

Практическое занятие № 4

Тема: Составление программ циклической структуры в IDE PyCharm Community. Размещение проекта на GitHub.

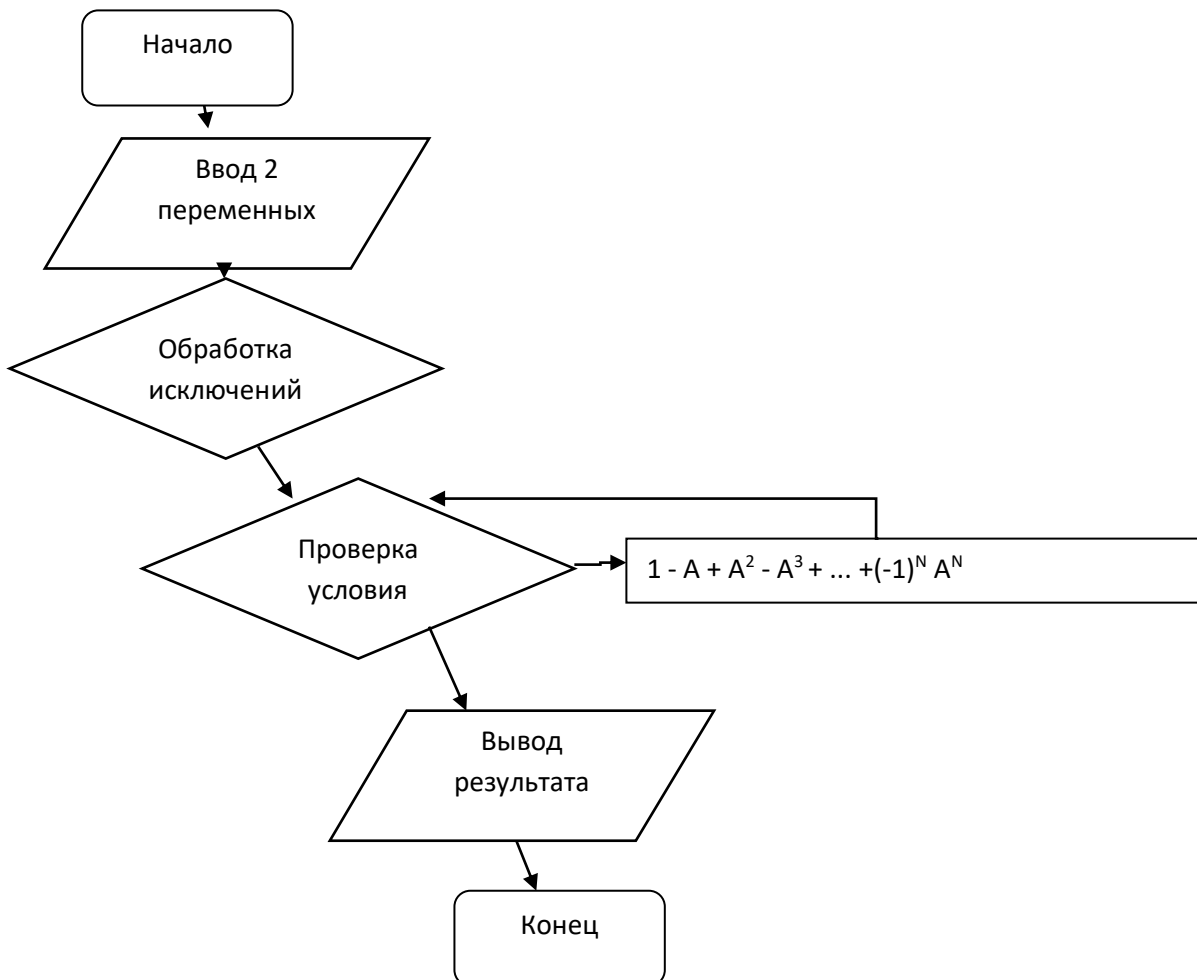
Цель: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ циклической структуры в IDE PyCharm Community, первичные навыки работы с сервисом GitHub.

