

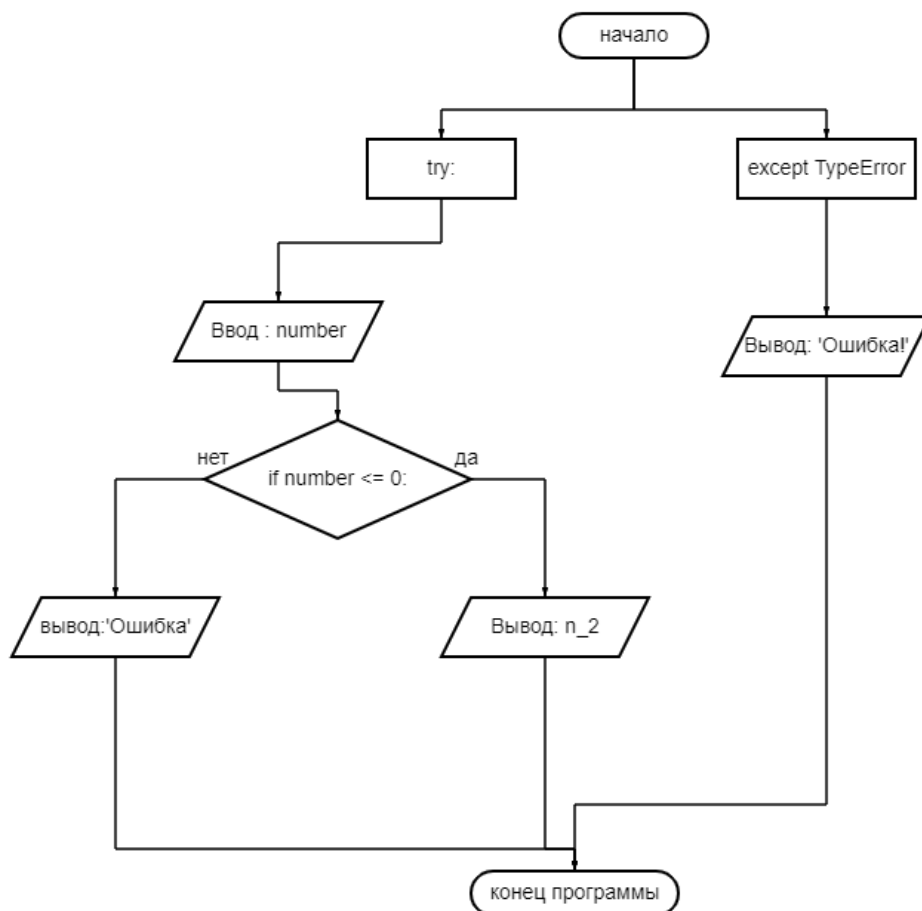
Отчет

Практическая задание №4

Тема: Составление программ структуры IDE PC Community Цель: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ в IDE PC Community

Постановка задания(4.1): Дано целое число N ($N > 0$). 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 $e = \exp(1)$.

Блок-схема (4.1):



Текст программы:

```
try: # обработчик исключений в котором выполняется блок при соответствии необходимого ввода данных
```

```
number = int(input('Введите целое число, оно должно быть больше 0: ')) # ввод целого положительного числа с клавиатуры
```

```
n = 1 # формула n-это переменная которой присвоено значение 1 и оно находится в типе
```

```

int
n_2 = 1 # формула(переменная)
n_3 = 1/n # переменной n_3 присваивается значение 1/n
if number <= 0: # условие, введенное число должно быть больше нуля
    print('Ошибка') # вывод ошибки если не удовлетворяет условию
else: # иначе выполняется цикл
    while n <= number: # цикл работает, пока n меньше, либо равен значению, которое
        # присвоено переменной number. Цикл прекратится, если n будет больше number
        n_3 *= 1/n # тело цикла,
        n_2 += n_3 # тело цикла,
        n += 1 # тело цикла,
    print(n_2) # вывод содержимого в конечной переменной n_2
except TypeError: # обработчик ошибки, в котором выводится ошибка, если введенные
    # нами данные не соответствуют формату int
    print('Ошибка!') # вывод ошибки если введен неверный тип данных

