

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.

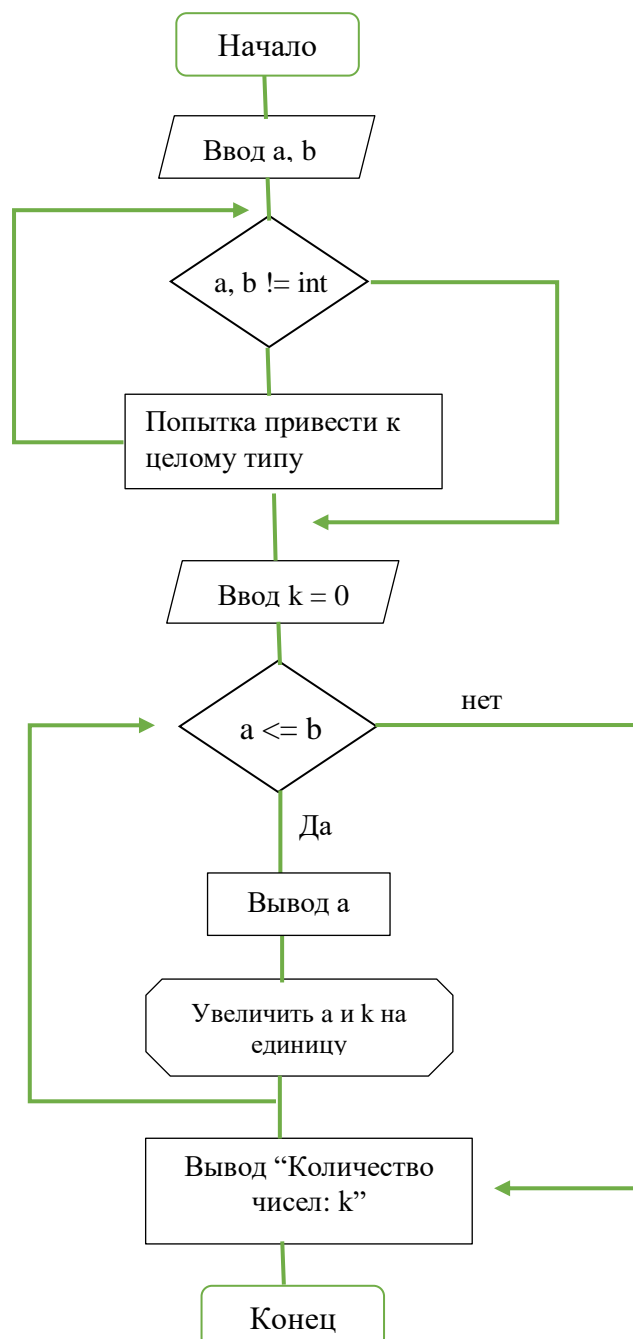
Задача 1:

Постановка задачи.

Разработать программу, где даны 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: Циклическая

Блок-схема алгоритма:



Текст программы:

```
#Даны два целых числа А и В (А < В). Вывести в порядке возрастания все целые  
#числа, расположенные между А и В (включая сами числа А и В), а также  
#количество  
#N этих чисел  
a, b = input("Введите первое число: "), input("Введите второе число: ")  
while type(a) != int: # обработка исключений  
    try:  
        a = int(a)  
    except ValueError:  
        print("Неправильно ввели!")  
        a = input("Введите первое число: ")  
while type(b) != int: # обработка исключений  
    try:  
        b = int(b)  
    except ValueError:  
        print("Неправильно ввели!")  
        b = input("Введите второе число: ")  
k = 0  
while a <= b:  
    print(a)  
    a += 1  
    k += 1  
print('Количество чисел: ', k)
```

Протокол работы программы:

Введите первое число: 4

Введите второе число: 9

4

5

6

7

8

9

Количество чисел: 6

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработала навыки составления программ цикличной структуры в IDE PyCharm Community. Была использована языковая конструкция while.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.

