**Задание:**

В игре-стратегии есть солдаты и герои. У всех есть свойство, содержащее уникальный номер объекта, и свойство, в котором хранится принадлежность команде, а также уровни силы (случайное значение от 10 до 100) и жизненных показателей (изначально 100). У солдат есть метод "перемещаться за героем", который в качестве аргумента принимает объект типа "герой" и "воевать" (в аргументе указывается один солдат из армии противника), в результате чего солдат наносит противнику и получает ущерб от него (при достижении показателя жизни 0 солдат исчезает).

У героев есть метод движения по карте (задаются новые координаты его расположения) и метод увеличения собственного уровня по итогу формирования его армии.

В основной части программы создайте по одному герою для каждой команды. В цикле сгенерируйте объекты-солдаты. Их принадлежность команде определяется случайно! Солдаты разных команд добавляются в разные списки.

Измеряется длина списков солдат противоборствующих команд и выводится на экран. У героя, принадлежащего команде с более длинным списком, поднимается уровень.

**Задание 2**

class Point:

def \_\_init\_\_(self, x, y):

self.x = x

self.y = y

@property

def x(self):

return self.\_x

@x.setter

def x(self, value):

try:

self.\_x = float(value)

except ValueError:

raise ValueError('"x" must be a number') from None

@property

def y(self):

return self.\_y

@y.setter

def y(self, value):

try:

self.\_y = float(value)

except ValueError:

raise ValueError('"y" must be a number') from None

Используя за основу идею класса Point, создать классы Figure (можно абстрактный, используя пакет и класс ABC https://docs.python.org/3/library/abc.html),

а также классы Circle, Triangle, Rectangle с методами move, square, center, bounds и экземпляры данных классов для демонстрации поведения