



UNIVERSIDAD
Popular del Cesar

Ingeniería de Sistemas

Patrones de Diseño de Software



Profesor: Deivis Martínez Acosta



Programación Orientada a Objetos

Agenda

Programación Orientada a Objetos (OOP)

Objetos y Clases

Características de la OOP

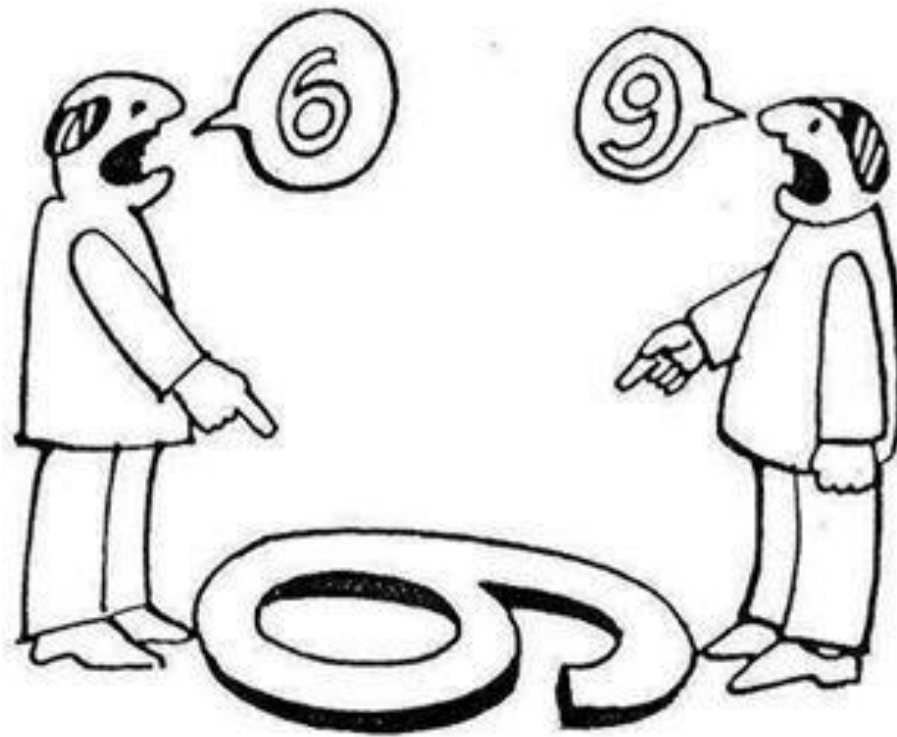
Conceptos más importantes



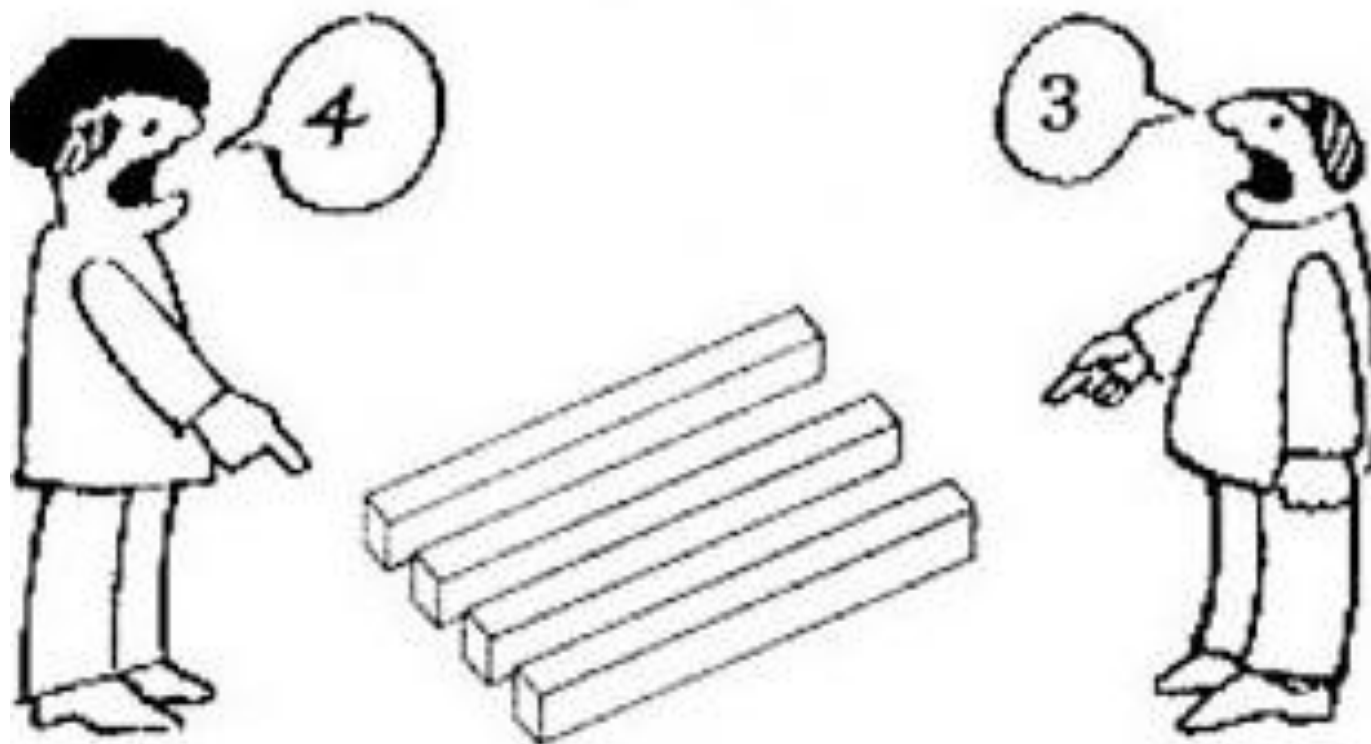
Paradigma de Programación



Paradigma de Programación



Paradigma de Programación



Paradigma de Programación

Paradigma de la Programación Orientada a Objetos (POO): Es quizás el paradigma de programación más utilizado y el que mas éxito tiene a día de hoy. Gracias a que es un paradigma que se acerca al mundo real y nos permite aumentar la comprensibilidad de los problemas. Este paradigma de programación lo soportan lenguajes como 'C++', 'Java' y 'C#'.

Paradigma de Programación

--<<Conjunto de teorías, estándares y métodos que juntos representan una forma de organizar el conocimiento>> [Kuhn 1970].

--“Un paradigma de programación indica un método de realizar cálculos y la manera en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa”.

Orientado a Objetos


Es una forma de programar que se acerca más a como nosotros expresamos las ordenes en nuestra vida normalmente.

Haciendo así la programación mucho más simple que las instrucciones secuenciales y de forma estructurada.



Programación Orientada a Objetos

La programación orientada a objetos consiste en ordenar datos en conjuntos modulares de elementos de información del mundo real (denominado un dominio). Estos elementos de datos se llaman objetos. Estos datos se agrupan de acuerdo a las características principales del mundo real de estos elementos.




Objetos





Objetos