Постановка задачи 1.

Дано вещественное число A и целое число $N (>0)$. Используя один цикл, найти значение выражения $1 - A + A^2 - A^3 + \dots + (-1)^N A^N$. Условный оператор не использовать.

Тип алгоритма: Циклический

Блок схема:



Текст программы:

```

# Дано вещественное число A и целое число N (N>0). Используя один цикл, найти значение
# выражения  $1 - A + A^2 - A^3 + \dots + (-1)^N A^N$ . Условный оператор не использовать.

A = float(input('Введите вещественное число: ')) # Ввод переменной
while type(A) != float: # Обработка исключений
    try:
        A = float(A)
    except ValueError:
        print('Неправильно ввели')
        A = input('Введите вещественное число: ')
N = input('Введите целое число больше нуля: ') # Ввод переменной
while type(N) != int: # Обработка исключений
    try:
        N = int(N)
    except ValueError:
        print('Неправильно ввели')
        N = input('Введите целое число больше нуля: ')
while N < 0:
    print('Неправильно ввели')
    N = input('Введите целое число больше нуля: ')
    while type(N) != int:
        try:
            N = int(N)
        except ValueError:
            print('Неправильно ввели')
            N = input('Введите целое число больше нуля: ')
C = 0
j = 0
K = -1
while K < N:
    a = (pow(-1, C))
    b = (pow(A, C))
    h = a * b
    j += h
    C += 1
    K += 1
print('Результат =', j)

```

Протокол работы программы:

Введите вещественное число: 12.3

Введите целое число больше нуля: 12

Результат = 11089572581896.168

Process finished with exit code 0

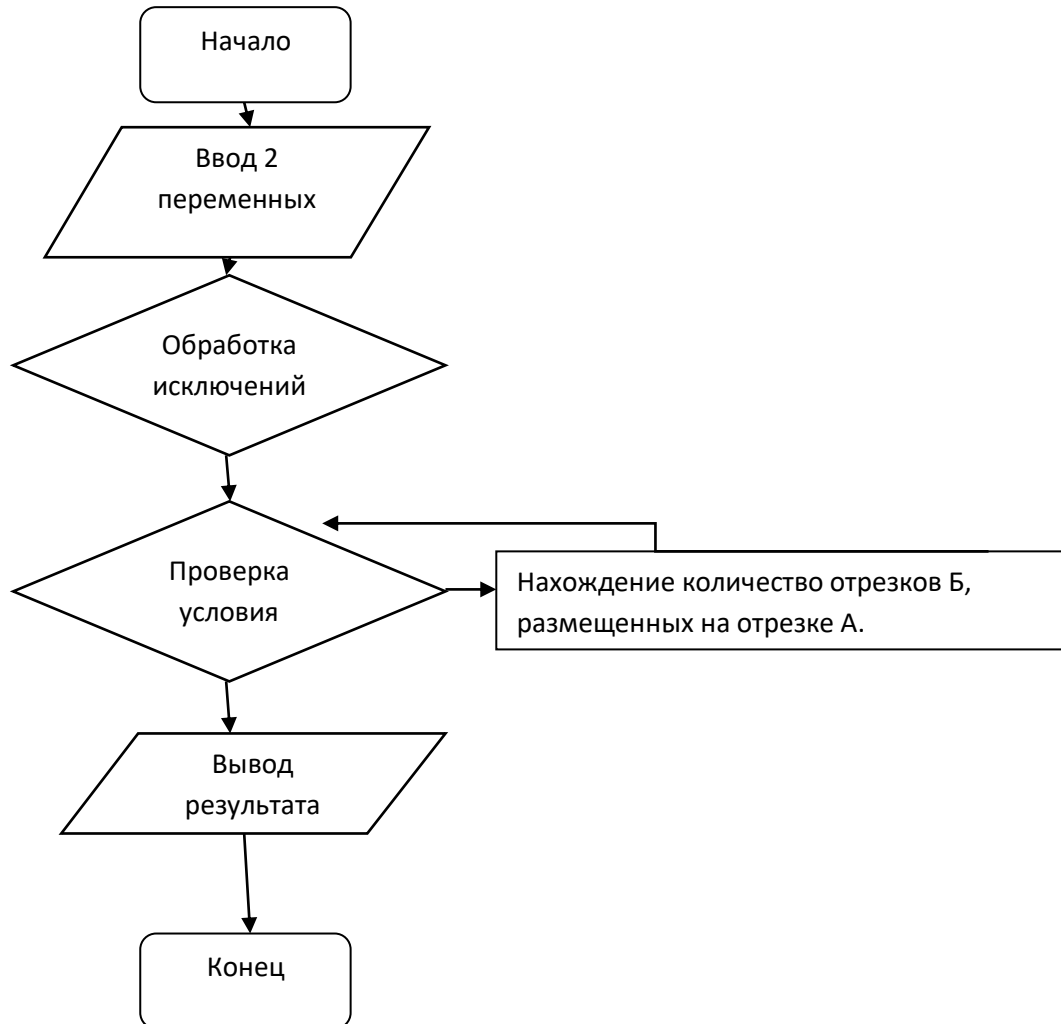
Постановка задачи 2.

