

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.

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 №1:



Текст программы №1:

```
n = input('Введите число больше нуля>> ') # Ввод числа

# Обработка исключений
while type(n) != int:
    try:
        n = int(n)
        if n < 0: # Обработка неположительных чисел
            print("Неправильно ввели!")
            n = input("Введите число больше нуля>> ")
        else:
            break
    except ValueError:
        print("Неправильно ввели!")
        n = input("Введите число больше нуля>> ")

sum_series = 1.0
factorial = 1.0

# Производим вычисления
for i in range(1, n + 1):
    factorial *= i
    sum_series += 1 / factorial

print(f"Приближенное значение константы e: {sum_series}") # Вывод результата
```

Протокол работы программы:

Введите число больше нуля>> 3

Приближенное значение константы e: 2.6666666666666665

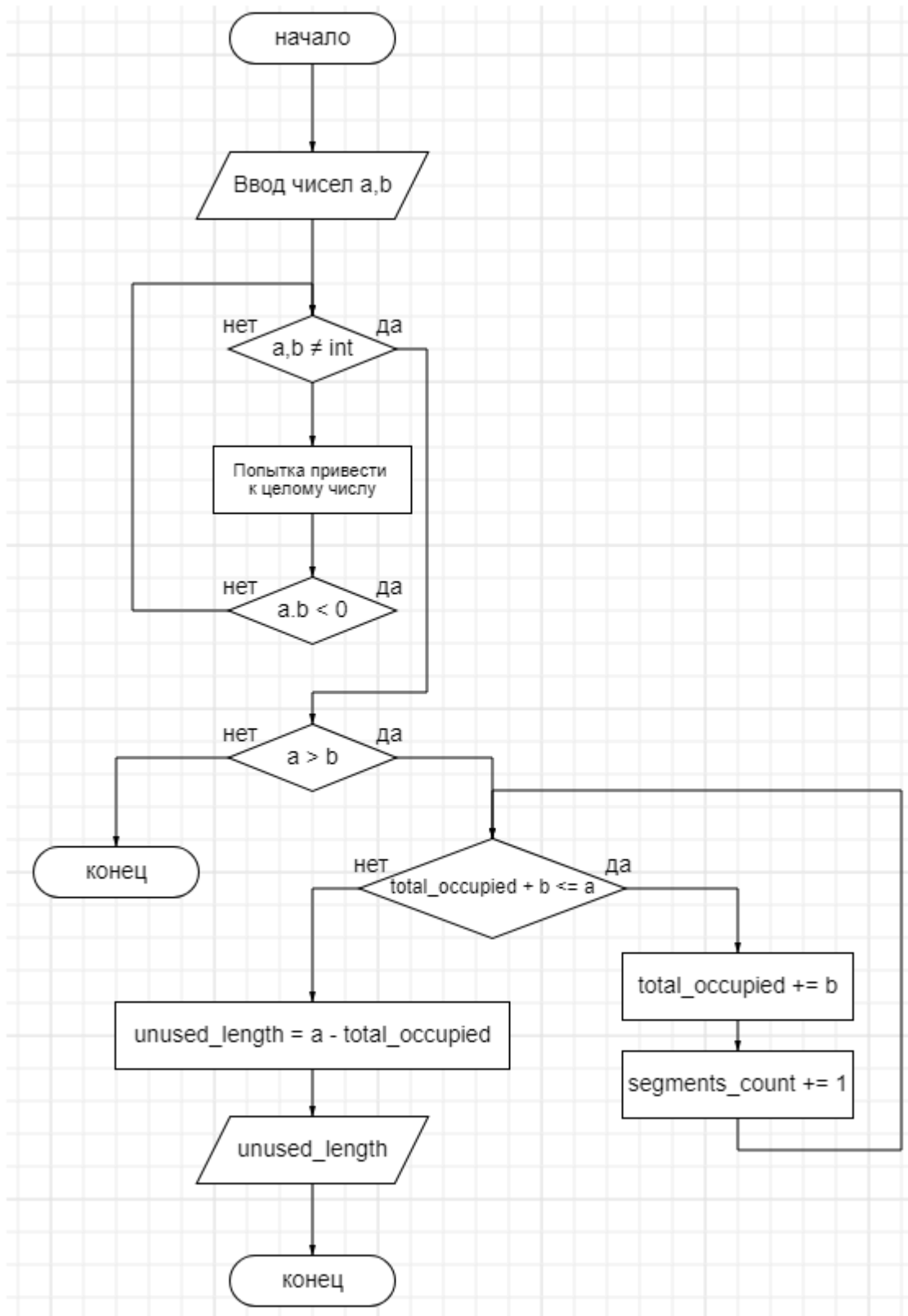
Process finished with exit code 0

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 №2:



Текст программы №2:

```

# Ввод длины большего отрезка A
a = input('Введите длину большего отрезка A>> ')

# Обработка исключений
while type(a) != int:
    try:
        a = int(a)
        # Проверка на положительность числа A
        if a < 0:
            print("Неправильно ввели!")
            a = input('Введите длину большего отрезка A>> ')
    except ValueError:
        print("Неправильно ввели!")
        a = input('Введите длину большего отрезка A>> ')

# Ввод длины меньшего отрезка B
b = input('Введите длину меньшего отрезка B>> ')

# Обработка исключений
while type(b) != int:
    try:
        b = int(b)
        # Проверка на положительность числа B
        if b < 0:
            print("Неправильно ввели!")
            b = input('Введите длину меньшего отрезка B>> ')
    except ValueError:
        print("Неправильно ввели!")
        b = input('Введите длину меньшего отрезка B>> ')

# Проверка, что число A больше B
if a > b:
    total_occupied = 0 # Переменная для хранения занятой длины
    segments_count = 0 # Переменная для подсчета отрезков

    # Цикл для размещения отрезков B в отрезке A
    while total_occupied + b <= a:
        total_occupied += b # Увеличиваем занятую длину на отрезок B
        segments_count += 1 # Подсчитываем кол-во отрезков

    # Вычисляем длину незанятой части отрезка A
    unused_length = a - total_occupied

    # Выводим результат
    print(f"Длина незанятой части отрезка A: {unused_length}")
else:
    print("Неправильно ввели! \nЧисло A должно быть больше числа B!")

```

Протокол работы программы №2:

Введите длину большего отрезка A>> 10

Введите длину меньшего отрезка B>> 3

Длина занятой части отрезка A: 1

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ цикличной структуры в IDE PyCharm Community. Были использованы языковые конструкции `while`, `if`, `for`. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовый программный код выложен на GitHub.