



RELACIÓN ENTRE CLASES I. AGREGACIÓN Y COMPOSICIÓN

Febrero / 2020

PROGRAMACIÓN ORIENTADA A OBJETOS
INGENIERÍA INFORMÁTICA

Objetivos

Identificar una relación de agregación entre clases.

Identificar una relación de composición entre clases.

Identificar la multiplicidad en una relación de clases.

Objetivos

Representar una relación de agregación entre clases en un diagrama de clases UML.

Representar una relación de composición entre clases en un diagrama de clases UML.

Representar la multiplicidad en la relación entre clases en un diagrama de clases UML.

Sumario

- ◇ Relaciones entre clases.
- ◇ Relación de Agregación.
- ◇ Relación de Composición.

Bibliografía

- ◇ *Como programar en Java.*
- ◇ *Aprenda Java como si estuviera en primero.*

Motivación

Existe o no una relación entre las siguientes entidades?

Punto2D
-m_coordX : double -m_coordY : double
+Punto2D() +Punto2D(_x : double, _y : double) +getCoordX() : double +setCoordX(_coordX : double) : void +getCoordY() : double +setCoordY(_coordY : double) : void

Circunferencia
-m_centro : Punto2D -m_radio : double
+Circunferencia() +Circunferencia(_radio : double, _x : double, _y : double) +Circunferencia(_radio : double, _centro : Punto2D) +getCentro() : Punto2D +setCentro(_centro : Punto2D) : void +getRadio() : double +setRadio(_radio : double) : void

Relaciones entre clases

Una clase puede tener referencias a objetos de otras clases como miembros. A dicha capacidad se le conoce como **relación** y algunas veces como **relación “tiene un”** o **relación “tiene muchos”**.

Relaciones entre clases

Una relación entre clases es una relación estructural que describe una conexión entre objetos.

Relaciones entre clases

Aunque las relaciones pueden ser bidireccionales (se pueden recorrer en ambos sentidos), en ocasiones es deseable hacerlas unidireccionales (restringir su navegación en un único sentido).

Relaciones entre clases

Gráficamente, cuando la asociación es unidireccional, la línea termina en una punta de flecha que indica el sentido de la asociación.

Multiplicidad de las relaciones

La multiplicidad de una asociación determina cuántos objetos de cada tipo intervienen en la relación:

El número de instancias de una clase que se relacionan con UNA instancia de la otra clase.

Multiplicidad de las relaciones

Cada asociación tiene dos multiplicidades (una para cada extremo de la relación).

Para especificar la multiplicidad de una asociación hay que indicar la multiplicidad mínima y la multiplicidad máxima (mínima..máxima)

Multiplicidad de las relaciones

Multiplicidad	Significado
1	Uno y sólo uno
0..1	Cero o uno
N..M	Desde N hasta M
*	Cero o varios
0..*	Cero o varios
0..*	Cero o varios
1..*	Uno o varios (al menos uno)

Multiplicidad de las relaciones

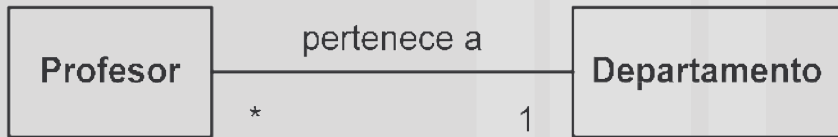
- ✓ Cuando la multiplicidad mínima es 0, la relación es opcional. (**Agregación**)
- ✓ Una multiplicidad mínima mayor o igual que 1 establece una relación obligatoria (**Composición**).

Multiplicidad de las relaciones



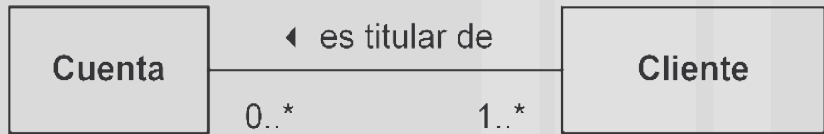
Todo departamento tiene un director.
Un profesor puede dirigir un departamento.

