**Подвиг 8.** С помощью модуля abc можно определять не только абстрактные методы, но и абстрактные объекты-свойства (property). Делается это следующим образом:

from abc import ABC, abstractmethod

class Transport(ABC):

@abstractmethod

def go(self):

"""Метод для перемещения транспортного средства"""

@property

@abstractmethod

def speed(self):

"""Абстрактный объект-свойство"""

Используя эту информацию и информацию о модуле abc из подвига 6, объявите базовый класс с именем **CountryInterface** со следующими абстрактными методами и свойствами:

name - абстрактное свойство (property), название страны (строка);  
population - абстрактное свойство (property), численность населения (целое положительное число);  
square - абстрактное свойство (property), площадь страны (положительное число);

get\_info() - абстрактный метод для получения сводной информации о стране.

На основе класса CountryInterface объявите дочерний класс Country, объекты которого создаются командой:

country = Country(name, population, square)

В самом классе Country должны быть переопределены следующие свойства и методы базового класса:

name - свойство (property) для считывания названия страны (строка);  
population - свойство (property) для записи и считывания численности населения (целое положительное число);  
square - свойство (property) для записи и считывания площади страны (положительное число);

get\_info() - метод для получения сводной информации о стране в виде строки:

"<название>: <площадь>, <численность населения>"

Пример использования классов (эти строчки в программе писать не нужно):

country = Country("Россия", 140000000, 324005489.55)

name = country.name

pop = country.population

country.population = 150000000

country.square = 354005483.0

print(country.get\_info()) # Россия: 354005483.0, 150000000

P.S. В программе требуется объявить только классы. На экран выводить ничего не нужно.

from abc import ABC, abstractmethod

# здесь объявляйте классы