

Fundamentos de la Tecnología Orientada a Objetos



Fundamentos de la Orientación a Objetos

ENLACE
DINAMICO

TERMINOS TECNICOS

ABSTRACCION

MODULARIDAD

ENCAPSULAMIEN.

JERARQUIA

TIPOS

CONCURRENCIA

PERSISTENCIA

POLIMORFISMO

PRINCIPIOS

OBJETO

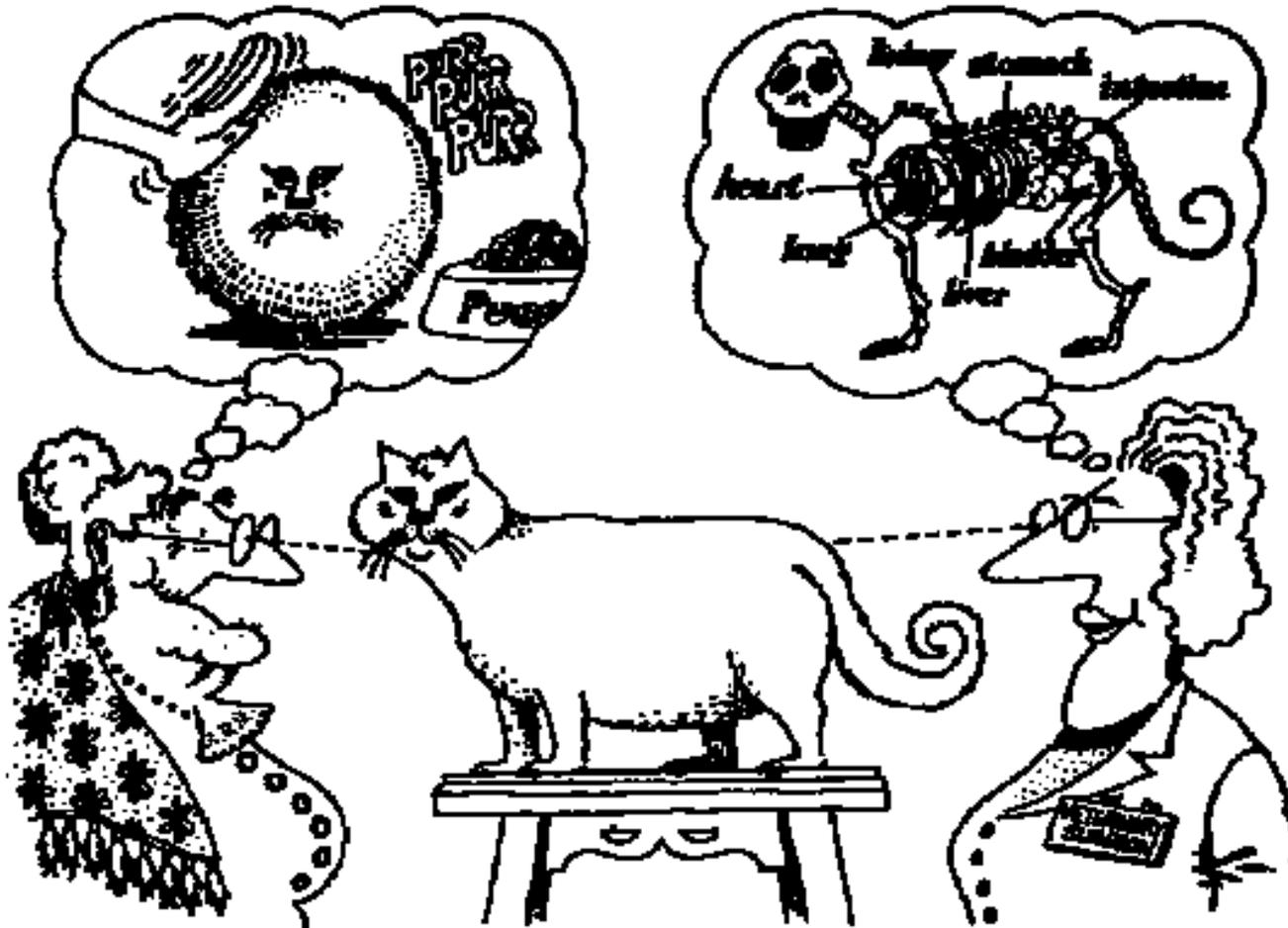
CLASE

HERENCIA

MECANISMOS BASICOS

Principios de la Orientación a Objetos

✓ Abstracción



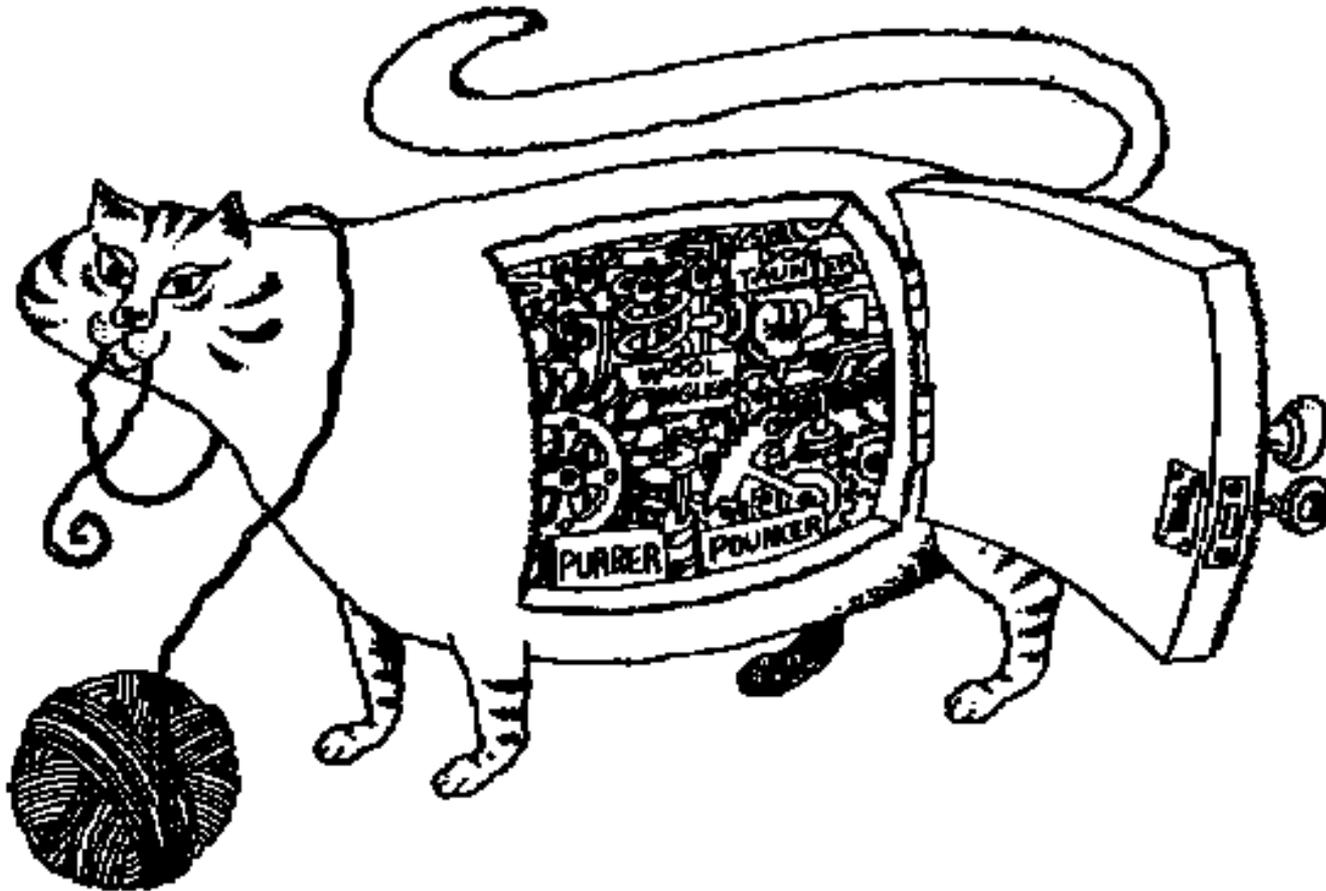
Principios de la Orientación a Objetos

✓ Abstracción

- Capacidad de adquirir conceptos (idea particular o lo que vemos y comprendemos del mundo), construir jerarquías de generalizaciones y componer objetos mediante la configuración de otros.
- Permite manejar la complejidad de los objetos.
- Enfatizar sobre los detalles importantes y olvidar los aspectos no relevantes desde perspectiva del usuario.
- Permite separar el comportamiento esencial de un objeto de su implementación.

Principios de la Orientación a Objetos

✓ Encapsulamiento



Principios de la Orientación a Objetos

✓ Encapsulamiento

- Esconder todos los detalles de un objeto que no contribuyen al entendimiento de sus características esenciales. Información escondida.
- Se refiere a incluir dentro de un objeto todo lo que éste necesita y hacerlo de forma tal que ningún otro objeto vea su estructura interna.
- Permite realizar cambios a los programas con poco esfuerzo.
- La idea principal es proteger los datos de un objeto del uso arbitrario.

Principios de la Orientación a Objetos

✓ Modularidad



Principios de la Orientación a Objetos

✓ Modularidad

- Descomponer un sistema en un conjunto de unidades discretas.
- Está asociado al concepto de programación modular / estructurada.
- Permite empaquetar la abstracción en pequeñas unidades.
- Está íntimamente relacionado con el encapsulamiento.
- Los módulos son generalmente unidades indivisibles.



Principios de la Orientación a Objetos

✓ Modularidad

Regla 1 – Correspondencia directa:

La estructura modular obtenida en el proceso de construcción de software debe seguir siendo compatible con cualquier otra estructura modular obtenida en el proceso de modelado del dominio del problema.

Regla 2 – Pocas interfaces:

Cada módulo debe comunicarse con el menor número de módulos posible.

Principios de la Orientación a Objetos

✓ Modularidad

Regla 3 – Pequeñas interfaces:

Si dos módulos se comunican deben intercambiar la menor información posible.

Regla 4 – Interfaces explícitas:

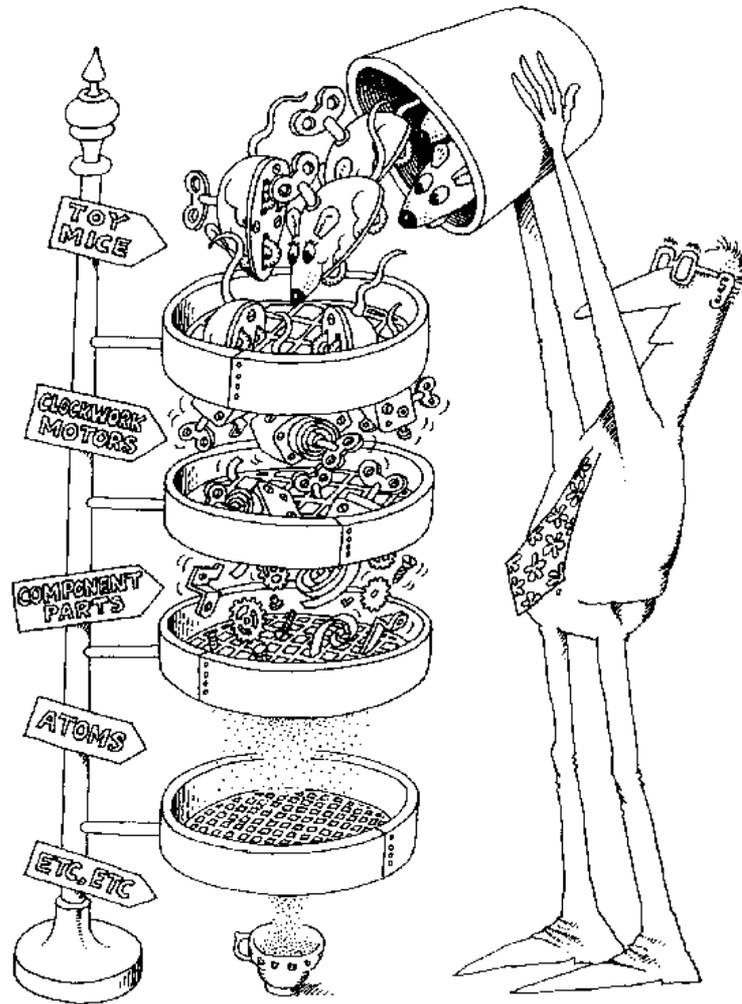
Siempre que dos módulos A y B se comuniquen, esto debe ser obvio a partir del contexto de A, de B o de ambos.

Regla 5 – Ocultamiento de información:

Cada módulo debe tener seleccionada un subconjunto de propiedades como información oficial del módulo para ponerla a disposición de otros módulos.

Principios de la Orientación a Objetos

✓ Jerarquía



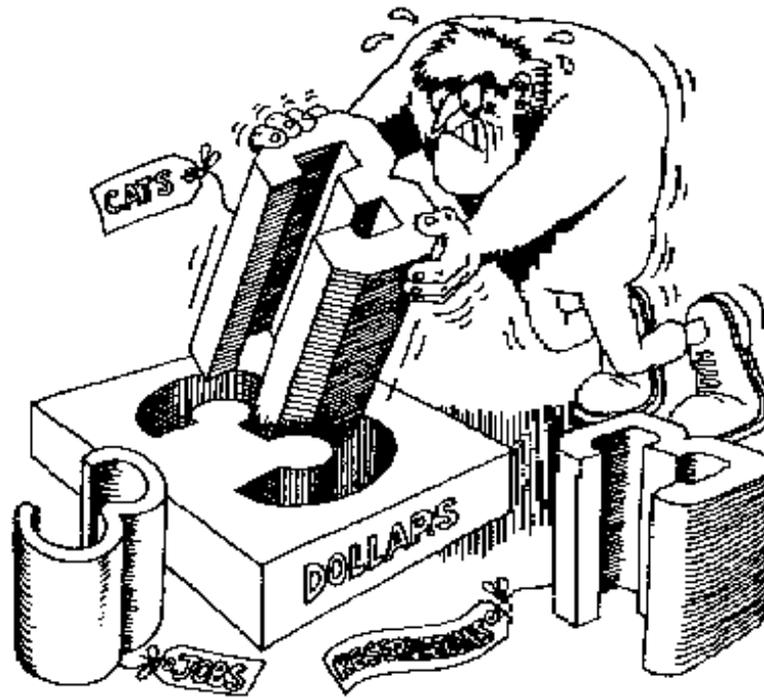
Principios de la Orientación a Objetos

✓ Jerarquía

- Es el acto o resultado de distinguir un concepto que es más general que otro.
- Permite examinar si los conceptos tienen algo en común.
- Nos ayuda a percibir que todas las instancias de un concepto más específico son también instancias de un concepto general.
- Permite empaquetar la abstracción en pequeñas unidades.
- Representa la organización de una abstracción.
- Su uso simplifica el entendimiento del problema.
- Se implementa mediante las relaciones de herencia y agregación entre clases.