Representación detallada, concreta y particular de un algo. Tal representación determina su identidad, su estado y su comportamiento particular en un momento dado. Se trata de un ente abstracto usado en programación que permite separar los diferentes componentes de un programa, simplificando así su elaboración, depuración y posteriores mejoras.

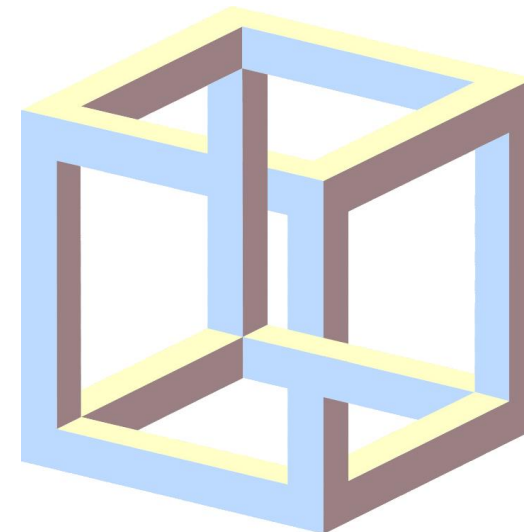


Objetos

Los objetos son/representan cosas

Los objetos pueden ser simples o complejos

Los objetos pueden ser reales o imaginarios






Objetos

Atributos: estos son los datos que caracterizan al objeto. Son variables que almacenan datos relacionados al estado de un objeto.


Identidad: El objeto tiene una identidad, que lo distingue de otros objetos, sin considerar su estado. Por lo general, esta identidad se crea mediante un identificador que deriva naturalmente de un problema





Métodos

Métodos (usualmente llamados funciones de miembro): Los métodos de un objeto caracterizan su comportamiento, es decir, son todas las acciones (denominadas operaciones) que el objeto puede realizar por sí mismo.






Métodos

Estas operaciones hacen posible que el objeto responda a las solicitudes externas (o que actúe sobre otros objetos).


Además, las operaciones están estrechamente ligadas a los atributos, ya que sus acciones pueden depender de, o modificar, los valores de un atributo.





Mensaje


Aunque los objetos se han de diferenciar claramente en una aplicación, estos se han de poder comunicar para trabajar en conjunto y construir así aplicaciones.





Mensaje

Esto se hace posible a través de lo que se denomina paso de mensajes. Cuando un objeto quiere comunicarse con otro lo que hace es enviarle un mensaje con los datos que desea transmitir.



