

Практическое занятие № 4.1

Тема: Составление программ циклический в IDE PyCharm Community.

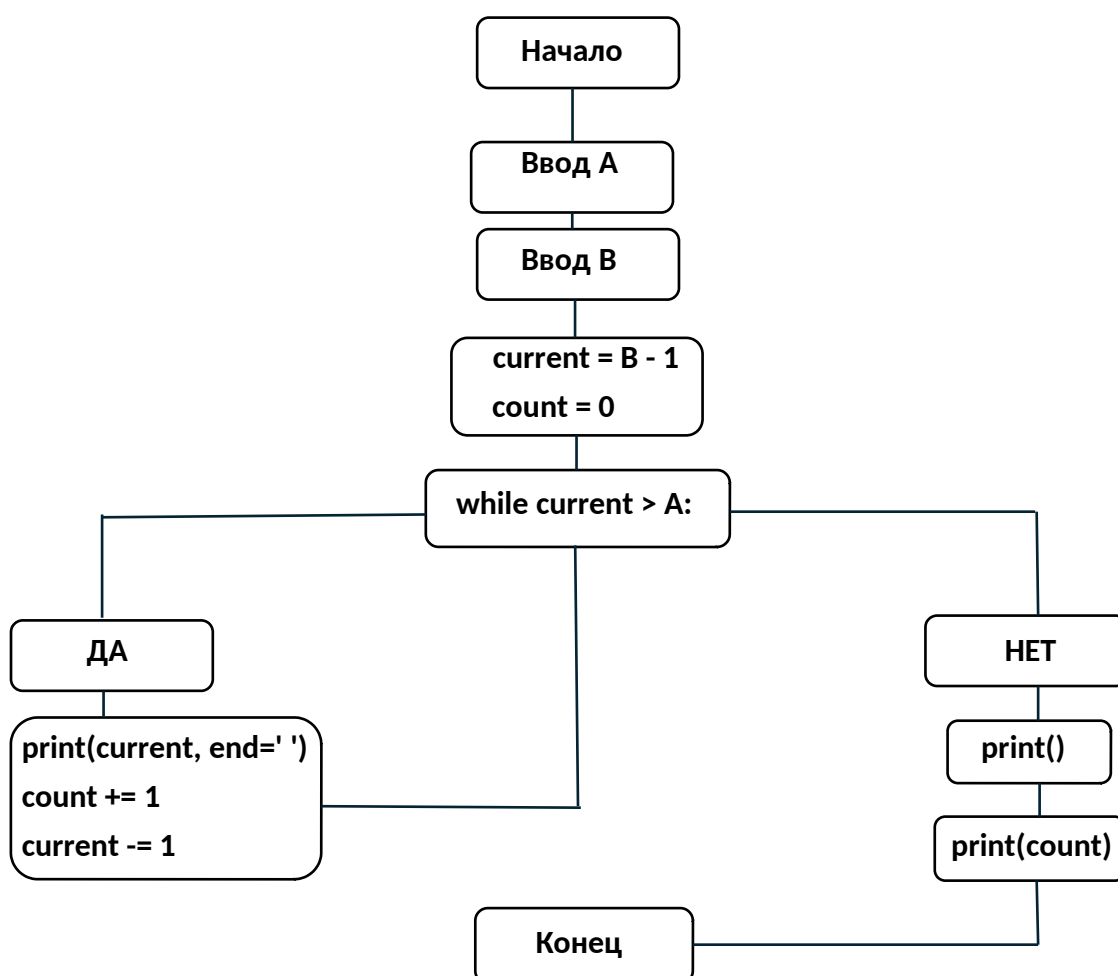
Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ циклической структуры в IDE PyCharm Community.

Постановка задачи.

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:

```
##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.  
  
try:  
    A = int(input("Введите A: "))  
    B = int(input("Введите B: "))  
  
    current = B - 1 # Инициализация переменной current, начиная с числа, предшествующего B  
    count = 0 # Инициализация счетчика для подсчета чисел  
  
    # Цикл, который выполняется, пока current больше A  
    while current > A:  
        print(current, end=' ') # Выводим текущее значение 'current' без перехода на новую строку  
  
        # Увеличиваем счетчик чисел  
        count += 1  
  
        # Переходим к следующему меньшему числу  
        current -= 1  
  
    print() # Переход на новую строку после вывода всех чисел  
    print(count)  
  
except ValueError:  
    print("Ошибка: Пожалуйста, введите целые числа.")
```

Протокол работы программы:

Введите A: 10

Введите B: 20

19 18 17 16 15 14 13 12 11

9

Process finished with exit code 0

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

Практическое занятие № 4.2

Тема: Составление программ циклический в IDE PyCharm Community.

