

Практическое занятие №3

Тема: составление программ ветвящейся структуры в IDEPyCharmCommunity.

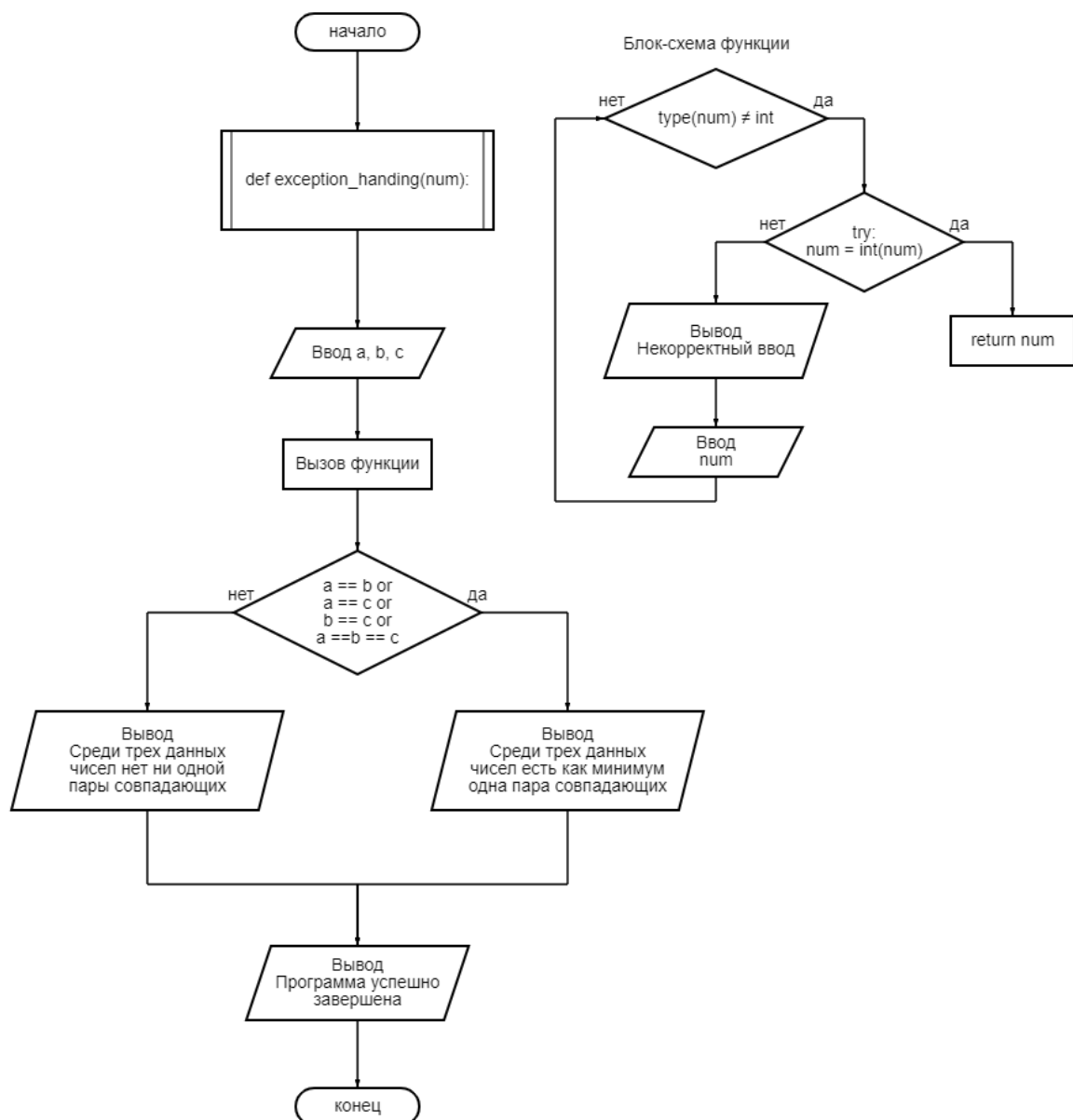
Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ ветвящейся структуры в IDEPyCharmCommunity.

Постановка задачи №1:

Проверить истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара совпадающих».

Тип алгоритма: ветвящийся.

Блок-схема алгоритма:



Текст программы:

```
# Проверить истинность высказывания: "Среди трех данных целых чисел
есть хотя бы одна пара совпадающих".

# Обработка исключений
def exception_handling(num):
    while type(num) != int:
        try:
            num = int(num)
        except ValueError:
            print(f'Некорректный ввод - {num}!')
            num = input('Введите новое целое число вместо этого: ')
    else:
        return num

# Ввод данных
a, b, c = input('Введите первое целое число: '), input('Введите второе
целое число: '), input('Введите третье целое число: ')

a = exception_handling(a)
b = exception_handling(b)
c = exception_handling(c)

# Проверка истинности высказывания
if a == b or a == c or a == b == c or b == c:
    print('Среди трех данных чисел есть как минимум одна пара
совпадающих')
else:
    print('Среди трех данных числе нет ни одной пары совпадающих')
print('Программа успешно завершена!')
```

Протокол работы программы:

Введите целое число: 5

Введите целое число: 2

Введите целое число: 5

Среди трех данных чисел есть как минимум одна пара совпадающих

Программа успешно завершена!