Principios de la Orientación a Objetos

✓ Tipos



Principios de la Orientación a Objetos

✓ Tipos

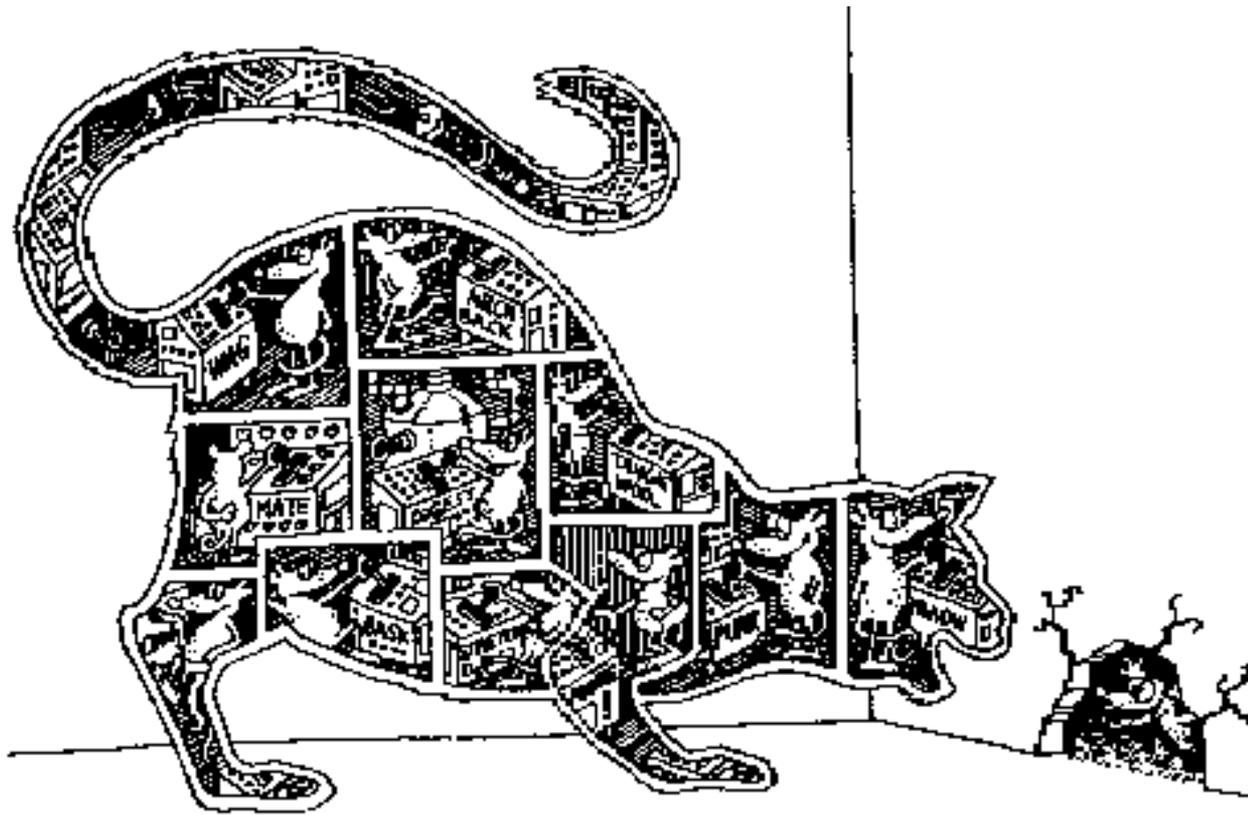
- Permite agrupar o formar un conjunto de objetos con características o comportamientos similares.
- Dos cosas que tienen la misma forma abstracta son análogas => son del mismo tipo.
- Se implementa mediante la descripción o identificación de los elementos que describen una clase:

datos / información => atributos

operaciones / comportamiento => métodos

Principios de la Orientación a Objetos

✓ Concurrencia



Principios de la Orientación a Objetos

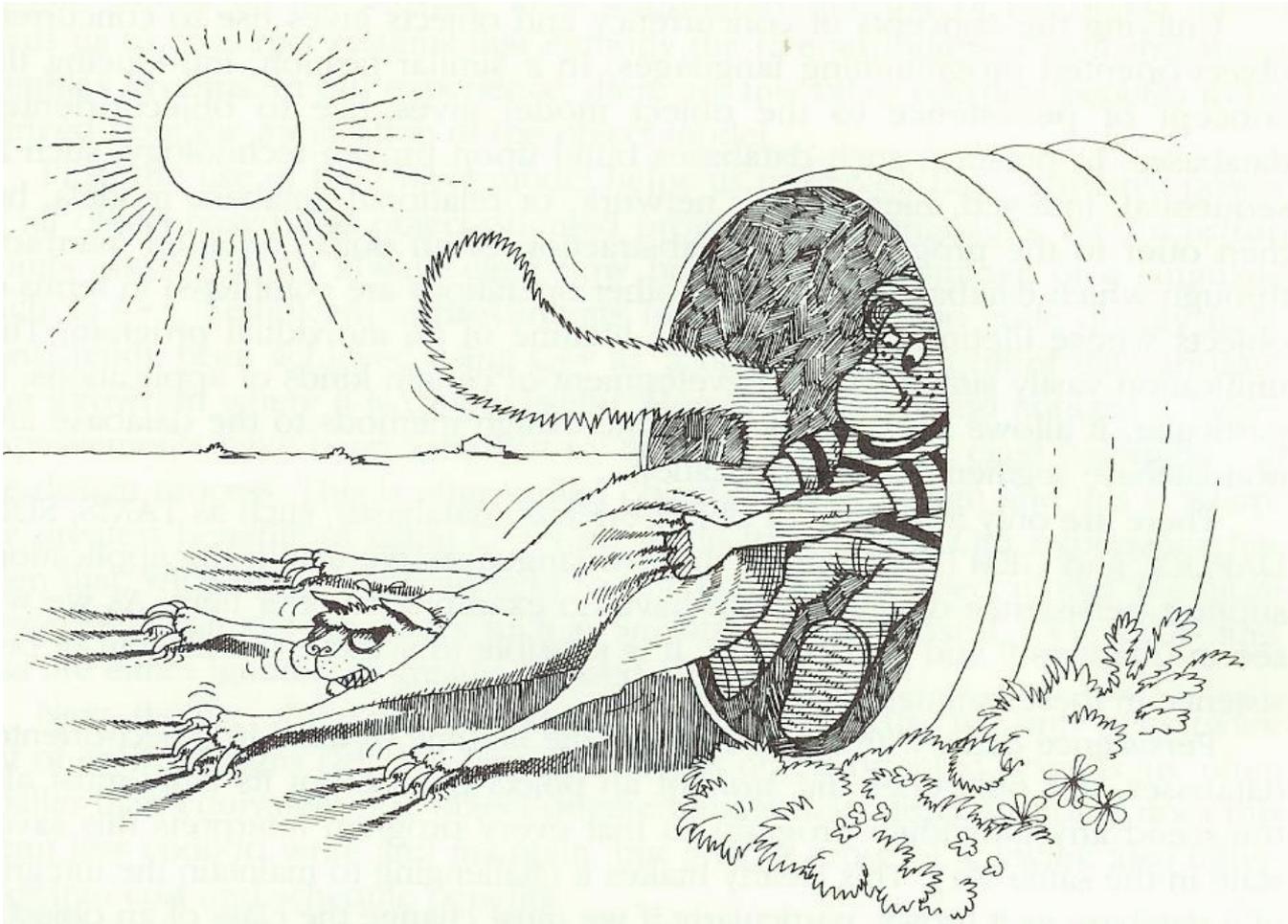
✓ Concurrencia

- Es la propiedad que distingue un objeto activo de uno que no lo es.
- Permite a diferentes objetos actuar al mismo tiempo; cada uno de ellos tiene autonomía.



Principios de la Orientación a Objetos

✓ Persistencia



Principios de la Orientación a Objetos

✓ Persistencia

- Es la propiedad que un objeto tiene de existir en el tiempo y en el espacio.
- Un objeto existe hasta que no sea más necesitado y el espacio que ocupa es reutilizado.



Principios de la Orientación a Objetos

✓ Polimorfismo



Principios de la Orientación a Objetos

✓ Polimorfismo

- Es el fenómeno mediante el cual una operación adopta varias formas de implementación.
- Habilidad para usar el mismo símbolo para propósitos diferentes cuando el contexto es claro.
- Para el usuario, la visión del operador es una sola, el objeto se encarga de decidir qué hacer; los detalles de implementación quedan ocultos.
- Dos operaciones completamente diferentes comparten el mismo nombre.
- Ventajas: extensibilidad, código compacto, claridad.
- Se implementa mediante la sobrecarga de funciones y operadores.

Términos Técnicos de la Orientación a Objetos

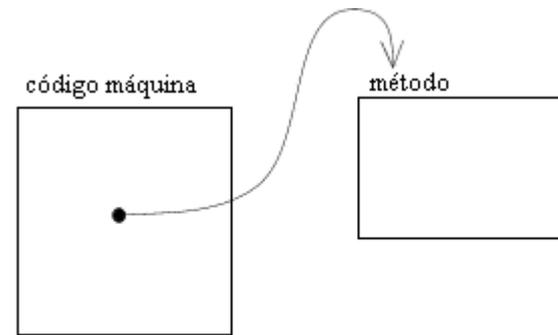
✓ Enlace Dinámico

- Un enlace es conectar una llamada de función al cuerpo de la función; cuando el enlace se lleva a cabo antes de que el programa se ejecute se llama enlace temprano o estático; el enlace posterior o dinámico ocurre en tiempo de ejecución.
- Representa uno de los mayores avances de la Orientación a Objetos y es una propiedad común (no necesaria) de los lenguajes orientados a objetos.
- Es un elemento necesario para poder realizar el polimorfismo.
- Para su implementación, debe existir un mecanismo que determine el tipo de objeto en tiempo de ejecución y haga la llamada a la función apropiada.

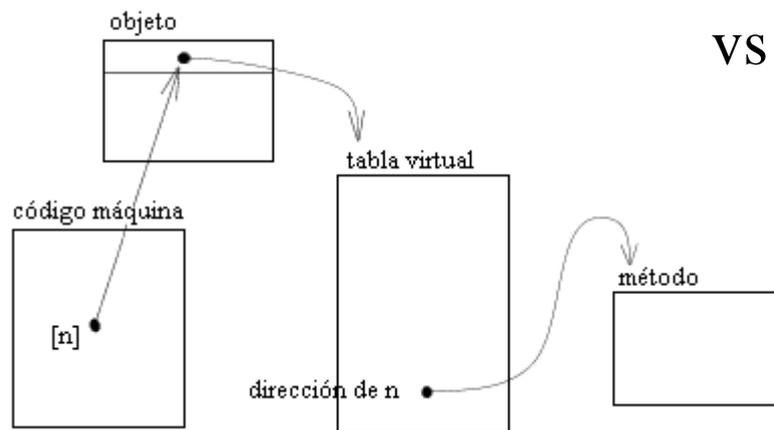
Términos Técnicos de la Orientación a Objetos

✓ Enlace Dinámico

enlace estático →



VS



← enlace dinámico

Mecanismos básicos de la Orientación a Objetos

✓ Objeto

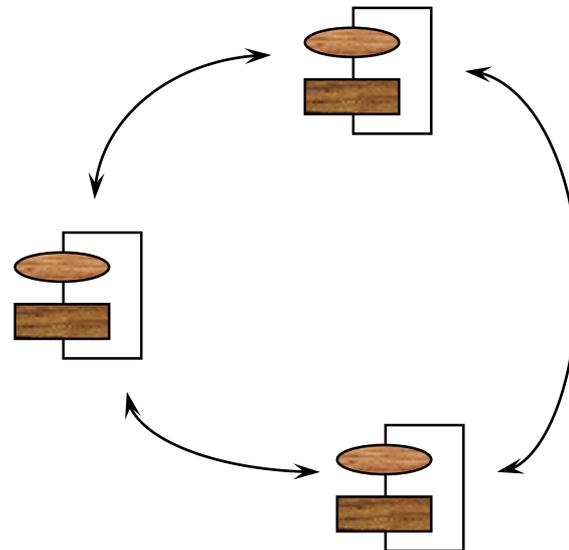
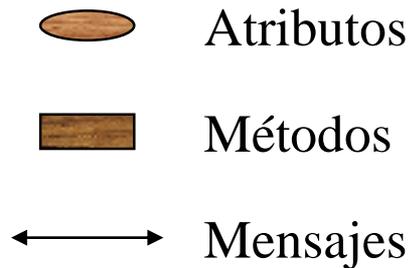
- Son las unidades básicas de construcción, para conceptualización, diseño o programación.
- Son instancias organizadas en clases con características comunes.
- Desde la perspectiva de un programa, los objetos son módulos que contienen datos (atributos) y las instrucciones u operaciones (métodos) que operan sobre esos datos y trabajan juntos para proveer funcionalidad.
- La habilidad para reconocer objetos es una herramienta que los humanos aprenden desde muy temprana edad.
- 3 características: estado, identidad y comportamiento.

Mecanismos básicos de la Orientación a Objetos

✓ Objeto

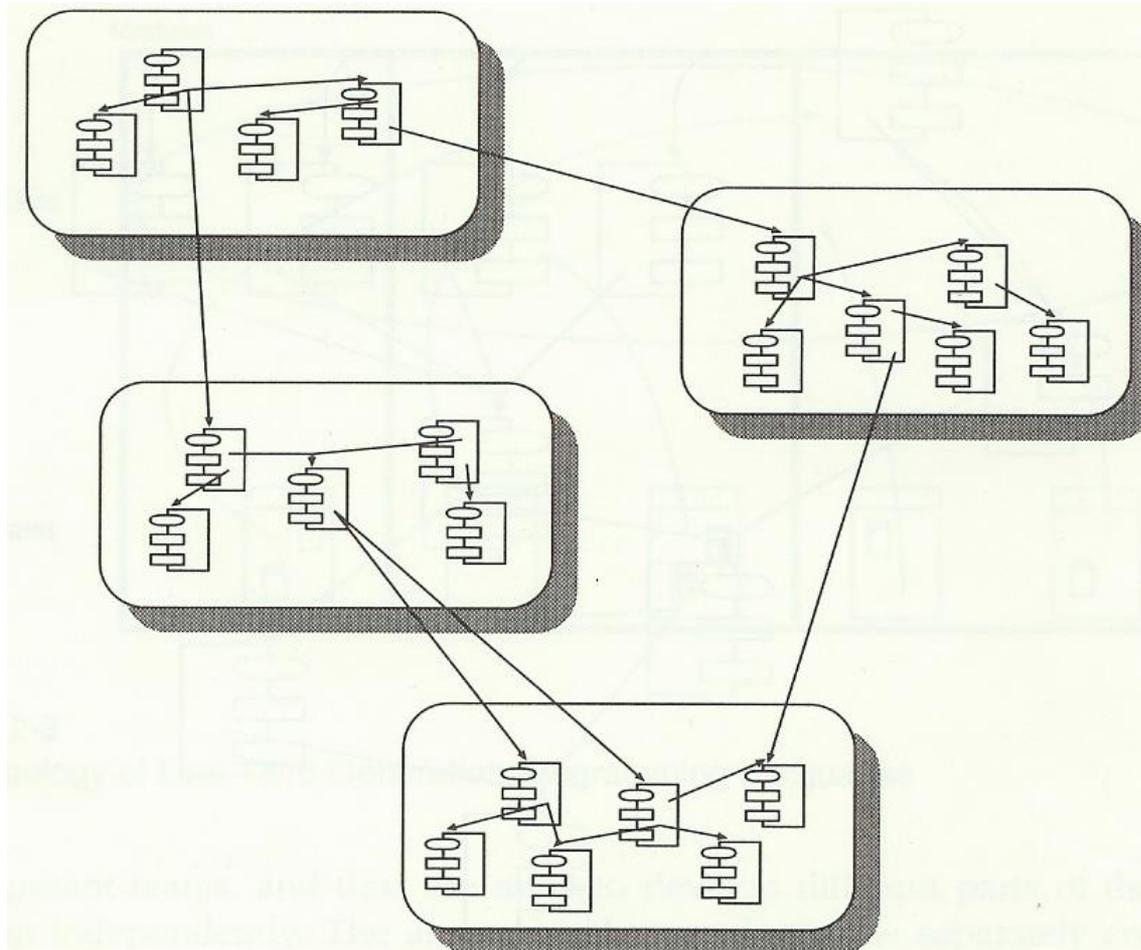
Fórmula conceptual 2:

$$\text{Objeto} = \text{Atributos} + \text{Métodos}$$



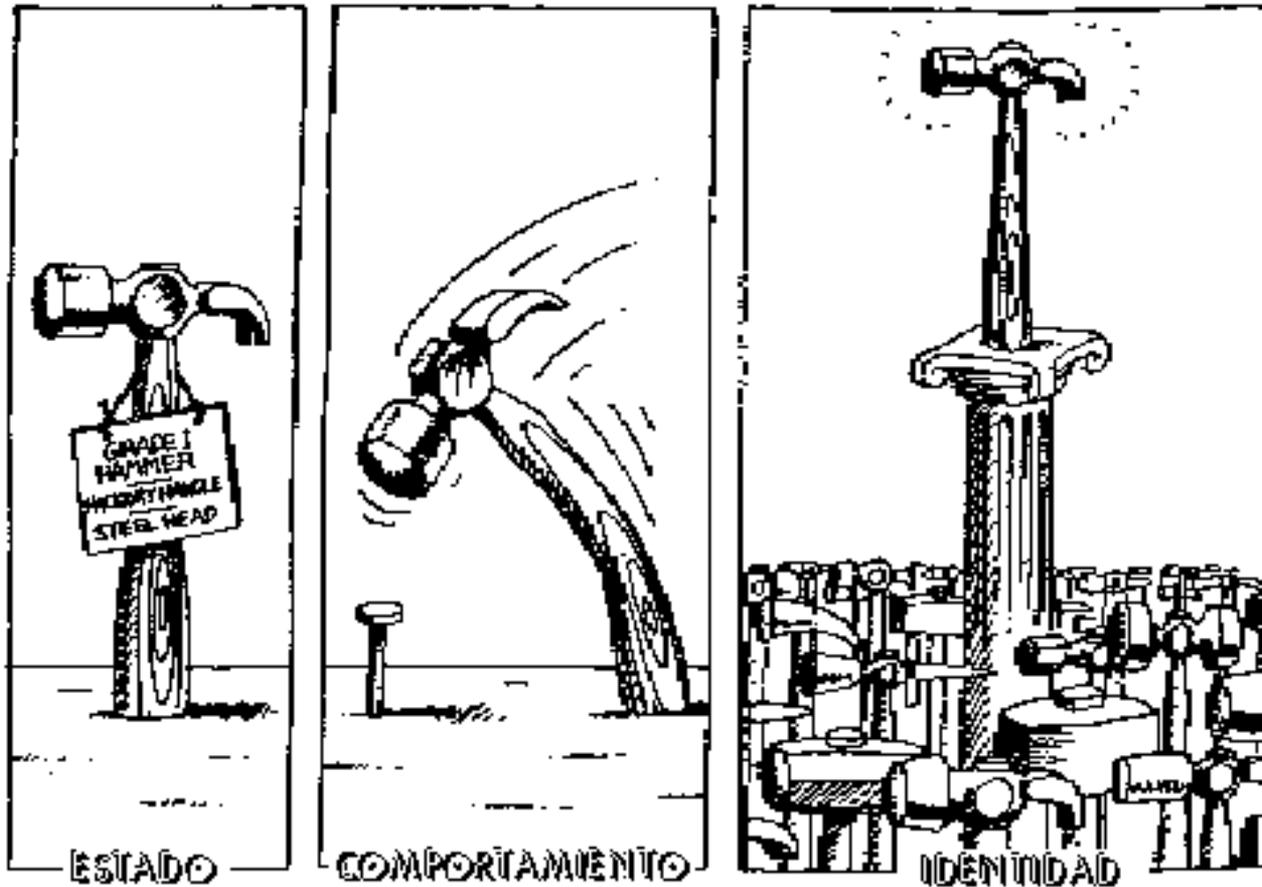
Mecanismos básicos de la Orientación a Objetos

✓ Objeto



Mecanismos básicos de la Orientación a Objetos

✓ Objeto



Mecanismos básicos de la Orientación a Objetos

✓ Objeto - Estado

- Contiene todas las propiedades (usualmente estáticas), además de los valores actuales de estas propiedades (usualmente dinámicas).
- Hace que cada objeto sea único.
- La respuesta que un objeto puede dar, luego de la llamada de un mensaje, depende de su estado actual.
- Ejemplos: color – rojo; peso – 75.4; edad – 24; número de ítems – 190; profesión – “Arquitecto”.

Mecanismos básicos de la Orientación a Objetos

✓ Objeto - Identidad

- Distingue a un objeto del resto.
- Es independiente del estado.
- Se conserva durante toda la vida del objeto, aún cuando se realicen cambios de estado.
- Es una propiedad muy importante para las Bases de Datos Orientadas a Objetos (OODB).



Mecanismos básicos de la Orientación a Objetos

✓ Objeto - Comportamiento

- Indica la forma como el objeto actúa y reacciona en término de los cambios de estado.
- Está totalmente definido por sus acciones u operaciones.
- 2 conceptos importantes: método y mensaje.



Mecanismos básicos de la Orientación a Objetos

✓ Objeto - Comportamiento

Método:

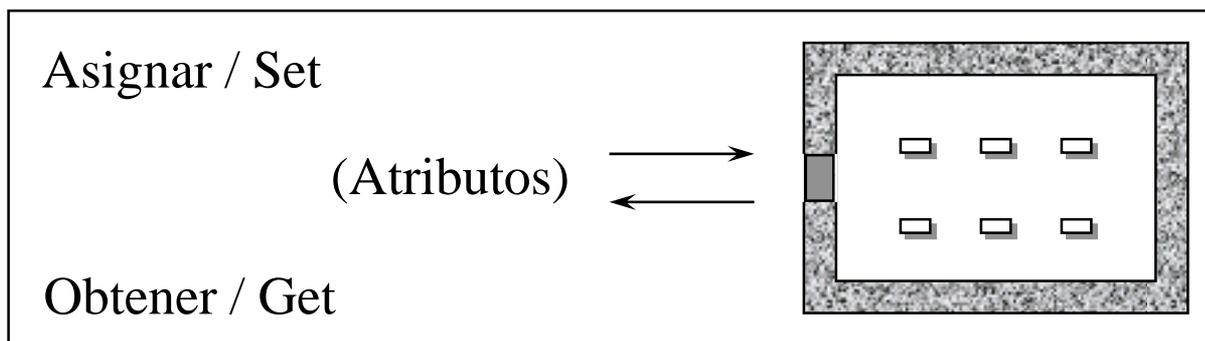
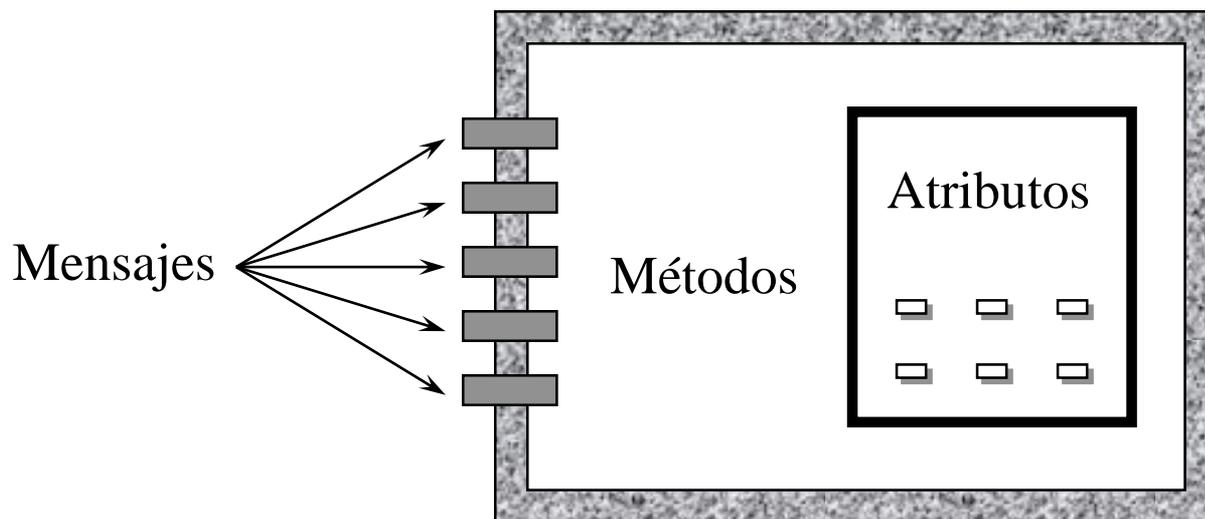
Procedimiento que reside en el objeto y determinan cómo éste actuará cuando reciba un mensaje. Su ejecución puede cambiar el estado del objeto, crear objetos nuevos, enviar mensajes a otros objetos, etc.

Mensaje:

Canal de comunicación que usan los objetos para ejecutar acciones. Sólo cuando se recibe un mensaje, el objeto ejecuta una acción. Todo proceso es activado por mensajes entre objetos.

