


Programación Orientada a Objetos

A large yellow square is positioned in the center of the slide, serving as a background for the 'JS' text.

JS

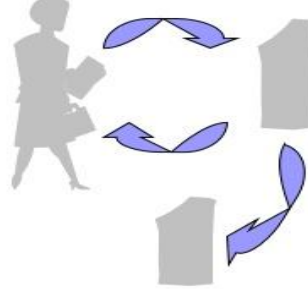


Introducción

La programación orientada a objetos es un paradigma de programación basado en el concepto de “objetos”, donde éstos interactúan entre sí para resolver un problema computacional.



PROCEDIMENTAL



**Retirar,
depositar,
transferir**

OBJETOS



**Cliente,
dinero, cuenta**

Objeto



Atributos:

- color
- raza
- edad
- energía

Métodos:

- jugar
- dormir
- comer

Un objeto es una entidad que tiene propiedades particulares, llamadas atributos, y formas de operar sobre sus propios atributos y los de otros objetos, llamadas métodos.



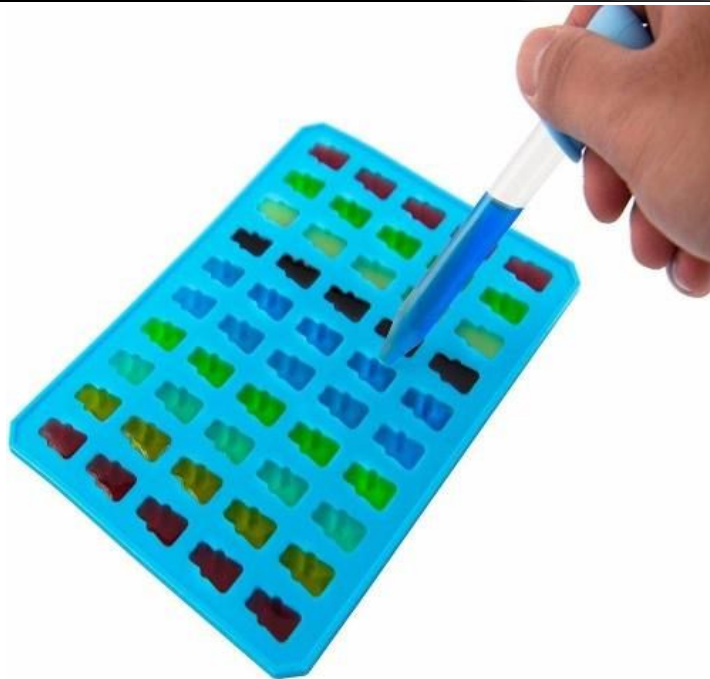
Clase

Una clase es un modelo definido por el usuario a partir del cual se crean objetos. En ellas se define cuáles serán los métodos y atributos que definirán a los objetos de su tipo.

Cuando creamos un objeto, se dice que se “instancia” una clase. En ese momento, asignamos valores a los atributos de una clase para crear objetos diferentes.



