

Практическое занятие № 4

Тема: Составление программ циклической структуры в IDE PyCharm Community.

Цель: выработка первичных навыков работы с IDE PyCharm Community, составление программ линейной структуры.

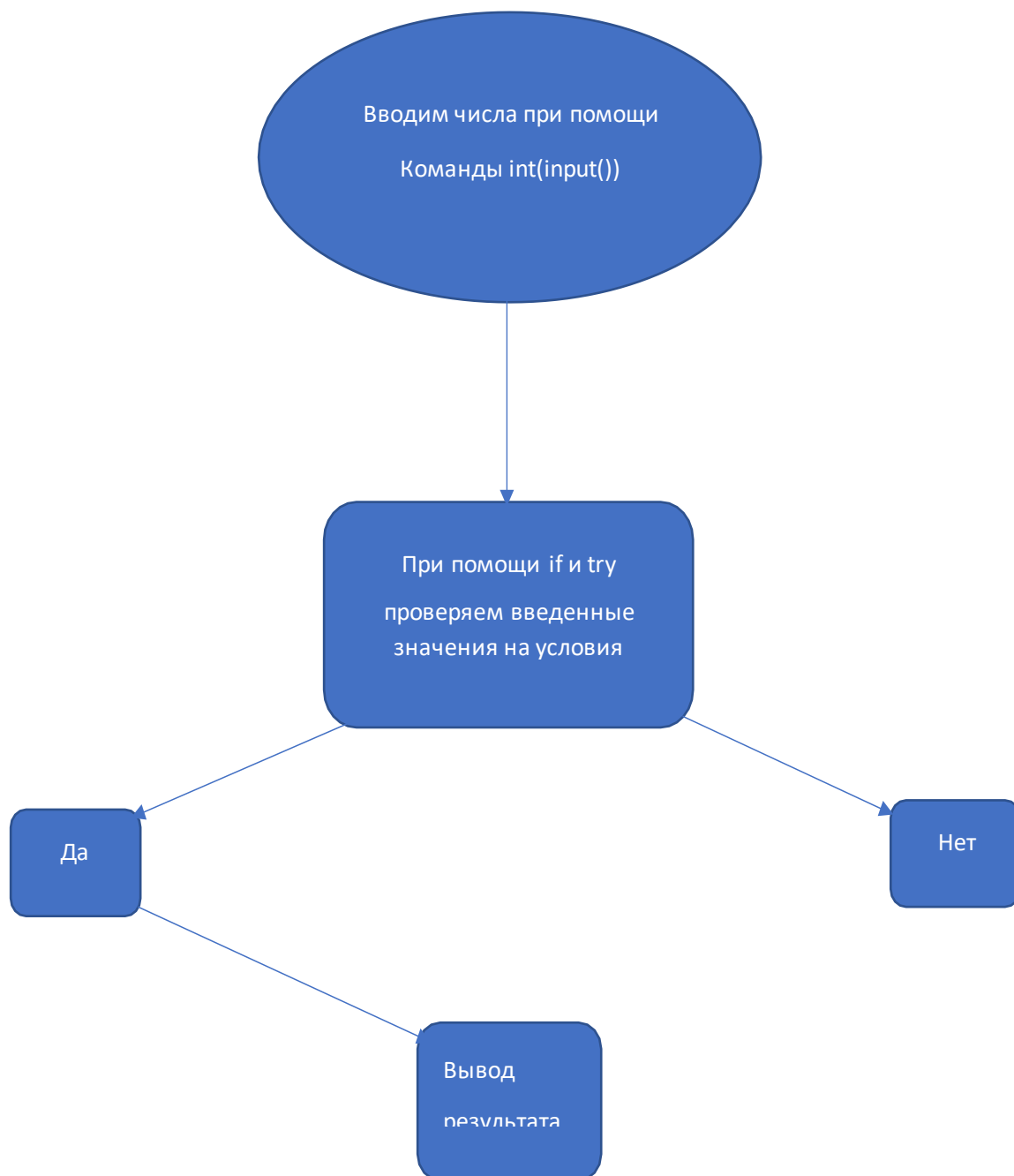
Задача 1

Постановка задачи.

Дано два целых числа A и B ($A < B$). Вывести в порядке возрастания все целые числа, расположенные между A и B (включая сами числа A и B), а также количество N этих чисел

Тип алгоритма: линейный.

Блок-схема алгоритма:



Текст программы:

```
# -*- coding: utf8 -*-
# ПЗ 4 вариант 26 задание 1
# Дано два целых числа А и В( $A < B$ ). Вывести в порядке возрастания все
# целые числа, расположенные между А и В(включая сами числа А и В), а также
# количество N этих чисел

def numbers_between(a, b):
    while a >= b:
        a = int(input("Первое число должно быть меньше второго. Пожалуйста, введите первое число заново: "))
        b = int(input("Теперь введите второе число: "))

    # Генератор списка для создания списка чисел
    number_list_descending = list(range(b, a - 1, -1))

    print("Числа между А и В в порядке убывания:", *number_list_descending)
    print("Количество чисел между А и В включительно:", len(number_list_descending))

a = int(input('Первое число: '))
b = int(input('Второе число: '))

numbers_between(a, b)
```

Протокол работы программы:

Первое число: 12

Второе число: 33

Числа между А и В в порядке убывания: 33 32 31 30 29 28 27 26 25 24 23 22
21 20 19 18 17 16 15 14 13 12

