

Практическое задание №3

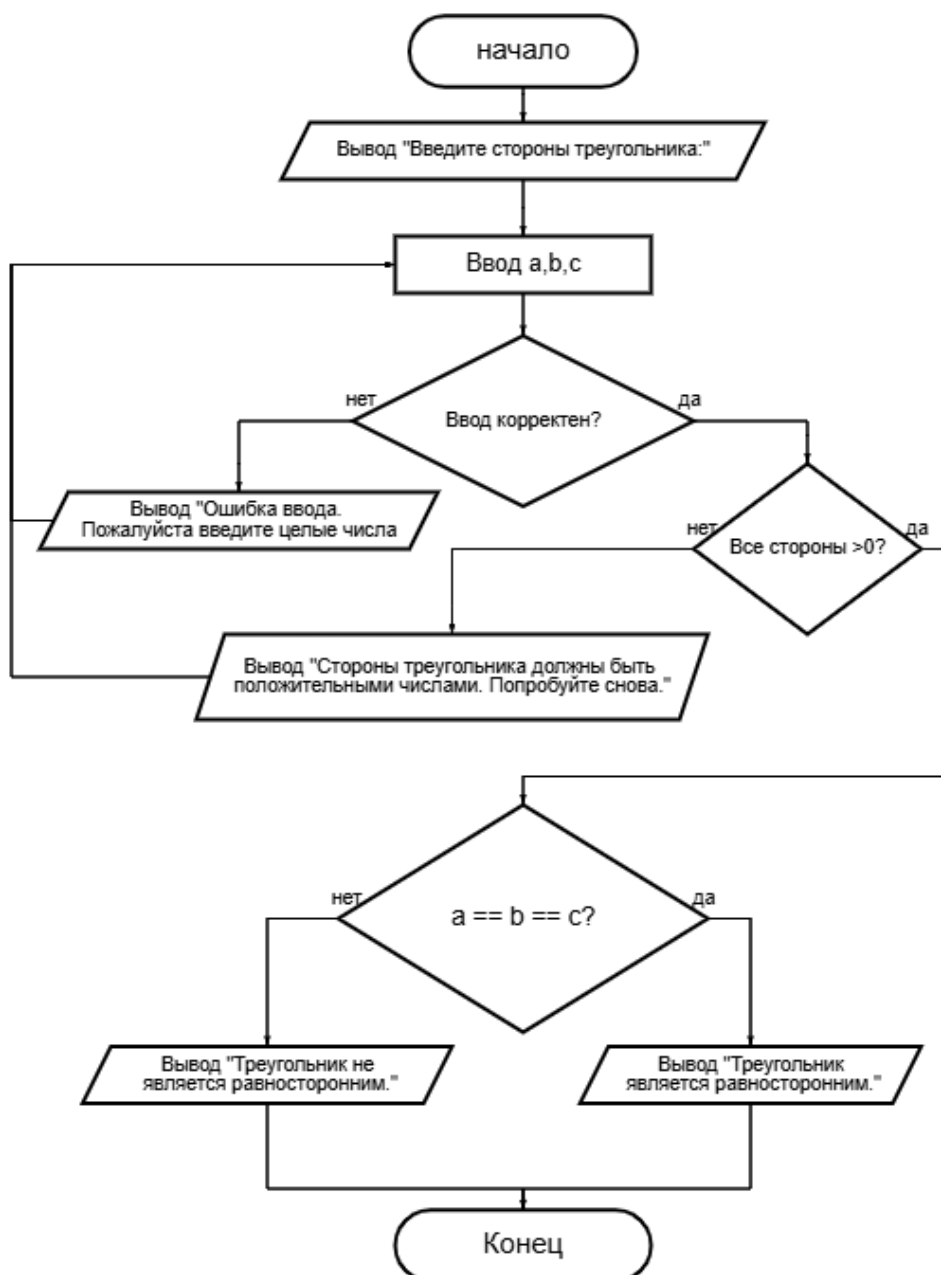
Тема: составление программ ветвящейся структуры в IDE PyCharm Community.

Цели практического занятия: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления составление программ ветвящейся структуры в IDE PyCharm Community.

Постановка задачи №1: Даны целые числа a , b , c , являющиеся сторонами некоторого треугольника. Проверить истинность высказывания: «Треугольник со сторонами a , b , c является равносторонним».

Тип алгоритма: ветвящейся

Блок-схема алгоритма:



Текст программы:

```
1  #Даны целые числа a, в, с, являющиеся сторонами некоторого треугольника.
2  #Проверить истинность высказывания: «Треугольник со сторонами a, b, c является равносторонним».
3
4  while True:
5      try:
6          print("Введите стороны треугольника (целые положительные числа):")
7          a = int(input("Сторона a: "))
8          b = int(input("Сторона b: "))
9          c = int(input("Сторона c: "))
10
11         if a > 0 and b > 0 and c > 0:
12             break
13         else:
14             print("Стороны треугольника должны быть положительными числами. Попробуйте снова.")
15     except ValueError:
16         print("Ошибка ввода. Пожалуйста, введите целые числа.")
17
18     if a == b == c:
19         print("Треугольник является равносторонним.")
20     else:
21         print("Треугольник не является равносторонним.")
```

Протокол работы программы:

Введите стороны треугольника (целые положительные числа):

Сторона a: -6

Сторона b: 0

Сторона c: 10

Стороны треугольника должны быть положительными числами. Попробуйте снова.