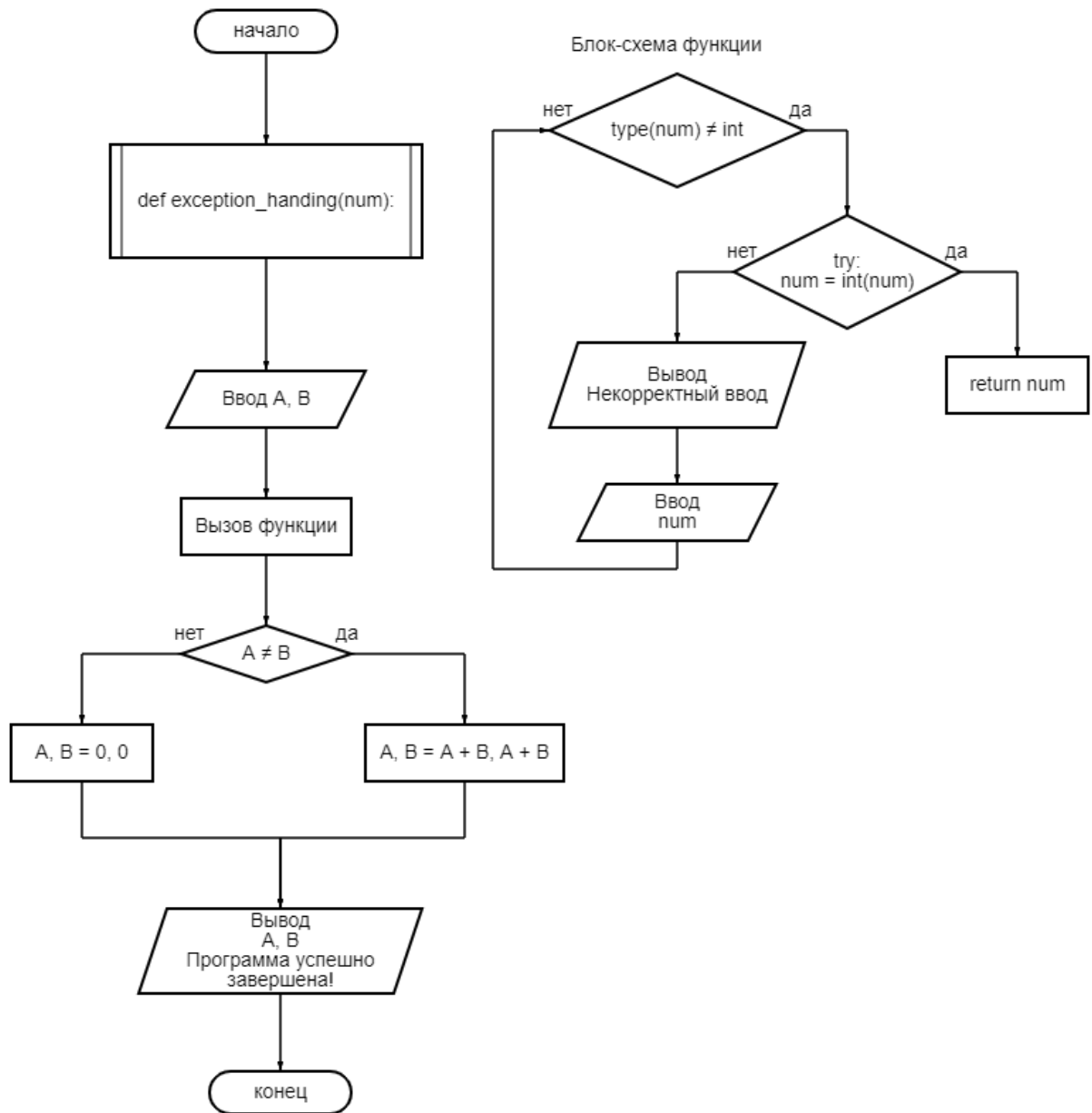
Process finished with exit code 0

Постановка задачи №2:

Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных A и B.

Тип алгоритма: ветвящийся.

Блок-схема алгоритма:



Текст программы:

Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной сумму этих значений, # а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных A и B.

Обработка исключений

```
def exception_handling(num):  
    while type(num) != int:  
        try:  
            num = int(num)  
        except ValueError:  
            print(f'Некорректный ввод - {num}!')  
            num = input('Введите новое целое число вместо этого: ')
```

```
        else:
            return num

# Ввод данных
A, B = input('Введите целое число: '), input('Введите целое число: ')

A = exception_handling(A)
B = exception_handling(B)

# Проверка условий
if A != B:
    A, B = A + B, A + B
else:
    A, B = 0, 0

print(f'A = {A}, B = {B}\n'
      f'Программа успешно завершена!')
```

Протокол работы программы:

Введите целое число: 3

Введите целое число: 6

A = 9, B = 9

Программа успешно завершена!

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал первичные навыки составления программ ветвящейся структуры.

Выполнена разработка, отладка, тестирование и оптимизация программного кода.