

**Практическое занятие №4**

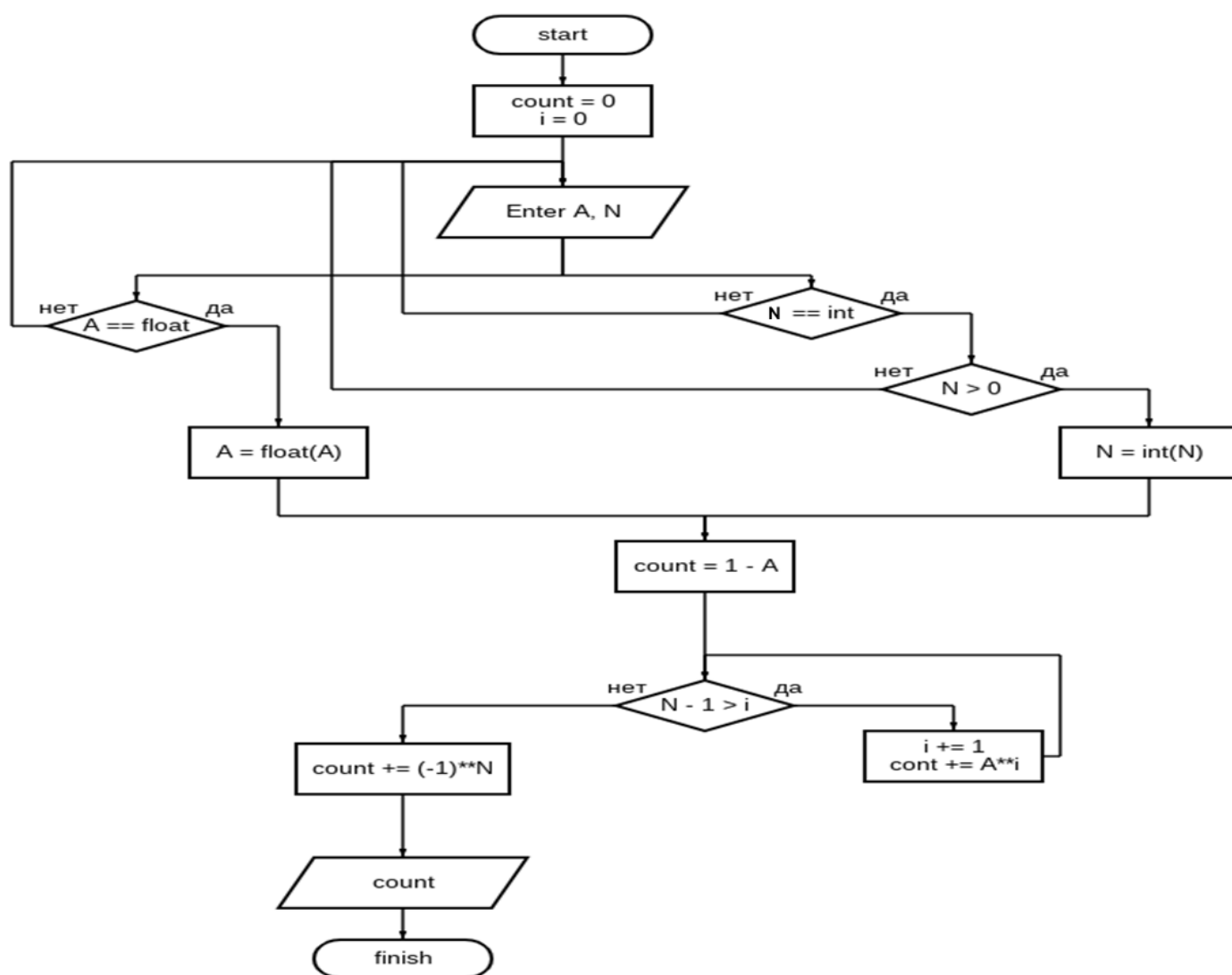
**Тема:** составление программ циклической структуры в IDE PyCharm Community.

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

**Постановка задачи.**

1. Разработать программу для ввода вещественного числа A и целого числа N ( $>0$ ). Используя один цикл, найти значение выражения  $1 - A + A^2 + A^3 + \dots + (-1)^N$ . Условный оператор не использовать.
2. Разработать программу для ввода положительных чисел 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 количество отрезков B, размещенных на отрезке A.

