



Departamento de Ciencia de la Computación
Escuela de Ingeniería UC
Pontificia Univesidad Católica de Chile

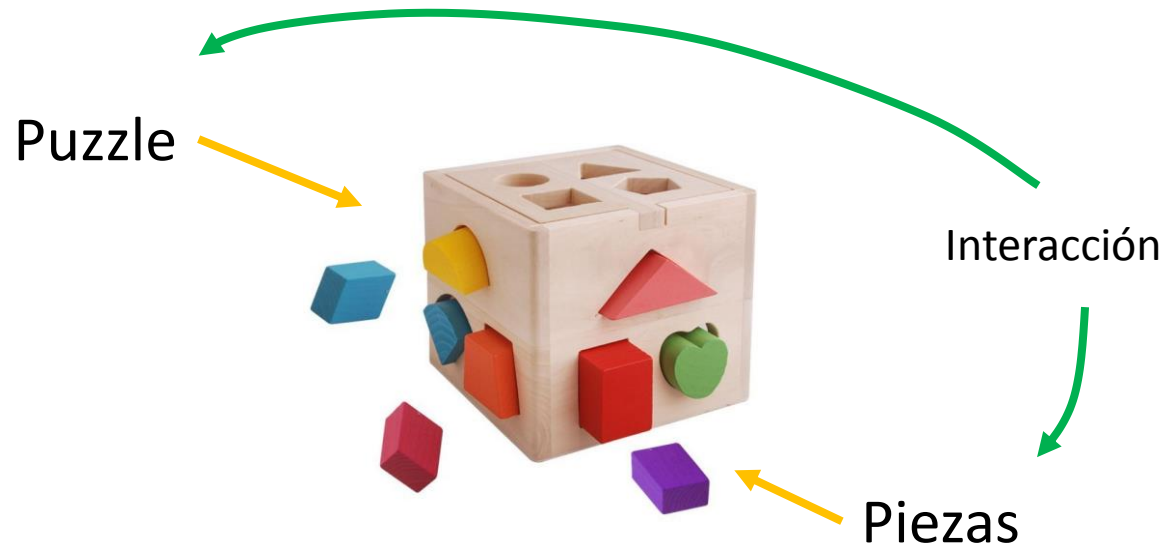
Programación Orientada a Objetos

Programación Avanzada

IIC-2233

¿Qué son los objetos?

- En el mundo real los objetos son tangibles que se pueden manipular y sentir, que representan algo que tiene significado para nosotros.



¿Qué son los objetos?

- En el área de desarrollo de software, un objeto es una colección de **datos** que además tiene asociado **comportamientos**.
 - Datos: **describen** a los objetos
 - Comportamientos: **representa acciones** que ocurren en los objetos

Ejemplo: Datos y Comportamiento

Objeto: Auto	
Datos	Comportamiento
Marca	Calcular próxima mantención
Modelo	Calcular distancia a alguna dirección
Color	Pintarlo de otro color
Año	Realizar nueva mantención
Motor	
Kilometraje	
Ubicación Actual	
Mantenciones	

¿Qué es OOP?

- Programación Orientada a Objetos quiere decir que los programas estarán orientados a modelar las funcionalidades a través de la interacción entre objetos por medio de sus datos y comportamiento.

