

Notación de Clases en UML

Georg Lehner

22 de octubre de 2002

Orientación a Objetos: OO se refiere a un paradigma de análisis, diseño e implementación con un enfoque en *objetos*, integrando *atributos* y *operaciones*.

Eso contrasta con paradigmas centrado en operaciones (análisis funcional) o en atributos (bases de datos).

Aspectos generales de OO son:

Identidad cada objeto es una identidad *distinguible*. Un objeto tiene referencia de si mismo. Dos objetos se distinguen aun cuando muestran atributos identicos.

Clasificación cada objeto es asignado a una clase específica. La clase es una *abstracción* del objeto concreto, indicando los atributos y operaciones comunes de todos los objetos pertenecientes a ella.

Polimorfismo la “misma” operación puede ser implementado diferente en diferentes clases.

Herencia crea una *jerarquía* de sub-clases, o clases derivados y super-clases. sub-clases *comparten* los *atributos y operaciones* de su super-clase(s) y presentan una *especialización* de ella.

Hablamos de una *clase* cuando nos referimos a una clase de objetos, y de un *objeto*, o una *instancia* de una clase, cuando nos referimos a un objeto específico de esta clase de objetos.

Notación:

El diagrama de *clases* presenta un conjunto de clases de un sistema/subsistema y las relaciones entre ellos.

El símbolo de la clase (fig.1) es un *rectángulo* sólido, con el nombre de la clase escrito en ella.

Lavadora

Figura 1: Símbolo de Clase



Figura 2: Clase con detalles

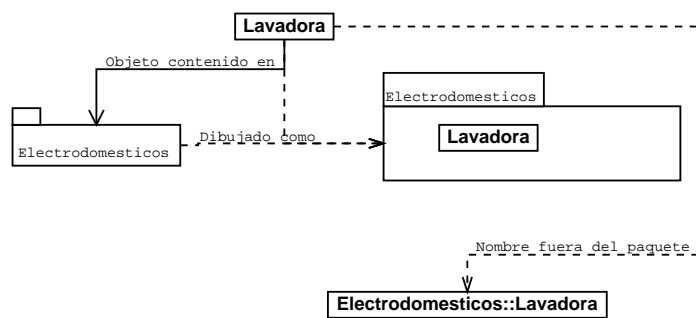
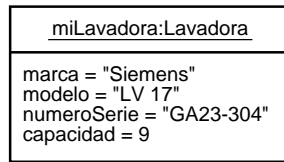


Figura 3: via de acceso a una clase

Opcionalmente pueden detallarse (fig.2) los *atributos* y las *operaciones* en divisiones verticales, divididas por una línea sólida. Si se detallan las operaciones, pero no los atributos, se dibuja una división vacía para ellos.

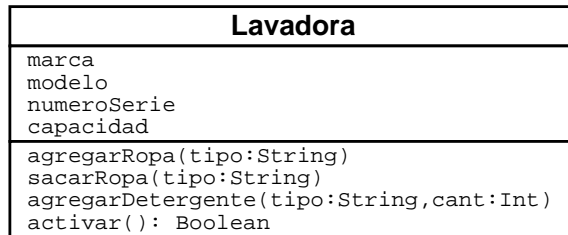
Cuando clases forman partes de paquetes (fig. 3), se puede especificar su pertenencia mediante la notación “classpath” – vía de acceso en diagramas donde no aparece el paquete mismo.



Instancia de la Clase Lavadora
 Instancias anónimas ":Lavadora"

Figura 4: Diagrama de objeto

El diagrama de objetos (Fig. 4) muestra una instancia específica de una clase, en especial cuando algunos atributos tienen valores específicos en el diagrama.



Firma (Signature) de operaciones

Figura 5: Firma de las operaciones en una clase

Se puede agregar detalle a atributos y operaciones, indicando su tipo de datos y en el caso de operaciones los parámetros con sus datos respectivos (fig. 5). Se llama *firma* a la plantilla de una función, que contiene el tipo de datos de retorno, los nombres de sus parámetros y sus tipos de datos respectivos.

Visibilidad

Los atributos y operaciones pueden ser de acceso restringido, en cuanto se refiere a su uso desde operaciones de otras clases:

- + **público, visible** cualquier clase ajena puede acceder al atributo o la operación de un objeto de esta clase. En el diagrama se utiliza el símbolo “+”, antepuesto al nombre del atributo o la operación.
- # **protegido** solamente sub-clases (clases derivadas) tienen acceso a este atributo u operación. El símbolo es “#”.
- **privado** El atributo o la operación solamente son definidos para un mismo objeto de esta clase.

Interfases

La utilidad de una clase consiste en que *realiza* una cierta función en el sistema. Las operaciones (públicas) de una clase son el Interfaz de esta clase.