Conceptos Básicos

The Four Pillars



Conceptos Básicos



Programación Orientada a Objetos

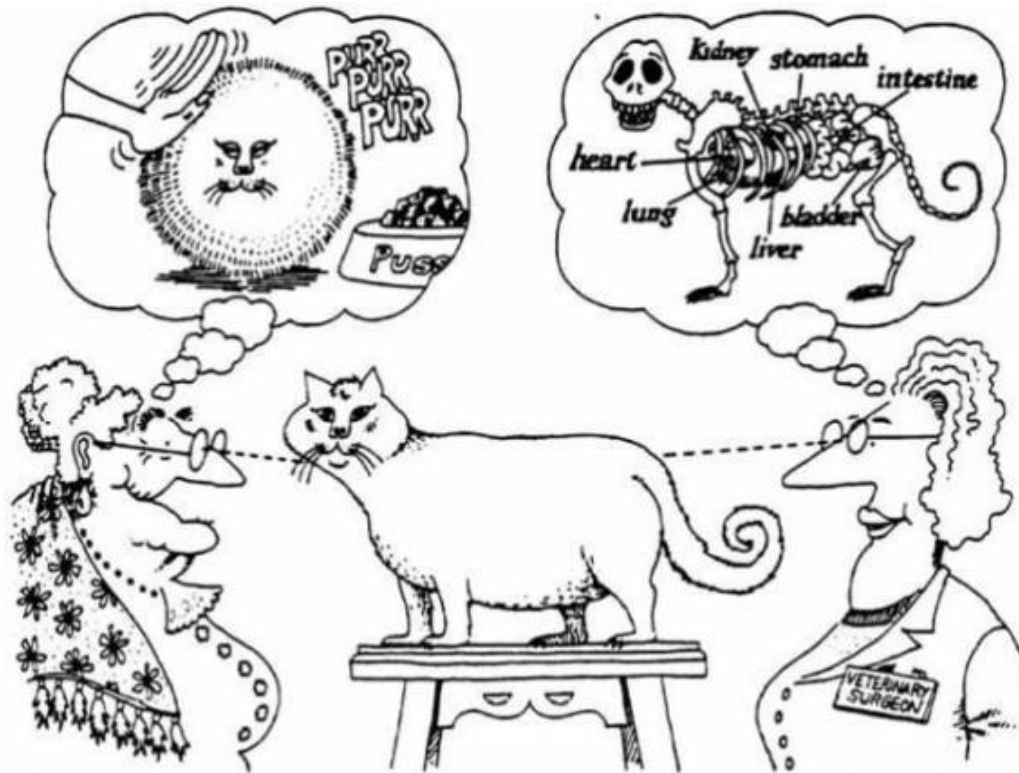


Abstracción

Un proceso que involucra reconocer y centrarse en las características importantes de una situación o un objeto, filtrando o ignorando los detalles no esenciales.

