



# **Programación II**

## Programación orientada a objetos (POO)

*La programación orientada a objetos es un método de implementación en la cual los programas se organizan como colecciones colaborativas de objetos, cada uno de los cuales representan una instancia de alguna clase, donde todas las clases son miembros de una jerarquía de clases unidas por relaciones de herencia.*

- Usan objetos como bloque básico de construcción
- Cada objeto pertenece a una clase
- Las clases se relacionan por herencias

## Elementos del modelo de objetos

Elementos principales (necesarios):

- Abstracción
- Encapsulamiento
- Modularidad
- Jerarquías

Elementos secundarios (útiles pero no esenciales):

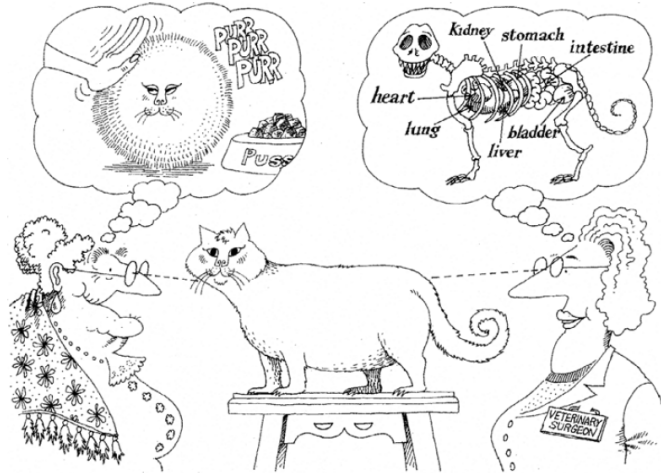
- Typing
- Concurrency
- Persistence

## Abstracción

Una abstracción representa las características esenciales de un objeto que permiten distinguirlo de todos los otros tipos de objetos. Este conjunto de características proporcionan fronteras conceptuales claramente definidas, relativas a la perspectiva del observador.

Una abstracción se centra en la vista externa de un objeto, por lo tanto permite separar el comportamiento esencial de un objeto de su implementación.

Nos vamos a centrar en la abstracción de entidades porque se deduce directamente del dominio del problema.



Abstraction focuses on the essential characteristics of some object, relative to the perspective of the viewer.

## Ejemplo de abstracción. Sensor Pasivo y Activo.

<b>Abstraction:</b> Temperature Sensor
<b>Important Characteristics:</b> temperature location
<b>Responsibilities:</b> report current temperature calibrate

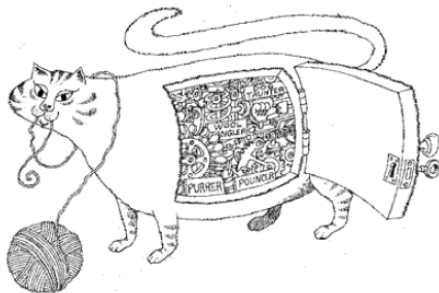
<b>Abstraction:</b> Active Temperature Sensor
<b>Important Characteristics:</b> temperature location setpoint
<b>Responsibilities:</b> report current temperature calibrate establish setpoint

## Encapsulamiento

El encapsulamiento se refiere al ocultamiento de toda información que no expresa las características esenciales de una abstracción.

**Abstracción y Encapsulamiento serían conceptos complementarios.**

**Abstracción** se focaliza en el comportamiento observable de un objeto  
**Encapsulamiento** se focaliza en ocultar la implementación o los mecanismos que materializan esos comportamientos.



Encapsulation hides the details of the implementation of an object.



## Modularidad

Modularizar consiste en dividir un programa en unidades que pueden ser compiladas independientemente, módulos, pero que mantienen conexiones entre sí.

Las conexiones entre módulos son las suposiciones que cada módulo hace sobre los restantes.

## Jerarquías

Nos permiten pensar en varias abstracciones como si fueran una sola.

Las jerarquías nos permiten ordenar las abstracciones.

## ¿Qué es (y no es) un objeto?

Un objeto representa un individuo, un ítem identificable, una unidad o una entidad, real o abstracta, con un rol bien definido en el dominio del problema

Ej. de objetos: una factura, un proceso, una figura, etc.

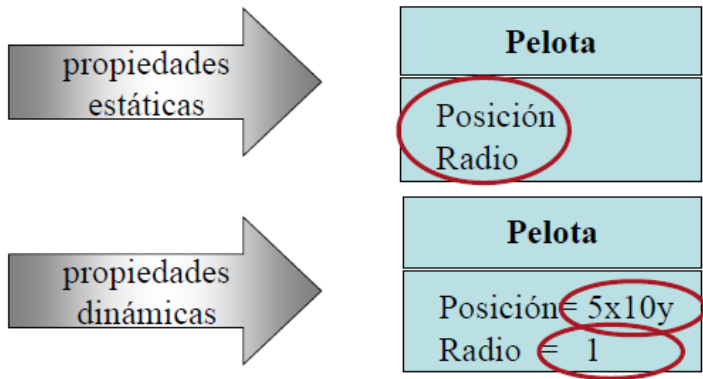
Hay que tener cuidado de no identificar como objetos cosas que son propiedades de los objetos, por ej: Colores, sentimientos, emociones.

## Definición de objeto

Es una entidad que posee estado, comportamiento e identidad. El estado y comportamiento de objetos similares están definidos en sus clases. El término *instancia* y *objeto* tienen el mismo significado.

## Estado.

El estado de un objeto abarca todas las propiedades (usualmente estáticas) del objeto más los valores actuales (usualmente dinámicos) de cada una de esas propiedades.



## Comportamiento.

El comportamiento es como un objeto actúa y reacciona, en término de su estado y el paso de mensajes.

El estado de un objeto representa la acumulación de resultados del comportamiento.

Las operaciones se denominan **métodos**

Operaciones:

- Modificadores: alteran el estado del objeto
- Selectores: acceden al estado de un objeto pero no lo alteran
- Iteradores: permiten acceder a todas las partes de un objeto en un orden determinado
- Constructores: crea o inicializa el objeto
- Destruidores: libera el estado de un objeto y lo destruye

### **Protocolo.**

El protocolo de un objeto define el flujo permitido de mensajes que un objeto puede recibir.

### **Identidad.**

Es la propiedad que distingue un objeto del resto de los objetos.

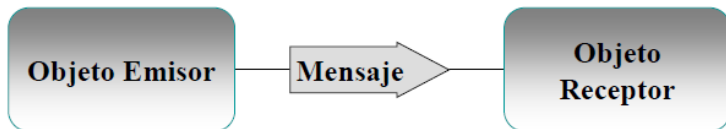
## ¿Cómo interactúan los objetos entre si?

*Los objetos contribuyen al comportamiento de un sistema colaborando unos con otros.*

Todas las operaciones en POO resultan de enviar **mensajes** entre objetos

Un mensaje es una solicitud para que un objeto ejecute una acción. Este objeto responde ejecutando métodos, es decir, usando sus propios procedimientos. El paso de mensajes es **sincrónico**.

En el envío de mensajes interactúan dos objetos:  
*el emisor del mensaje y el receptor.*





El objeto emisor debe conocer la identidad del objeto receptor. La puede obtener de alguna de las siguientes maneras:

- Estado objeto emisor
- Argumento del método desde donde se emite el mensaje
- Retorno del envío de un mensaje previo

### **El envío de mensajes soporta un importante principio: la abstracción**

El estado interno de un objeto solo puede modificarse mediante el envío de algún mensaje válido.

Esto implica que puede cambiarse la implementación de un objeto sin que varíe el comportamiento del mismo, ni el programa que lo contiene.

## **Bibliografía y enlaces útiles.**

- Booch, Grady et al: Object Oriented Analysis and Design with applications - Third Edition