Количество чисел между А и В включительно: 22

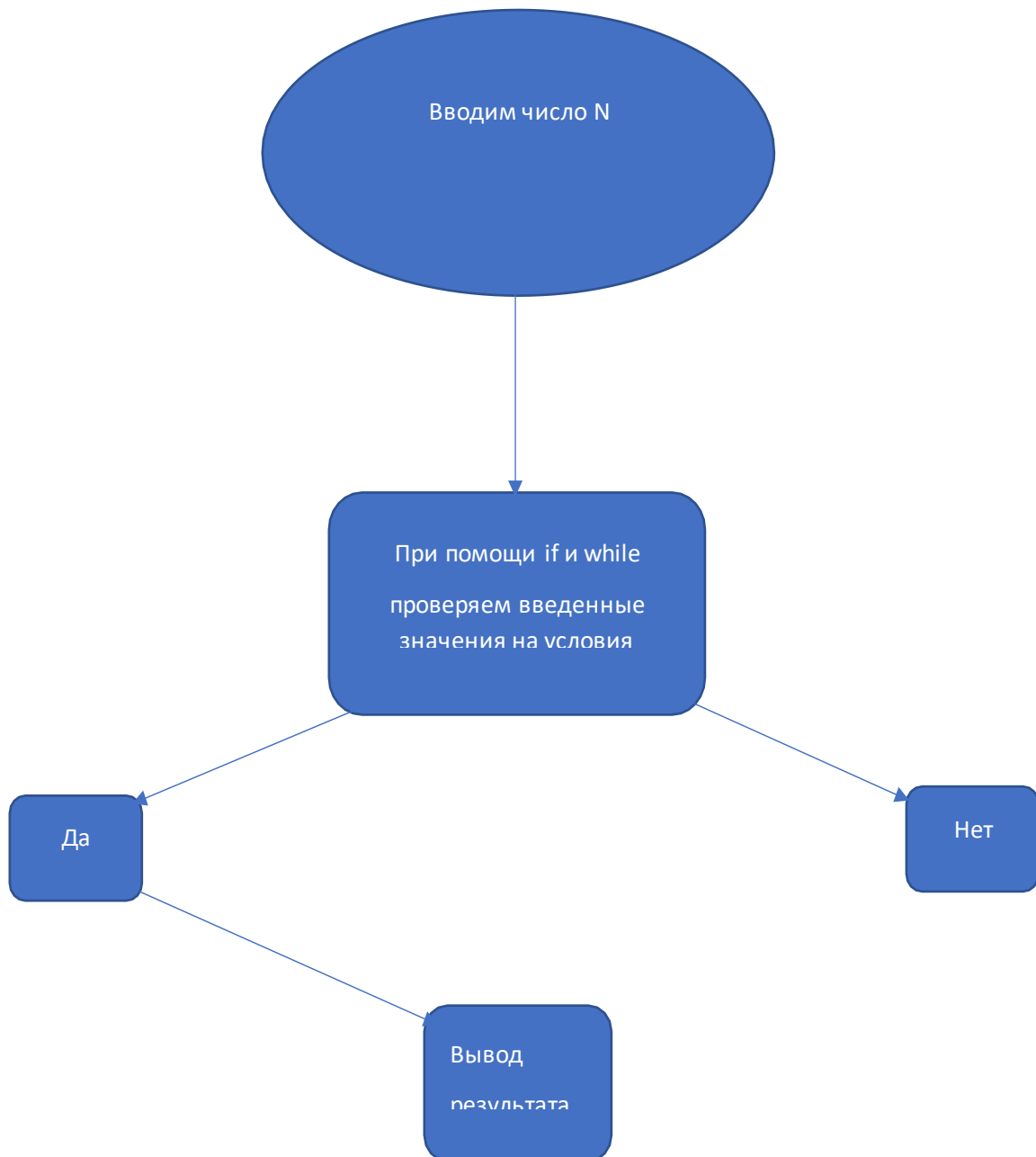
Программа 2

Постановка задачи.

Дано целое число N (>1). Вывести наибольшее из целых чисел K , для которых сумма $1 + 2 + \dots + K$ будет меньше или равна N , и саму эту сумму.

Тип алгоритма: линейный.

Блок-схема алгоритма:



Текст программы:

```
# -*- coding: utf8 -*-
# ПЗ 4 вариант 26 задание 2
# Дано целое число N(>1). Вывести наибольшее из целых чисел K, для которых сумма 1 + 2 + ... + K
# будет меньше или равна N, и саму эту сумму..

def find_largest_K(N):
    K = 0
    total_sum = 0
    while total_sum + (K + 1) <= N:
        K += 1
        total_sum += K
    return K, total_sum

N = 23 # твое целое число N
largest_K, sum_below_N = find_largest_K(N)
print("Наибольшее K:", largest_K)
print("Сумма до K:", sum_below_N)
```

Протокол работы программы:

Наибольшее K: 6

Сумма до K: 21

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ линейной структуры в IDE PyCharm Community В этой программе был использован цикл `while` для проверки условия $a \geq b$, чтобы гарантировать, что первое число меньше второго. При этом при несоблюдении условия запрашивались новые значения для переменных `a` и `b`. Функция возвращает наибольшее значение `K` и сумму до этого значения `K`.

Затем программа использует эту функцию, передавая целое число `N` в качестве аргумента, и выводит наибольшее значение `K` и сумму до этого значения `K` с помощью `print`.

Таким образом, вы использовали функции, цикл `while`, арифметические вычисления для достижения поставленной цели. Программа также запрашивает у пользователя ввод значений `a` и `b` для начала выполнения. Выполнены разработка кода, отладка, тестирование, оптимизация кода. Готовые программные коды выложены на [GitHub](#).