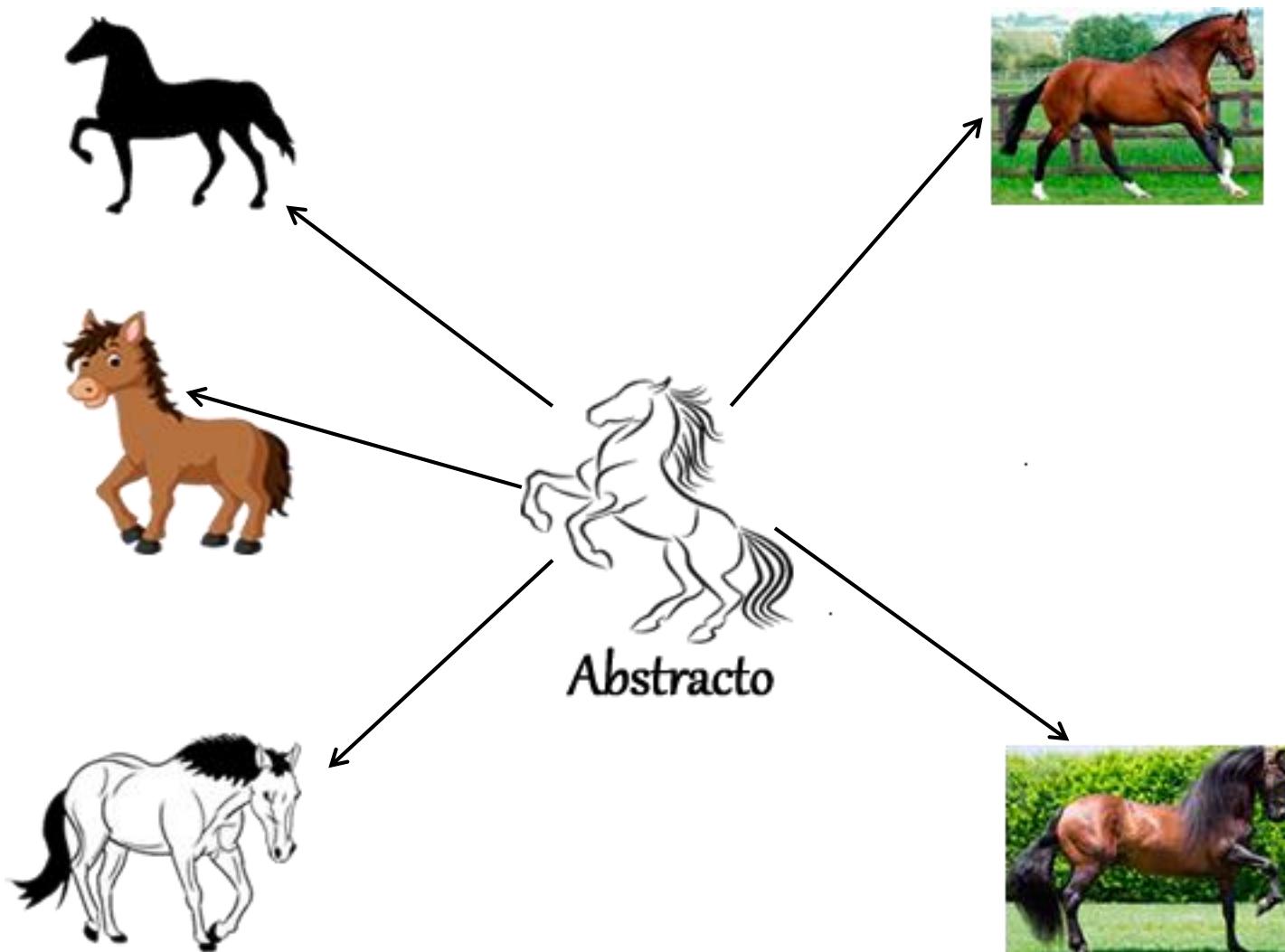


Abstracción



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

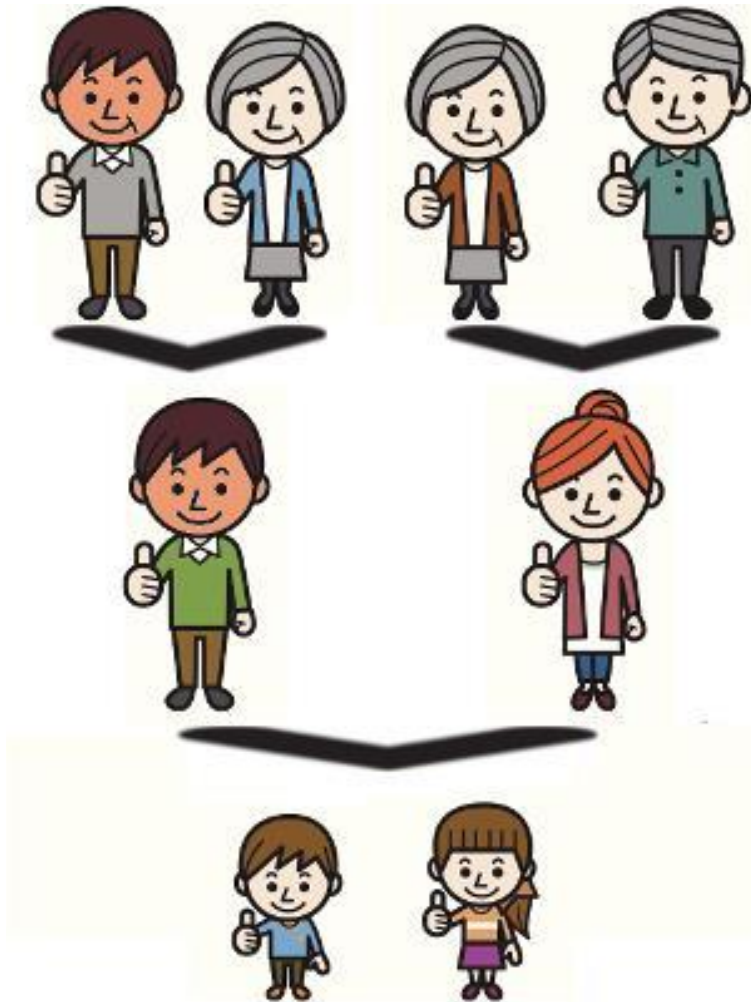
Abstracción



Herencia



Herencia



Herencia


La herencia es específica de la programación orientada a objetos, donde una clase nueva se crea a partir de una clase existente. La herencia (a la que habitualmente se denomina subclase) proviene del hecho de que la subclase (la nueva clase creada) contiene los atributos y métodos de la clase primaria. La principal ventaja de la herencia es la capacidad para definir atributos y métodos nuevos para la subclase, que luego se aplican a los atributos y métodos heredados.



