

Отчет о практическом занятии

Практическое занятие №4. Вариант 17.

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

Задание 1.

Постановка задачи.

Дано целое число $N(>0)$.

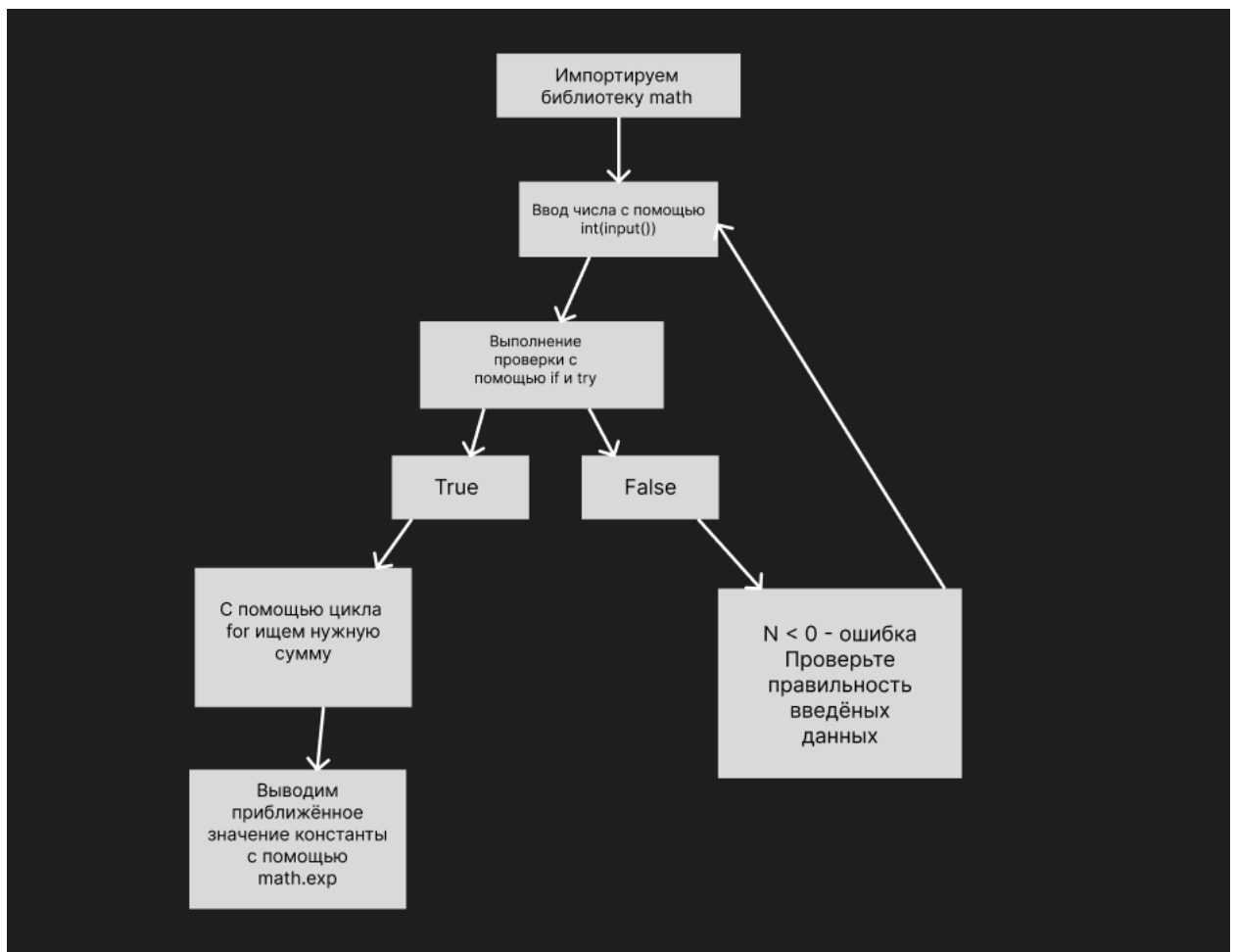
Используя один цикл, найти сумму $1 + 1/(1!) + 1/(2!) + 1/(3!) + \dots + 1/(N!)$

(выражение $N!$ – это N -факториал (обозначает произведение всех целых чисел от 1 до N : $N! = 1 * 2 * \dots * N$))

Полученное число является приближенным значением константы $e = \exp(1)$.

