

TRABAJO PRÁCTICO N°4

Asignatura: Programación Orientada a Objetos

Importante: recordar que utilizaremos un guión bajo inicial en cada nombre de atributo como convención para indicar que deben tratarse como atributo privado, mientras que todos los métodos serán considerados públicos.

Ejercicio 1

Implementar una clase Caja, con atributos correspondientes a su material y sus medidas de ancho, largo (profundidad) y alto. Generar métodos de clase para constructores específicos de cajaGrande, cajaChica, cajaCarton, cajaPlastica y definir al menos dos opciones más.

Implementar además un método estático, que permita determinar si una caja entra o no en una estantería de 50cm de ancho, 40cm de profundidad y 30cm de alto.

Ejercicio 2

Implementar una clase Vector, que permita almacenar una representación de vectores en dos dimensiones (x,y). Incorporar los métodos necesarios para que sea posible ejecutar las siguientes sentencias:

- a) `v1 = Vector(2, 3)`
- b) `v2 = Vector(1, 4)`
- c) `print(v1)` # resultado esperado: imprime 'Vector = (x : 2, y : 3) '
- d) `v3 = v1 + v2` # resultado esperado: `v3 = Vector(3,7)`
- e) `v4 = 5*v1` # resultado esperado: `v4 = Vector(10, 15)`
- f) `v5 = v1.invertir()` # resultado esperado: `v5 = Vector(-2, -3)`
- g) `for i in v2:`
 `print(i)` # resultado esperado: imprime 1, luego imprime 4
- h) `v6 = v2 - v1` # resultado esperado: `v6 = Vector(-1,1)`
- i) `v7 = len(v2)` # resultado esperado: 2 (cantidad de dimensiones del v2, es decir, x e y)
- j) `v8 = v1 + 4` # resultado esperado: `Vector(6,7)`
- k) `v9 = 9 + v1` # resultado esperado: `Vector(11, 12)`

Ejercicio 3

Implementar la clase Sobre, que permita modelar un sobre de papel con sus dimensiones de ancho y alto. Incorporar los métodos necesarios para que sea posible ejecutar las siguientes sentencias:

- a) `s1 = Sobre(10, 20)`
- b) `s2 = Sobre(8, 15)`
- c) `a = s2 < s1` # resultado esperado: True, ya que el área de s2 es menor al área de s1
- d) `b = s1 in s2` # resultado esperado: False, ya que s1 no podría colocarse dentro de s2
- e) `s3 = s1 / 2` # resultado esperado: `[Sobre(10,10), Sobre(10,10)]`, ya que recorta s1 en dos sobres más chicos, respetando el ancho