Clases vs Objetos

- En OOP los objetos son descritos por **clases**

Cada objeto es una **instancia** de la clase Auto

Objeto 1



Objeto 2



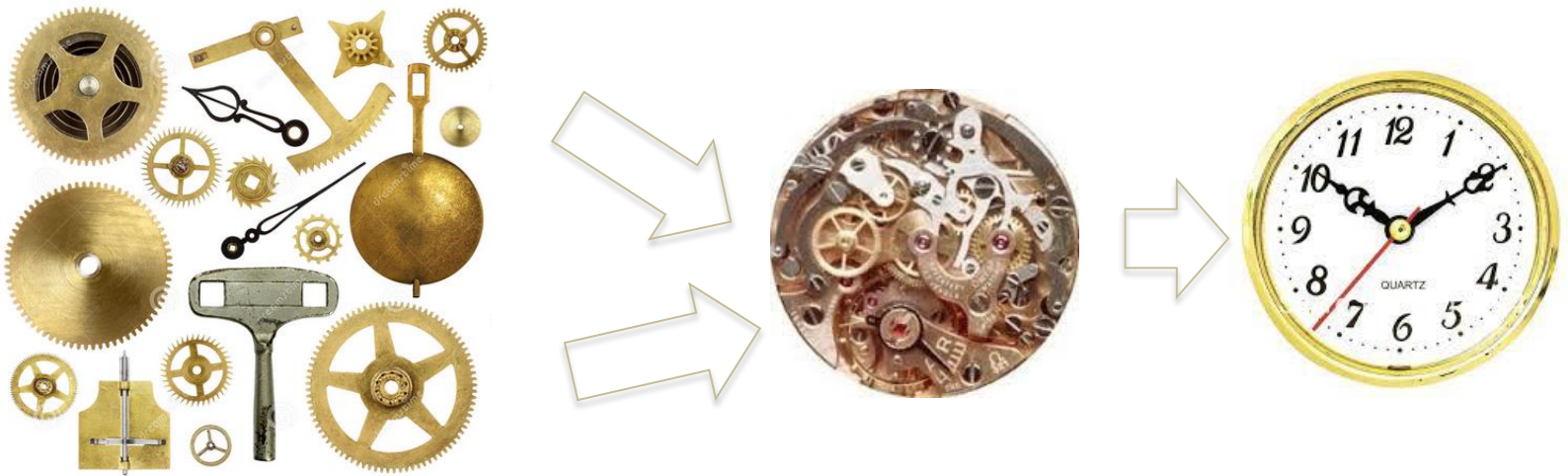
Objeto 3



Clase **Auto**

Encapsulamiento

- Existen atributos de los objetos que no necesitan ser visualizados ni accedidos por los otros objetos con que se interactúa



Interfaz



Interface
Turn on
Turn off
Volume up
Volume down
Switch to next channel
Switch to previous channel
Current channel
Volume level



Interfaz

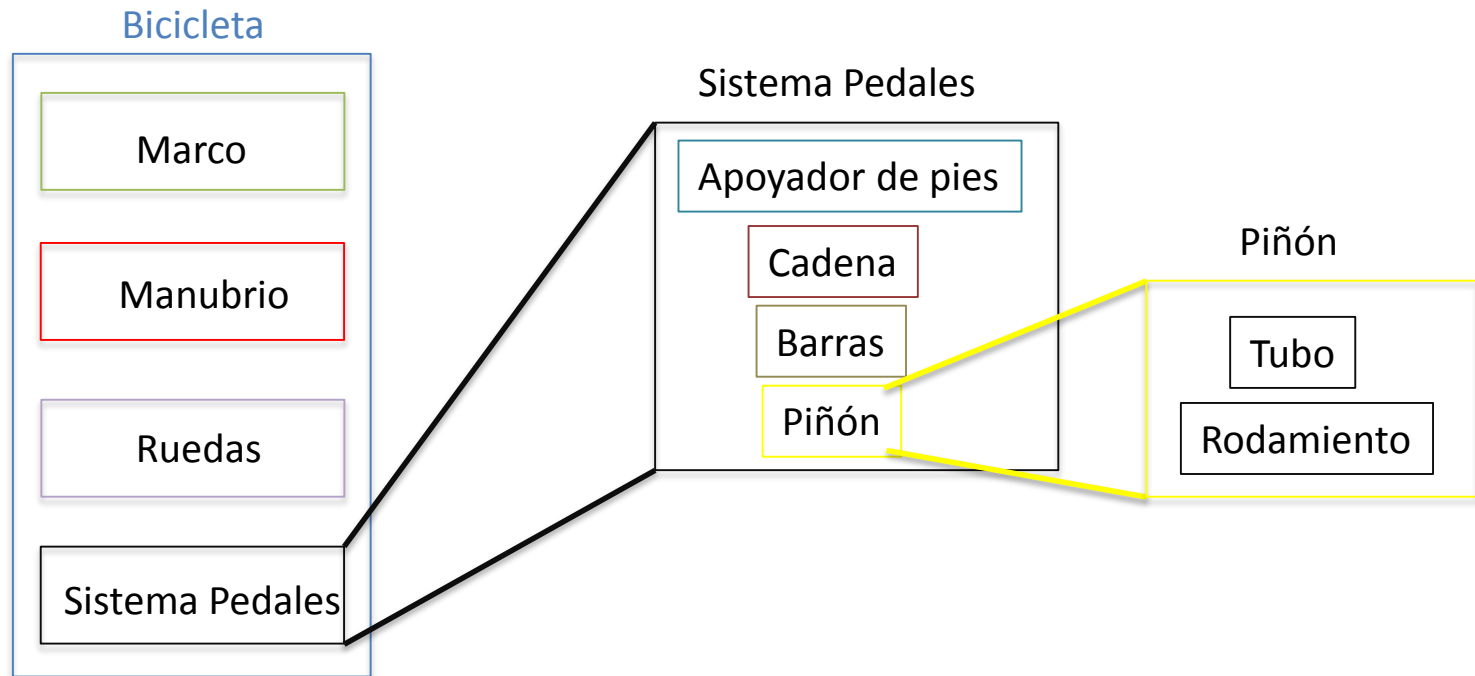
- Proceso de extracción de la interfaz de un objeto a partir de sus detalles internos
- El nivel de detalle de la interfaz se denomina **abstracción**



