EL ORIGEN DE UML

UML está basado en la orientación de objetos, sistema que vio la luz mucho antes que el UML en el campo de los lenguajes de programación.

En un lenguaje de programación la descripción de los objetos se realiza de manera formal utilizando una sintaxis rigurosa. Dicha sintaxis resulta ilegible para los no programadores y difícil de descifrar para los programadores. A diferencia de las máquinas, los humanos prefieren utilizar lenguajes gráficos para representar abstracciones, ya que dominan este tipo de lenguaje con mayor facilidad y obtienen una visión de conjunto de los sistemas en mucho menos tiempo.

La versión 1.0 de UML se publica en 1997. La OMG (Object Management Group) adopta la notación UML en noviembre de 1997 en su versión 1.1 y crea un Task Force encargada de la evolución del UML. En marzo de 2003, la versión 1.5 incorpora la posibilidad de describir acciones gracias a una extensión de UML llamada *Action Semantics* o semántica de acciones.

La versión 2.0 se publicó en julio de 2005. Constituye la primera evolución importante desde la aparición del UML en 1997. A ella se han ido añadiendo numerosos diagramas y los ya existentes se han enriquecido con nuevas construcciones. Desde julio de 2005 esta versión 2.0 ha sido mejorada. En febrero de 2009, la versión 2.2 se publicó incluyendo la taxonomía oficial de los perfiles UML. La última versión es la 2.5, publicada en junio de 2015, que incorpora principalmente la posibilidad de presentar los atributos y métodos heredados de una clase.

CONCEPTOS DE LA ORIENTACIÓN A OBJETOS

1. EL OBJETO

Un objeto es una entidad identificable del mundo real. Puede tener una existencia física (un caballo, un libro) o no tenerla (un texto de ley). *Identificable* significa que el objeto se puede designar.

Ejemplo

Mi yegua Jorgelina
Mi libro sobre UML
El artículo 293B del código de impuestos

En UML todo objeto posee un conjunto de atributos (estructura) y un conjunto de métodos (comportamiento). Un atributo es una variable destinada a recibir un valor. Un método es un conjunto de instrucciones que toman unos valores de entrada y modifican los valores de los atributos o producen un resultado.

Todo sistema concebido en UML está compuesto por objetos que interactúan entre sí y realizan operaciones propias de su comportamiento.

Ejemplo

Una manada de caballos es un sistema de objetos que interactúa entre sí, cada objeto posee su propio comportamiento.

2. LA ABSTRACCIÓN

La abstracción es un principio muy importante en modelado. Consiste en tener en cuenta únicamente las propiedades pertinentes de un objeto para un problema concreto. Los objetos utilizados en UML son abstracciones del mundo real.

Ejemplo

Si nos interesamos por los caballos en su actividad de carrera, las propiedades aptitud, velocidad, edad, equilibrio mental y casta de origen son pertinentes para dicha actividad y se tienen en cuenta.

Si nos interesamos por los caballos en su actividad de bestia de tiro, las propiedades edad, tamaño, fuerza y corpulencia son pertinentes para dicha actividad y se tienen en cuenta.

3. CLASES DE OBJETOS

Un conjunto de objetos similares, es decir, con la misma estructura y comportamiento, y constituidos por los mismos atributos y métodos, forma una clase de objetos. La estructura y el comportamiento pueden entonces definirse en común en el ámbito de la clase.

Todos los objetos de una clase, llamada también instancia de clase, se distinguen por tener una identidad propia y sus atributos les confieren valores específicos.

Ejemplo: El conjunto de caballos constituye la clase CABALLO, que posee la estructura y el comportamiento descritos a continuación…

Caballo (clase)
nombre edad tamaño peso (estructura compuesta de atributos)
correr() (comportamiento compuesto de métodos)

El caballo Jorgelina es una instancia de la clase CABALLO cuyos atributos y valores se ilustran a continuación…

Jorgelina: Caballo
nombre: Jorgelina edad=8 tamaño=1,72 peso=550

4. ENCAPSULACIÓN

La encapsulación consiste en ocultar los atributos y métodos del objeto a otros objetos. En efecto, algunos atributos y métodos tienen como único objetivo tratamientos internos del objeto y no deben estar expuestos a los objetos exteriores. Una vez encapsulados, pasan a denominarse atributos y métodos privados del objeto.