Multiplicidad de las relaciones



Todo profesor pertenece a un departamento.
A un departamento pueden pertenecer varios profesores.

Multiplicidad de las relaciones



Relación opcional

Un cliente puede o no ser titular de una cuenta

Relación obligatoria

Una cuenta ha de tener un titular como mínimo

Agregación y Composición

Casos particulares de relaciones: Relación entre un todo y sus partes

Gráficamente, se muestran como asociaciones con un rombo en uno de los extremos.

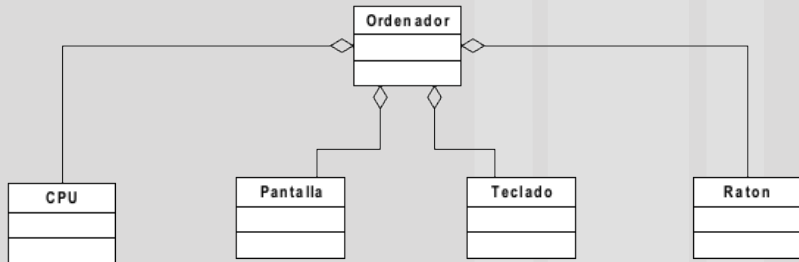
Agregación

Las relaciones de agregación se basan en la idea de observar o entender un objeto como una composición de otros objetos. Desde nuestro punto de vista, las relaciones de agregación se entenderán como relaciones en las cuales una serie de clases aparecen como tipos de los atributos de otra clase.

Agregación

La mejor forma de identificar si nos encontramos ante una relación de agregación es preguntarnos si la clase que queremos definir “tiene un” (en inglés, “has - a”) atributo de la otra clase que estemos usando (de ahí que en ciertas referencias se definan como relaciones “has - a”).

Agregación



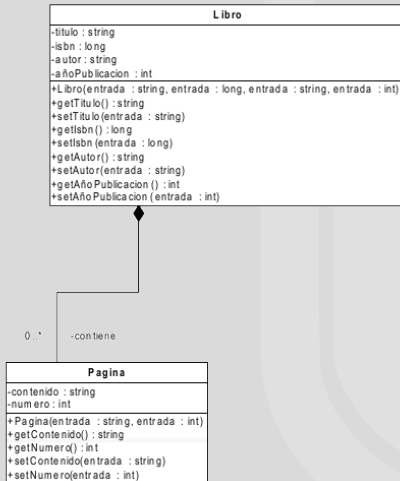
Composición

Basado en esta idea, existe un tipo particular de agregación, llamada **composición (o también agregación fuerte)**, en la cual los objetos agregados no tienen sentido fuera del objeto resultante.

Composición

También se puede entender la composición como una relación en la que, los objetos siendo agregados, deben dejar de existir cuando lo hace el objeto compuesto.

Composición



Debate

Entonces agregación o composición?

Punto2D
-m_coordX : double -m_coordY : double
+Punto2D() +Punto2D(_x : double, _y : double) +getCoordX() : double +setCoordX(_coordX : double) : void +getCoordY() : double +setCoordY(_coordY : double) : void

Circunferencia
-m_centro : Punto2D -m_radio : double
+Circunferencia() +Circunferencia(_radio : double, _x : double, _y : double) +Circunferencia(_radio : double, _centro : Punto2D) +getCentro() : Punto2D +setCentro(_centro : Punto2D) : void +getRadio() : double +setRadio(_radio : double) : void

Conclusiones

- ◇ Algunas relaciones pueden ser consideradas agregaciones o composiciones, en función del contexto en que se utilicen.

Conclusiones

- ◇ Mucho se ha discutido a cerca de las agregaciones y las composiciones, el debate es casi tan caliente como el de los include y extends de los casos de uso. Ya que algunos sostienen que los lenguajes orientados a objetos, tienen garbage collector, por lo que no necesitan métodos de destrucción de los objetos (relacionados a los ciclos de vida en la composición).

Conclusiones

- ◇ La programación es la misma para las composiciones y las agregaciones, y que la diferencia es meramente conceptual entre una y otras.

UNIVERSIDAD DE MATANZAS

cosechando el saber

FIN