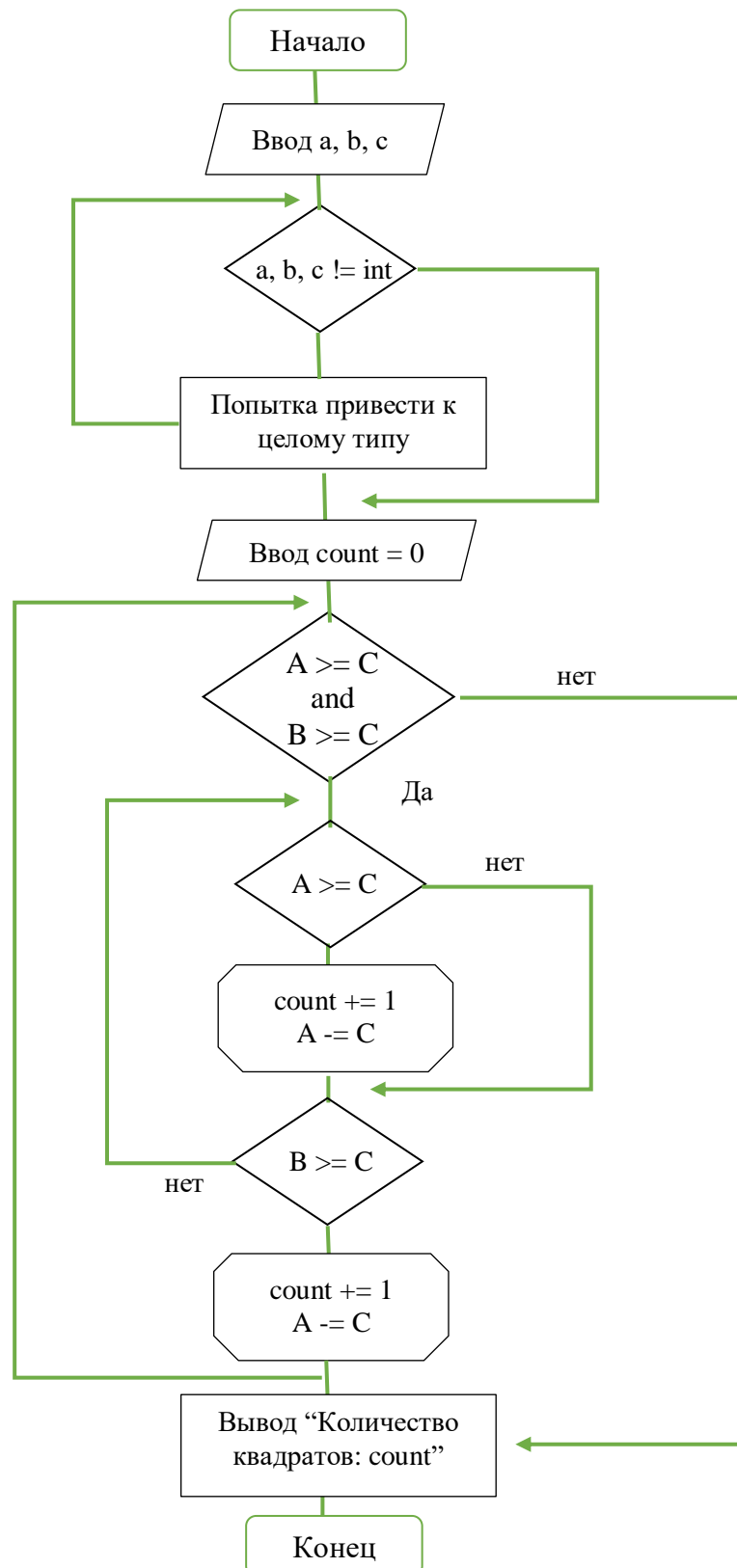
Задача 2:

Постановка задачи.

Разработать программу, где даны положительные числа A , B , C . На прямоугольнике размера $A \times B$ размещено максимально возможное количество квадратов со стороной C (без наложений). Найти количество квадратов, размещенных на прямоугольнике. Операции умножения и деления не использовать.

Тип алгоритма: Циклическая

Блок-схема алгоритма:



Текст программы:

```
#Найти количество квадратов, размещенных на прямоугольнике. Операции
#умножения и деления не использовать.

A = input("Введите длину прямоугольника: ")
B = input("Введите ширину прямоугольника: ")
C = input("Введите длину стороны квадрата: ")
while type(A) != int: # обработка исключений
    try:
        A = int(A)
    except ValueError:
        print("Неправильно ввели!")
        A = input("Введите длину прямоугольника: ")
while type(B) != int: # обработка исключений
    try:
        B = int(B)
    except ValueError:
        print("Неправильно ввели!")
        B = input("Введите ширину прямоугольника: ")
while type(C) != int: # обработка исключений
    try:
        C = int(C)
    except ValueError:
        print("Неправильно ввели!")
        C = input("Введите длину стороны квадрата: ")
# вводим счетчик квадратов
count = 0
# Размещаем квадраты на прямоугольнике
while A >= C and B >= C:
    # Размещаем квадрат по горизонтали
    while A >= C:
        count += 1
        A -= C
    # Размещаем квадрат по вертикали
    while B >= C:
        count += 1
        B -= C
# Выводим результат
print("Количество размещенных квадратов:", count)
```

Протокол работы программы:

Введите длину прямоугольника: 700
Введите ширину прямоугольника: 436
Введите длину стороны квадрата: 30
Количество размещенных квадратов: 37
Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработала навыки составления программ цикличной структуры в IDE PyCharm Community. Была использована языковая конструкция while.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.