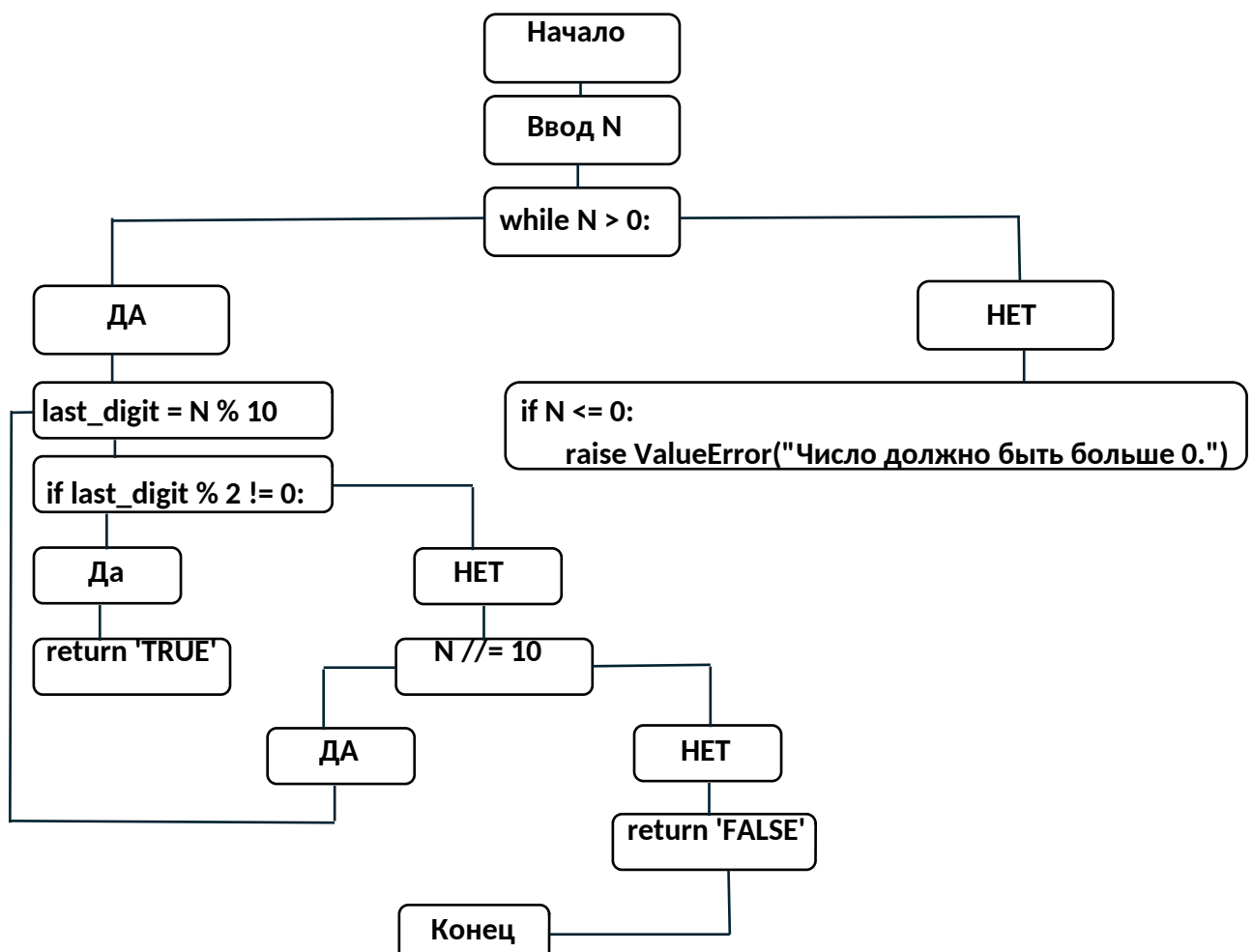
Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ циклической структуры в IDE PyCharm Community.

Постановка задачи.

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:

```
##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.

def has_odd_digit(N): 1 usage  NiceKirill
    # Цикл, который выполняется, пока N больше 0
    while N > 0:
        # Извлечение последней цифры числа N с помощью операции остатка от деления
        last_digit = N % 10

        # Проверка, является ли последняя цифра нечетной
        if last_digit % 2 != 0:
            # Если найдена нечетная цифра, возвращаем 'TRUE'
            return 'TRUE'

        # Убираем последнюю цифру из N, используя целочисленное деление
        N //= 10

    # Если все цифры были четными, возвращаем 'FALSE'
    return 'FALSE'

try:
    # Чтение целого числа N из ввода
    N = int(input("Введите целое число N (>0): "))

    # Проверка, что N положительное
    if N <= 0:
        raise ValueError("Число должно быть больше 0.")

    # Вывод результата проверки на наличие нечетных цифр
    print(has_odd_digit(N))

except ValueError as e:
    # Обработка ошибки, если ввод некорректен
    print('Ошибка:', e)
```

Протокол работы программы:

Введите целое число N (>0): 2

FALSE

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ ветвящейся структуры в IDE PyCharm Community. Были использованы языковые конструкции Try, Except, While, if, raise, return. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.