

Практическое занятие №4

Тема: составление программ циклической структуры в IDE PyCharm Community.

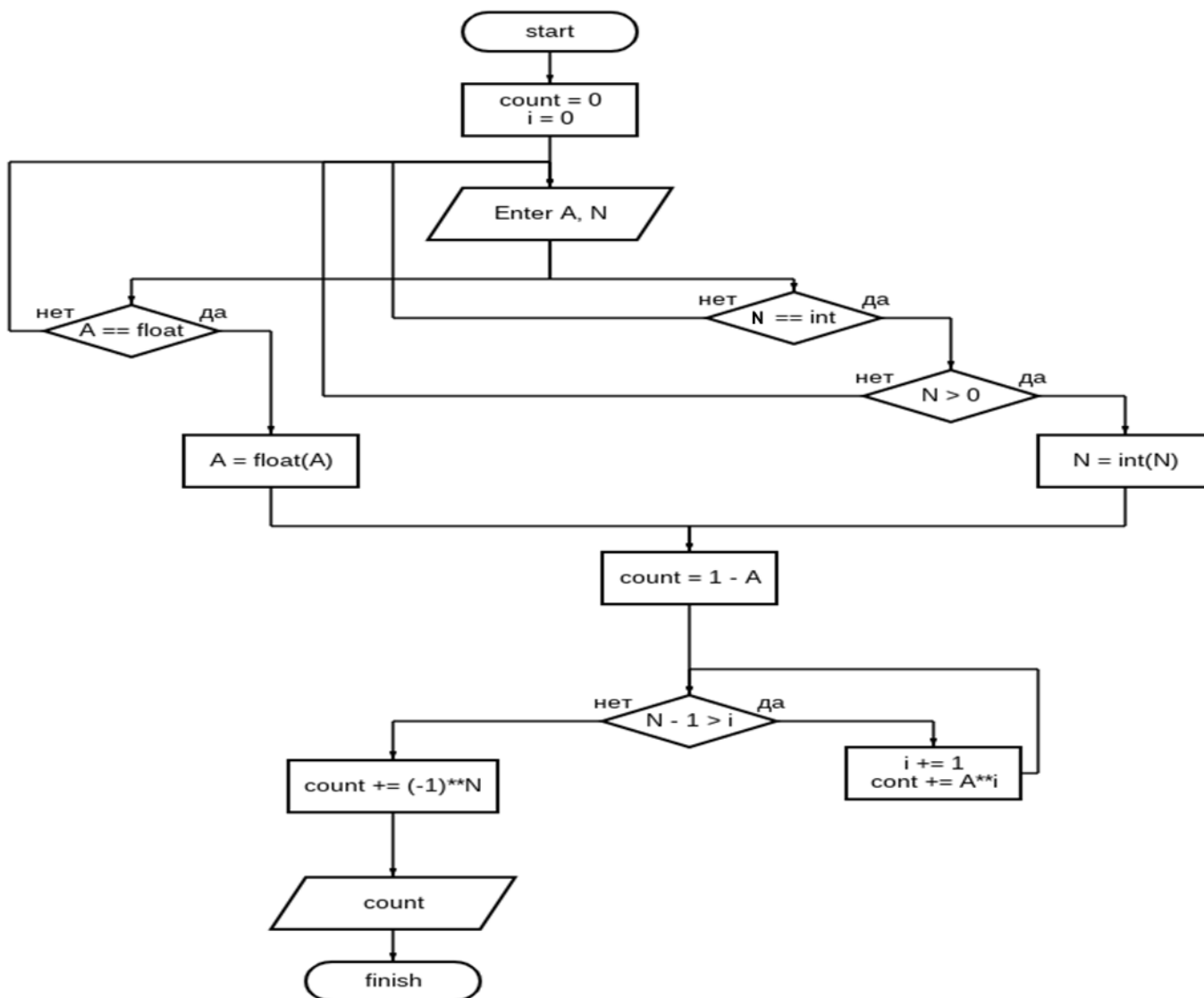
Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ циклической структуры в IDE PyCharm Community.

Постановка задачи.

1. Разработать программу для ввода вещественного числа A и целого числа N ($N > 0$). 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 1:



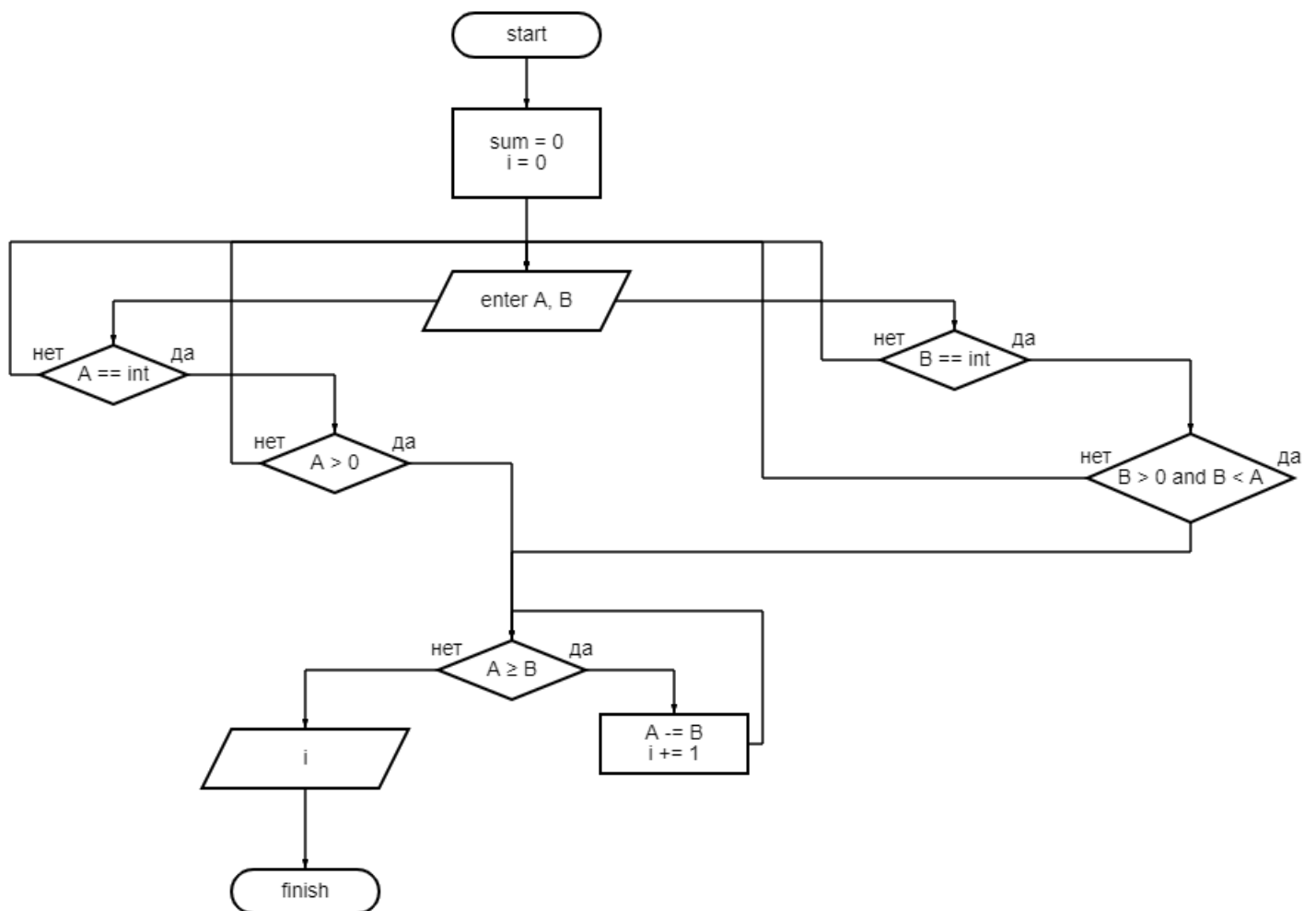
Текст программы 1:

```

1  # Дано вещественное число A и целое число N (N>0). Используя один цикл,
2  # найти значение выражения  $1 - A + A^2 + A^3 + \dots + (-1)^N N$ . Условный оператор не использовать.
3
4  # -> 5.0 4
5  # <- 147.0
6
7  count = 0 # значение выражения
8  i = 1     # счётчик
9  A = input("Enter A: ")
10 N = input("Enter N: ")
11
12
13 # обработка исключений
14 while type(A) != float:
15     try:
16         A = float(A)
17     except ValueError:
18         A = input("Enter A: ")
19
20 while type(N) != int:
21     try:
22         N = int(N) if int(N) > 0 else int(input("Enter N (N > 0): "))
23     except ValueError:
24         N = input("Enter N: ")
25
26 count = 1 - A
27
28 while N - 1 > i:
29     i += 1
30     count = count + A**i
31
32 count = count + (-1)**N
33
34 print("expression value: ", count)

```

Блок-схема алгоритма 2:



Текст программы 2:

```
1  # даны положительные числа A и B (A > B).
2  # На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений).
3  # Не используя операции умножения и деления, найти количество отрезков B, размещённых на отрезке A.
4
5  # -> 6 2
6  # <- 3
7
8  sum = 0      # число, которое стремится к длине A
9  i = 0        # максимально возможное количество отрезков длины B
10 A = input("Enter A: ")
11 B = input("Enter B: ")
12
13
14 # обработка исключений
15 while type(A) != int:
16     try:
17         A = int(A) if int(A) > 0 else int(input("Enter A (A > 0): "))
18     except ValueError:
19         A = input("Enter A: ")
20
21 while type(B) != int:
22     try:
23         B = int(B) if int(B) > 0 and int(B) <= A else int(input("Enter B: "))
24     except ValueError:
25         B = input("Enter B: ")
26
27 while A >= B:
28     A -= B
29     i += 1
30 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.