

Programación orientada a objetos

(POO – OOP)

Que vemos?

Indicar que vemos en las siguientes imágenes.

Tomamos nota de lo observado.









Observamos

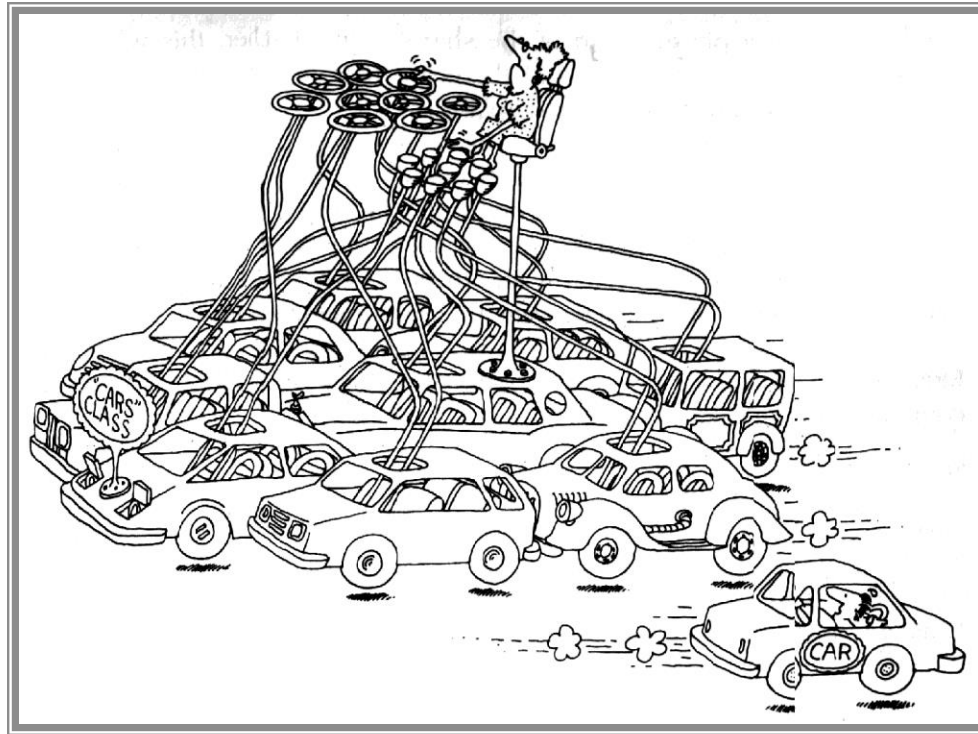
Vimos colores, modelos, tamaños, años, tipos de autos. Estamos reconociendo características.

Algunos se mueven, otros están detenidos, son conducidos por un chofer.

Estamos descubriendo comportamiento.

Todos los autos tienen cosas en común, tanto es su características como es su comportamiento y son objetos únicos.

CLASE - Definición



Una clase representa a un conjunto de objetos que:

- comparten la misma características
- se comportan de la misma forma.

Objeto - Definición

Un objeto es una instancia de una clase; una entidad tangible

Posee ciertas características y Comportamiento, **que se lo da su clase**

Características de un objeto

Una identidad (son únicos aunque tengan los mismos datos)

Estado, sus datos pueden ser modificados durante su ciclo de vida; son guardados en variables privadas y usados mediante una property

Comportamiento: determinado por los métodos

Objeto

Atributos: se escriben en minúscula para distinguirlo de las properties (getters y setters) (encapsular)

Se especifican como atributos solo las características relevantes del objeto que interesan para resolver nuestro problema. (abstracción)

Clase

Define las características y comportamiento de un objeto.

Cada clase usualmente se codifica en un archivo separado. (cs)

Los atributos se escriben en minúscula para distinguirlo de las properties (getters y setters) (encapsular)

Para describir su comportamiento utiliza métodos públicos

Programación Orientada a objetos

Es una manera de construir software

Propone resolver problemas de la realidad identificando objetos y relaciones entre ellos.

