



Pilares de la Orientación a Objetos

www.profmatiasgarcia.com.ar

Pilares de la Orientación a Objetos

Abstracción

Relaciones

Herencia

Encapsulamiento

Abstracción

- ❖ La Abstracción es la propiedad que permite seleccionar las características esenciales y relevantes de un objeto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real.
- ❖ La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

Abstracción

- ❖ Ignorancia Selectiva
 - La abstracción nos ayuda a trabajar con cosas complejas
 - Se enfoca en lo importante
 - Ignora lo que no es importante (simplifica)
- ❖ Una clase es una abstracción en la que:
 - Se enfatizan las características relevantes
 - Se suprimen otras características
- ❖ Una clase debe capturar una y solo una abstracción clave

Abstracción

- ❖ El proceso de abstracción es completamente subjetivo y dependiente del contexto; Por ejemplo, tomando como objeto de análisis el concepto de automóvil, un observador interesado en la mecánica podría caracterizarlo como un vehículo con cierto tipo de motor, mientras que otro observador, en un contexto diferente, podría caracterizar al mismo automóvil como un medio de transporte, haciendo énfasis en su capacidad de transportar personas o cosas, sin importar el tipo de motor que tenga (simplificación).

Tipos de abstracciones

- ❖ Las nociones de *dividir en partes* y *dividir en especializaciones* representan las dos formas de abstracción más importantes usadas en la programación orientada a objetos. Comúnmente se las conoce como las abstracciones *es-un* y *tiene-un*.
- ❖ La idea de división en partes es la abstracción *tiene-un*. El significado de este término es fácil de comprender; un auto 'tiene-un' motor, 'tiene-una' transmisión, etc.

Tipos de abstracciones

- ❖ Al concepto de especialización nos referimos como la abstracción es-un. Nuevamente, el término surge de las sentencias que pueden ser usadas para ilustrar las relaciones. Utilizando este concepto, una bicicleta 'es-un' vehículo rodado, el cual a su vez 'es-un' medio de transporte..."

Encapsulamiento

- ❖ El Encapsulamiento es la propiedad que permite asegurar que el contenido de la información de un objeto está oculta al mundo exterior.
- ❖ El encapsulamiento sostiene que el único que puede manipular el estado interno de un objeto es el mismo objeto. En otras palabras, un objeto no puede invadir el espacio privado (estado interno) de otro objeto.
- ❖ Permite la modificación de éste sin que se tengan que modificar las aplicaciones que lo utilizan.

Encapsulamiento

- ❖ Principio que establece que los atributos propios de un objeto no deben ser visibles desde otros objetos.
 - Deben ser declarados como privados.
- ❖ Permite abstraer al resto del mundo de la complejidad de la implementación interna.
- ❖ Permite exponer el estado del objeto sólo a través del comportamiento que le hayamos definido mediante miembros públicos.
- ❖ ¿Por qué es útil?
 - Punto de Control/Validación
 - Mejor respuesta ante los Cambios

Relaciones

- ❖ Todo sistema abarca muchas clases y objetos
- ❖ Los objetos contribuyen en el comportamiento de un sistema colaborando entre si
 - La colaboración se logra a través de las relaciones
- ❖ Existen dos tipos principales de relaciones
 - Asociación
 - Agregación

Relaciones de Asociación

- ❖ Una asociación es una conexión entre dos clases que representa una comunicación
 - Una asociación puede tener nombre
 - La comunicación puede ser tanto uni como bi-direccional (por defecto)
 - La multiplicidad es el número de instancias que participan en una asociación
- ❖ Ejemplo:
 - Una Persona es Dueña de uno o varios Vehículos
 - Un Vehículo Pertenece a una Persona



Relaciones de Agregación

- ❖ La agregación es una forma especial de asociación donde un todo se relaciona con sus partes
 - También se conoce como “una parte de” o una relación de contención
- ❖ Ejemplo:
 - Una Puerta es una parte de un Vehículo
 - El Vehículo es azul, la Puerta es Azul
 - Mover el Vehículo implica mover la Puerta

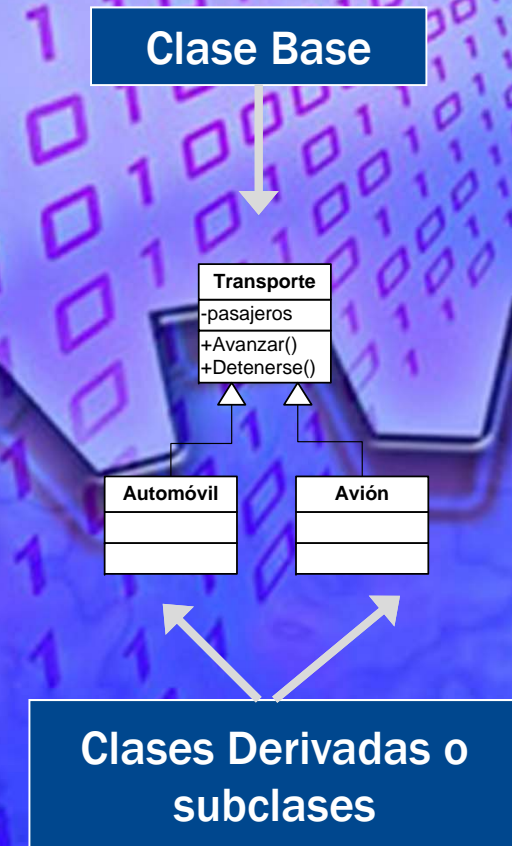


Herencia

- ❖ El propósito principal de la herencia es el de organizar mejor las clases que componen una determinada realidad, y poder agruparlas en función de atributos y comportamientos comunes a la vez que cada una se especializa según sus particularidades.
- ❖ Es la relación entre clases de objetos que permite rehusar los atributos y operaciones definidas en otra clase más general de la cual se hereda o deriva

