

Conceptos Fundamentales de la Programación Orientada a Objetos

Programación III
Lic. Gladys Chuquimia
gchuquimia@gmail.com

Programación Orientada a Objetos

“La POO es un método de implementación en el que los programas se organizan como *colecciones cooperativas de objetos*, cada uno de los cuales *representan una instancia* de alguna *clase*, y cuyas clases son todas miembro de una jerarquía de clases unidas mediante *relaciones de herencia*” – Grady Booch

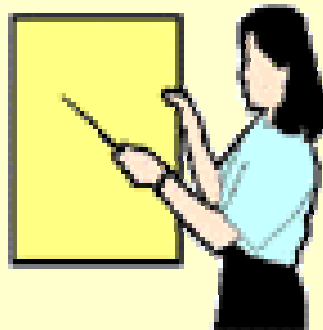
Identificación de los Objetos según Shlaer, Mellor y Coad/Yourdon

- # **Cosas Tangibles** – avión, libro
- # **Roles o papeles** – gerente, cliente, empleado
- # **Organizaciones** – Equipo, empresa, división
- # **Incidentes** – (Sucesos, Eventos) vuelo, accidente
- # **Interacciones** – (Transacción, Contratos)
Compras, Vendedor, Contrato Laboral
- # **Especificaciones** – Inventarios, Proformas
- # **Lugares** – Sala de Espera, Parqueo, Oficina

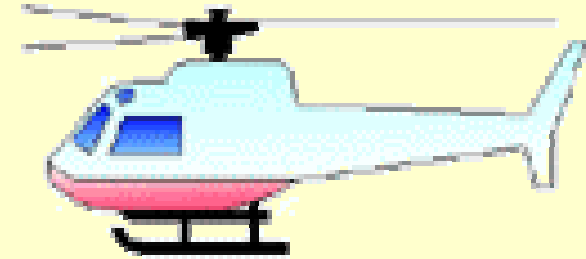
¿Cómo se encuentran los objetos?



Antena Parabólica



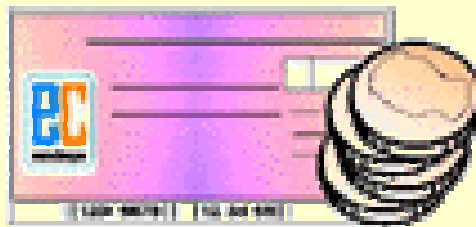
Objeto Profesor



Helicóptero

$a + bi$

Número Complejo



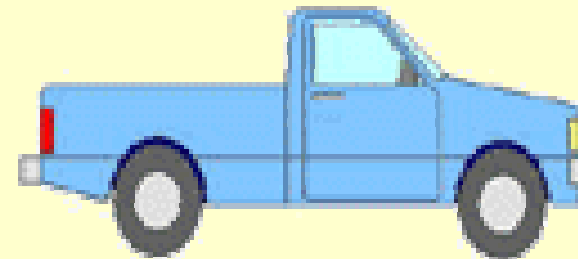
Cuenta Bancaria



Automóvil

“Los objetos están ahí para usarlos”

Los objetos similares se agrupan en **clases**



Objetos de la clase **Coche**

Clases

- ✦ Son una descripción de un **conjunto de objetos**.
Consta de Métodos y Atributos.

Es un **TDA definido** que determina las **estructuras de datos** y **operaciones** asociadas con ese tipo.

En OO, una clase no solo tiene el objetivo de categorizar sino también el de servir como **plantilla para crear objetos**.

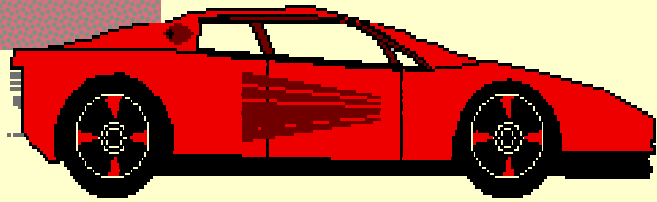
Miembros Dato - atributos

Características especiales que distinguen a un objeto.

Funciones Miembro - Operaciones

- # Describen el comportamiento asociado a un objeto.
- # Representan las acciones que pueden realizarse con el objeto ó sobre un objeto.
- # Cada método tiene un nombre y un cuerpo que realiza la acción o comportamiento asociado con el nombre del método.

Paso inicial en la definición del objeto coche



Coche

Representación Simplificada

Clase:

Coche
color: <i>Cadena</i> modelo: <i>Int</i> marca: <i>Cadena</i> placa: <i>Cadena</i>
Ir() Parar() Girar_a_la_D() Girar_a_la_I()

D
A
T
O
S

Tiene las **características**:

- Color
- Modelo
- Marca
- Placa

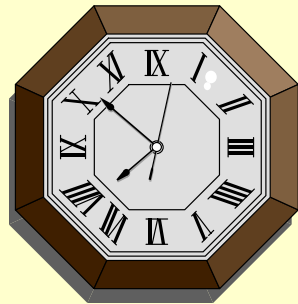
Funciones que puede realizar:

- Ir
- Parar
- Girar a la derecha
- Girar a la izquierda

M
E
T

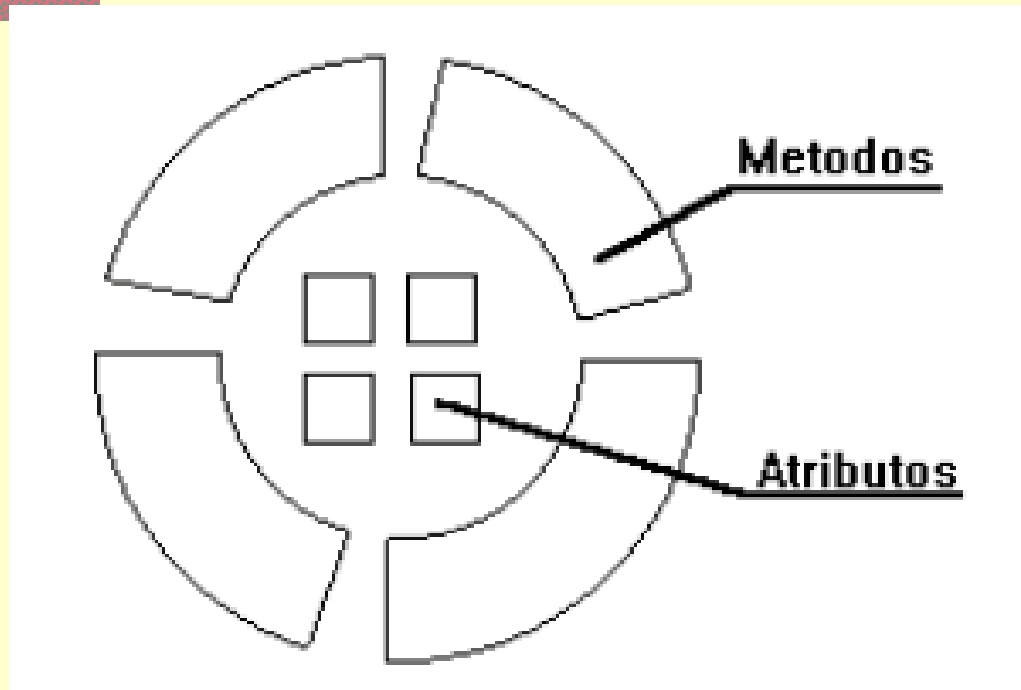
Hablemos de Objetos

Que es un objeto ?

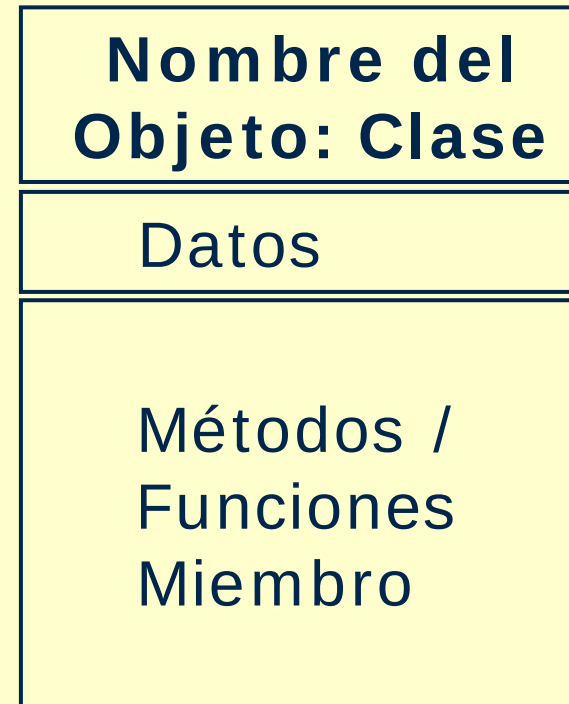


- Un objeto es una **instancia de una clase** (una categoría). Un objeto tiene una estructura. Tiene **atributos** (propiedades - **datos**) y **funciones** que operan sobre estos datos (actividades – comportamiento – **métodos**).

Representación gráfica de un objeto



TAYLOR



UML

Definición del **objeto** coche



Definición Completa

Toyo:Coche

Rojo
2007
Gold
ABC123

Ir()
Parar()
Girar_a_la_D()
Girar_a_la_I()

Definición Especifica

Toyo2: Coche

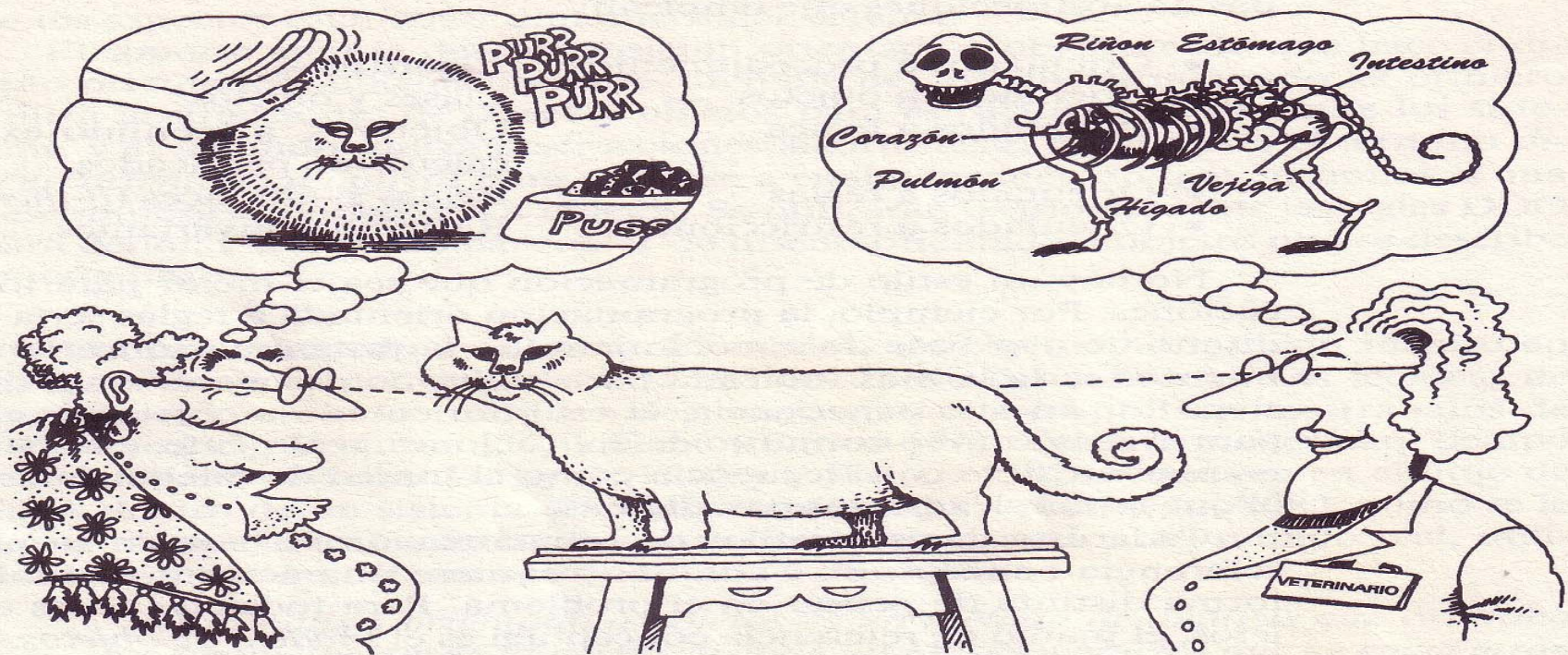
Blanco
2001
Peta
QWS123

Toyo2: COCHE

Representación Simplificada
de un objeto previamente
definido.

Abstracción (2)

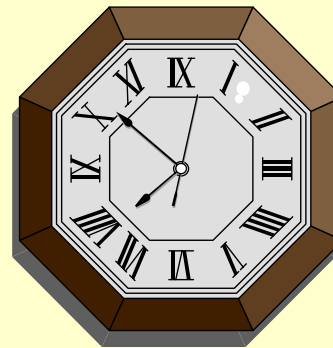
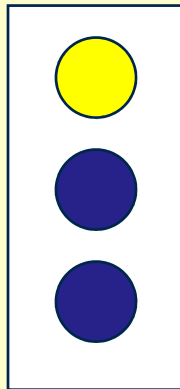
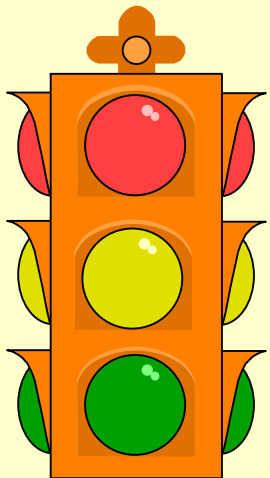
“Supresión intencionada, u ocultamiento, de algunos detalles de un proceso o artefacto, con el objeto de destacar de manera más clara otros aspectos, detalles o estructuras”



La abstracción se centra en las características esenciales de algún objeto, en relación a la perspectiva del observador.

Abstracción

- # En palabras sencillas “abstracción” significa filtrar las propiedades y operaciones de un objeto hasta que queden solamente aquellas que necesitamos. (Generalización)



HH:mm:ss

Paso inicial en la definición del **objeto** coche para su venta



Coche

Representación Simplificada

Clase:

Coche	
color:	<i>Cadena</i>
modelo:	<i>Int</i>
marca:	<i>Cadena</i>
precio:	<i>Float</i>
RegistroCoche()	
Mostrar()	
ModificaPrecio()	
Modificar()	

D
A
T
O
S

Tiene las **características**:

- Color
- Modelo
- Marca
- Precio

Funciones que puede realizar:

- Registro del coche
- Mostrar datos del coche
- Modificar el precio
- Modificar otros datos

M
E
T

Definición del **objeto** coche



Definición Completa

Car1: Coche

Rojo
2007
Gold
8000

RegistroCoche()
Mostrar()
ModificaPrecio()
Modificar()

Definición Especifica

Car2: Coche

Blanco
2001
Peta
4500

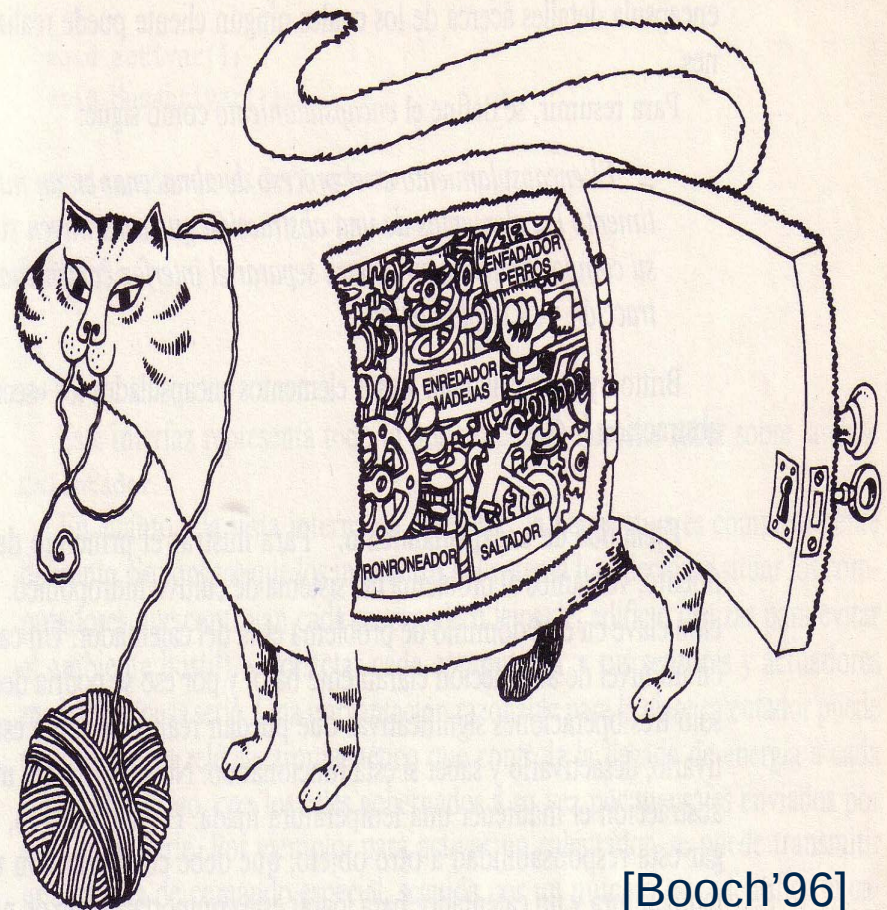
Car1: COCHE

Representación Simplificada
de un objeto previamente
definido.

Encapsulación (2)

“Proceso de almacenar en un mismo compartimento los elementos de una abstracción que constituyen su estructura y su comportamiento”
[Booch'96]

16/08/2010



[Booch'96]

El encapsulamiento oculta los detalles de la implementación de un objeto.

Encapsulación

- # Es el “empaquetar” en una clase los datos (propiedades) y el código (métodos) que opera en esos datos.
- # El acceso a esos datos solo es permitido a través de los componentes públicos del objeto.(Principio de ocultamiento)



La calculadora oculta las operaciones que realiza internamente de la persona que la utiliza.

Control de acceso a miembros



Clase

- Atributo privado
 - + Atributo público
 - # Atributo protegido
-
- Método privado
 - + Método público
 - # Método Protegido

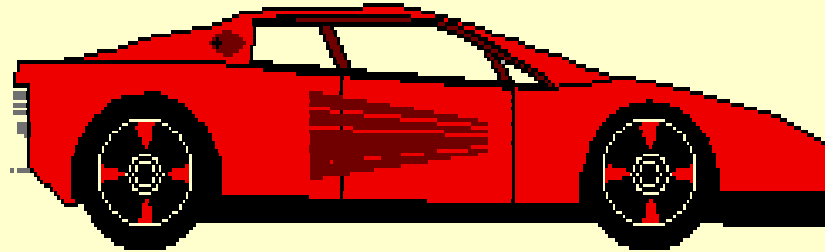
Coche

- color: *Cadena*
- + modelo: *Int*
- marca: *Cadena*
- precio: *Float*

- + RegistroCoche()
- + Mostrar()
- + ModificaPrecio()
- + Modificar()

Se accede sólo a miembros públicos

Definición del objeto coche



Definición Completa

Car1: Coche

- Rojo
- + 2007
- Gold
- 8000

- + RegistroCoche()
- + Mostrar()
- + ModificaPrecio()
- + Modificar()

Definición Especifica

Car2: Coche

- Blanco
- + 2001
- Peta
- 4500

Envío de Mensajes

- # Los objetos trabajan entre si por medio del envío de mensajes.



Mensajes

Sintaxis:

Objeto.método()

Objeto.atributo público

En el ejemplo:

Car1.RegistroCoche()

Car1.Mostrar()

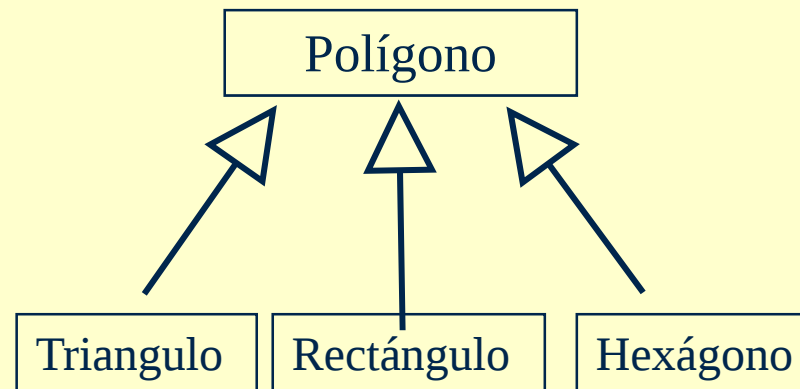
Car1.Modelo = 2008

Herencia

- # La herencia podemos definirla como la capacidad para crear nuevas clases a partir de clases existentes.
- # Las nuevas clases podrán contar con otros atributos y operaciones.
- # El principio en que basa la división de clases es la jerarquización (características comunes)
- # Ventaja principal: reutilización del código.

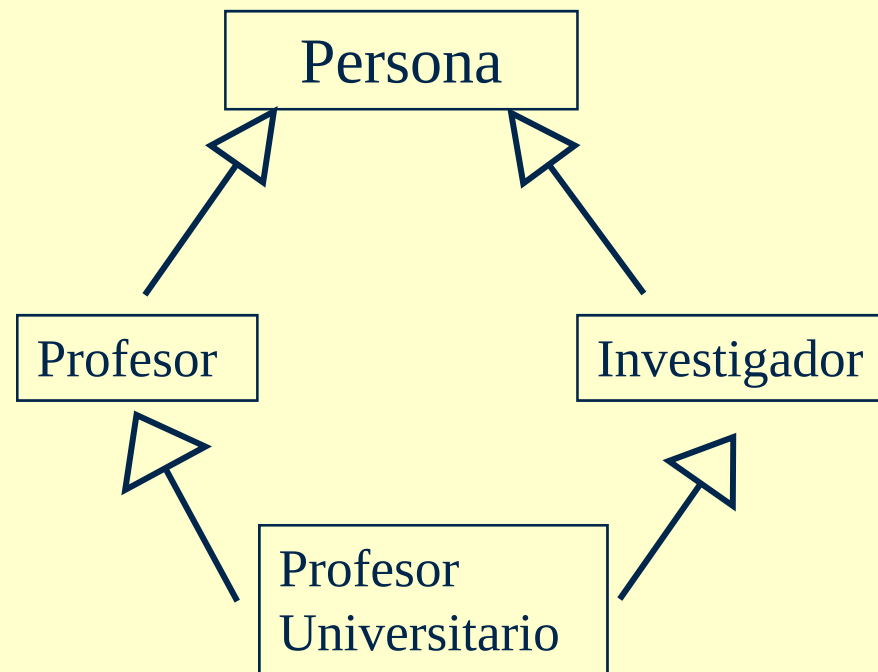
Herencia simple

- # Clase Derivada hereda datos y métodos de una sola clase base.



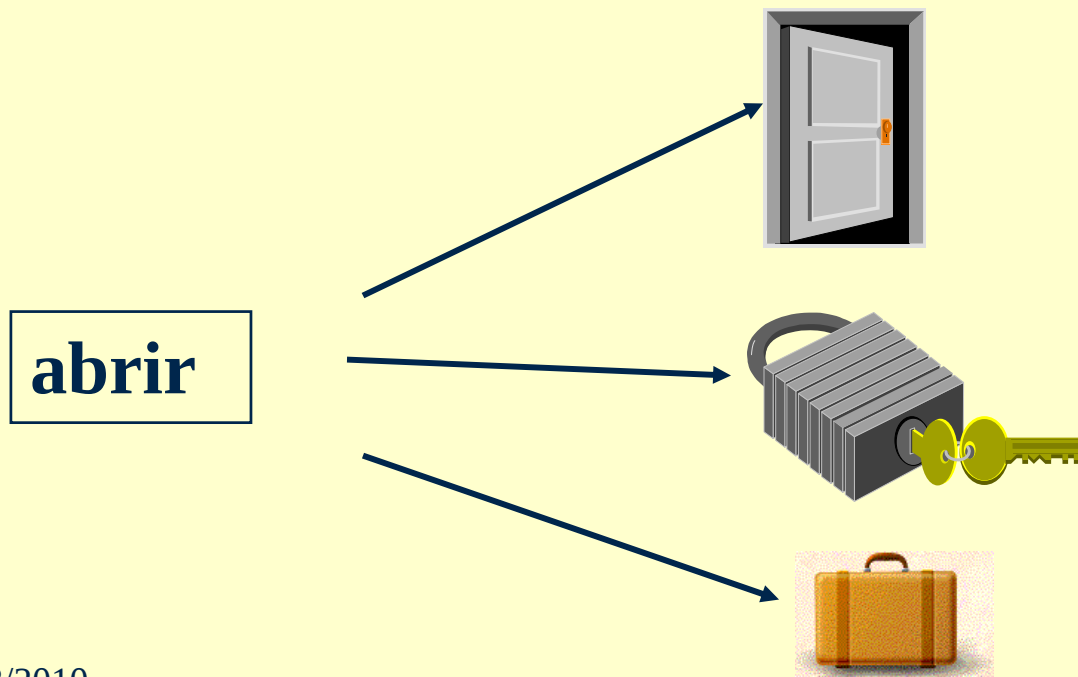
Herencia Múltiple

- # Clase Derivada hereda datos y métodos de más de una clase base.



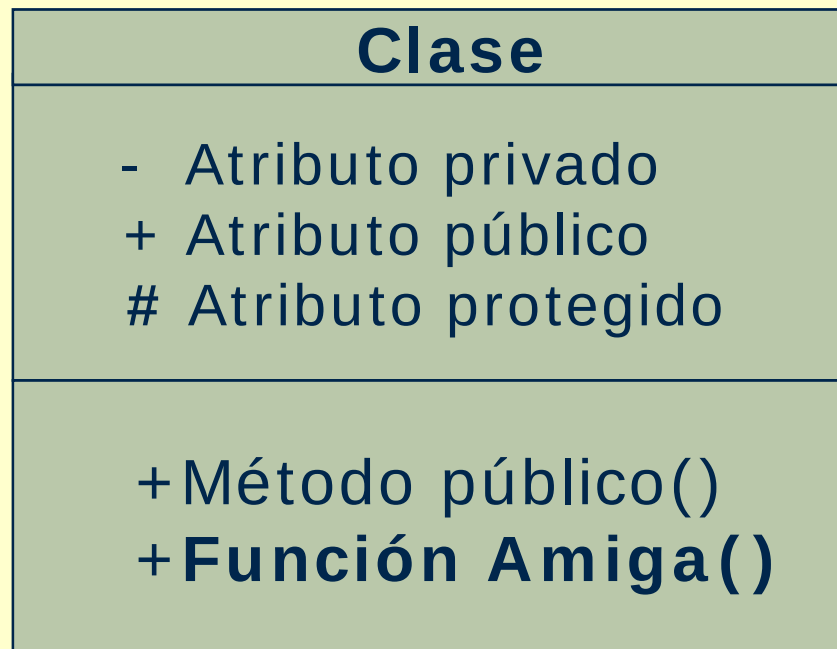
Polimorfismo

- Mediante esta propiedad los métodos de los objetos pueden tener el mismo nombre, y funcionan distinto cuando se les envía un mensaje.



Función Amiga

- Todo módulo que no sea miembro de la clase, pero que tiene acceso irrestricto a todos los miembros de la clase.



```
//Programa
//función Amiga
{
//Instrucciones
}
```

Lenguajes Orientados a Objetos

- # El lenguaje Simula (*Simulation Language*), fue el primero que se diseñó para facilitar la programación orientada a objetos.
- # Lenguaje C++, JAVA, Smalltalk
- # JAVA, desarrollado por la empresa Sun Microsystems en 1995 y que se ha extendido ampliamente en World Wide Web.



Fin de la presentación

• ***Gracias por su atención!***