Un mundo lleno de objetos

Cualquier entidad tangible o visible del problema, entidad abstracta que tenga una estructura y un comportamiento, entidad receptora de acciones

Ej. Círculo, empleado, factura
temperatura, trayectoria, venta

Programación Orientada a objetos

De este modo se “comparten” responsabilidades, en forma similar a como se comparten las responsabilidades en una empresa:

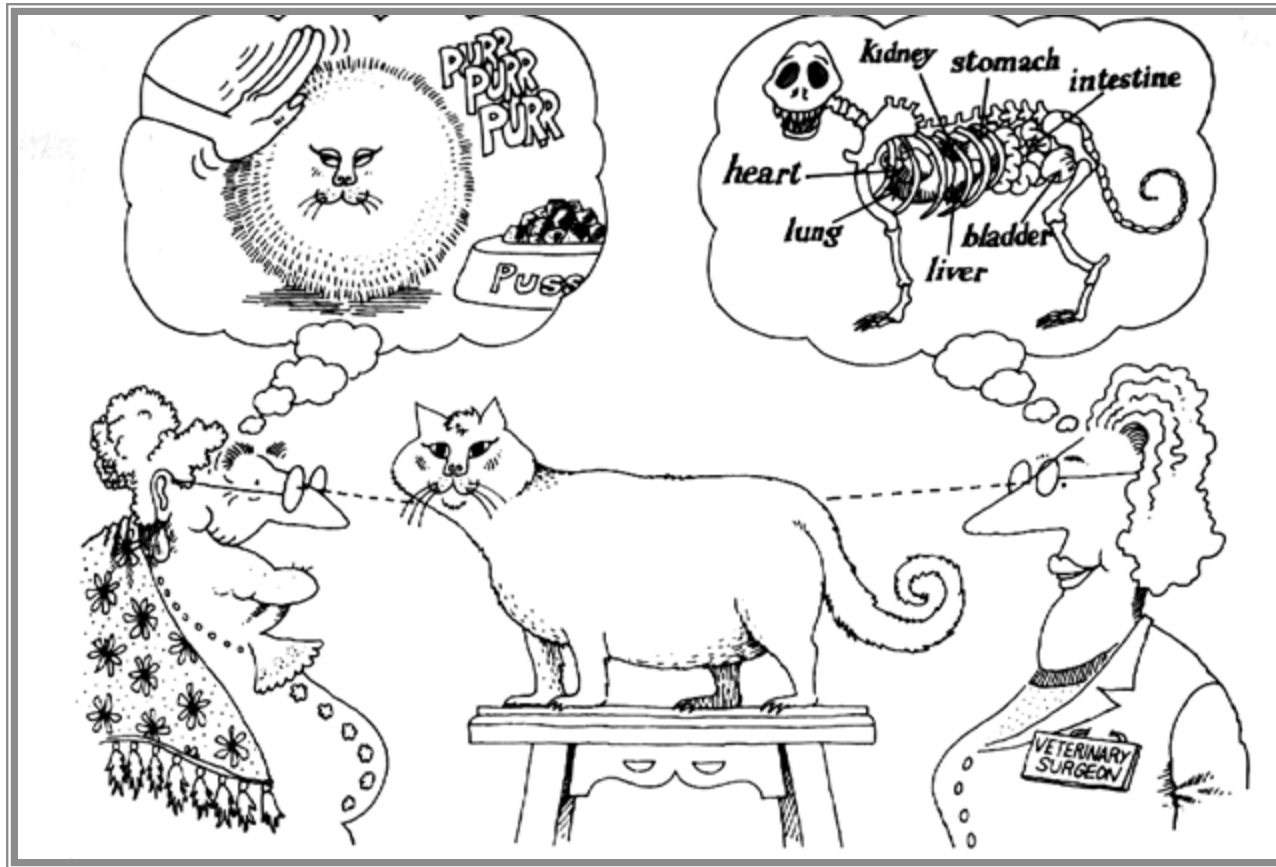
Por ej. , en un taller mecánico ¿le pedirían al contador que arregle el motor de su auto? ¿o al mecánico que presente el balance de la empresa?

En un sistema orientado a objetos, “repartir” responsabilidades significa que cada objeto contiene todo el código de lo que sabe hacer, y nada más que ese código. Dentro del código correspondiente al círculo no vamos a encontrar cómo resolver el área de un triángulo, y viceversa

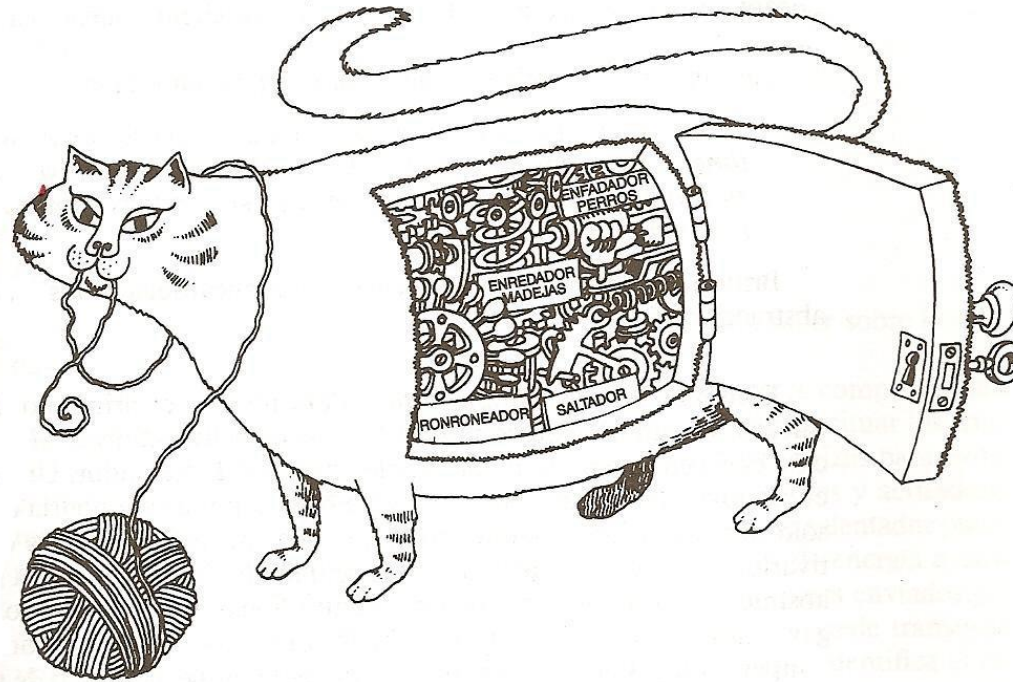
Pilares POO

- Abstracción
- Encapsulamiento
- Herencia
- Polimorfismo

Abstracción



Encapsulamiento



La encapsulación esconde la complejidad

Encapsulamiento

ENCAPSULAR en unos artefactos denominados **Objetos** sus características y su comportamiento

Cada objeto “sabe resolver” solamente lo que le corresponde, y todo lo que le corresponde.

Cuando se necesita “algo” del objeto, se le envía una solicitud para que lo resuelva. Se le envía un MENSAJE

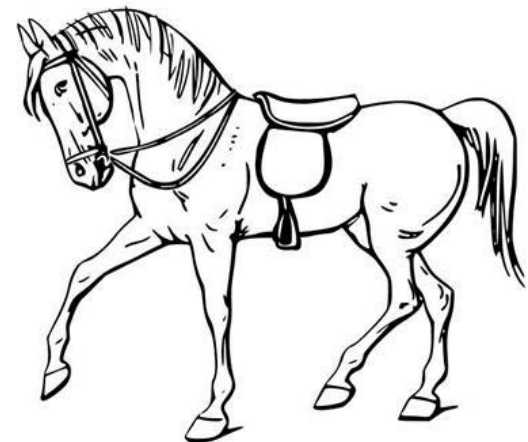
Encapsulamiento

Un grupo de atributos, métodos y otros miembros relacionados se tratan como si de una sola unidad u objeto se tratase.

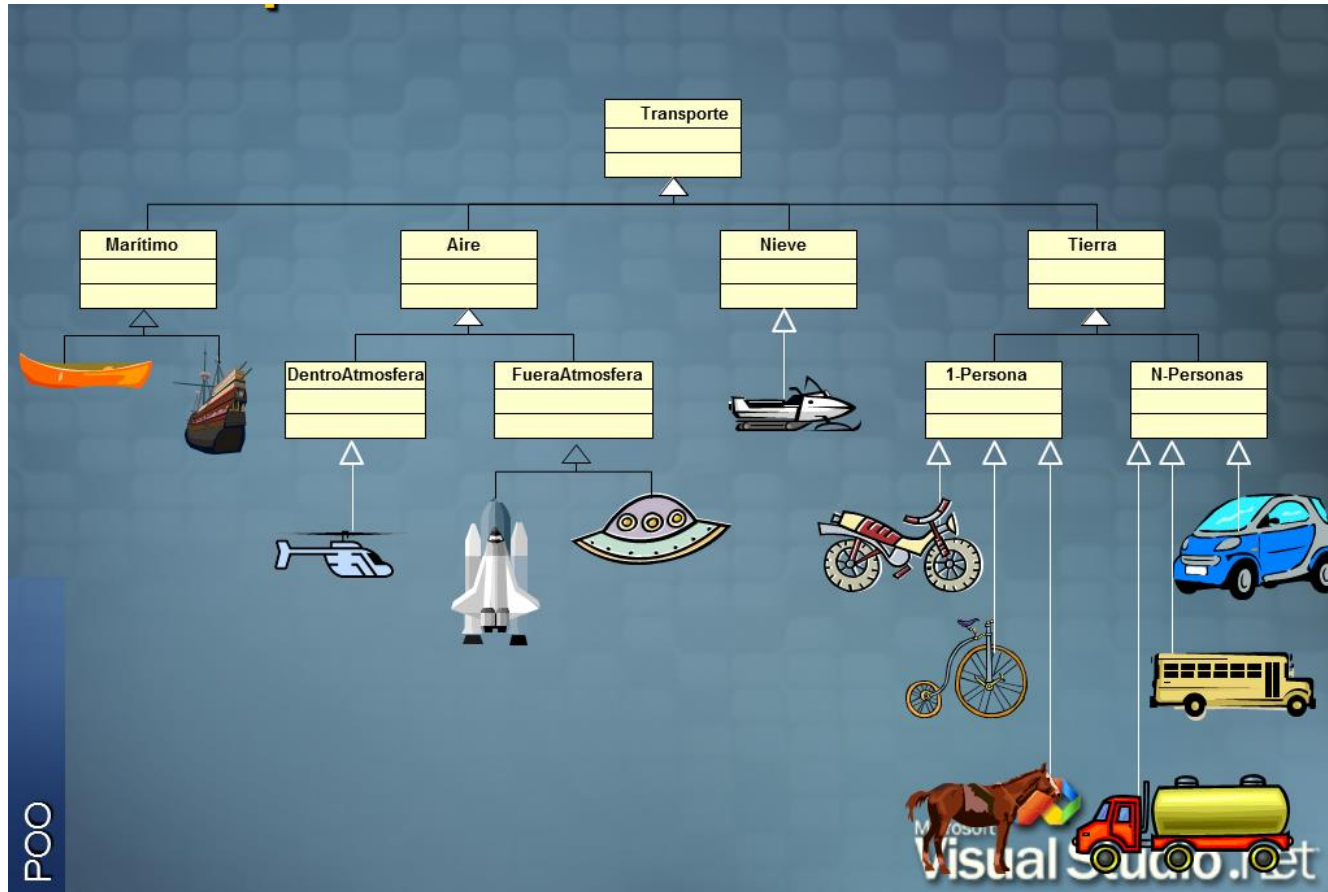
Los objetos pueden controlar cómo se cambian atributos o se ejecutan métodos.

OCULTA los detalles de implementación de los objetos dentro de los mismos.

¿Que tienen en común ?



Herencia - Jerarquia



Herencia

Una clase nueva se crea a partir de una clase existente.

Concepto de Subclase y SuperClase

Herencia

Mecanismo que permite especificar nuevas clases basadas en una clase existente (clase Base, por ejemplo Coche).

Las clases derivadas heredan todas las propiedades, métodos que expone la clase base.

Las derivadas (por ejemplo CocheDiesel) pueden personalizarse con propiedades y métodos adicionales.

Fomenta el Reuso de código.

Polimorfismo



Polimorfismo

Polimorfismo significa que múltiples clases se pueden utilizar de forma intercambiable, si bien cada clase implementa las mismas propiedades o los mismos métodos de maneras diferentes.

Por ejemplo, dada una clase base "Coche", el polimorfismo permite definir diferentes métodos "CalcularConsumo" para sus clases derivadas (ej.: CocheNafta, CocheDiesel, Sulky). Y cuando se invoca el método, no interesa qué tipo de coche es.

Está asociado a la herencia o a las interfaces (conceptos que se estudiarán más adelante en la carrera)

Objeto volador

Pájaro – Avión – Helicoptero

todos saben volar, despegar y aterrizar, pero cada uno internamente define como realiza esa acción.