Vendedor
Nombre
Nº autos vendidos
Comisión asignada

Composición

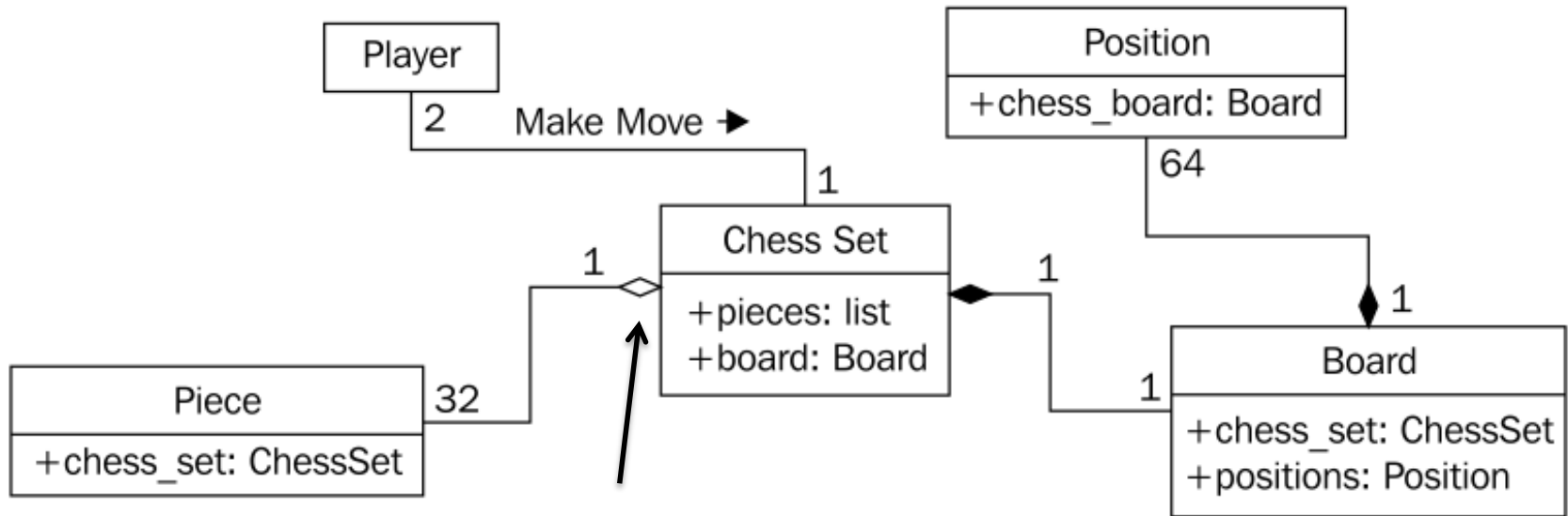
- Colección de varios objetos para formar uno nuevo



Composición vs Agregación



■ Juego de Ajedrez



Agregación ocurre cuando el objeto contenido en otro puede tener sentido como objeto único (hollow diamond)

Herencia:

Vehículo

Ruedas
Tamaño
Color
Encender
Mover



Marca
Modelo
Airbag
Abrir maleta



Nº estudiantes
Filas asientos
Lista Colegios



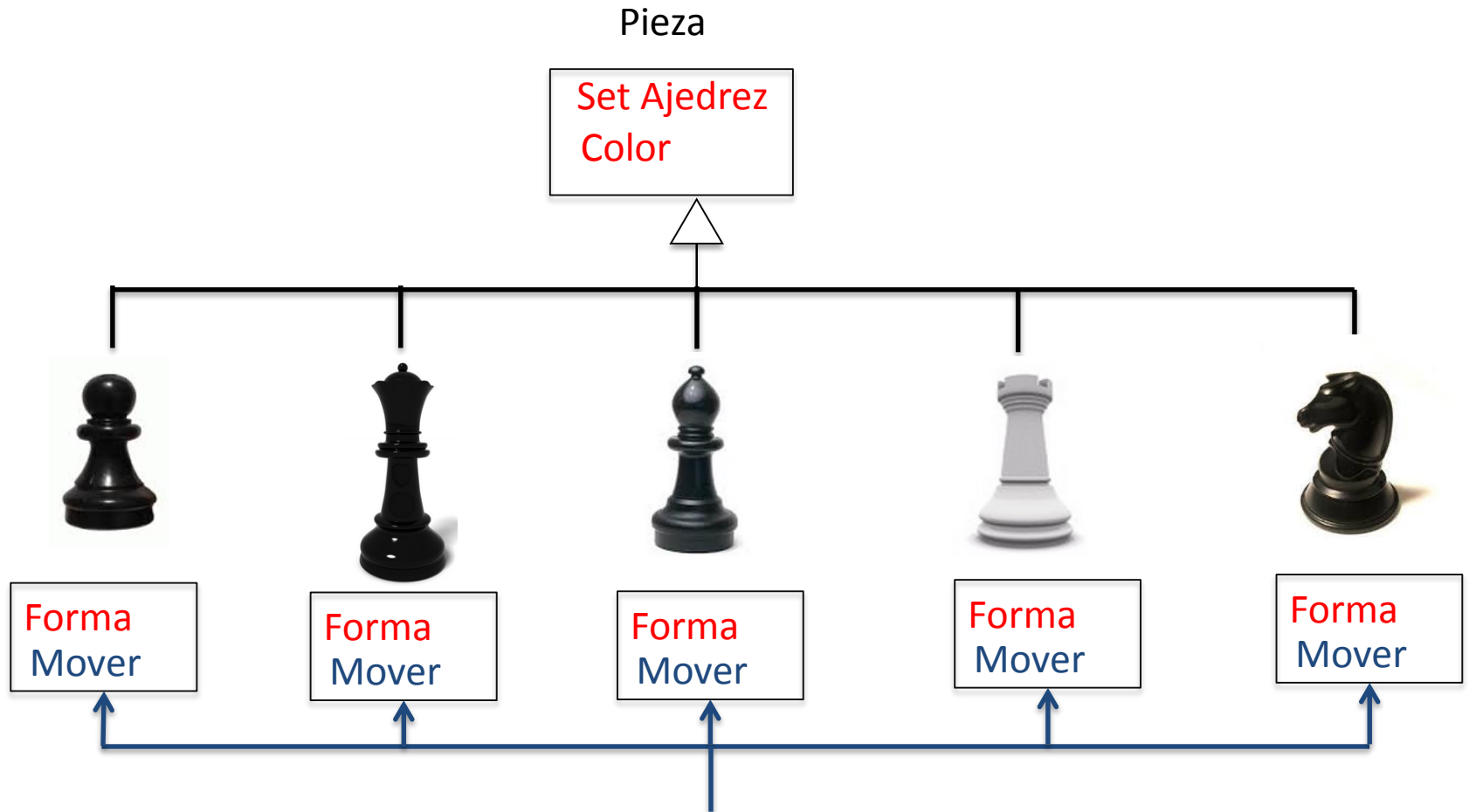
cilindros
Tipo Llantas
Encender



armamento
