

POO 2

Juan Carlos Figueroa Duran

Otoño-2014

Universidad del Bío-Bío
Departamento de Ciencias de la Computación y Tecnología de la
Información

1 Interpretación de Diagramas de Clase

Una Clase es una agrupación de objetos de un mismo tipo, que tienen un conjunto de características y/o un comportamiento común. Una clase es una representación abstracta de algo.

Asignatura	Persona
Rectángulo	Libro
Producto	Electrodoméstico
Medicamento	Estrella
Automóvil	Préstamo (en una biblioteca, p.e.)

Las características de una clase corresponden a los atributos comunes que poseen todos los objetos que pertenecen a ella. Estos atributos asumen valores dentro de un conjunto dependiendo de la naturaleza de los mismos (por ejemplo: numérico, string, etc.):

- Una clase Rectángulo podría tener los atributos: largo, alto.
- Una clase Producto podría tener los atributos: código, nombre, stock actual, stock mínimo.
- Una clase Libro podría tener los atributos: nombre, número de páginas y género.

El comportamiento de una clase son las operaciones que la clase tiene disponible para otras clases que interactúan con ella. Estas operaciones se llaman métodos y, en general, utilizan los atributos de la clase.

- La clase Rectángulo podría tener las operaciones: calcula área, calcula perímetro, obtiene largo, cambia tamaño.
- La clase Producto podría tener las siguientes operaciones: obtiene nombre, aumenta stock, disminuye stock.
- La clase Libro que podría tener las siguientes operaciones: obtiene nombre, cambia género.

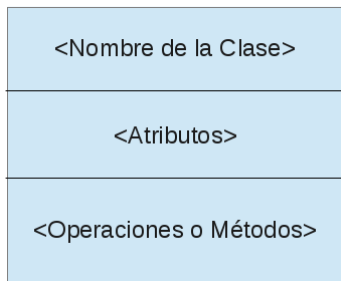
Unified Modeling Language (UML):

- Es un lenguaje gráfico estándar de modelado de sistemas de software.
- Mediante su uso es posible visualizar, especificar, construir y documentar un sistema.
- UML permite describir, por medio de diferentes tipos de diagramas, diversos aspectos de un sistema incluyendo aquellos conceptuales como procesos de negocio y aspectos concretos como esquemas de bases de datos.
- Nosotros conoceremos un tipo de diagrama en particular llamado “Diagrama de Clases”.

Diagrama de clases:

- Diagrama que describe la estructura de un sistema de software.
- Describe las clases (con sus atributos y métodos) y relaciones que existen entre ellas.
- Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas.

Representación UML de una Clase:



Mediante un rectángulo dividido en tres secciones:

<Nombre de la Clase>: Contiene el nombre de la Clase

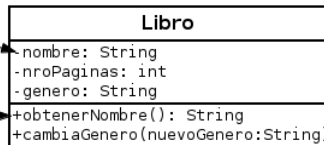
<Atributos>: Contiene los atributos (o variables de instancia) que caracterizan a la Clase y su accesibilidad o visibilidad (private, protected o public).

<Operaciones o Métodos>: Contiene los métodos u operaciones y su visibilidad o accesibilidad. Estos son la forma como interactúa el objeto con su entorno dependiendo de la visibilidad (private, protected o public).

Ejemplo:

Atributo que contiene el nombre
del Libro

Método que permite obtener el
nombre del Libro



Descripción de los Atributos de una Clase:

< controldeacceso > < nombreatributo > : < tipodedato >

- -nombre: String

Visibilidad de los Atributos de una Clase:

public (+): Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.

private (-): Indica que el atributo sólo será accesible desde dentro de la clase. Es decir, sólo los métodos de la clase lo pueden usar (accesar).

protected (#): Indica que el atributo no será accesible desde fuera de la clase, excepto por aquellas clases que son herederas de esta clase (ver relación de herencia).

Descripción de los Métodos de una Clase:

`< controldeacceso >< nombremtodo > ([< listadeparametros >]) :< tipodedatoretornado >`

- `+obtieneNombre(): String`

Visibilidad de los Atributos de una Clase:

public (+): Indica que el método será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.

private (-): Indica que el método sólo será accesible desde dentro de la clase. Es decir, sólo los métodos de la clase lo pueden usar (Invocar).

protected (#): Indica que el método no será accesible desde fuera de la clase, excepto por aquellas clases que son herederas de esta clase (ver relación de herencia).