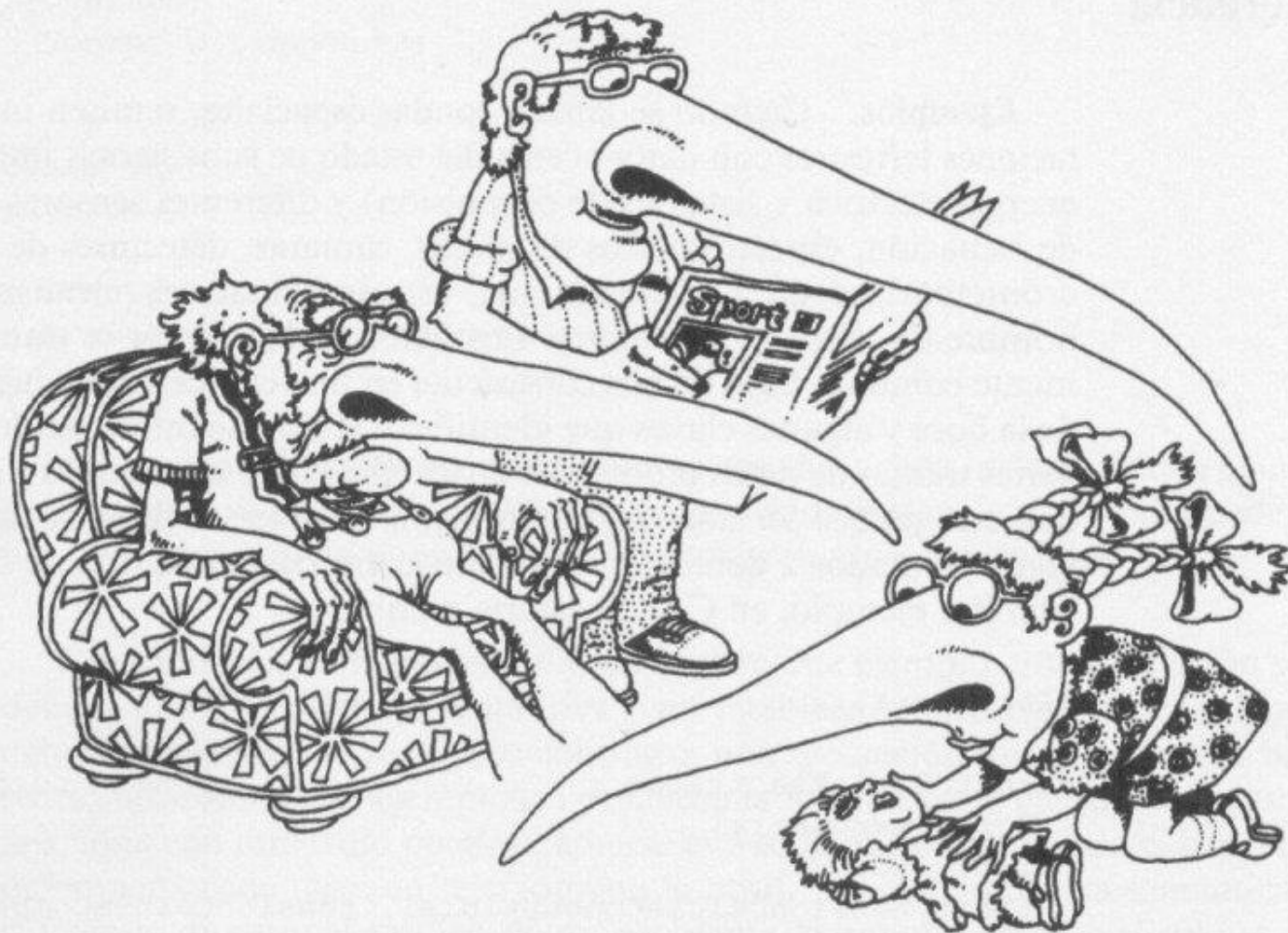
Herencia

Las características no privadas son las que se obtienen al heredar.

La herencia permite crear una estructura jerárquica de clases cada vez más especializada. La gran ventaja es que uno ya no debe comenzar desde cero cuando desea especializar una clase existente.



Herencia



Una subclase puede heredar la estructura y comportamiento de su superclase.

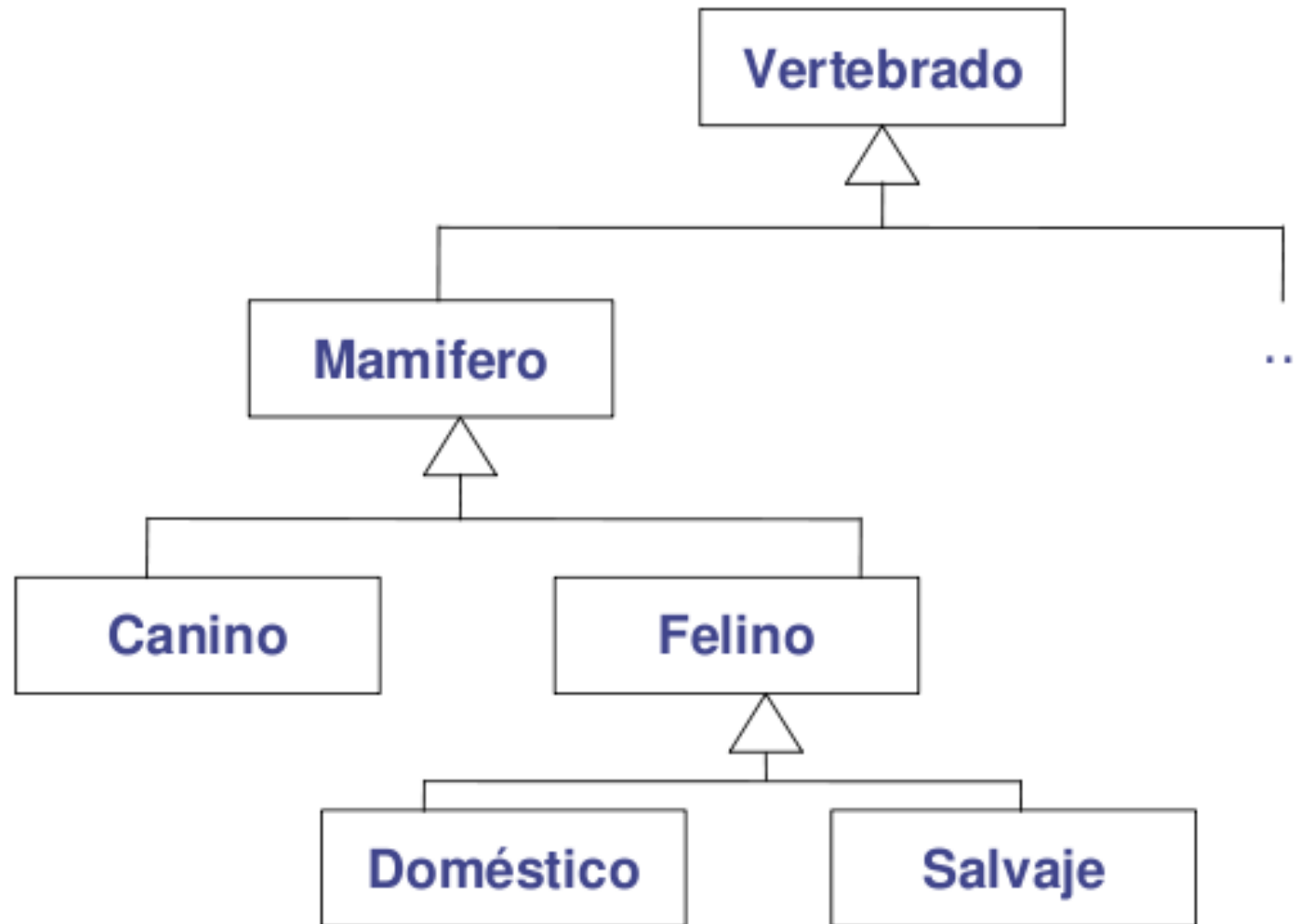
Jerarquía de clase

La relación padre-hijo entre clases puede representarse desde un punto de vista jerárquico, denominado vista de clases en árbol. La vista en árbol comienza con una clase general llamada superclase (a la que algunas veces se hace referencia como clase primaria, clase padre, clase principal, o clase madre; existen muchas metáforas genealógicas).

Jerarquía de clase

Las clases derivadas (clase secundaria o subclase) se vuelven cada vez más especializadas a medida que van descendiendo en el árbol. Por lo tanto, se suele hacer referencia a la relación que vincula una clase secundaria con una clase primaria mediante la frase es una x o y.

