



Java Polimorfismo



Octavio Robleto



octavio.robleto@gmail.com

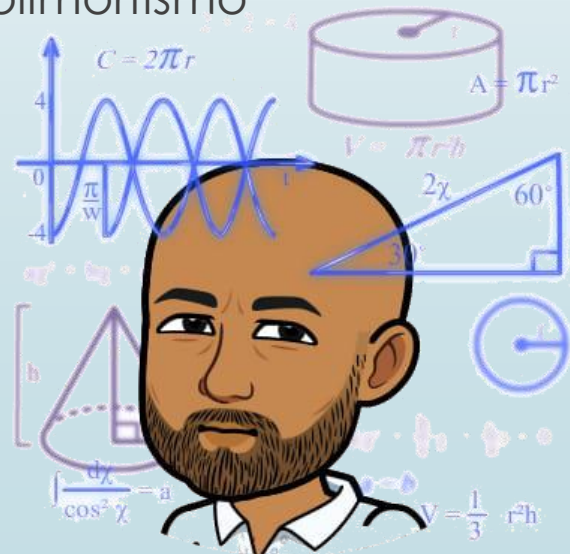


<https://octaviorobleto.com>



Polimorfismo

- El polimorfismo es una de las características fundamentales de la Programación Orientada a Objetos y está estrechamente relacionado con la herencia.
- Una jerarquía de clases, los métodos y clases abstractas, la sobreescritura de métodos y las conversiones entre clases de la jerarquía sientan las bases para el polimorfismo. Es necesario entender bien estos conceptos para comprender el polimorfismo

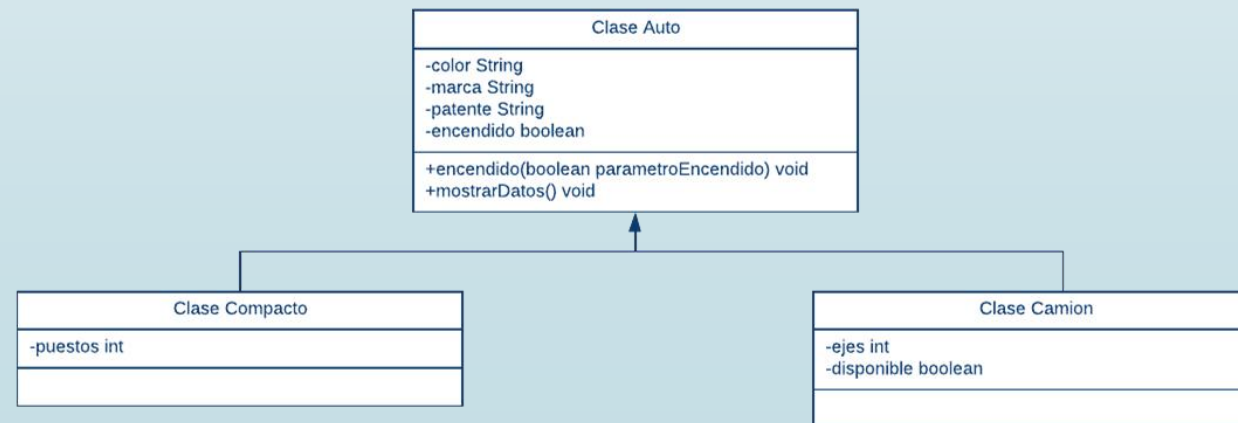


Como logramos esto?

- Los métodos deben estar declarados (métodos abstractos) y a veces también pueden estar implementados (métodos no abstractos) en la clase base.
- Los métodos debes estar redefinidos en las clases derivadas.
- Los objetos deben ser manipulados utilizando referencias a la clase base

Diagrama de clases UML

OCTAVIO ROBLETO | August 3, 2019



Por Asignación

- En java, una misma variable (Clases, interfaces...) puede hacer referencia a más de un tipo de Clase. El conjunto de las que pueden ser referenciadas **está restringido por la herencia o la implementación**

```
Auto compacto1 = new Compacto("Azul", "Ford", "ANZ-963", true, 3);
```

Declaración del
Objeto de tipo Auto

Se instancia como
de tipo compacto



Polimorfismo de sobrecarga

- Varias implementaciones del mismo método con distintos parámetros (cambio de interfaz). Muy habitual en constructores

```
Auto[] estacionamiento = new Auto[2];
estacionamiento[0] = new Compacto("Rojo", "Daewoo", "XZD-856", true, 3);
estacionamiento[1] = new Camion("Verde", "Mercedes Benz", "CAM-7896", false, 16, true);

for (int i = 0; i < estacionamiento.length; i++) {
    estacionamiento[i].mostrarDatos();
}
```

**Tipo Auto
comportamiento
diferente al mostrar
Datos**