INSTITUTO KHIPU

Semestre III

Sesión 11

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

ANÁLISIS Y DISEÑO DE SISTEMAS

Tema:

DIAGRAMA DE CLASES - RELACIONES ENTRE CLASES: HERENCIA.

DIAGRAMA DE CLASES - RELACIONES ENTRE CLASES: HERENCIA.

Introducción

En la entrada anterior se introdujo el **concepto de interfaz** asociado a los **Diagramas de Clases** como herramientas de documentación de la estructura estática de una aplicación informática según los principios de **UML**. En esta entrada se introducirá la el **concepto de herencia** en UML.

Si el paso de la **Programación Estructurada** a la **Programación Orientada a Objetos** supuso dotar al programador de mayor potencia y control sobre su código, la evolución de la **Programación Modular** en la **Herencia Orientada a Objetos** ha supuesto un mecanismo muy potente para aumentar la productividad evitando las continuas reinvenciones de la rueda.

Contexto

Técnicamente hablando la **herencia** en el contexto de la programación orientada a objetos tiene muchas cualidades, pero siendo pragmáticos la herencia supone un **mecanismo muy eficaz para reducir y simplificar el código**.

Para que demuestre todo su potencial hay que partir de una **concepción del sistema como composición/colaboración/interacción de entidades**, clases o interfaces, huyendo de la concepción puramente funcional.

Identificación

Partiendo de las **especificaciones** hay que identificar las **entidades del sistema** y, acto seguido, averiguar las **relaciones** que hay entre ellas analizando cuales están vinculadas entre sí, de una forma o otra.

Una vez **identificadas las relaciones** existentes entre las entidades de un sistema hay que averiguar cuales de ellas pueden ser **modelizadas en forma de herencia**.

Para **detectar una relación de herencia** hay que examinar las entidades involucradas en cada una de las diferentes relaciones binarias que existan en el sistema y, para cada relación, averiguar si existe una **derivación semántica entre ambas entidades**.

Dicho de otro modo, para detectar una relación de herencia entre dos entidades hay que averiguar si una de ellas es una **consecuencia o evolución** de la otra.

En ese caso es muy probable que esa relación se pueda modelizar en forma de herencia.

Ventajas

Y … ¿En qué beneficia la detección de una relación de herencia entre dos entidades de un sistema?.

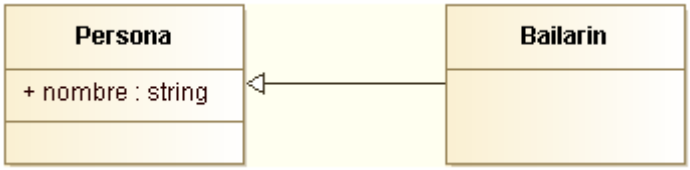
Pues, en primer lugar permite expresar la **relación natural** existente entre ellas de forma correcta. Si en general, expresar las cosas de forma incorrecta genera muchos problemas, en los diagramas de clases más.

Y en segundo lugar supone que no hay que duplicar los recursos existentes en la entidad base dentro de la entidad especializada, lo que ayuda a **simplificar el diagrama** y a **clarificar su significado**.

Clases

Es posible encontrar una **relación de herencia entre dos clases**. En este caso la **clase derivada** posee todos los recursos de la **clase matriz** a **excepción** de aquellos recursos marcados con **visibilidad privada**.

En el Diagrama de Clases siguiente se muestra un ejemplo de **relación de herencia** entre dos clases: **Persona** y **Bailarin**.



Como se puede observar la **linea** que vincula ambas entidades es **dirigida**, de **trazo continuo** y acaba en una **punta de flecha cerrada**. **Empieza en la entidad derivada y temina en la entidad base**.

En la sintaxis UML el diagrama anterior expresa que la clase **Persona** es la **generalización** de la clase **Bailarin,** lo cual, puesto por pasiva, también significa que la clase **Bailarin** es la **especialización** de la clase **Persona**.