Тип алгоритма: линейный.

Блок-схема алгоритма:



Текст программы:

```
# Вариант 17 Задача 1
# Дано целое число N(>0) .
# Используя один цикл, найти сумму 1+1/(1!)+1/(2!)+1/(3!)+...+1/(N!)
# (выражение N! - N-факториал - обозначает произведение всех целых
чисел от 1 до N: N! = 1·2·...·N)
# Полученное число является приближенным значением константы e=exp(1)
import math
try:
    N = int(input("Введите целое число N (>0): "))
    sum = 0
    factorial = 1
    if N <= 0:
        print("N - должно быть больше 0")
    else:
        for i in range(N+1):
            sum += 1 / factorial
            factorial *= (i + 1)
        print("Сумма равна:", sum)
        print("Приближенное значение константы e:", math.exp(1))
except ValueError:
    print("Проверьте правильность введенных данных!")
```

Задание 2

Постановка задачи.

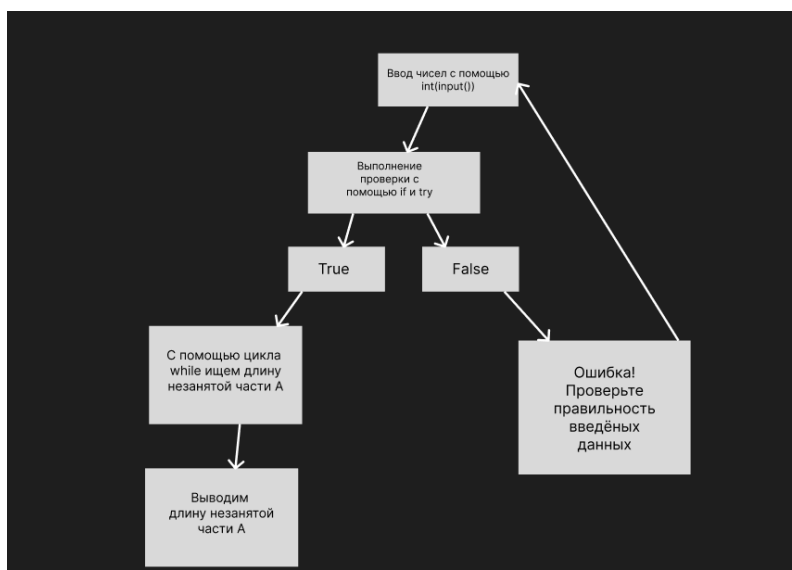
Даны положительные числа A и B ($A > B$).

На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений).

Не используя операции деления и умножения, найти длину занятой части отрезка A

Тип алгоритма: циклический.

Блок-схема алгоритма:



Текст программы:

```
# Вариант 17 Задача 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 деления и умножения, найти длину
незанятой части отрезка A
try:
    A = float(input("Введите положительное число A: "))
    B = float(input("Введите положительное число B (B должно
быть меньше A): "))
    dlina = A
    if A < 0:
        print("Число A - должно быть положительным")
    elif B > A:
        print("Число B - должно быть меньше, чем число A")
    elif B < 0:
        print("Число B - должно быть положительным")
    else:
        while dlina >= B:
            dlina -= B
        print("Длина незанятой части отрезка A =", dlina)
except ValueError:
    print("Проверьте правильность введённых данных!")
```

Вывод: в процессе выполнения практического занятия выработал навыки составления программ циклической структуры в IDE PyCharm Community. Были использованы циклы for и while, а также Команды print(), int(), input(), if/else, try/except. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.