Radar
Mover

Overriding

Polimorfismo



Mismo método pero cada uno actúa distinto

Duck Typing



Forma
Mover



Forma
Mover



Forma
Mover



Forma
Mover

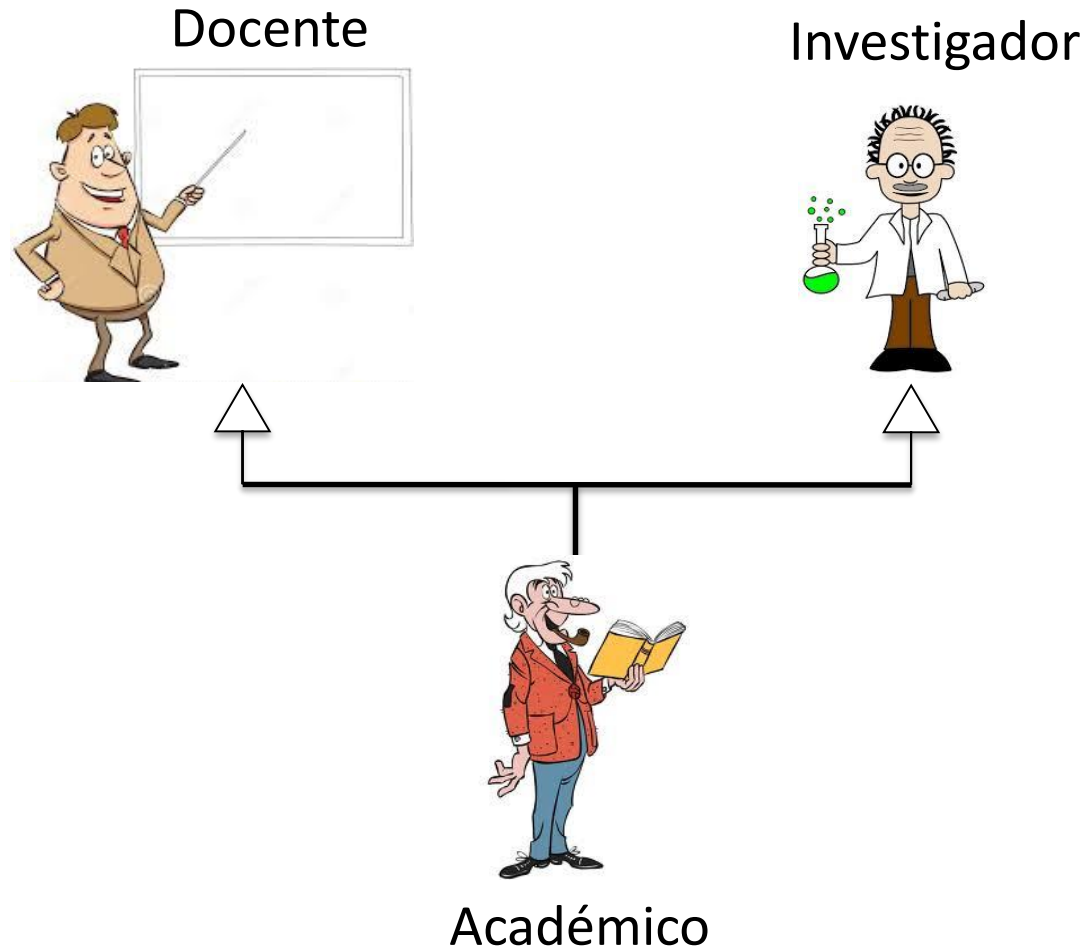


Raza

...