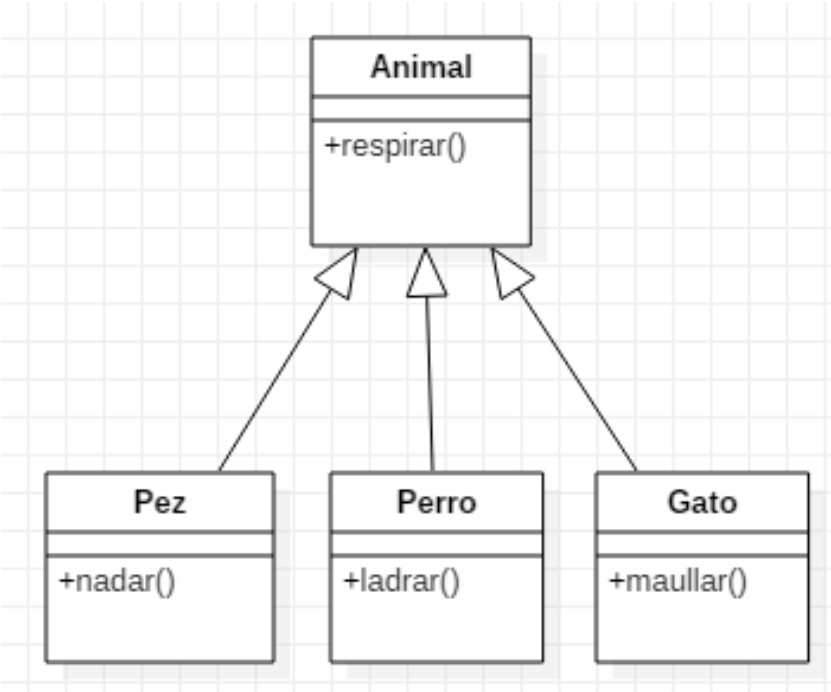
En el ejemplo, la clase **Persona** dispone de un atributo llamado **nombre** de tipo **String** que es **público**. Como consecuencia de la herencia la clase **Bailarin** también tiene un atributo llamado **nombre** de tipo **String** que hereda de la clase **Persona**. Como se puede observar **los recursos heredados no se representan**.

El diagrama anterior se puede **codificar en Java** dando lugar a un código fuente parecido al siguiente.

Herencia

Otra relación muy común en el diagrama de clases es la herencia. Este tipo de relaciones permiten que una clase (clase hija o subclase) reciba los atributos y métodos de otra clase (clase padre o superclase)**.** Estos atributos y métodos recibidos se suman a los que la clase tiene por sí misma. Se utiliza en relaciones «es un».

Un ejemplo de esta relación podría ser la siguiente: Un pez, un perro y un gato son animales.



Ejemplo de herencia

En este ejemplo, las tres clases (Pez, Perro, Gato) podrán utilizar la función respirar, ya que lo heredan de la clase animal, pero solamente la clase Pez podrá nadar, la clase Perro ladrar y la clase Gato maullar. La clase Animal podría plantearse ser definida abstracta, aunque no es necesario.

Ejemplo:

Un profesor emérito puede ser un profesor como también un conferenciante

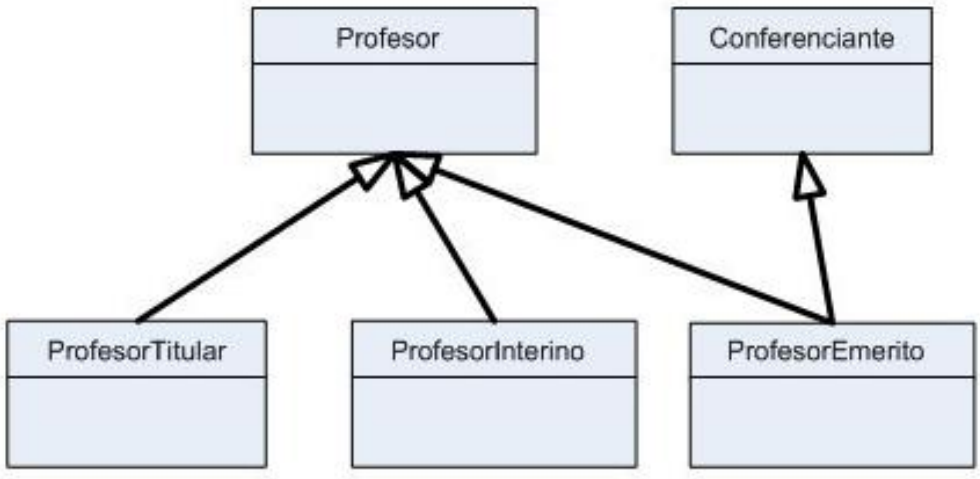
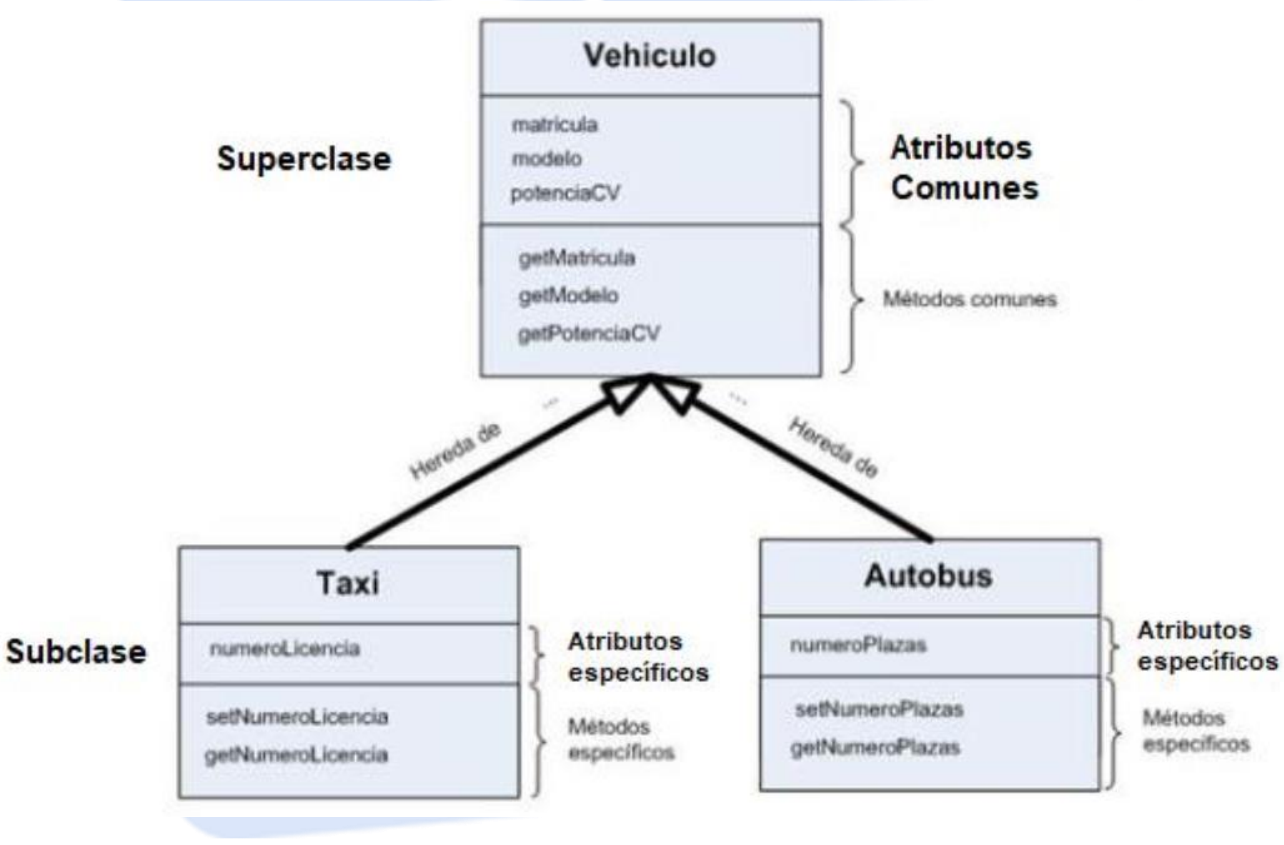