Mecanismos básicos de la Orientación a Objetos

✓ Objeto - Comportamiento



Mecanismos básicos de la Orientación a Objetos

✓ Objeto - Comportamiento

Fórmula conceptual 3:

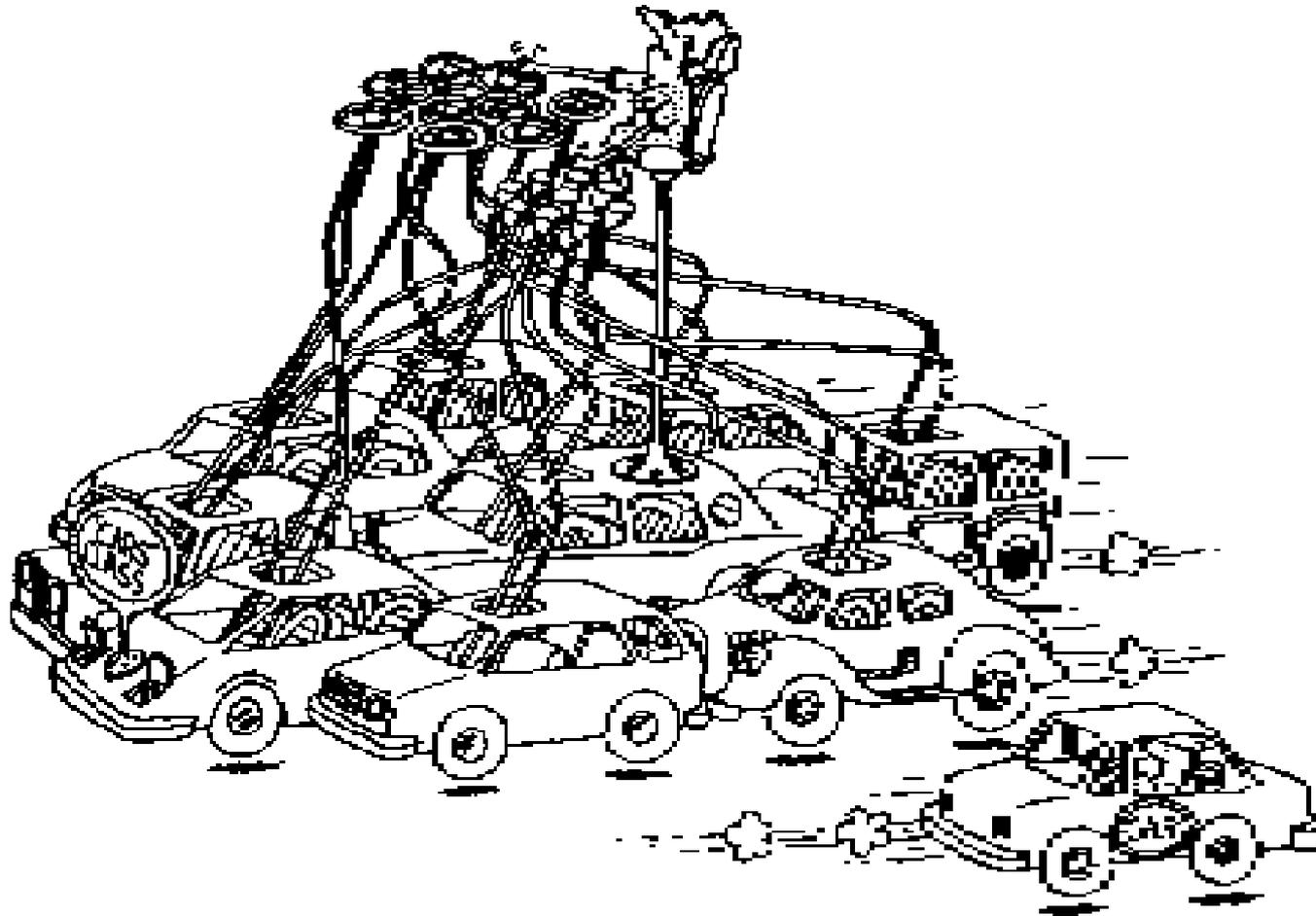
Mensaje = Identidad del Objeto + Método + Parámetros

Protocolo del Objeto:

Es el conjunto de mensajes a los cuales un objeto responde o puede responder. Nótese que no necesariamente es igual al conjunto de métodos del objeto (encapsulamiento y polimorfismo).

Mecanismos básicos de la Orientación a Objetos

✓ Clase



Mecanismos básicos de la Orientación a Objetos

✓ Clase

- Define la estructura y el comportamiento de una forma abstracta o concepto para darle vida a los objetos.
- Es un patrón (plantilla) que define los atributos y métodos a ser incluidos es un tipo particular de objeto.
- Se dice que un objeto es una instancia particular de una clase.
- Consta de dos partes: una declaración y una implementación. La declaración lista o identifica los miembros de la clase (atributos y métodos). La implementación define el cuerpo de los métodos de la clase.

Mecanismos básicos de la Orientación a Objetos

✓ Clase - interfaz

- Determina la forma en la cual los elementos de la clase (atributos y métodos) pueden o no ser usados (son visibles) por cada uno de los tres tipos diferentes de usuarios:
 - elementos de la misma clase,
 - elementos de clases relacionadas (herencia) y
 - otros elementos externos.
- Tres niveles de interfaz:
 - Privada visibilidad sólo para la misma clase.
 - Protegida visibilidad para la misma clase y clases relacionadas.
 - Pública visibilidad para cualquiera.

Mecanismos básicos de la Orientación a Objetos

✓ Clase - interfaz

PUBLICA

PROTEGIDA

PRIVADA

Clase actual

Clases relacionadas

Otros usuarios externos

Mecanismos básicos de la Orientación a Objetos

✓ Clase - relaciones

- Basado en el principio de jerarquía, para la definición de una clase nueva se pueden usar clases existentes, ya sea incorporando un objeto como atributo de la clase (agregación) o estableciendo una generalización (herencia).

Para dos clases **A** y **B**:

A es una agregación de **B** si y sólo si

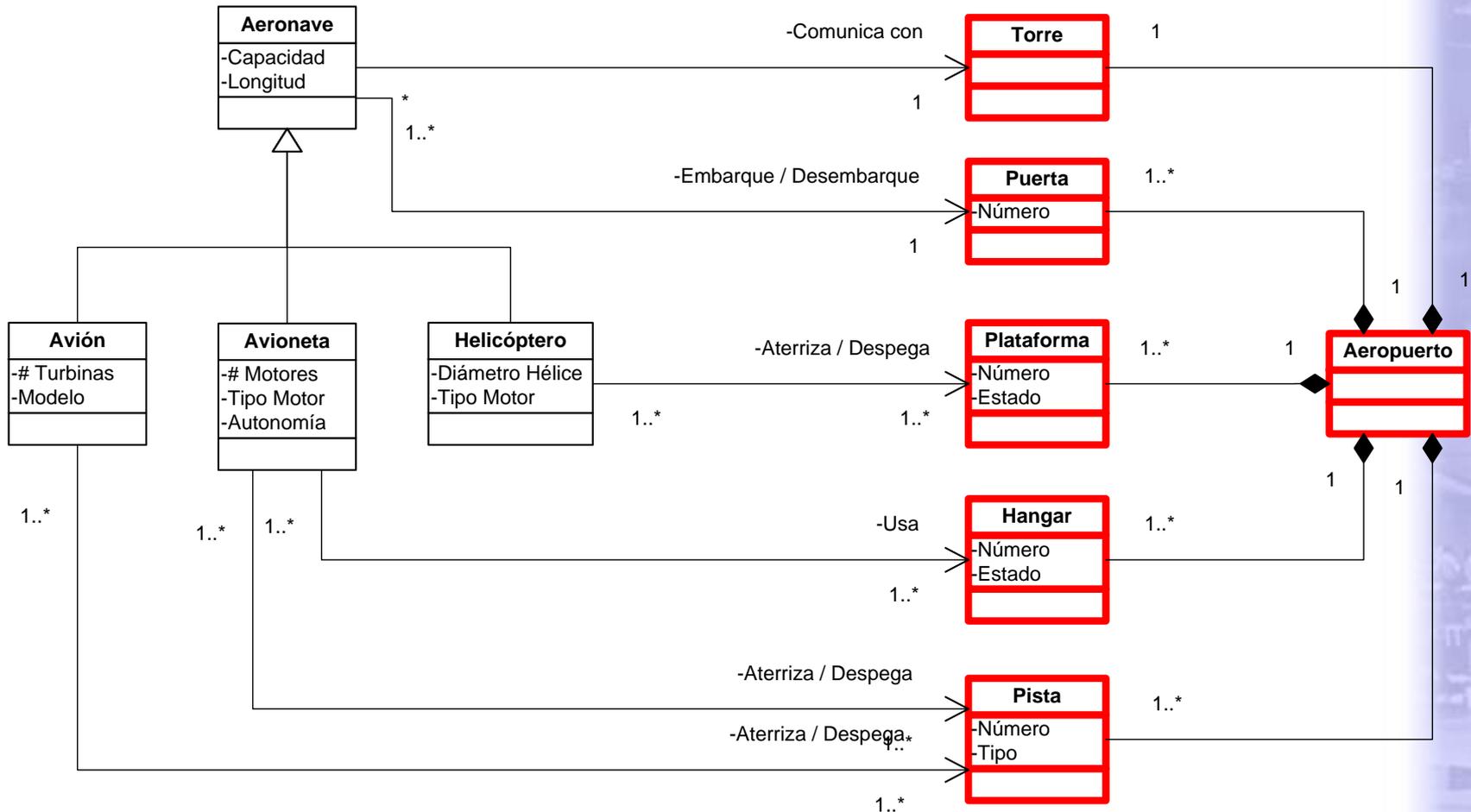
A forma parte de B

A hereda de **B** (B es una generalización de A) si y sólo si

A es un caso particular de B

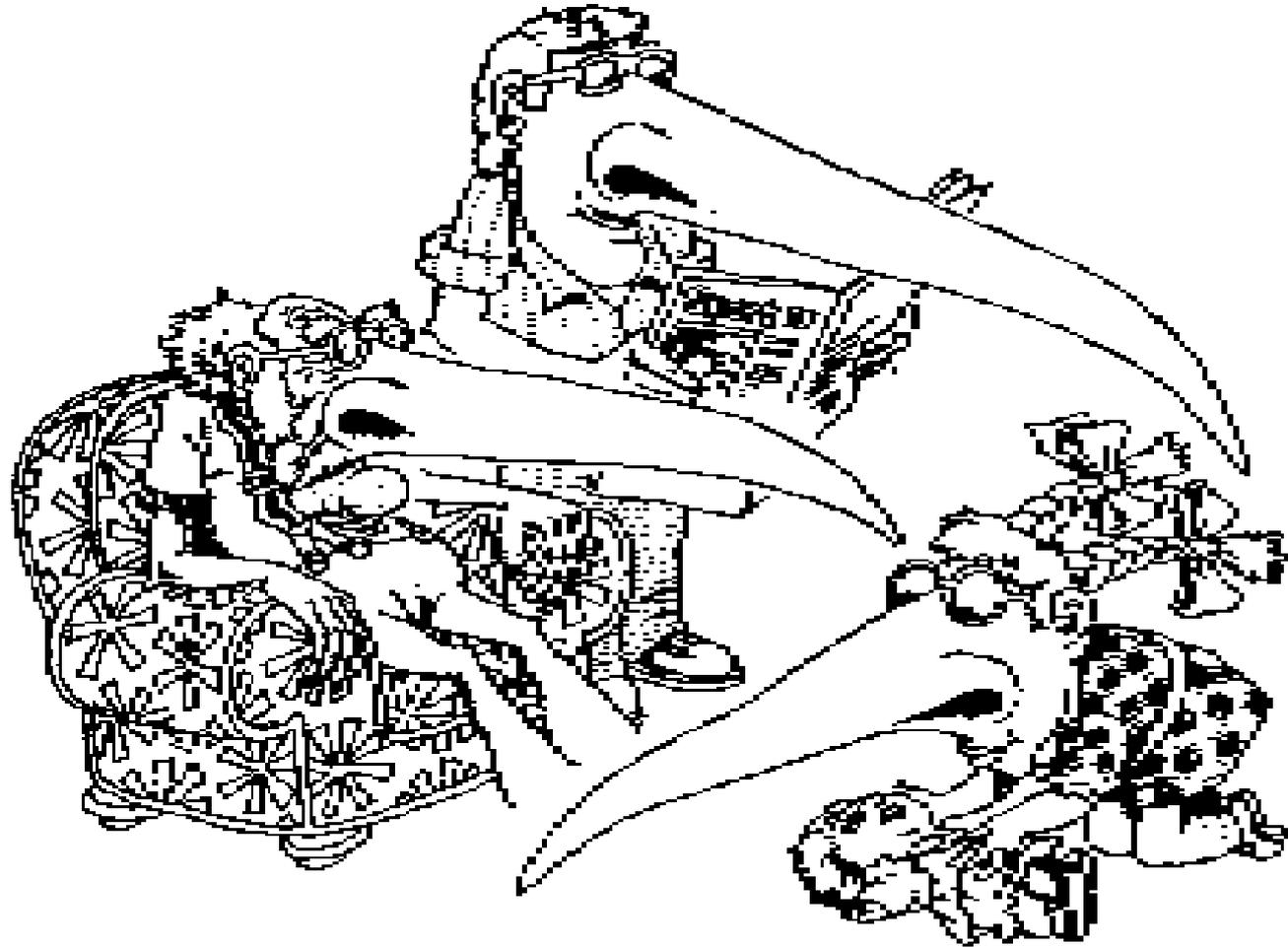
Mecanismos básicos de la Orientación a Objetos

✓ Clase – relaciones (ejemplo)



Mecanismos básicos de la Orientación a Objetos

✓ Herencia



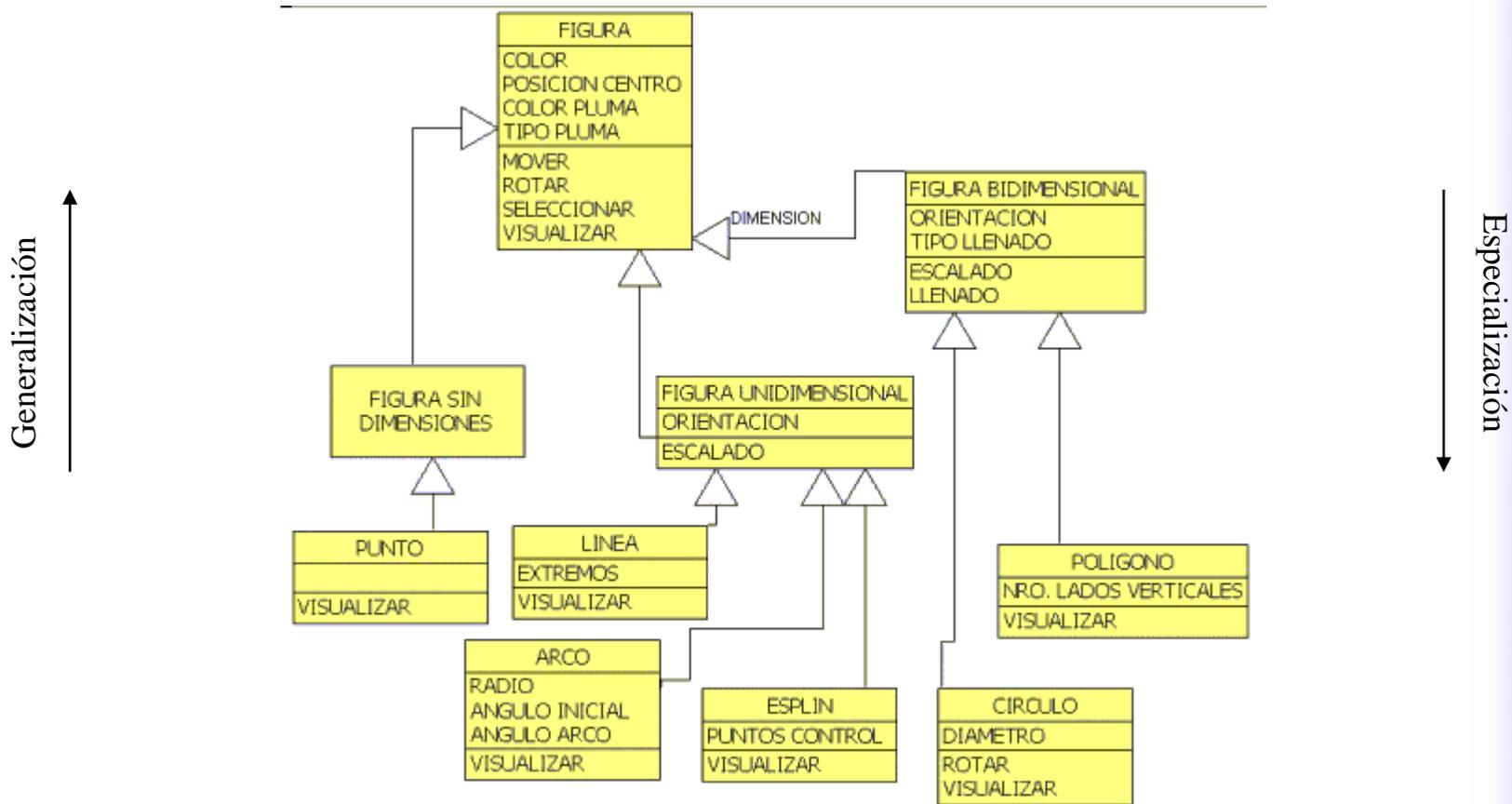
Mecanismos básicos de la Orientación a Objetos

