

Практическое занятие № 4

Тема: составление программ циклической структуры в IDE PyCharm Community.
Размещение проекта на GitHub.

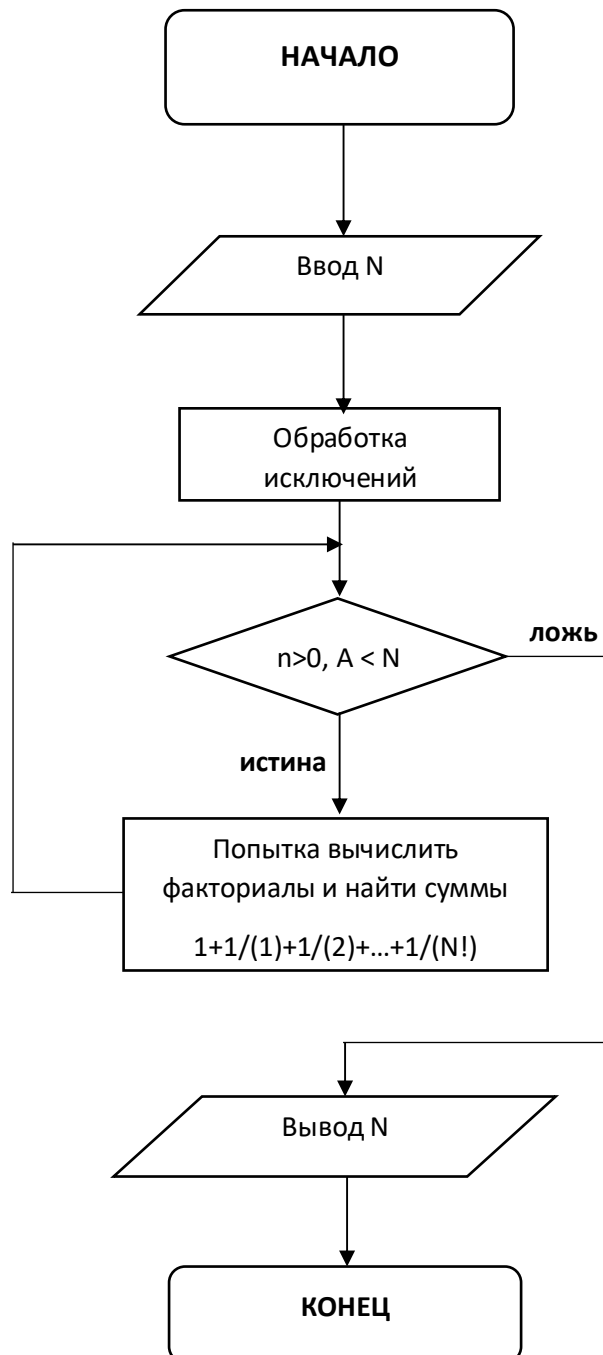
Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ циклической структуры в IDE PyCharm Community, первичные навыки работы с сервисом GitHub.

Постановка задачи.

1) Дано целое число $N (>0)$. Используя один цикл, найти сумму $1 + 1/(1!) + 1/(2!) + 1/(3!) + \dots + 1/(N!)$ (выражение $N!$ — N -факториал — обозначает произведение всех целых чисел от 1 до N : $N! = 1 \cdot 2 \cdot \dots \cdot N$). Полученное число является приближенным значением константы $e = \exp(1)$.

Тип алгоритма: циклический.

Блок-схема алгоритма:



Текст программы:

```
# вариант 17
# Дано целое число N(>0). Используя один цикл, найти сумму  $1 + 1/(1!) + 1/(2!) + \dots + 1/(N!)$ .
# Выражение  $N!$  – N-факториал – обозначает произведение всех целых чисел от 1 до N:  $N! = 1 \cdot 2 \cdot \dots \cdot N$ .
# Полученное число является приближенным значением константы  $e = \exp(1)$ .

try:
    # обработка исключений
    n = int(input("Введите целое число N: "))
    a = 0
    c = 0
    b = 1
    if n > 0:
        e = a + 1
        while a < n:
            # цикл while выполняется то тех пор, пока a < n
            a += 1
            e *= a
            b += 1 / e
        print(b)
    else:
        print("Вы ввели неправильное число")
except ValueError:
    print("Что-то пошло не так, попробуйте снова!")
```

Протокол работы программы:

Введите целое число N: 5

Сумма равна: 2.7166666666666663

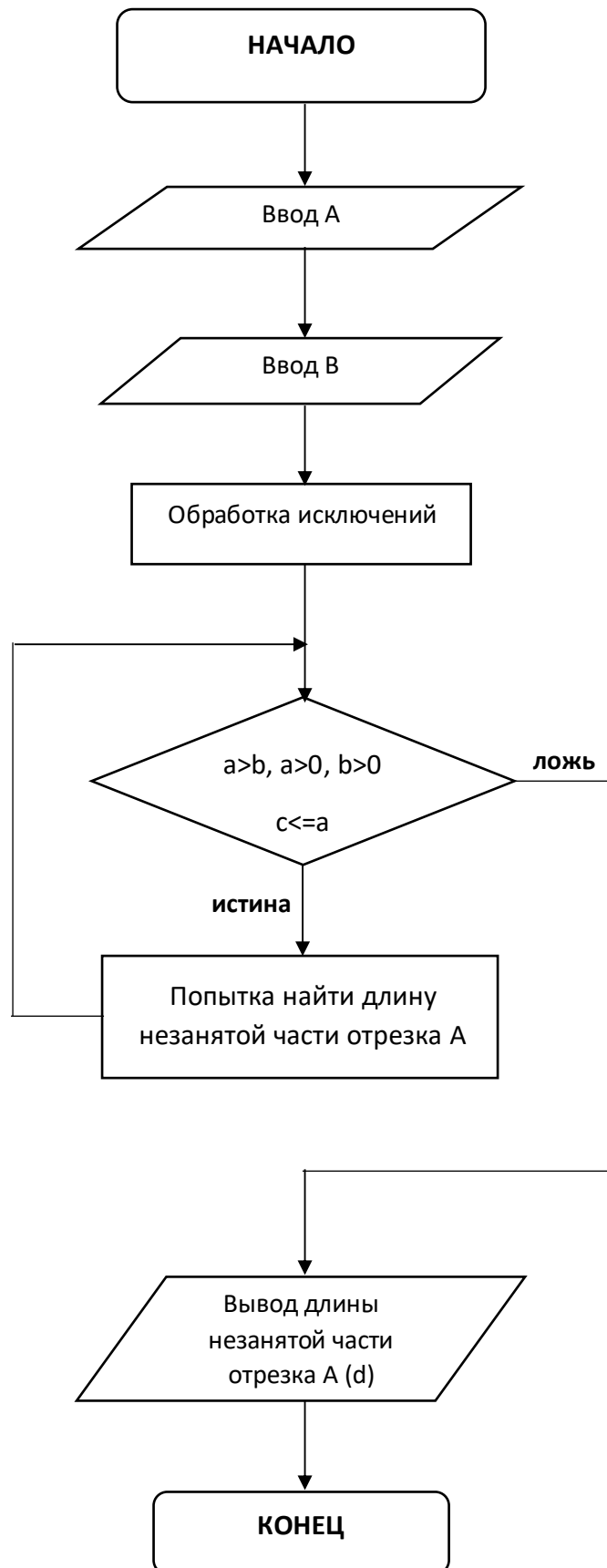
Process finished with exit code 0

Постановка задачи.

2) Даны положительные числа A и B ($A > B$). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений). Не используя операции умножения и деления, найти длину незанятой части отрезка A.

Тип алгоритма: циклический.

Блок-схема алгоритма:



Текст программы:

```

# вариант 17
# Даны положительные числа a и b ( $a > b$ ). На отрезке a размещено максимально возможное
# количество отрезков b.
# Не используя операции умножения и деления, найти длину незанятой части отрезка a.

try:
    # обработка исключений
    a = float(input("Введите положительное число a: "))
    b = float(input("Введите положительное число b: "))
    c = 0
    if 0 < a > b > 0:
        while c <= a:
            c += b
            c = c - b
            d = a - c
            d = round(d, 1)          # round(d,1) округляет число d до 1 знака после запятой
            print(f"Длина незанятой части отрезка a: {d}")
    else:
        print("Вы ввели неправильное число")
except ValueError:
    print("Что-то пошло не так, попробуйте снова")

```

Протокол работы программы:

Введите положительное число a: 2.2

Введите положительное число b: 1

Длина незанятой части отрезка a: 0.2

Process finished with exit code 0

Вывод:

В процессе выполнения практического занятия выработал навыки составления программ циклической структуры в IDLE PyCharm Community. Были использованы языковые конструкции **while**, **if**.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.