Herencia



Herencia

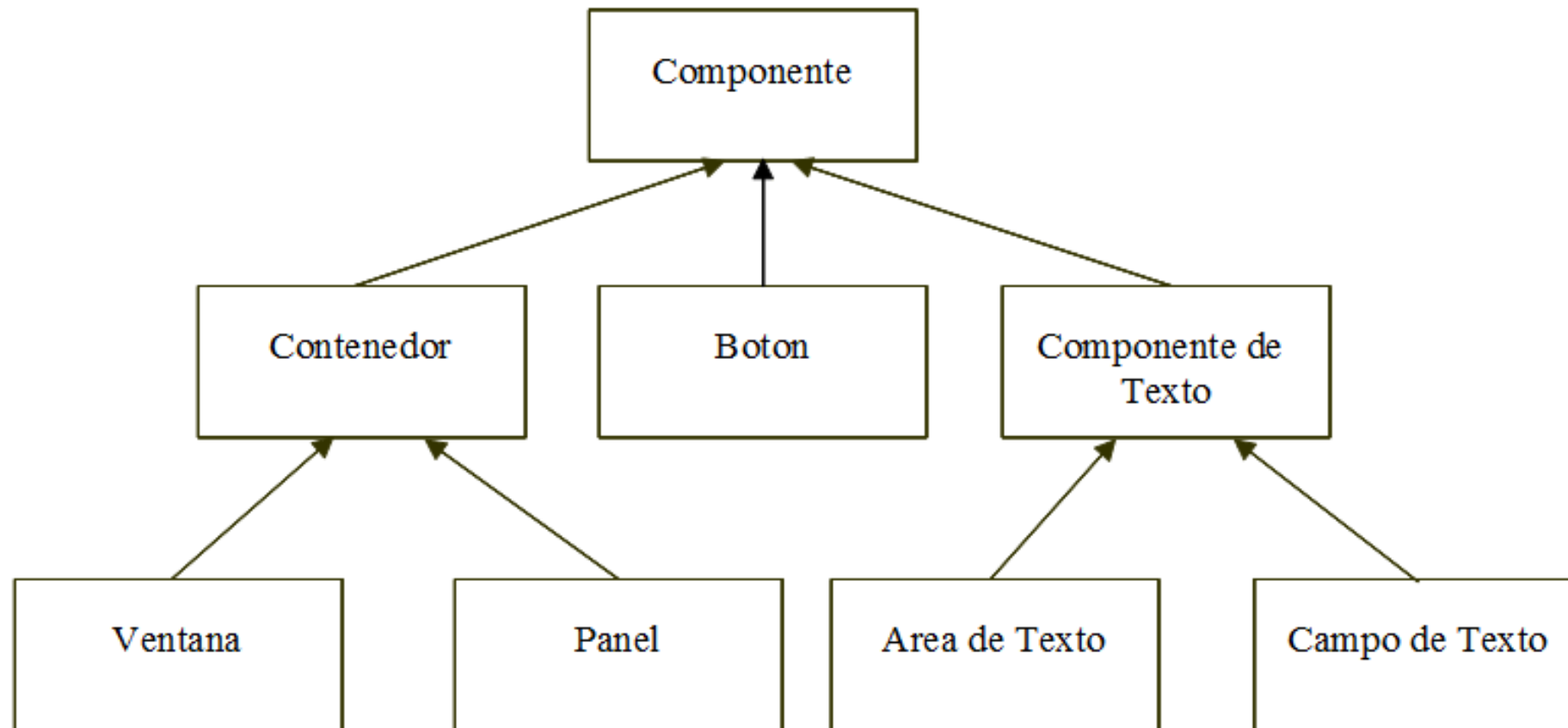



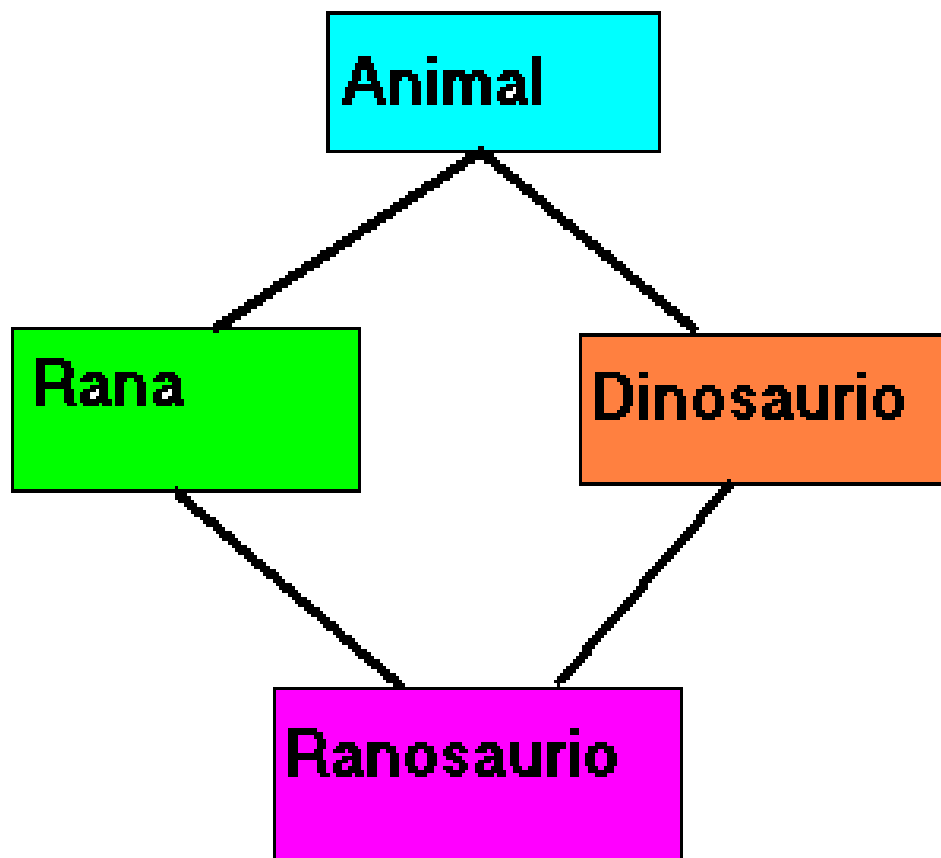
Diagrama de Herencia de Componentes

Herencia múltiple

Algunos lenguajes orientados a objetos, como C++ permiten herencias múltiples, lo que significa que una clase puede heredar los atributos de otras dos superclases. Esta forma puede utilizarse para agrupar atributos y métodos desde varias clases dentro de una sola.

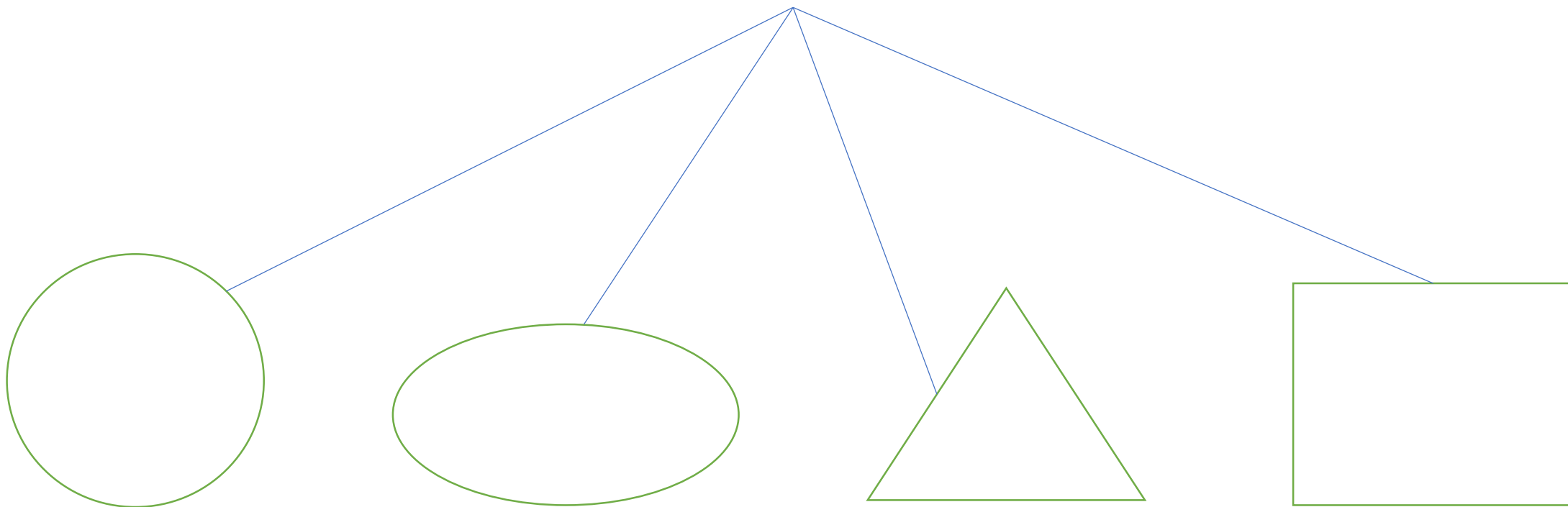


Herencia



Polimorfismo


Figura Geométrica





Polimorfismo


Es la capacidad que tienen los objetos de una clase para responder a los mensajes o eventos en función de los parámetros utilizados durante su invocación. Es una característica que resulta de gran ayuda en programación pues facilita la claridad y consistencia del código, aunque es un concepto bastante genérico (y frecuentemente malinterpretado).





Polimorfismo

Se conoce con este nombre el hecho de que un método tiene el mismo nombre y resulta en el mismo efecto básico pero está implementado de forma distinta en las distintas clases de una jerarquía.





Polimorfismo

Los objetos pueden tener un comportamiento diferente ante distintos mensajes, es decir, pueden realizar la misma acción pero de distinta forma según la interacción que tenga con otros objetos.

El polimorfismo está muy ligado a la herencia.




Encapsulamiento





Encapsulamiento


Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema.





Encapsulamiento

La utilidad de encapsulamiento va por la facilidad para manejar la complejidad, ya que tendremos a las clases como cajas negras donde sólo se conoce el comportamiento pero no los detalles internos, y esto es conveniente porque nos interesará conocer qué hace la clase pero no será necesario saber cómo lo hace.



BIBLIOGRAFÍA

PROGRAMACION ORIENTADA A OBJETOS EN JAVA, Francisco Blasco
Programación Orientada a Objetos, Juan Carlos Moreno



UNIVERSIDAD
Popular del Cesar

Vamos a la práctica