Figura 7: Ejemplo de herencia múltiple

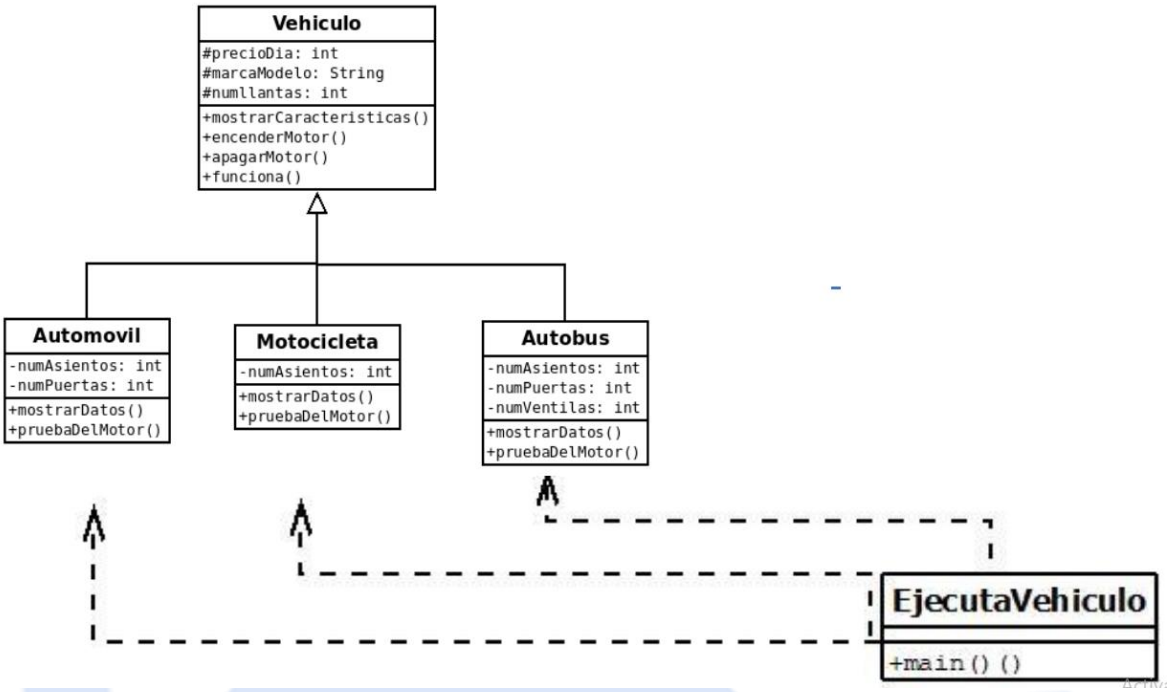
La herencia es una relación entre dos clases donde una denominada clase base o general (padre) y la otra derivada o subclase (hija) La herencia se conoce como generalización o especialización incluso se observa una jerarquía entre las clases, en ejemplo:

La clase Vehículo es la superclase (clase padre) y clase taxi (clase hija o derivada) es una subclase. Así tambiénla clase Autobús (clase hija o derivada) es una subclase



Programa de Herencia de Vehículos y tipos de vehículos

a) Realiza el modelado de las clases y agrega otro tipo de vehículo al diagrama



FUENTE:

- https://diagramasuml.com/diagrama-de-clases/
- https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-clases/
- https://ecosistema.buap.mx/forms/files/dspace-23/14_herencia.html
- https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-clases/
- http://sel.unsl.edu.ar/licenciatura/ingsoft2/UML-DiagramaClaseObjeto.pdf
- BURCH, John; GRUDNISKY, Gary. "Diseño de Sistemas de Información", Grupo Noriega editores.
- SENN, James A. "Análisis y diseño de sistemas de información", 2da. ed., McGraw-Hill.


INSTITUTO
KHIPU