Método constructor de una Clase:

- Generalmente una clase incluye al menos un método llamado **“constructor”**.
- Este método permite la creación de objetos de dicha clase.
- Un constructor posee las siguientes características:
 - Es un método que lleva el mismo nombre de la clase.
 - Se invoca una sola vez, cuando se crea un nuevo objeto
 - Puede usarse para inicializar los atributos de un objeto. Sin embargo, puede no contener instrucción alguna, en este caso, su propósito es permitir la obtención de memoria para las variables de instancia.
 - Puede incluir parámetros
 - Una clase puede incluir varios constructores con distintas definiciones de parámetros formales.
- Un constructor siempre debe ser definido como público, de manera que sea posible crear objetos de dicha clase desde cualquiera otra clase.

Ejemplo:

Libro
-nombre: String -nroPaginas: int -genero: String
+<<constructor>> Libro() +<<constructor>> Libro(nombre:String,numeroDePagina:int,genero:String) +obtenerNombre(): String +cambiaGenero(nuevoGenero:String)

Interpretemos la siguiente clase:

Fecha
-dia: int -mes: int -año: int
+asignar(dia:int,mes:int,año:int) +obtieneDia(): int +obtieneMes(): int +obtieneAño(): int +toString(): String +igual(): boolean +mayor(): boolean

Una aplicación en ejecución (más aún un sistema de software conformado por múltiples aplicaciones) se puede describir como un conjunto de objetos colaborando para desarrollar una tarea dada.

Dichas interacciones se producen debido a las relaciones que existen entre tales objetos.

Las relaciones entre objetos son: **Asociación**, **Agregación/composición** y **Herencia**.

Asociación:

- Se produce entre dos o más objetos.
- La más simple es una asociación entre dos objetos, denominada asociación binaria.
- Una asociación es la más general de las relaciones y se produce cuando un objeto interactúa con otro debido a una transacción propia del ámbito del problema, por lo tanto, el nombre de la asociación será el verbo que represente tal transacción.
- La multiplicidad indica el número de objetos que pueden estar involucrados en la relación.

Multiplicidad:

1 Un elemento relacionado.

0..1 Uno o ningún elemento relacionado.

0..* Ninguno o varios elementos relacionados.

1..* Varios elementos relacionados pero al menos uno.

* Varios elementos relacionados.

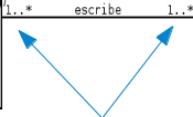
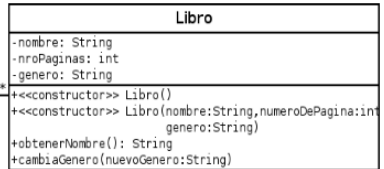
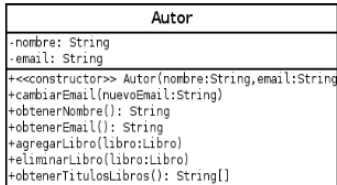
M..N Entre M y N elementos relacionados.

Asociación:

Nombre de la asociación

Se lee: Un libro es (o fue)
escrito por uno o más autores

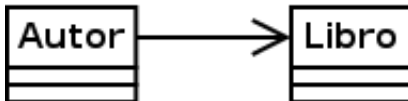
Se lee: Un autor escribe (o ha
escrito) uno o más libros



Multiplicidad

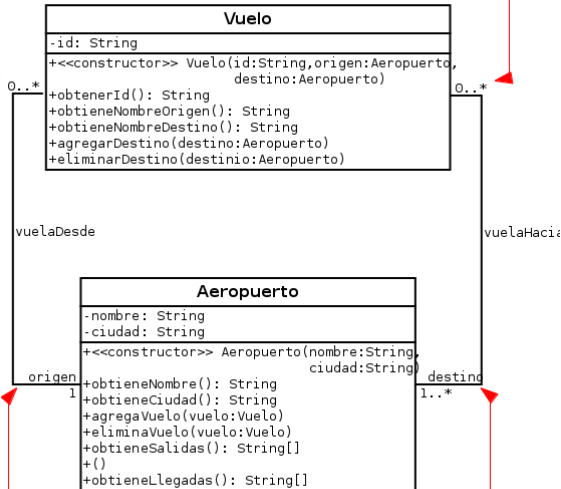
Direccionalidad:

- En principio, se asume que una asociación es bidireccional, es decir que se puede navegar desde cualquiera de clases implicadas a la otra,
- Pero también es posible indicar que la navegación ocurrirá en una sola dirección



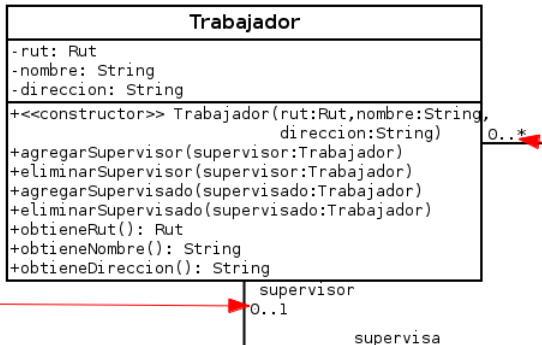
A un aeropuerto llega ninguno
(porque está inhabilitado) o
mucho vuelos

¿Como se interpreta esto?



