

Практическое занятие №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.

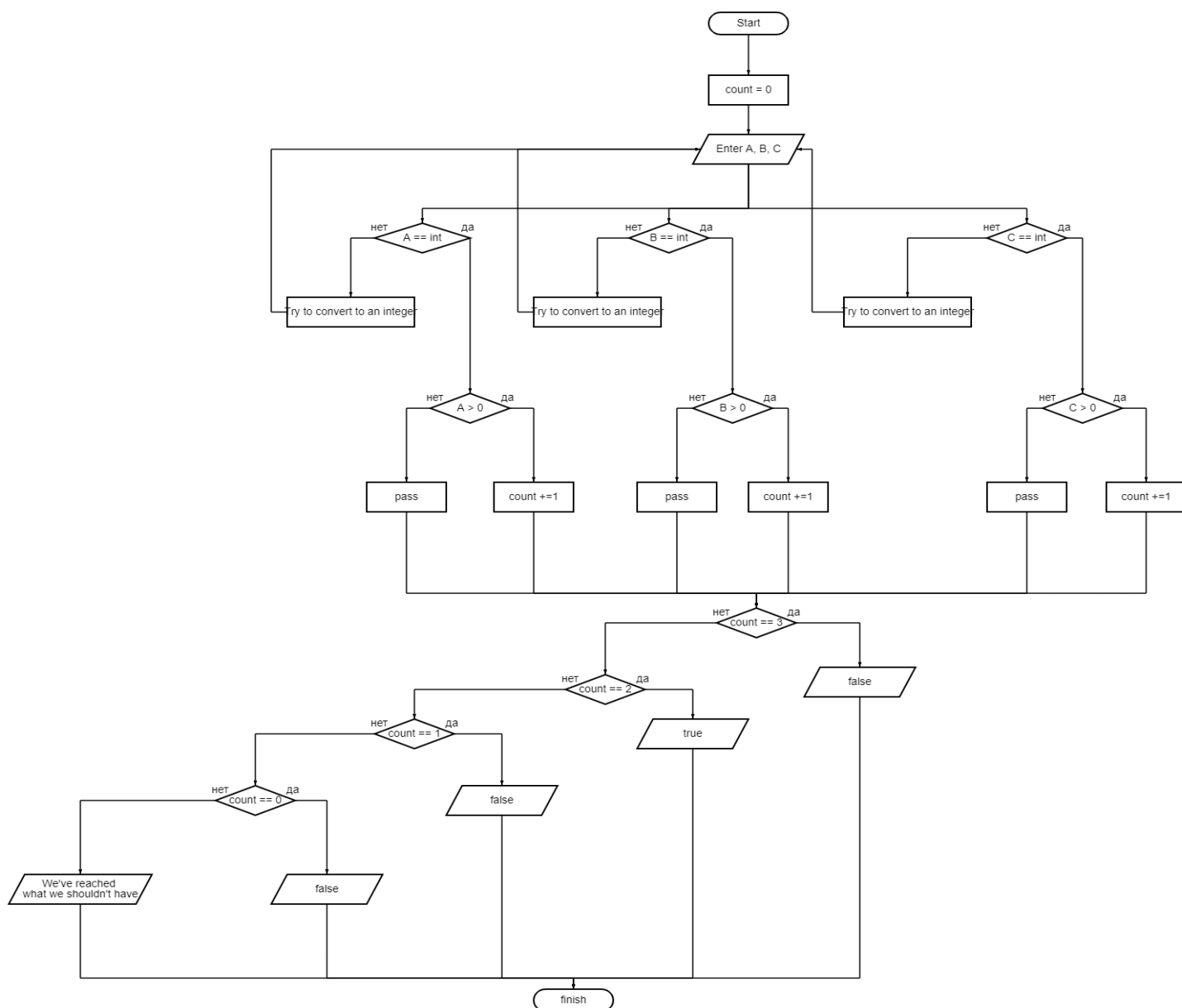
Постановка задачи.

1. Разработать программу для ввода трёх чисел (A, B, C), и вывода результата высказывания (высказывание: Ровно два из чисел A, B, C являются положительными).
2. Разработать программу для ввода трёх чисел, одно из которых отлично от двух других, равных между собой, и вывода порядкового номера числа, отличного от остальных.

Тип алгоритма: алгоритм с ветвлением.

Блок-схема алгоритма:

1.



Текст программы:

1.

```

# Задача: Даны три целых числа: A, B, C.
# Проверить истинность высказывания: «Ровно два из чисел A, B, C являются положительными»
# -> 1 5 -10
# <- true

# примечание: в этом решении 0 (ноль) воспринимается как отрицательное число

# переменные
count = 0
a = input("Enter one number: ")
b = input("Enter two number: ")
c = input("Enter three number: ")

# обработка исключений
while type(a) != int:

```

```

try:
    a = int(a)
    if a > 0: count += 1
except ValueError:
    a = input("Enter one number: ")

while type(b) != int:
    try:
        b = int(b)
        if b > 0: count += 1
    except ValueError:
        b = input("Enter two number: ")

while type(c) != int:
    try:
        c = int(c)
        if c > 0: count += 1
    except ValueError:
        c = input("Enter three number: ")

# проверка на истинность условия и вывод результата
if count == 3:
    print("(^o^o)^ ~ ^ false")
    print("Exactly three out of three numbers are positive")

elif count == 2:
    print("(^o^o)^ ~ ^ true")
    print("Exactly two out of three numbers are positive")

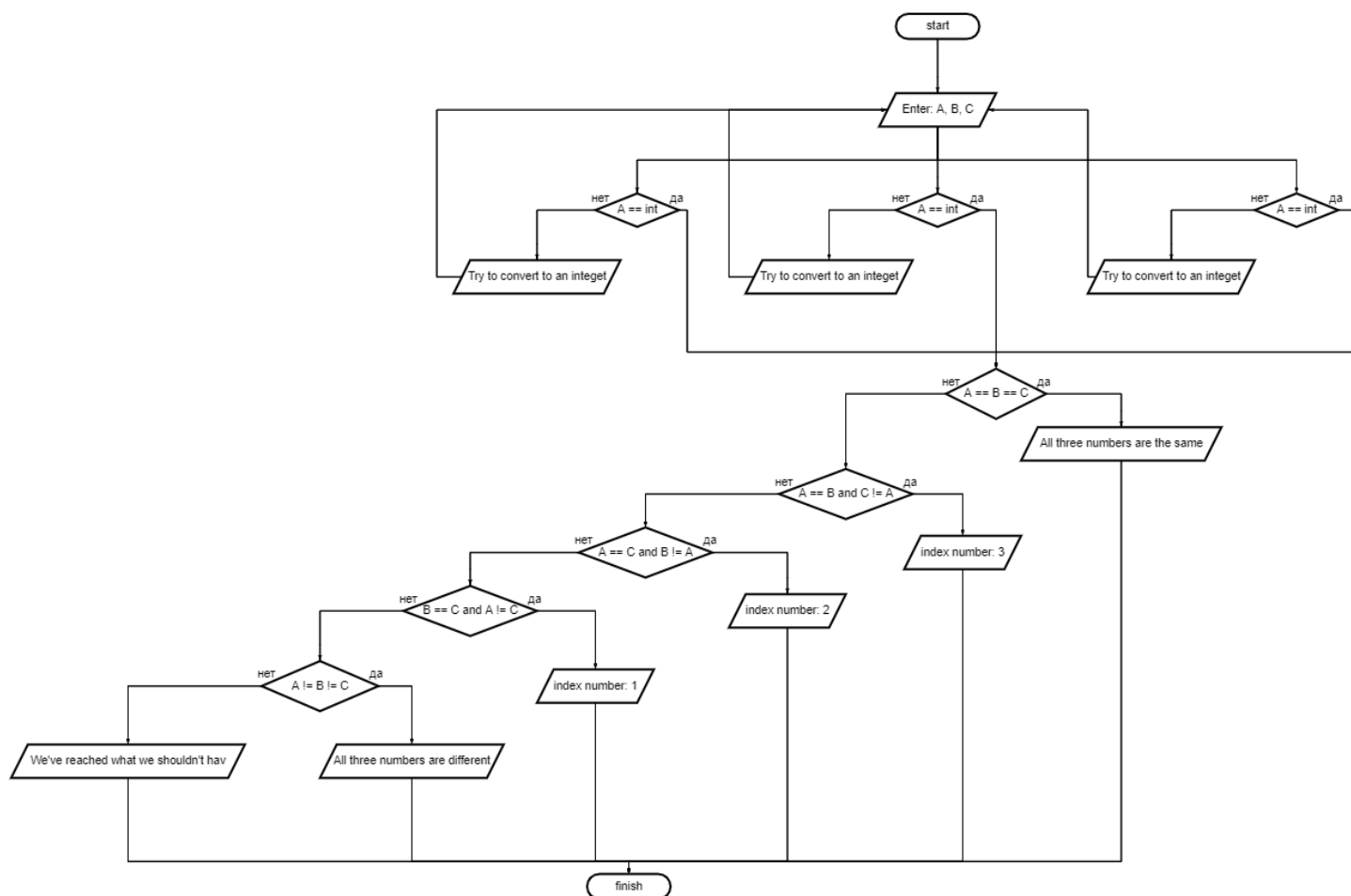
elif count == 1:
    print("(^o^o)^ ~ ^ false")
    print("Exactly one out of three numbers are positive")

elif count == 0:
    print("(^o^o)^ ~ ^ false")
    print("None of the three numbers is positive")
else:
    print("We've reached what we shouldn't have")

```

Блок-схема алгоритма:

2.



Текст программы:

2.

```

# Даны три целых числа, одно из которых отлично от двух других, равных между собой.
# Определить порядковый номер числа, отличного от остальных

# -> 6 6 4
# <- 3

# примечание: в этом решение индексация (порядковый номер) начинается с 1 (единицы)

# переменные
a = input("Enter one number: ")
b = input("Enter two number: ")
c = input("Enter three number: ")

# обработка исключений
while type(a) != int:
    try:
        a = int(a)
    except ValueError:
        a = input("Enter one number: ")

while type(b) != int:
    try:
        b = int(b)
    except ValueError:
        b = input("Enter two number: ")
  
```

```
while type(c) != int:
    try:
        c = int(c)
    except ValueError:
        c = input("Enter three number: ")

# проверка на истинность условия и вывод результата
print("`•^•`")

if a == b == c:
    print("All three numbers are the same")
elif a == b and c != a:
    print("index number: 3")
elif a == c and b != a:
    print("index number: 2")
elif b == c and a != c:
    print("index number: 1")
elif a != b != c:
    print("All three numbers are different")
else:
    print("We've reached what we shouldn't have")
```

Вывод: в процессе выполнения практического занятия выработала навыки работы составления алгоритмов ветвления. Были использованы языковые конструкции try...except...else; if...else.

Выполнено: разработка, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.