

Практическое занятие № 4

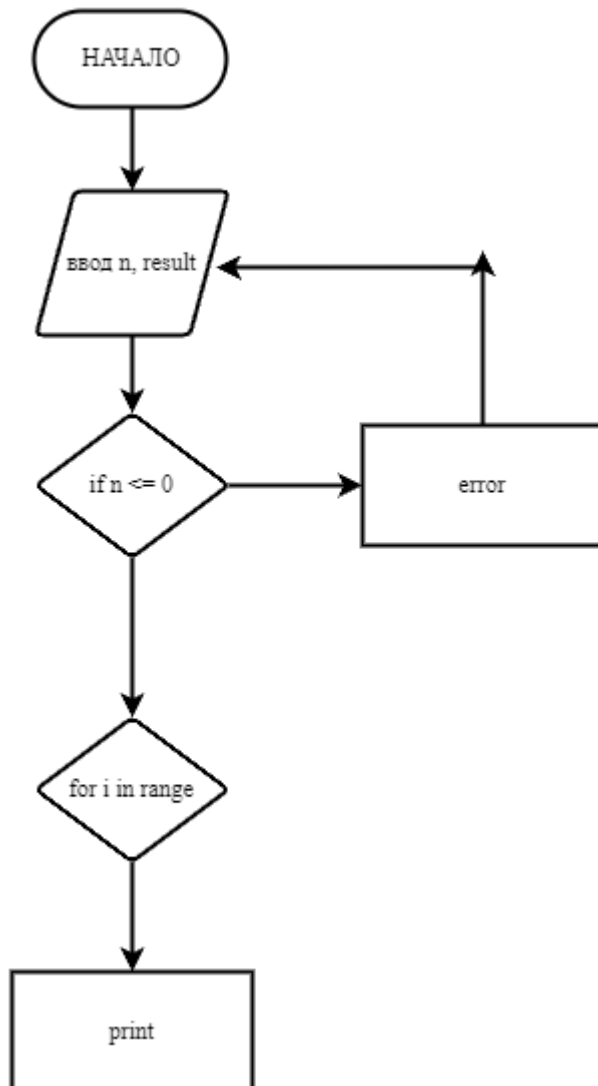
Тема: составление программ циклической структуры в IDE PyCharm Community.

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

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

Тип алгоритма: циклический.

Блок-схема алгоритма N1:



Текст программы:

Дано целое число N (>0).
Найти произведение $1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$ (N сомножителей).

```
def fun():  
    try:  
        N = int(input("Введите целое число N: " + "\n"))  
  
        if N <= 0:  
            raise ValueError("N должно быть больше 0")  
  
        result = 1.0  
        for i in range(1, N + 1):  
            result *= 1.0 + (i / 10) # i / 10 = дробное значение  
  
        print("Произведение 1.1 * 1.2 * 1.3 * ... * " + "\n" f"{N} = {result:.2f}")  
  
    except ValueError as e:  
        print(f"[-] Ошибка:", e)  
        fun()
```

fun()

Протокол работы программы:

Введите целое число N:

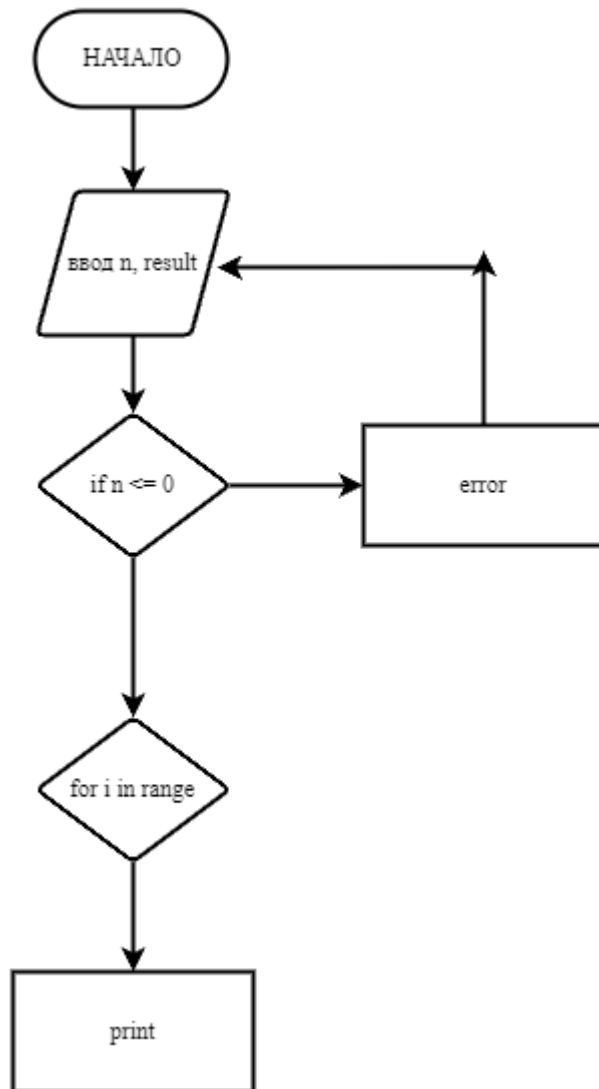
5

Произведение $1.1 \cdot 1.2 \cdot 1.3 \cdot \dots \cdot$

$5 = 3.60$

Process finished with exit code 0

Блок-схема алгоритма N2:



Текст программы:

```

def xam():
    try:
        N = int(input("Введите целое число N: "))

        if N <= 1:
            raise ValueError("N должно быть больше 1")

        K = 0
        while 3 ** K <= N:
            print(f"3^{K} = {3 ** K}") # принцип работы цикла
            K += 1

        print(f"Наименьшее целое число K, при котором выполняется неравенство, равно: {K}")

    except ValueError as e:
        print("[-] Ошибка:", e)
  
```

хам()

хам()

Протокол работы программы:

Введите целое число N: 3

$3^0 = 1$

$3^1 = 3$

Наименьшее целое число K, при котором выполняется неравенство, равно: 2

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал(а) навыки составления программ циклической структуры в IDE PyCharm Community.

Были использованы языковые конструкции `try`, `if`, `raise`, `except`, `while`, `def`, `for`, `in`. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.