**Тип алгоритма:** циклический.



**Блок-схема алгоритма:**

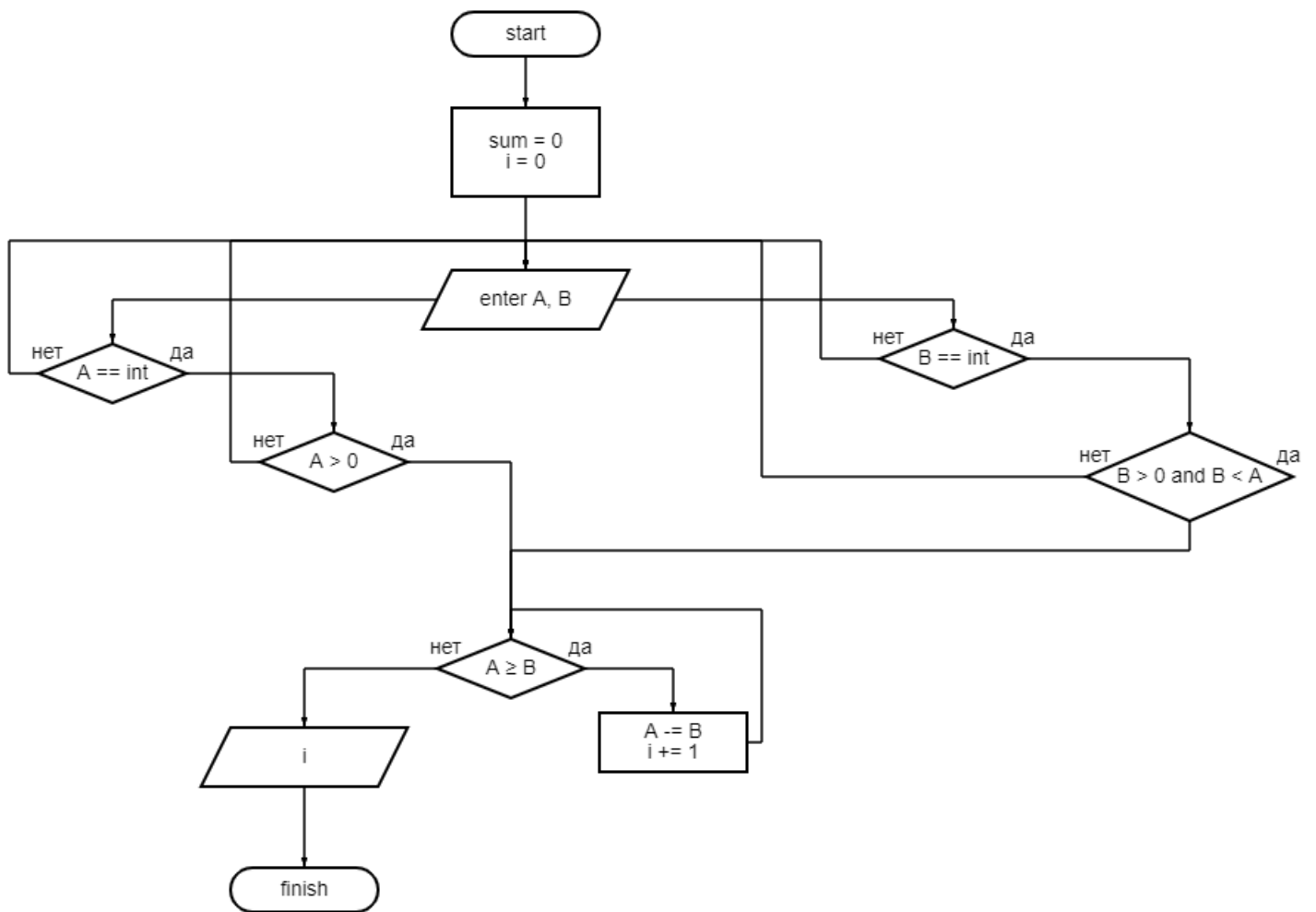
## Текст программы:

1.

```
#Дано вещественное число A и целое число N (N>0). Используя один цикл,  
#найти значение выражения  $1 - A + A^2 + A^3 + \dots + (-1)^N$ . Условный оператор не использовать.  
  
#-> 5.0 4  
#<- 147.0  
  
count = 0    #значение выражения  
i = 1        #счётчик  
A = input("Enter A: ")  
N = input("Enter N: ")  
  
# обработка исключений  
while type(A) != float:  
    try:  
        A = float(A)  
    except ValueError:  
        A = input("Enter A: ")  
  
while type(N) != int:  
    try:  
        N = int(N) if int(N) > 0 else int(input("Enter N (N > 0): "))  
    except ValueError:  
        N = input("Enter N: ")  
  
count = 1 - A  
  
while N - 1 > i:  
    i += 1  
    count = count + A**i  
  
count = count + (-1)**N  
  
print("expression value: ", count)
```

## Блок-схема алгоритма:

2.



Текст программы:

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 количество отрезков B, размещённых
на отрезке A.

# -> 6 2
# <- 3

A = input("Enter A: ")
B = input("Enter B: ")
sum = 0      #число, которое стремится к длине A
i = 0       #максимально возможное количество отрезков длины B

# обработка исключений
while type(A) != int:
    try:
        A = int(A) if int(A) > 0 else int(input("Enter A (A > 0): "))
    except ValueError:
        A = input("Enter A: ")
  
```

```
while type(B) != int:
    try:
        B = int(B) if int(B) > 0 and int(B) < A else int(input("Enter B: "))
    except ValueError:
        B = input("Enter B: ")

while A >= B:
    A -= B
    i += 1
print("number of segments B placed on segment A: ", i)
```

**Вывод:** в процессе выполнения практического занятия выработала навыки работы составления циклических алгоритмов. Были использованы языковые конструкции try...except; while; if...else.

**Выполнено:** разработка, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.