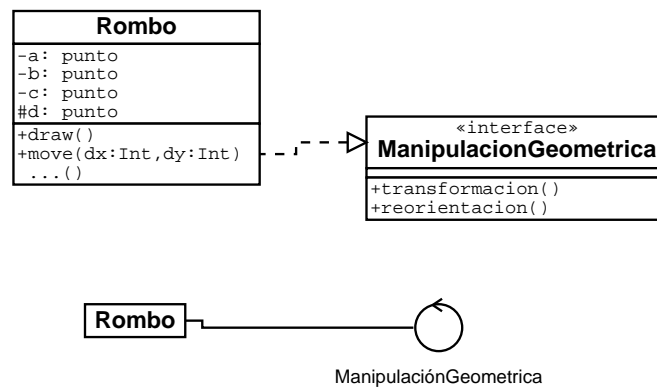


Figura 6: Clase realiza Interfaz

Como generalización, un *conjunto de operaciones* que realizan una función en un sistema se llama *Interfaz*. Más que una clase puede realizar el mismo interfaz.

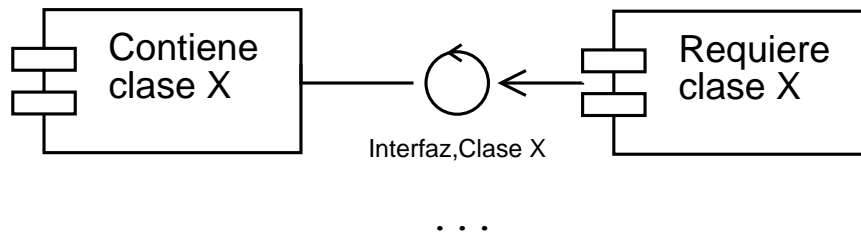


Figura 7: Componente realiza Interfaz

Componentes en un sistema OO son conjuntos de clases, que realizan varias funciones en el sistema. Una componente realiza (a través de sus clases) uno o varios interfaces (fig. 7).

La figura 6 muestra como se *especifica* un interfaz de una clase mediante un *estereotipo*, y como en otros diagramas el interfaz puede ser representado mediante el símbolo de un círculo y el nombre del interfaz.

Estereotipo es una extensión de un símbolo UML a una nueva semántica. Se anota el estereotipo respectivo (=significado) del símbolo con letra cursiva encima del nombre del elemento con doble parentesis aguda: <<*estereotipo*>>

Relaciones entre clases - Calificación

A los conectores presentados en la primera unidad solo se agrega la *realización*, mostrado en figura 6.

Asociación es una línea sólida. Se puede detallar la relación entre clases asociadas mediante



Figura 8: Detalle de una asociación

la descripción del rol de cada clase en las puntas respectivas, el nombre de la relación y una flecha indicando la *dirección*, de cual clase a cual otra se refiere la calificación.

Composición (fig. 9) es una forma especial de Agregación (fig. 10): Los componentes

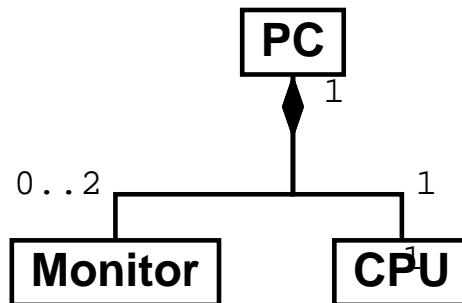


Figura 9: Composición con multiplicidad

por sí mismos no tienen validez.

A través de la *calificación* de relaciones se agrega semántica (sentido, significado):

Multiplicidad una cuantificación en ambos extremos restringe la cantidad de objetos de una clase que pueden ser relacionados con objetos de la otra (ejemplo en fig. 9). Existen las siguientes notaciones:

0 ninguno

1 uno

n un número específico

n..m un intervalo, desde n a m objetos. Uno de los dos límites puede faltar, indicando de “0 a m”, o de “m a infinito” respectivamente.

a,b,c enumeración, o “a” objetos, o “b”, objetos, o “c” objetos. Pueden enumerarse una cantidad arbitraria de números.

***** el asterisco indica el intervalo 0 a infinito.

Restricción una restricción arbitraria de la relación entre dos clases se denota con su

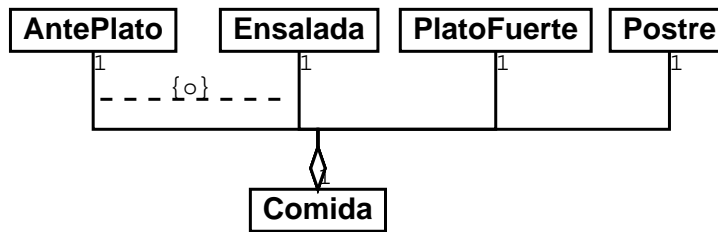


Figura 10: Agregación con restricción

nombre o detalle encerrado en llaves “{...}”. En el ejemplo en figura 10 se detalla, que para una comida se puede escoger entre un Anteplato o una Ensalada (pero no se puede pedir ambos).

Nota: al contrario con la relación “composición”, un Postre por si solo ya conforma una Comida en nuestro ejemplo.