



## Trabajo practico N.º 2

### Herencia y composición

1- Realizar una clase CuentaBancaria con los siguientes atributos y métodos:

- Atributos: nombre y apellido del titular, saldo de la cuenta
- Métodos:
  - constructor: con saldo inicial en cero.
  - `__str__()`: devolver todos los atributos.
  - `saldo()`: mostrar el saldo de la cuenta
  - `ingresar(monto)`: ingresar un monto a la cuenta (verificar monto positivo)
  - `retirar(monto)`: retirar un monto a la cuenta (verificar monto positivo)

Las clases CuentaAhorro y CuentaCorriente heredan los atributos y métodos de CuentaBancaria.

La clase CuentaAhorro tiene además un atributo para determinar si la cuenta esta inactiva si el saldo es menor a 10000 y activa si es mayor. Tiene los siguientes métodos:

- `ingresar(monto)`: Se puede ingresar dinero invocando al método heredado.
- `retirar(monto)`: Se puede retirar si la cuenta esta activa. Debe invocar al método heredado.
- `__str__()`: devolver todos los atributos incluyendo si esta activa.

La clase CuentaCorriente tiene además un atributo de sobregiro inicializada en cero. Además los siguientes métodos:

- `ingresar(monto)`: Se puede ingresar, pero si hay sobregiro se reduce la cantidad agregada.
- `retirar(monto)`: Se puede retirar un monto superior al saldo, el dinero que se debe queda como sobregiro. Invocar al método heredado.
- `__str__()`: devolver todos los atributos incluyendo el sobregiro. Invocar al método heredado.

2- Realizar las siguientes clases:

Clase Inmueble:

- Atributos:
  - `identificador_inmobiliario` (numero)
  - `cantidad_habitaciones` (numero)
  - `area` (numero) en metros cuadrados
  - `direccion` (texto)
  - `precio_venta` (numero)
- Métodos:
  - `calcular_precio(valor_area)`: calcular y devolver `precio_venta = area * valor_area`
  - `__str__()`: devolver todos los atributos.

Clase Casa, hereda de Inmueble:

- Atributos:
  - `cantidad_pisos` (numero)
- Métodos:
  - `__str__()`: devolver todos los atributos.

Clase Departamento, hereda de Inmueble:

- Atributos:



- numero\_piso (numero)
- Métodos:
  - \_\_str\_\_(): devolver todos los atributos.

Clase CasaRural, hereda de Casa:

- Atributos:
  - distancia (numero) a la municipalidad mas cercana
  - altitud (numero)
- Métodos:
  - \_\_str\_\_(): devolver todos los atributos.

Clase CasaUrbana, hereda de Casa:

- Atributos:
  - tiene\_pileta (boolean)
- Métodos:
  - \_\_str\_\_(): devolver todos los atributos.

Clase DepartamentoFamiliar, hereda de Departamento:

- Atributos:
  - cantidad\_familiares (numero)
- Métodos:
  - \_\_str\_\_(): devolver todos los atributos.

Clase DepartamentoEstudio, hereda de Departamento:

- Atributos:
  - cantidad\_empleados (boolean)
- Métodos:
  - \_\_str\_\_(): devolver todos los atributos.

3- Definir una clase Punto con atributos x e y. Luego definir una clase Linea con dos atributos: punto\_a y punto\_b a partir de la clase Punto.

Agregar los siguientes métodos a la clase Linea:

- \_\_init\_\_(a, b): Constructor que recibe los puntos a y b instanciados previamente.
- \_\_str\_\_(): Retornar la posición de la linea en el siguiente formato ((xa,ya), (xb, yb))
- distancia(): Calcular y retornar la distancia entre los puntos.
- mover\_derecha(cantidad): Mover linea a la derecha
- mover\_izquierda(cantidad): Mover linea a la izquierda
- mover\_arriba(cantidad): Mover linea hacia arriba
- mover\_abajo(cantidad): Mover linea hacia abajo

4- Crea las siguientes clases:

- Motor:
  - Atributos: nivel\_aceite, temperatura, encendido
  - Métodos: arrancar(), detener()
- Rueda:



- Atributos: rodado, presión
- Métodos: inflar(), desinflar()
- Ventana:
  - Atributos: polarizado,
  - Métodos: abrir(), cerrar()
- Puerta: con una Ventana
  - Atributos: color
  - Métodos: abrir(), cerrar()
- Auto: con un motor, cuatro ruedas y dos puertas
  - Métodos: \_\_str\_\_ (con todos los atributos)

5- Desarrollar las siguiente clases en Python:

- Cancion:
  - Atributos: titulo, autor, duracion
  - Métodos: \_\_str\_\_()
- Album:
  - Atributos: titulo, canciones (lista con objetos 'Cancion')
  - Métodos: agregar\_cancion(), numero\_canciones(), eliminar\_cancion(), duracion\_total(), \_\_str\_\_()

6- Realizar las siguientes clases en Python:

- Libro:
  - Atributos: titulo, autor, paginas
  - Métodos: \_\_str\_\_()
- Biblioteca:
  - Atributos: libros (lista con objetos 'Libro')
  - Métodos: agregar\_libro(), ordenar\_lista\_por\_titulo(), numero\_libros(), eliminar\_libro(), \_\_str\_\_().

7- Se requiere desarrollar un programa que modele un sistema de peaje. Las estaciones de peaje tienen un nombre, departamento en que están ubicadas y una lista de vehículos que pasaron.

Los vehículos que llegan a un peaje tienen una placa (tipo String). El peaje cobra diferentes valores de peaje según el tipo de vehículo. Si es un auto, el valor del peaje es de \$300. Si es una moto, \$150. Si es un camión, el valor del peaje depende del número de ejes, se cobra \$500 por cada eje.

Se requiere que la estación de peaje calcule el valor del peaje de cada vehículo que llegue y el total de peajes recolectados. Así, al finalizar, el sistema debe imprimir en pantalla un listado con los vehículos que llegaron al peaje y el total acumulado.

Atributos y métodos de las clases:

- Vehiculo:
  - Atributos: placa
- Moto: Hereda de vehiculo
  - Atributos: valor\_peaje
  - Métodos: \_\_init\_\_(placa)
- Auto: Hereda de vehiculo:



- Atributos: valor\_peaje
- Métodos: \_\_init\_\_(placa)
- Camion: Hereda de vehiculo
  - Atributos: valor\_peaje, cantidad\_ejes
  - Métodos: \_\_init\_\_(placa, cantidad\_ejes)
- Peaje:
  - Atributos: nombre, departamento, vehiculos
  - Métodos: agregar\_vehiculo(), calcular\_total\_peaje(), cantidad\_vehiculos()