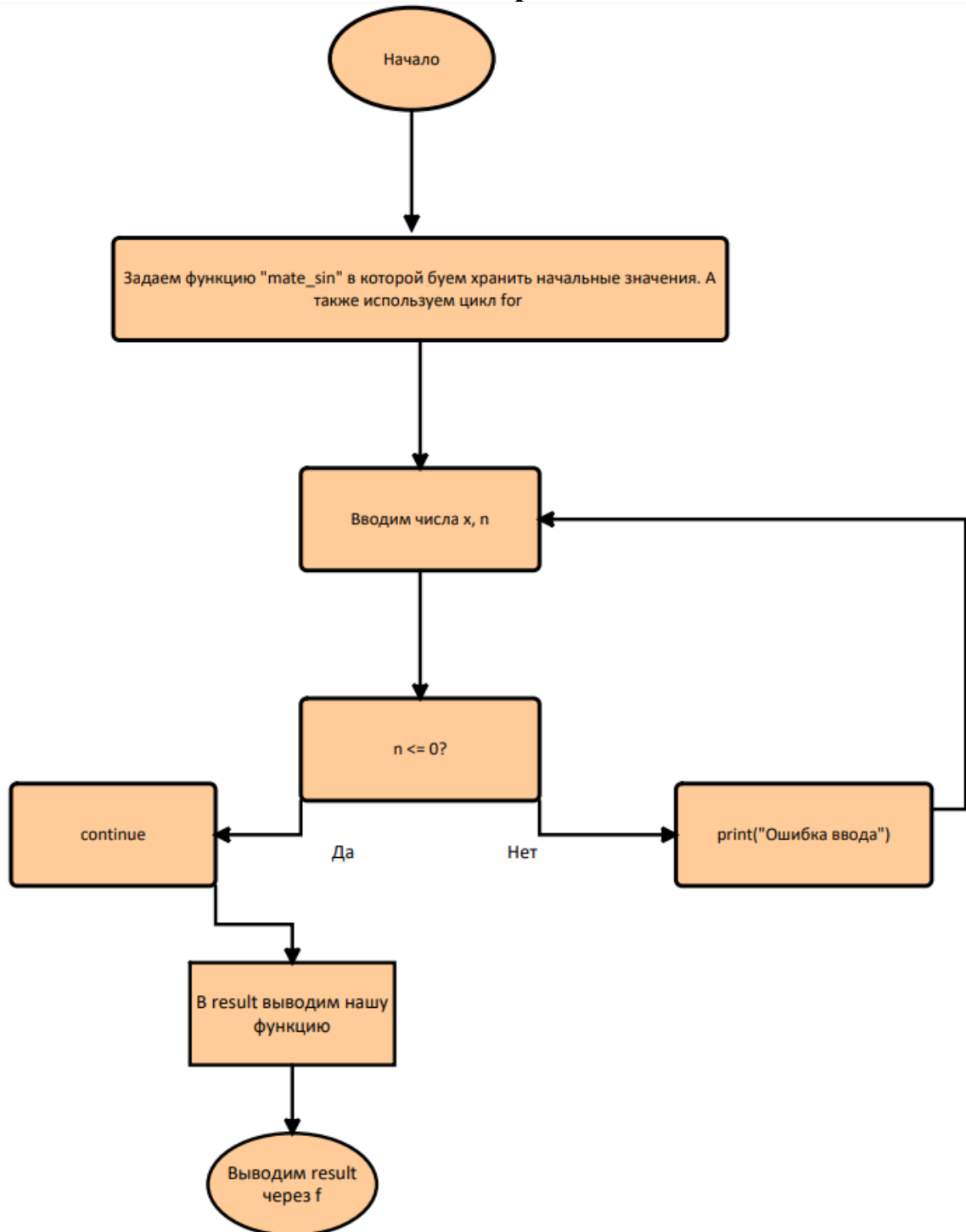
Введите вещественное число X: 6.3

Введите целое число N (> 0): 10

Приближенное значение $\sin(6.3)$ с помощью 10
членов ряда: 0.015705646851907677

Process finished with exit code 0

Блок схема алгоритма:



Постановка задачи:

Дано целое число N (> 0). Найти сумму $1N + 2N-1 + \dots + N1$

Тип алгоритма: Циклический

Текст программы:

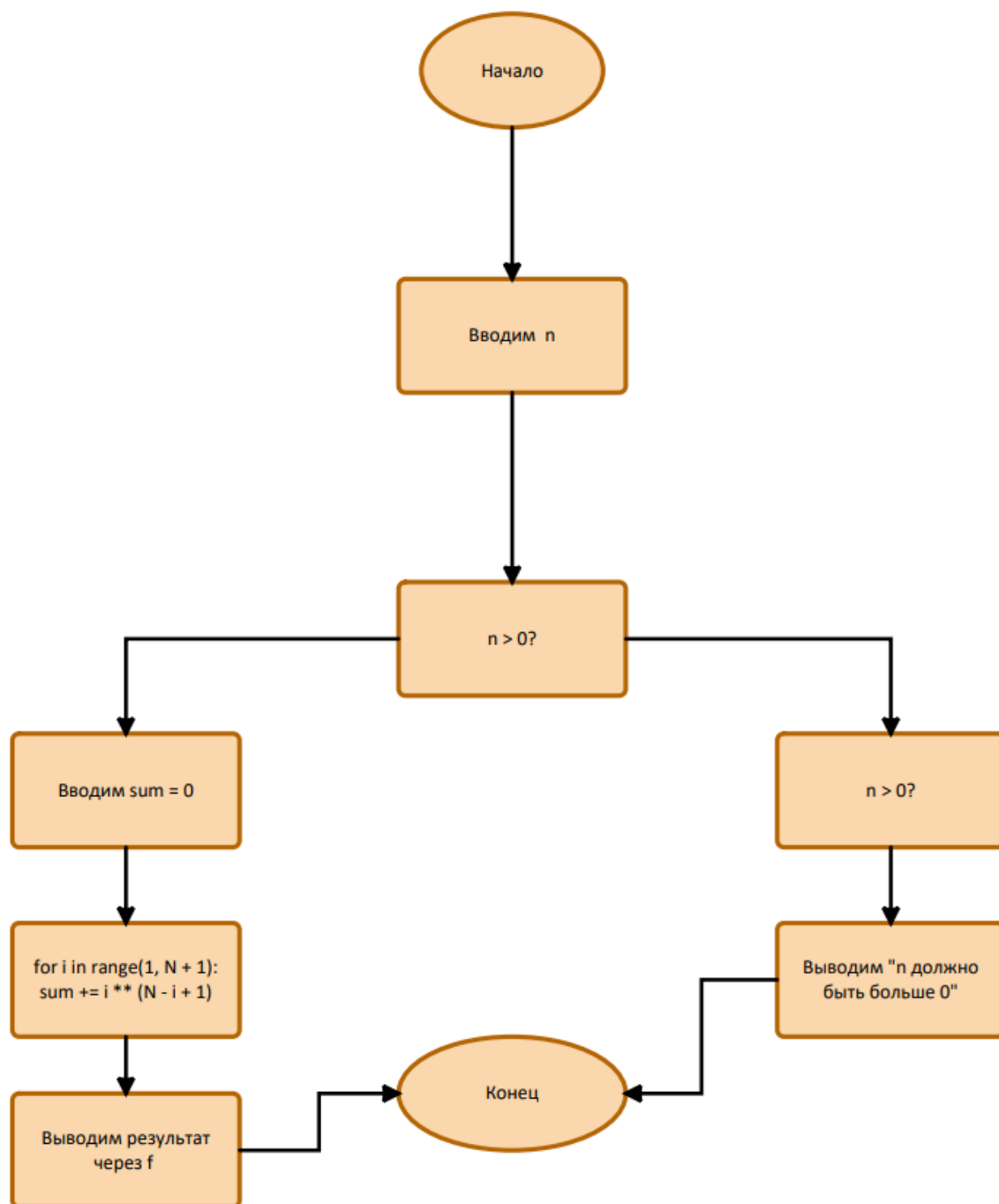
```
#Дано целое число N (> 0). Найти сумму 1N + 2N-1 + ... + N1

while True:
    try:
        N = int(input('Введите целое число N (> 0): '))
        break
    except ValueError:
        print("Число должно быть целым")

if N > 0:
    sum = 0 #Вводим начальное значение суммы
    for i in range(1, N + 1):
        sum += i ** (N - i + 1) # Вычисление текущего члена и добавление к сумме

    print(f"Сумма последовательности для N={N}: {sum}") #Выводим результат через f
else:
    print('Ошибка: N должно быть больше 0.')
```

Блок схема алгоритмов:



Протокол программы:

Введите целое число N (> 0): 5

Сумма последовательности для $N=5$: 65

Process finished with exit code 0

Блок схема алгоритмов:

Вывод:

В процессе работы я закрепил полученные ранее навыки, приобрела новые навыки в использование цикла for научилась создавать программы с использованием циклов в IDE PyCharm Community.