Введите стороны треугольника (целые положительные числа):

Сторона a: 0.8

Ошибка ввода. Пожалуйста, введите целые числа.

Введите стороны треугольника (целые положительные числа):

Сторона a: 4

Сторона b: 4

Сторона c: 4

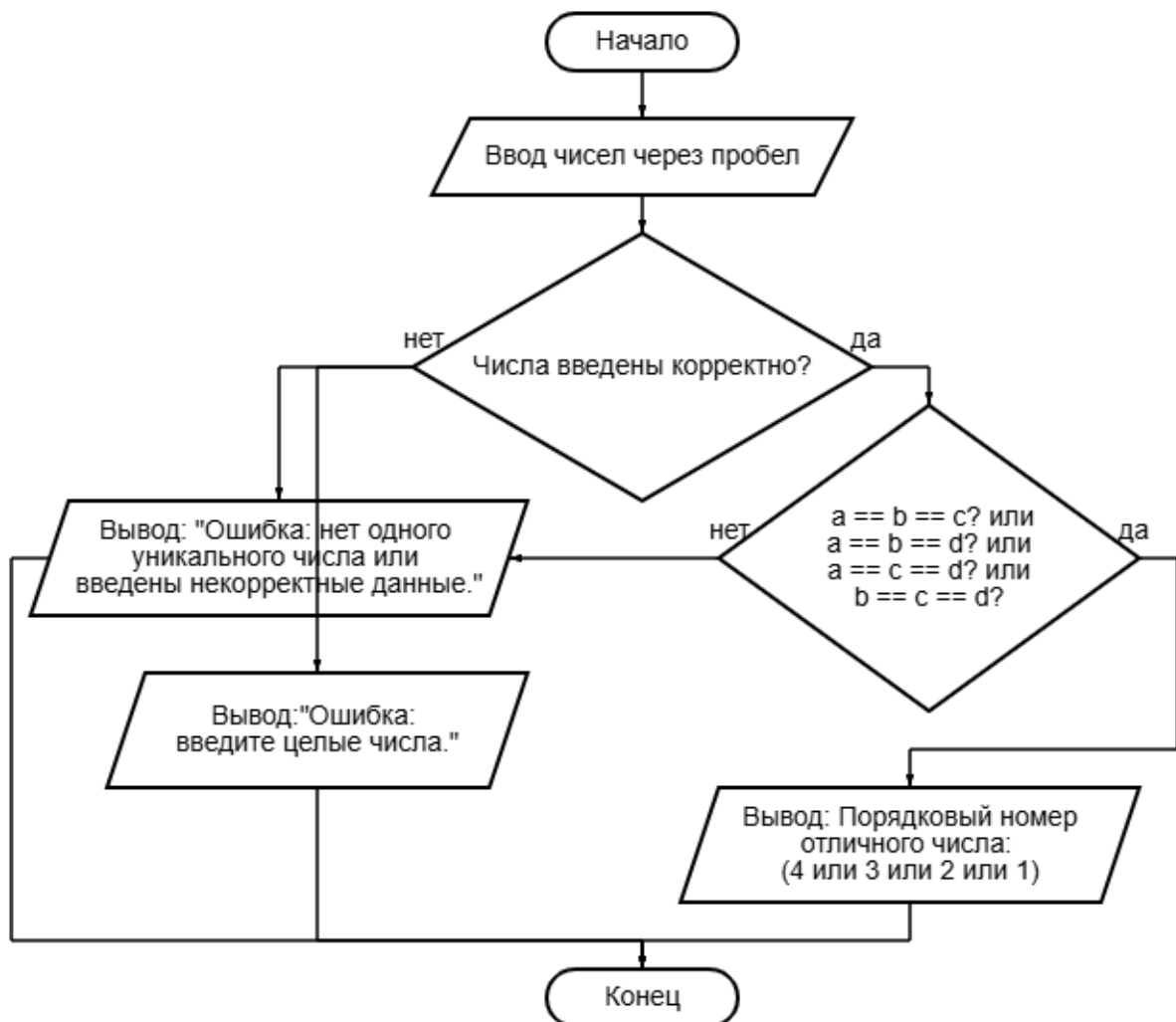
Треугольник является равносторонним.

Process finished with exit code 0

Постановка задачи №2: Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Определить порядковый номер числа, отличного от остальных.

Тип алгоритма: ветвящейся

Блок-схема алгоритма:



Текст программы:

```

1      # Даны четыре целых числа, одно из которых отлично от трех других, равных между собой.
2      #Определить порядковый номер числа, отличного от остальных.
3
4      try:
5          a, b, c, d = map(int, input("Введите четыре целых числа через пробел: ").split())
6
7          if a == b == c:
8              print("Порядковый номер отличного числа: 4")
9          elif a == b == d:
10             print("Порядковый номер отличного числа: 3")
11          elif a == c == d:
12             print("Порядковый номер отличного числа: 2")
13          elif b == c == d:
14             print("Порядковый номер отличного числа: 1")
15          else:
16             print("Ошибка: нет одного уникального числа или введены некорректные данные.")
17      except ValueError:
18          print("Ошибка: введите целые числа.")

```

Протокол работы программы:

Введите четыре целых числа через пробел: 1 2 2 2

Порядковый номер отличного числа: 1

Process finished with exit code 0

Вывод: я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления составление программ ветвящейся структуры в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.