

### Практическое задание №3

**Тема:** составление программ ветвящейся структуры в IDE PyCharm Community

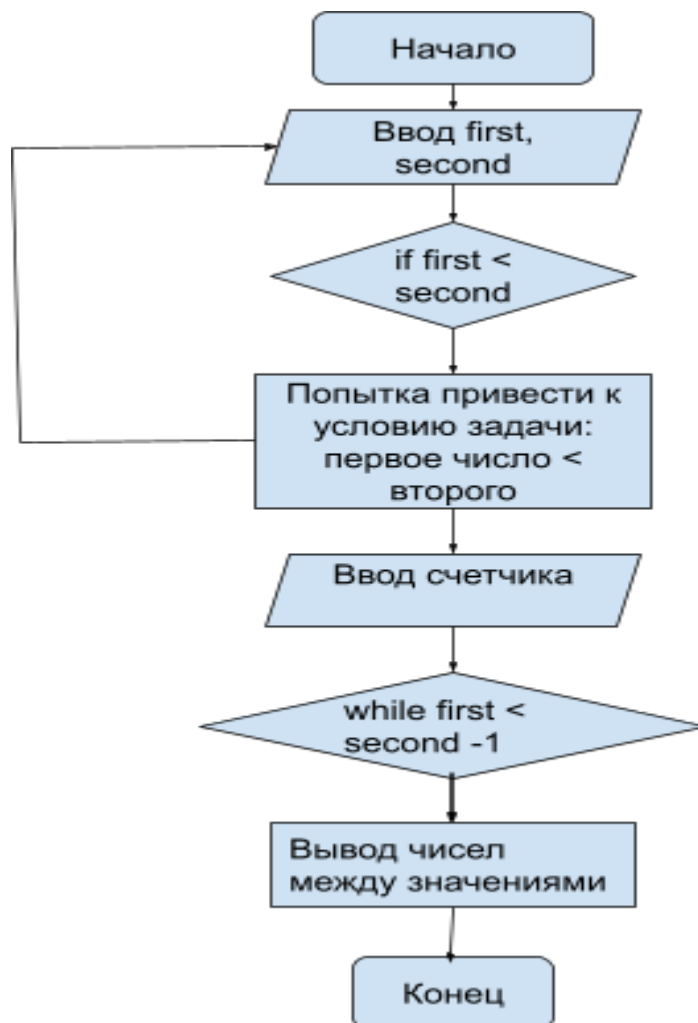
**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ ветвящейся структуры в IDE PyCharm Community

