

## Практическое занятие № 4

**Тема:** Составление программ циклической структуры в IDE PyCharm Community.

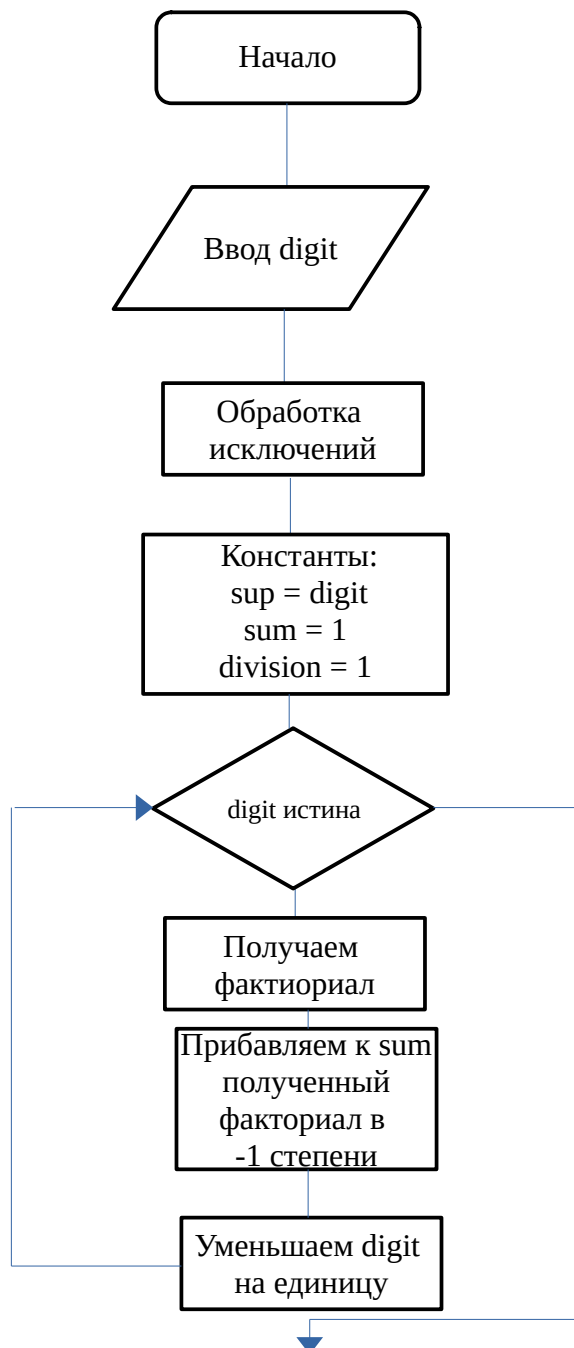
**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ циклической структуры в IDE PyCharm Community.

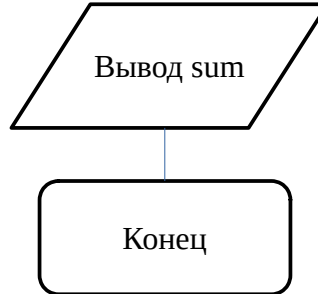
### Постановка задачи №1.

Дано 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:

```
# 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  $e = \exp(1)$ .  
  
digit = input("Введите целое число: ")  
while type(digit) != int: # обработка исключений  
    try:  
        digit = int(digit)  
    except ValueError:  
        print("Неправильно ввели!")  
        digit = input("Введите число: ")  
  
sum = 1  
division = 1  
sup = digit  
while digit:  
    division = division * (sup - digit + 1)  
    sum = sum + 1/division  
    digit -= 1  
print(sum)
```

### Протокол работы программы:

Введите целое число: 4  
2.7083333333333333

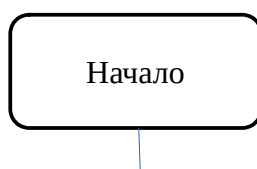
Process finished with exit code 0

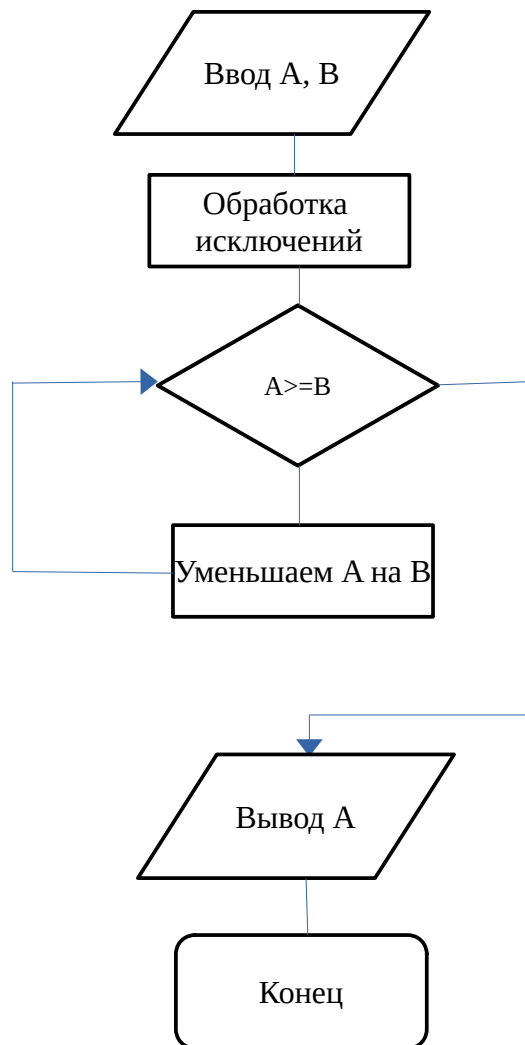
### Постановка задачи №2.

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:

```
# 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.
a = input("Введите первое положительное число: ")
b = input("Введите второе положительное число, которое меньше первого: ")

while type(a) != int: # обработка исключений
    try:
        a = int(a)
    except ValueError:
        print("Неправильно ввели!")
        a = input("Введите первое число: ")
while type(b) != int: # обработка исключений
    try:
        b = int(b)
        if b >= a:
            print("Неправильно ввели!")
            b = input("Введите второе число ещё раз: ")
    except ValueError:
        print("Неправильно ввели!")
        b = input("Введите второе число: ")

while a >= b:
    a -= b
```

```
print(f"Длина незанятой части отрезка A: {a}")
```

### Протокол работы программы:

Введите первое положительное число: 20

Введите второе положительное число, которое меньше первого: 3

Длина незанятой части отрезка A: 2

Process finished with exit code 0

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

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

Готовые программные коды выложены на GitHub.