Un vuelo sale desde un único
primer aeropuerto. Origen es
el nombre del rol.

Un vuelo tiene al menos
un destino, pero puede
tener destinos intermedios



Un trabajador
es supervisado
por ninguno o
por un
trabajador

Un trabajador
supervisa a
ninguno o
varios
trabajadores

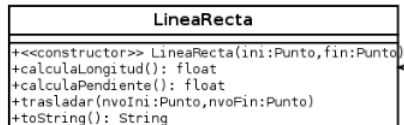
Agregación:

- Es una relación que permite indicar si un objeto (el todo) está conformado por otros objetos (sus partes).
- Por lo general, esta relación no lleva nombre en un diagrama.
- Los objetos que son parte de otros tienen un ciclo de vida independiente del ciclo de vida del o los objetos en que aparecen agregados.

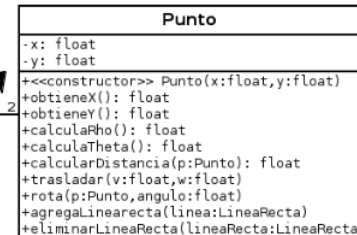
Clase que tiene una relación de agregación

Un objeto de la clase Punto puede no pertenecer a linea recta alguna

Un objeto de la clase LineaRecta se forma a partir de dos puntos



0..*



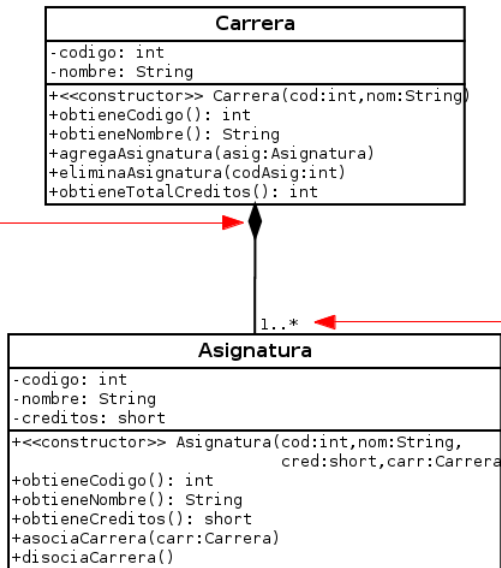
2

Composición:

- Una composición es una relación semejante a la agregación , pero más estricta.

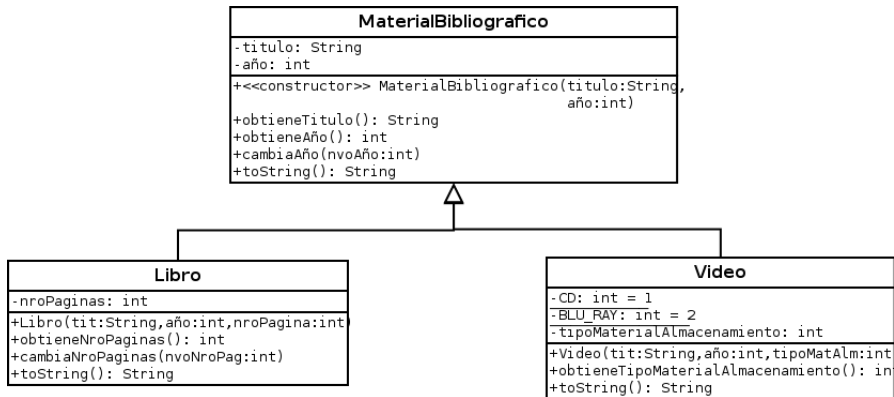
Indica que es la clase Carrera la que se compone de Asignaturas. En este caso, una Asignatura se relaciona con una única Carrera.

Una carrera se compone de una o más Asignaturas.



Herencia o Generalización:

- La herencia relaciona dos objetos indicando que uno se deriva o es una extensión del otro.
- De tal forma que uno se puede entender como una generalización y el otro como una especialización de un concepto común.
- El verbo asociado a esta relación es: “es un” o “es una”.



Herencia o Generalización:

- Nombres usados para las clases en la relación:
 - clase padre - clase hija.
 - superclase - superclase
 - clase base - clase derivada
- Las subclasses heredan propiedades de sus clases padre, es decir, atributos y operaciones (y asociaciones) de la clase padre están disponibles en sus clases hijas.
- La especialización es una técnica muy eficaz para la extensión y reutilización.