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 количество отрезков Б, размещенных на отрезке А.

Тип алгоритма: Циклический

Блок схема:



Текст программы:

```
# Даны положительные числа А и В (А > В). На отрезке длины А размещено
максимально
# возможное количество отрезков длины В (без наложений). Не используя
операции
# умножения и деления, найти количество отрезков В, размещенных на отрезке А.

A = input('Введите длину отрезка А: ') # Ввод переменной
while type(A) != int: # Обработка исключений
    try:
        A = int(A)
    except ValueError:
        print('Неправильно ввели')
        A = input('Введите длину отрезка А: ')
while A < 0:
    print('Неправильно ввели')
    A = input('Введите длину отрезка А: ')
while type(A) != int:
```

```

        try:
            A = int(A)
        except ValueError:
            print('Неправильно ввели')
            A = input('Введите длину отрезка A: ')
B = input('Введите длину отрезка B, которая меньше A: ') # Ввод переменной
while type(B) != int: # Обработка исключений
    try:
        B = int(B)
    except ValueError:
        print('Неправильно ввели')
        B = input('Введите длину отрезка B: ')
while B < 0:
    print('Неправильно ввели')
    B = input('Введите длину отрезка B: ')
    while type(B) != int:
        try:
            B = int(B)
        except ValueError:
            print('Неправильно ввели')
            B = input('Введите длину отрезка B: ')
while A < B:
    print('Неправильно ввели')
    A = input('Введите длину отрезка A: ')
    while type(A) != int:
        try:
            A = int(A)
        except ValueError:
            print('Неправильно ввели')
            A = input('Введите длину отрезка A: ')
    while A < 0:
        print('Неправильно ввели')
        A = input('Введите длину отрезка A: ')
        while type(A) != int:
            try:
                A = int(A)
            except ValueError:
                print('Неправильно ввели')
                A = input('Введите длину отрезка A: ')
B = input('Введите длину отрезка B: ')
while type(B) != int:
    try:
        B = int(B)
    except ValueError:
        print('Неправильно ввели')
        B = input('Введите длину отрезка B: ')
while B < 0:
    print('Неправильно ввели')
    B = input('Введите длину отрезка B: ')
    while type(B) != int:
        try:
            B = int(B)
        except ValueError:
            print('Неправильно ввели')
            B = input('Введите длину отрезка B: ')
C = 0
c = 0
while C < A:
    C += B
    if C < A:
        c += 1
print('Количество отрезков B на отрезке A =', c)

```

Протокол работы программы:

Введите длину отрезка A: 21

Введите длину отрезка B, которая меньше A: 20

Количество отрезков B на отрезке A = 1

Process finished with exit code 0

Вывод: В процессе выполнения практического занятия выработал навыки составления программ циклической структуры в IDE PyCharm Community. Были использованы языковые конструкции while. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.