

Практическое занятие № 4

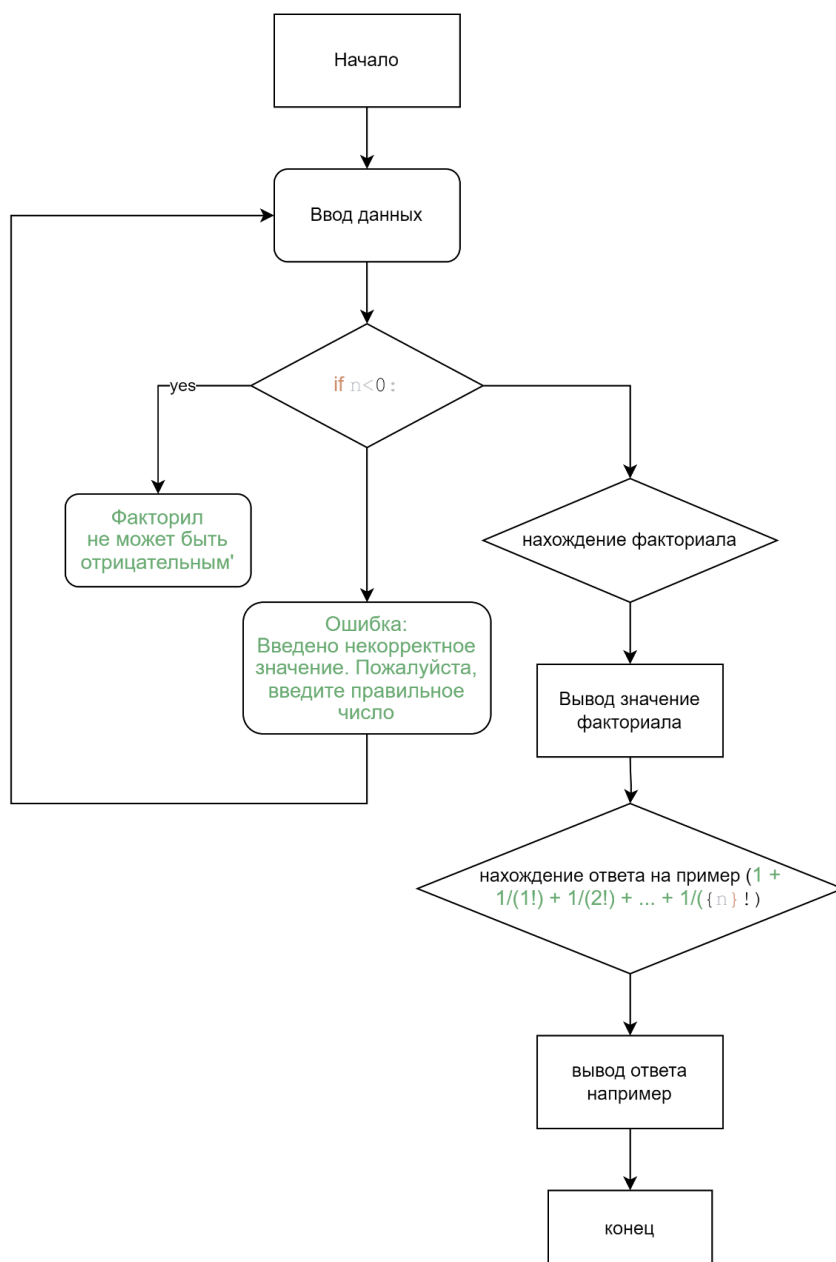
Тема: составление программ циклической структуры в IDE PyCharm Community

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

Постановка задачи1:

Тип алгоритма: циклический

Блок-схема алгоритма1:



Текст программы1:

```

# 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  $e = \exp(1)$ .
def factorial(n):
    if n == 0 or n == 1: # Базовый случай
        return 1
    return n * factorial(n - 1)

def example_factorial(n): #функция
    sum_factorials = 0.0
    factorial = 1
    i = 0
    while i <= n:
        if i > 0:
            factorial *= i
            sum_factorials += 1 / factorial
            i += 1
    return sum_factorials

while True:
    try:
        n = int(input("Введите n факториал"))
        if n < 0:
            print('Факториал не может быть отрицательным')
        else:
            print("-" * 35)
            print(f"Факториал {n} равен {factorial(n)}")
            print("-" * 35)
            print(f" $1 + 1/(1!) + 1/(2!) + \dots + 1/({n}!)$ ")
            print(f"Ответ примера= {example_factorial(n)}")
            break
    except ValueError:
        print("11.")

```

Терминал1:

C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.11.exe "D:\колледж\алгоритмы и программирование\Pz Sapronov Daniil\PZ_4+\PZ_4_1.py"

Введите n факториал5

Факториал 5 равен 120

$1 + 1/(1!) + 1/(2!) + \dots + 1/(5!)$

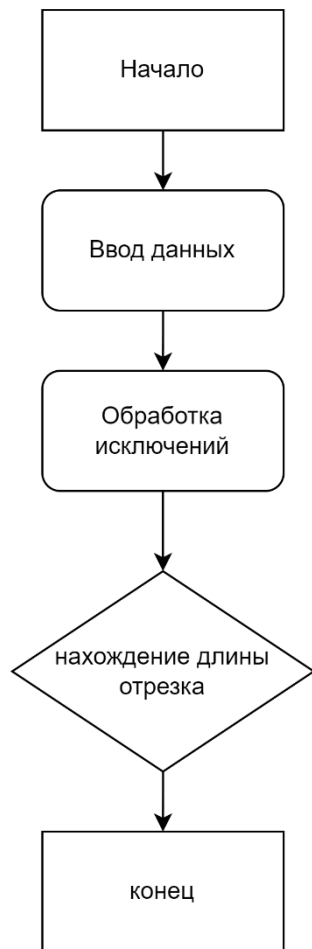
Ответ примера= 2.7166666666666663

Process finished with exit code 0

Постановка задачи2: 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 А.

Блок-схема алгоритма2:



Текст программы2:

```
# Даны положительные числа А и В (А > В). На отрезке длины А размещено
# максимально возможное количество отрезков длины В (без наложений). Не
# используя операции умножения и деления, найти длину незанятой части отрезка
# А.
a, b = input("Введите первое число: "), input("Введите второе число: ")

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)
    except ValueError:
        print("Неправильно ввели!")
        b = input("Введите второе число: ")

k = 0
while a <= b:
    print(a)
```

```
a += 1
k += 1
print('Количество чисел: ', k)
```

Терминал2:

C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.11.exe
"D:\колледж\алгоритмы и программирование\Pz Sapronov
Daniil\PZ_4+\PZ_4_2.py"

Введите первое число: 1

Введите второе число: 10

1

2

3

4

5

6

7

8

9

10

Количество чисел: 10

Process finished with exit code 0

Вывод: : в процессе выполнения практического занятия выработал навыки составления программ циклической структуры в PyCharm Community Edition . Были использованы языковые конструкции if, def, while, try, except. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub