UML. DIAGRAMA DE CLASES

CODE UD7 – Tema 1

IES Plurilingüe Antón Losada Diéguez

Adrián Fernández González

Tabla de contenido

UML. Diagrama de clases

1. Introducción

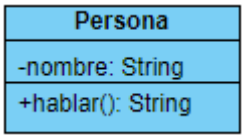
Los diagramas de clase son el tipo más utilizado de UML, ya que representan las clases, el bloque básico de todo sistema basado en orientación a objetos.

Las clases se representan en modo de cajas con su nombre, atributos y métodos y se relacionan entre sí mediante una serie de flechas.

Su representación es muy similar a la de un Modelo Entidad-Relación de una base de datos.

2. Caja de clase

Como se ha mencionado, una clase está representada por una caja dividida en tres bloques, el nombre de la clase, luego los atributos que contiene y por último los métodos.



Los atributos y los métodos se definen de igual forma, el accesor y el nombre y, en caso de los métodos, los () con los nombres de los parámetros.

De forma opcional, puede incluirse el tipo de los atributos y el tipo que devuelven los métodos si el lenguaje así lo requiere o permite.

2.1. Accesores

Los accesores indican el modo de acceso o visibilidad de atributos y métodos y se representa mediante un símbolo antes del nombre del mismo:

- privated

~ package (default)

protected

+ public

2.2. Interfaz

Las clases interfaz pueden representarse de dos modos, o bien poniendo <<interface>> sobre el nombre de la clase o rodeando el nombre con << >>.

3. Relaciones

Las clases se relacionan entre sí de seis formas, asociación, herencia, implementación, dependencia, agregación y composición.

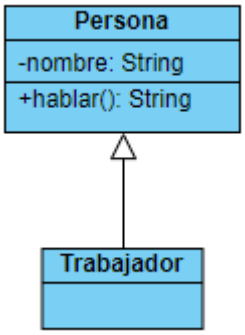
3.1. Asociación

La asociación es una relación básica, genérica, sin indicar el tipo en concreto. Se representa mediante una línea simple o una terminada en una flecha abierta '>'.

En este ejemplo, la clase Persona y la clase Trabajo están relacionadas de algún modo.

3.2. Herencia

La herencia es una relación de extensión, una clase hereda de la otra. Se representa mediante una flecha con punta blanca que apunta a la clase padre.

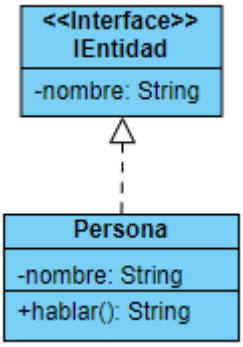


En este ejemplo, trabajador hereda de persona.

3.3. Implementación

La relación que implica la implementación de una interfaz. Se representa mediante una flecha discontinua con punta blanca apuntando a la interfaz que implementa.

También se denomina **dependencia** de tipo ***realization*** o ***realización***.



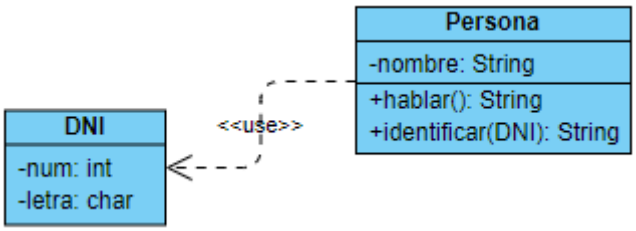
En este ejemplo, la clase Persona implementa la interfaz IEntidad.

3.4. Dependencia

Esta relación indica que existe una dependencia de una clase con otra. Una clase necesita de otra para su correcto funcionamiento.

Aunque hay varios tipos de dependencia, habitualmente es de uso. Este uso suele ser que una clase requiere de otra como parámetro en uno de sus métodos.

Se representa mediante una flecha discontinua con punta abierta '>'.



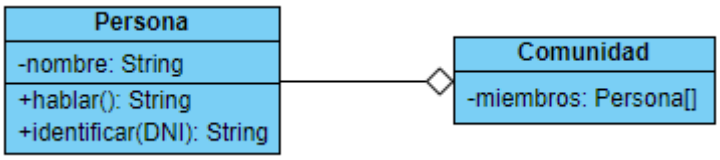
En este ejemplo, la clase Persona usa un DNI para identificarse.

Como añadido, el tipo de dependencia puede ser indicado mediante texto entre << >> sobre la propia línea.

3.5. Agregación

Esta relación indica que una clase forma parte de otra, por ejemplo, como atributo, pero que la agregada es independiente y funciona por sí misma.

Se representa mediante una flecha con un rombo blanco apuntando a la clase que agrega.

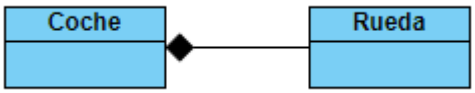


En este ejemplo, una comunidad está formada de personas, pero la persona es independiente y no tiene por qué pertenecer a una comunidad.

3.6. Composición

En una composición, una clase forma parte de otra de forma total, no puede existir sin la otra. Esto ocurre cuando una clase es parte de otra, si la completa desaparece, ella también.

Se representa mediante una flecha con un rombo negro apuntando a la clase que compone.



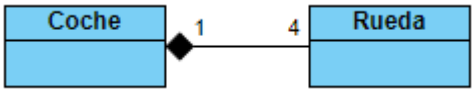
En este caso, el coche se compone de ruedas y estas no existen sin la existencia de la clase Coche.

3.7. Multiplicidad

Las relaciones pueden estar marcadas con una multiplicidad, indicando el número de elementos que intervienen en la relación. Esto es similar a la cardinalidad del Modelo Entidad-Relación.

Los más habituales son:

- n Un número en concreto.
- 0,1 O uno o ninguno.
- n..m Un rango determinado.
- n..* De un número a infinito.



En este ejemplo, se ve que un coche se compone de 4 ruedas y cada rueda solo está en un coche.

4. Abreviación

Para abreviar o como método de descripción general, las clases pueden ser definidas solo con su nombre, dejando para futuras fases del desarrollo el definir sus atributos y métodos.

También es habitual obviar los *getters* y *setters* a no ser que pueda ocasionar confusión si existen atributos que los tienen y otros que no.