```

Протокол работы:

Вводится число 5

Проверка на положительность

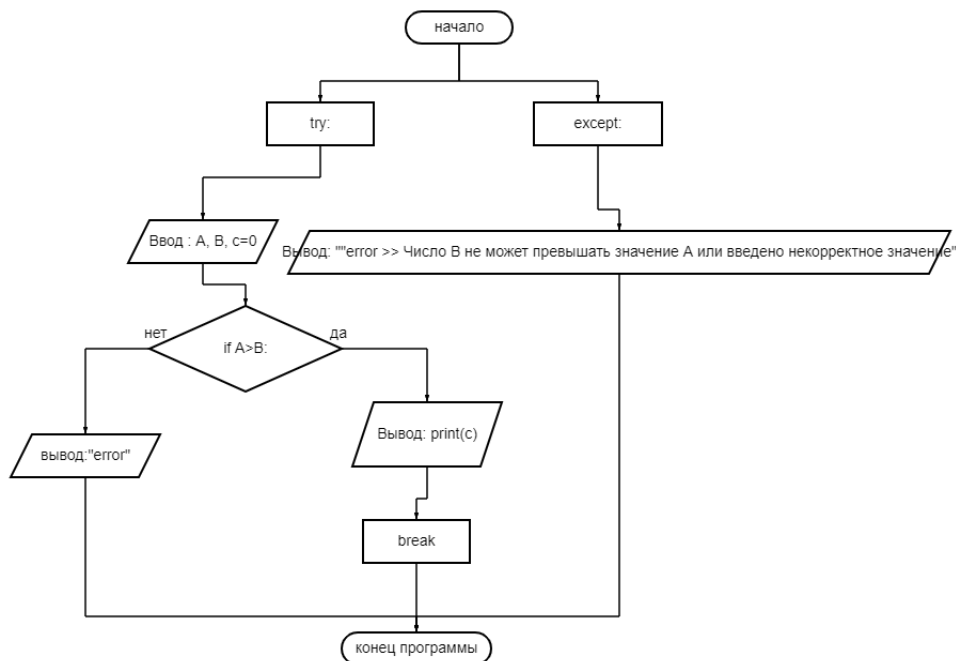
Переход к циклу

Тело цикла(формула $1 + 1/(1!) + 1/(2!) + 1/(3!) + \dots + 1/(N!)$)

Вывод: n_2 (148.41315910258)

Постановка задания(4.2): Даны положительные числа А и В ($A > B$). На отрезке длины А размещено максимально возможное количество отрезков длины В (без наложений). Не используя операции умножения и деления, найти длину занятой части отрезка А.

Блок-схема:



Текст программы:

```
try: #обработчик исключений
A = int(input("Введите положительное число A >> ")) # ввод целого числа с клавиатуры
B = int(input("Введите положительное число B >> ")) # ввод целого числа с клавиатуры
c = 0 # нулевое значение
if A>B: # условие первое число A должно быть больше B
while A>B: # цикл выполняется, если A>B.
A = A-B # тело цикла, в котором от числа A вычитается число B
c = A # тело цикла, остаток от числа A
print(c) # вывод c
break # завершает цикл
else: # если не выполняет условие if (A>B)
print("error") # вывод ошибки
except: # ошибка, если введены неверные значения
print("error >> Число B не может превышать значение A или введено некорректное
значение") # вывод ошибки
```

Протокол работы:

Вводятся числа A, B, c=0

5, 3

Проверка условия: A>B

Цикл while A>B:

A = A-B

C = A

Вывод: 2

Вывод : в ходе выполнения практического задания выработал навыки составления программ. Была использована языковая конструкция try-except, if-else, while-break. Выполнена разработка кода, отладка, тестирование, оптимизация кода. Готовая работа выложена на GitHub.