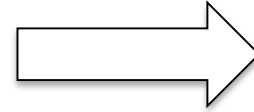
Mover

Herencia Múltiple



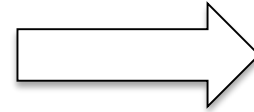
Objetos en Python

ipython notebook

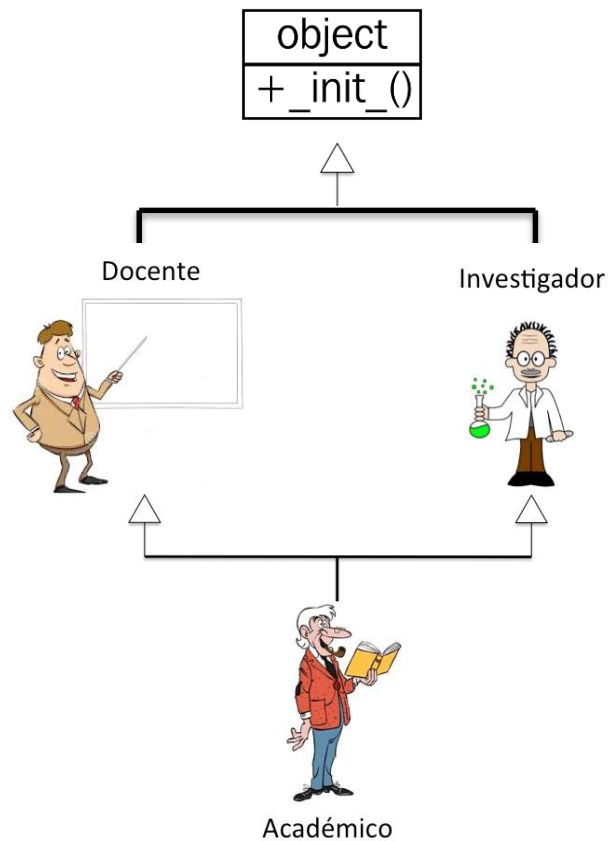


Herencia y Multi-Herencia en Python

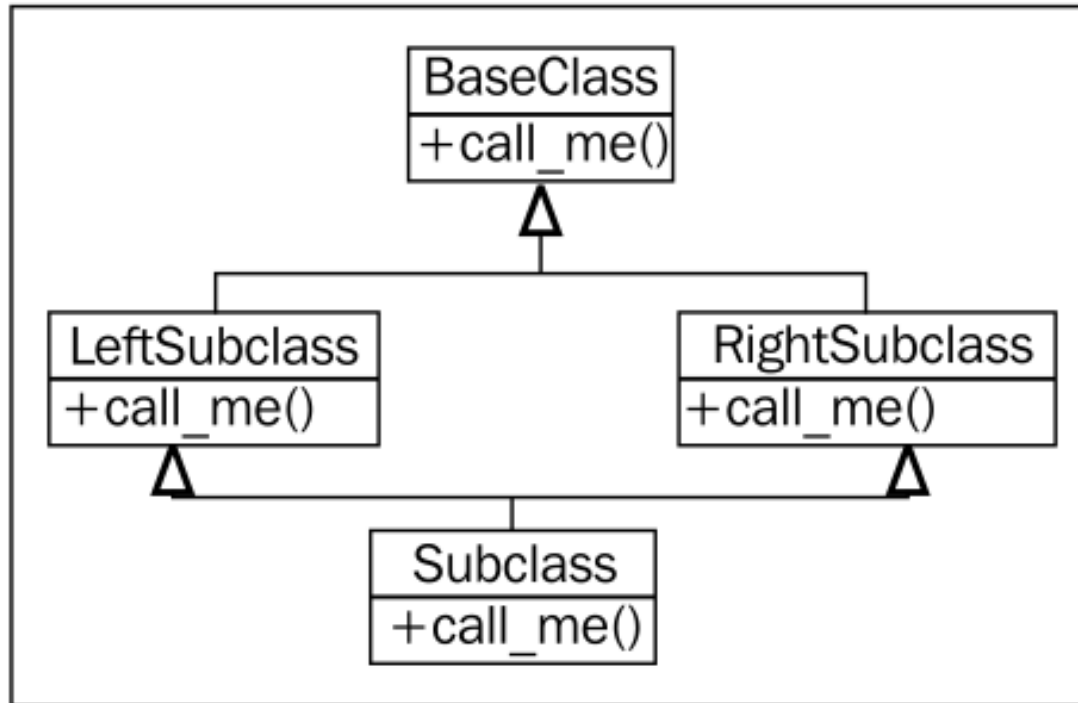
ipython notebook



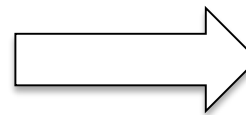
Multi-Herencia y el problema del diamante



Multi-Herencia y el problema del diamante

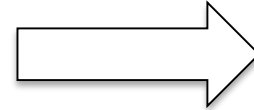


ipython notebook



Polimorfismo en Python

ipython notebook



Properties

- A veces necesitamos agregar nuevo código antes de “setear” o retornar el valor de un atributo de una clase
- El problema es que añadir métodos que seteen o retornen el atributo nos obligaría a modificar todas las llamadas que ya están apuntando a ese atributo

Properties

- En Python, con el keyword “property”, podemos agregar “setters” and “getters” y hacer que parezcan atributos

ipython notebook