Clase



Los 4 pilares de la POO

Pilares de la POO

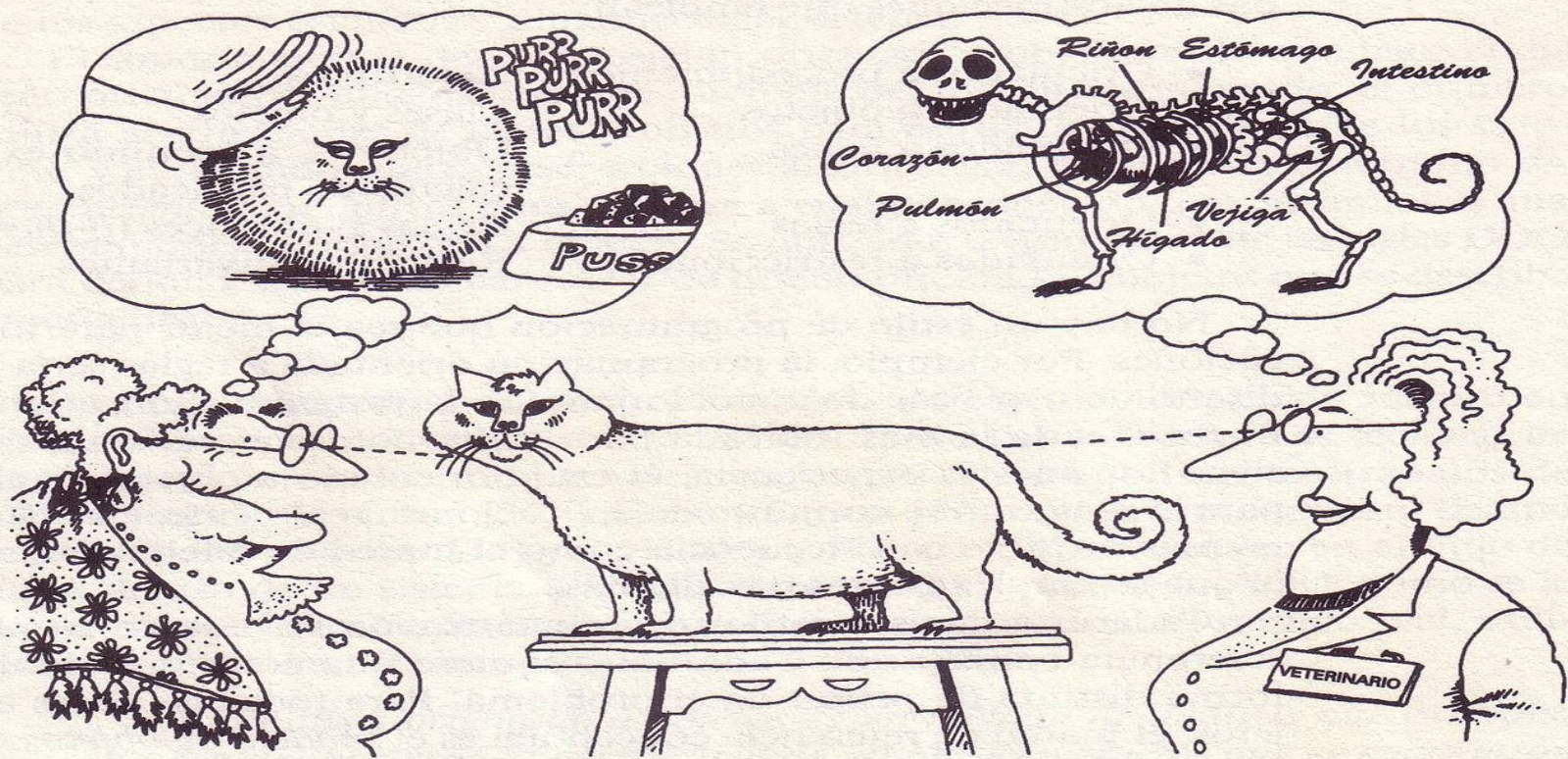


Abstracción

Consiste en aislar un elemento de su contexto como un individuo, y preguntarnos qué es lo que lo define.

En programación, el término se refiere al énfasis en el “¿qué tiene” y “¿qué hace?”, más que en el “¿cómo lo hace?”.

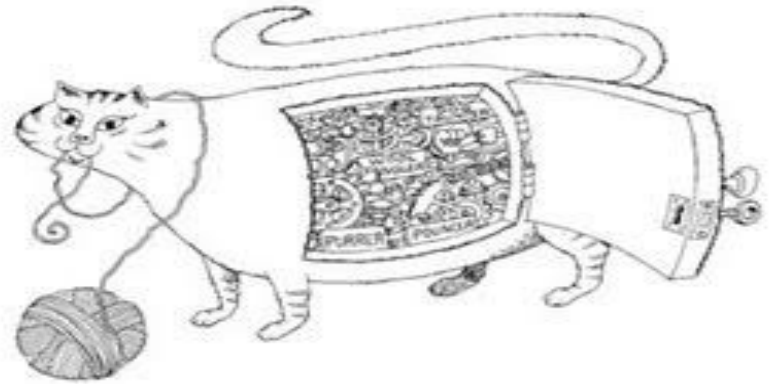




La abstracción se centra en las características esenciales de algún objeto, en relación a la perspectiva del observador.

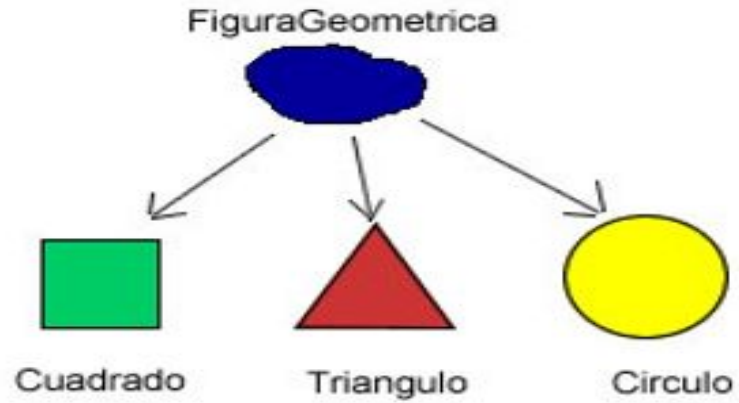
Encapsulamiento

El encapsulamiento o ocultamiento de la información son dos puntos básicos en la abstracción de datos. Todo lenguaje permite definir objetos ocultando alguna parte de ellos (datos y/o métodos).



Polimorfismo

Se denomina polimorfismo a la capacidad que tienen los objetos de una clase de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación.



Herencia

Es la facilidad mediante la cual una clase A (padre) hereda cada uno de sus atributos y métodos a una clase B (hija) de modo que pueda utilizarlos como si fueran suyos

Herencia de Clases

