

## Практическое занятие №4

Тема: составление программ циклической структуры в IDE PyCharm Community.

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

Вариант20.

### Номер 1

### Блок-схема для данного кода на Python:

```
graph TD
    A[Начало] --> B{Ввод x (вещественное)};
    B --> C{Ввод n (целое)};
    C --> D{n <= 0 ИЛИ x не вещественное?};
    D -- Да --> E[Вывод ошибки: "Только числа n > 0 и число x дробное число"];
    E --> F[Конец];
    D -- Нет --> G[count = 0];
    G --> H{Цикл for k in range(n + 1)};
    H --> I[term = ((-1) ** k) * (x ** (2 * k)) / math.factorial(2 * k)];
    I --> J[count += term];
    J --> H;
    H --> K[Конец цикла];
    K --> L[Вывод count];
    L --> F;
```

### Код программы

```
#Дано вещественное число X и целое число N (> 0). Найти значение выражения
# 1 - X2/(2!) + X4/(4!) - ... + (-1)-X2/((2-N)!) (N! = 12 ... N).
# Полученное число является приближенным значением функции
```

cos в точке X.

```
import math
try:
    x = float(input("Введите вещественное число X: "))
    n = int(input("Введите целое число N (> 0): "))
    if n <= 0 and float(x):
        raise ValueError
except ValueError:
    print("Только числа n > 0 и число x дробное число")
else:
    count = 0
    for k in range(n + 1):
        term = ((-1) ** k) * (x ** (2 * k)) /
math.factorial(2 * k)
        count += term
    print(count)
```

## Номер 2

Блок-схема кода Python:

Начало --> Ввод числа --> Преобразование в целое

\

\--> Обработка исключения (ValueError)

\

\--> Вывод ошибки --> Конец

\

\--> Инициализация count --> Цикл for

\

\--> Вычисление в цикле --> Конец цикла

\

\--> Вывод результата --> Конец

Код программы

```

# Найти сумму  $1 ** 1 + 2 ** 2 + \dots + N ** N$ 
try:
    number = int(input("Введи положительное целое число\n"))
    if number <= 0:
        raise ValueError
except ValueError:
    print("Только положительные числа")# если уже ввели
буквы
else:
    count = 0
    for i in range(1, number + 1):
        count = count + (i ** i)
    print(count, "count")

```

## 

1. Импортируется модуль math.
2. Пользователь вводит вещественное число X и целое число N.
3. Проверяется, является ли N положительным числом.
4. Если N не удовлетворяет условию, программа выводит сообщение об ошибке.
5. Если проверка прошла успешно, начинается расчёт суммы ряда.
6. Результат расчётов выводится на экран.

## 

1. Ожидание ввода числа.
2. Преобразование ввода в целое число.
3. Проверка числа на положительность, иначе — исключение.
4. Инициализация счетчика count = 0.
5. Цикл от 1 до числа включительно, возведение каждого числа в степень самого себя и добавление к count.
6. Вывод результата и завершение программы.

**Вывод:** в процессе выполнения практического занятия выработал навыки составления программ циклической структуры в ID PyCharm Community.