#### Постановка задачи №1:

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

**Тип алгоритма:** циклический

**Блок-схема алгоритма:**



**Текст программы:**

**def first\_task():**

**try:**

**first = int(input('Введите любое целое число A: '))**

**second = int(input('Введите любое целое число, которое больше A: '))**

**if first < second:**

**counter = 0**

**while first < second-1:**

**second -= 1**

**counter += 1**

**print(second)**

**print('Количество целых цифр между введенными значениями: ', counter)**

**else:**

**print('Введены некорректные данные!')**

**first\_task()**

**except:**

**print('Введены некорректные данные!')**

**first\_task()**

**first\_task()**

**Протокол работы программы:**

**Введите любое целое число A: 1**

**Введите любое целое число, которое больше A: 7**

**6**

**5**

**4**

**3**

**2**

**Количество целых цифр между введенными значениями: 5**

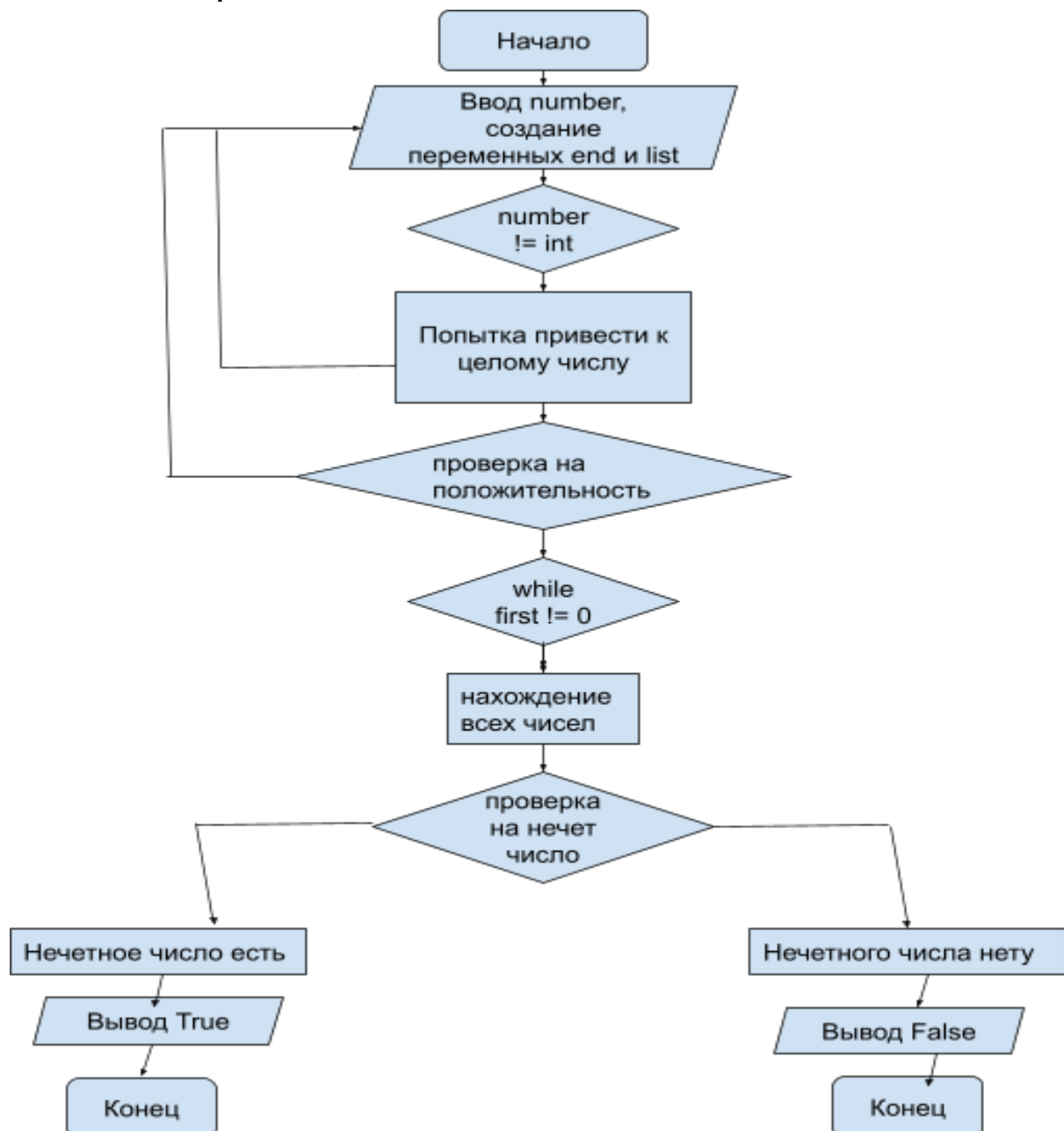
**Process finished with exit code 0**

## Постановка задачи №2:

Дано целое число  $N (>0)$ . С помощью операций деления нацело и взятия остатка от деления определить, имеются ли в записи числа  $N$  нечетные цифры. Если имеются, то вывести TRUE, если нет - вывести FALSE.

Тип алгоритма: циклический

Блок-схема алгоритма:



Текст программы:

```
def second_task():  
    list = []
```

```
end = False
try:
    number = int(input('Введите любое целое число N, которое больше 0: '))
    if number > 0:
        div = 10
        first = 1
        ost = number % 10
        list.append(ost)
        while first != 0:
            first = number // div
            div = div * 10
            ost = first % 10
            list.append(ost)
        for i in list:
            if i % 2 != 0:
                end = True
                print(end)
                exit()
        print(end)

    else:
        print('Введены некорректные данные')
        second_task()
except ValueError:
    print('Введены некорректные данные')
    second_task()
```

second\_task()

Протокол работы программы:

Текст программы: `def second_task():`

```
list = []
end = False
try:
    number = int(input('Введите любое целое число N, которое больше 0: '))
    if number > 0:
        div = 10
        first = 1
        ost = number % 10
        list.append(ost)
        while first != 0:
            first = number // div
            div = div * 10
            ost = first % 10
            list.append(ost)
        for i in list:
            if i % 2 != 0:
                end = True
                print(end)
                exit()
        print(end)

    else:
        print('Введены некорректные данные')
        second_task()
except ValueError:
    print('Введены некорректные данные')
    second_task()
```

`second_task()`

Протокол работы программы:

Введите любое целое число N, которое больше 0: 1  
True - В вашем числе присутствуют нечетные числа!

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 *try*, *except*, *if*, *def*.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.