✓ Herencia

- Relación entre clases, donde una clase comparte la estructura y/o el comportamiento de una clase base.
- Permite crear clases nuevas a partir de clases ya existentes, estableciendo niveles de jerarquía y programando sólo las diferencias.
- Permite refinar las estructuras sin necesidad de duplicar información.
- La clase nueva (jerarquía inferior) es conocida como subclase o clase derivada y la clase de mayor nivel se denomina superclase o clase base.
- Según el número de clases base, puede ser de dos tipos: simple (una sola clase base) y múltiple (mas de una clase base).

Mecanismos básicos de la Orientación a Objetos

✓ Herencia - ejemplo



Mecanismos básicos de la Orientación a Objetos

✓ Herencia

Fórmula conceptual:

$$\textit{Clase Derivada} = \sum \textit{Clases Base} + \textit{Otros Elementos}$$

Nota: Si el identificador de un método o atributo de una clase cualquiera es similar a alguno de los elementos de una posible clase base, cualquier referencia a este identificador estará asociado al elemento de la clase derivada, a menos que se indique lo contrario.

Nota: Cuando un objeto recibe un mensaje que contiene un método que no está definido en su clase, se busca automáticamente hacia arriba en su jerarquía de clases. Igual ocurre al acceder a un atributo del objeto.

Mecanismos básicos de la Orientación a Objetos

✓ Herencia – clases abstractas y métodos virtuales

Nota: Las clases abstractas son aquellas clases cuya descripción es incompleta y sólo sirven para ser usadas como clases bases en una relación de herencia. En términos de diseño, son útiles para permitir a un usuario refinar o particularizar un concepto. No es posible o no tiene sentido instanciar objetos con ellas. Por lo general contienen uno o más métodos virtuales.

Nota: Aunque pueden haber métodos genéricos definidos en algún nivel de la jerarquía de clases, no está prohibido que existan definiciones específicas o detalladas en los niveles inferiores. En este caso, ambos métodos pueden ser usados (como si fuera un polimorfismo), siempre y cuando no se trate de la redefinición de un método virtual.

Mecanismos básicos de la Orientación a Objetos

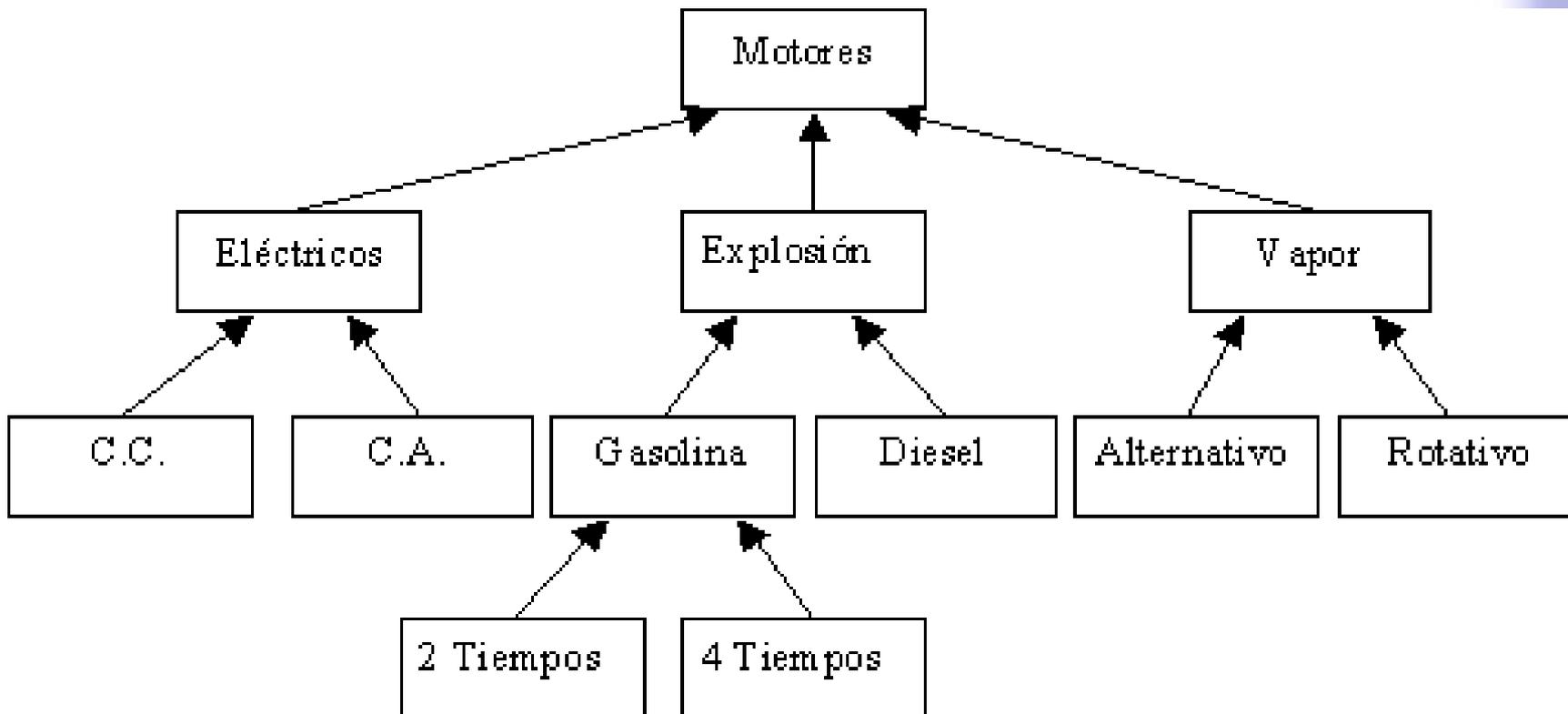
✓ Herencia – clases abstractas y métodos virtuales

Nota: Un método virtual es un método que pertenece a una clase abstracta y que debe ser obligatoriamente redefinido en las clases derivadas. Puede verse como una indicación a los usuarios de la clase base, que parte del trabajo a realizar en la clase derivada, es definir estos métodos a fin de que se complete el comportamiento del concepto.

Nota: El uso o no de métodos virtuales tiene implicaciones en los espacios de memoria ocupados por los objetos instanciados. Un método no virtual de una clase base y otro similar (polimórfico) de la clase derivada ocupan ambos espacios de memoria independientes. En un método virtual sólo se tiene el espacio de memoria ocupado por la redefinición en la clase derivada.

Mecanismos básicos de la Orientación a Objetos

✓ Herencia – clases abstractas y métodos virtuales



Mecanismos básicos de la Orientación a Objetos

✓ Herencia – interfaz

- Permite determinar qué criterio de interfaz (público, privado ó protegido) se va a usar para los usuarios de la clase derivada con los elementos públicos de la clase base.