Herencia

- ❖ Es una relación entre clases en la cual una clase comparte la estructura y comportamiento definido en otra clase (Grady Booch)
- ❖ Cada clase que hereda de otra posee:
 - Los atributos de la clase base además de los propios
 - Soporta todos o algunos de los métodos de la clase base
- ❖ Una subclase hereda de una clase base (superclase).



Herencia

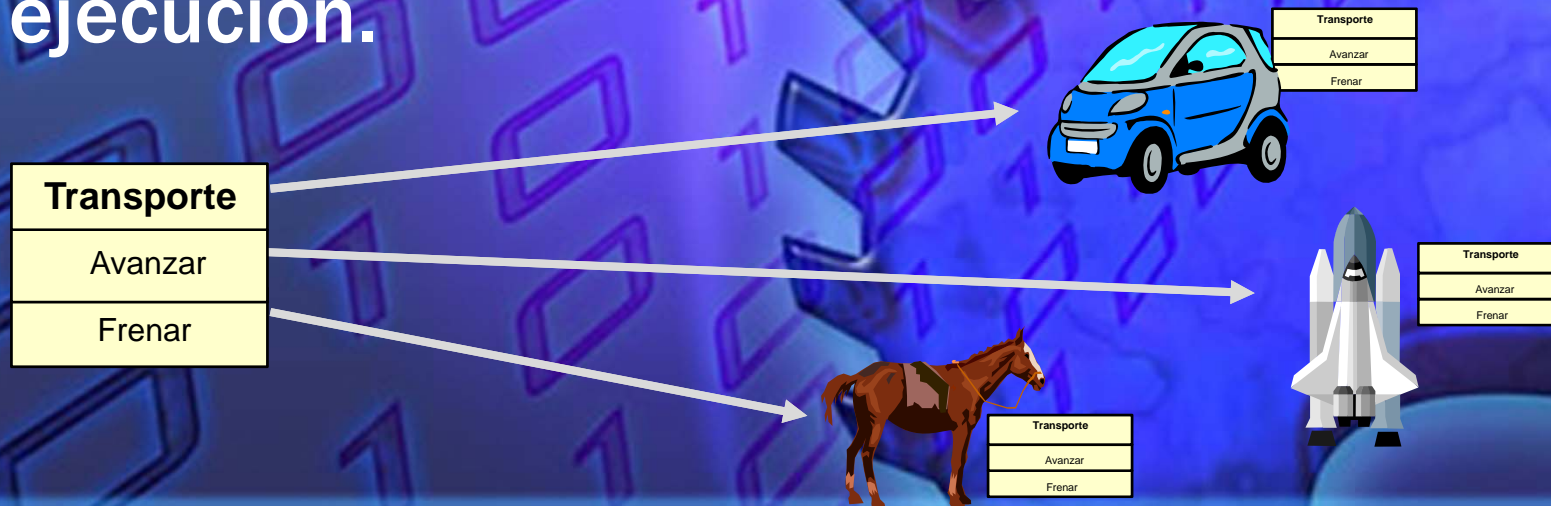
- ❖ Herencia “Es-Un”: herencia real, donde la subclase es un tipo específico de la superclase (generalización).
 - Un Cuadrado es un Rectángulo
 - Un perro es un mamífero
 - Un automóvil es un vehículo a motor
- ❖ hay dos tipos de herencias:
 - Herencia Simple: una clase derivada puede heredar sólo de una clase base.
 - Herencia Múltiple: una clase derivada puede heredar de una o más clases base.

Polimorfismo

- ❖ Es la propiedad que tienen los objetos de permitir invocar genéricamente un comportamiento (método) cuya implementación será delegada al objeto correspondiente recién en tiempo de ejecución.
- ❖ El polimorfismo tiende a existir en las relaciones de herencia, pero no siempre es así.
- ❖ Permite la posibilidad de desencadenar operaciones diferentes en respuesta a un mismo mensaje.

Polimorfismo - Ejemplo

- ❖ La definición del método reside en la clase base.
- ❖ La implementación del método reside en la clase derivada.
- ❖ La invocación es resuelta al momento de ejecución.



Webgrafía y Licencia:

- ❖ Textos tomados, corregidos y modificados de diferentes páginas de Internet, tutoriales y documentos.
- ❖ Este documento se encuentra bajo Licencia Creative Commons 2.5 Argentina (BY-NC-SA), por la cual se permite su exhibición, distribución, copia y posibilita hacer obras derivadas a partir de la misma, siempre y cuando se cite la autoría del *Prof. Matías E. García* y sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.
- ❖ Autor:

Matías E. García

Prof. & Tec. en Informática Aplicada

www.profmatiasgarcia.com.ar

info@profmatiasgarcia.com.ar

 creative
commons



www.profmatiasgarcia.com.ar