Ejemplo

Al correr, un caballo realiza diferentes movimientos como levantar las patas, levantar la cabeza o levantar la cola. Esos movimientos son internos al funcionamiento del animal y no tienen por qué ser conocidos en el exterior. Son métodos privados. Las operaciones acceden a una parte interna del caballo: sus músculos, su cerebro y su vista. La parte interna se representa en forma de atributos privados.

5. ESPECIALIZACIÓN Y GENERALIZACIÓN

Una clase de objetos puede introducirse de manera independiente a las demás clases o bien definirse como un subconjunto de alguna otra clase, este subconjunto debería constituir siempre un conjunto de objetos similares.

Hablamos entonces de subclases de otras clases que, por tanto, constituyen especializaciones de esas otras clases.

Ejemplo
La clase de los caballos es una subclase de la clase de los mamíferos.

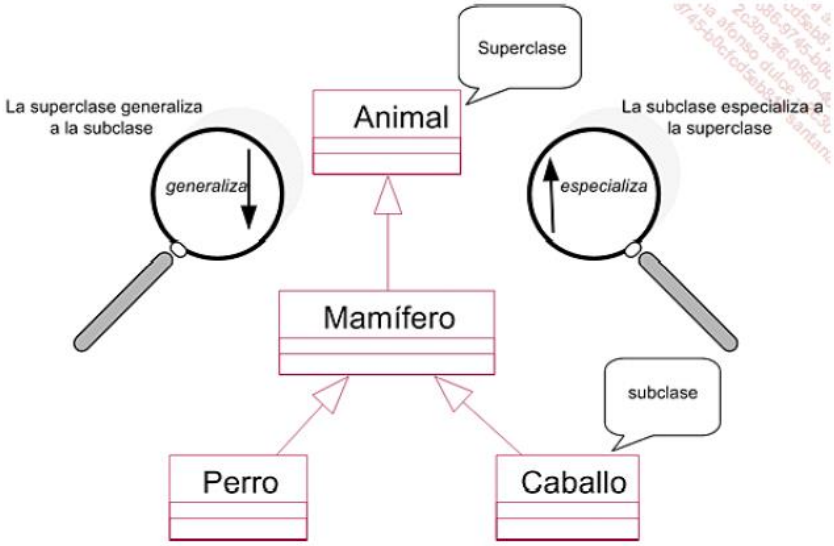
La generalización es la relación inversa a la especialización. Si una clase es una especialización de otra clase, ésta última es una generalización de la primera. Es su superclase.

Ejemplo
La clase de los mamíferos es una superclase de la clase de los caballos.

La relación de especialización puede aplicarse a varios niveles, dando lugar a la jerarquía de clases.

Ejemplo

La clase de los caballos es una subclase de la clase de los mamíferos, ella misma subclase de la clase de los animales. La clase de los perros es otra subclase de la clase de los mamíferos. La jerarquía de clases correspondiente aparece representada en la siguiente figura

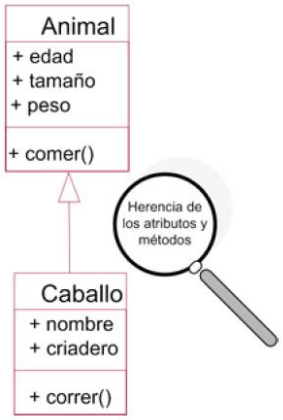


6. HERENCIA

La herencia es la propiedad que hace que una subclase se beneficie de la estructura y el comportamiento de su superclase. La herencia deriva del hecho de que las subclases son subconjuntos de las superclases. Sus instancias son asimismo instancias de la superclase y, por consiguiente, además de la estructura y el comportamiento introducidos en la subclase, se benefician también de la estructura y comportamiento definidos por la superclase.

Ejemplo

Tomamos un sistema en el que la clase Caballo es una subclase directa de la clase Animal. El caballo se describe entonces mediante la combinación de la estructura y del comportamiento derivados de las clases Caballo y Animal, es decir, mediante los atributos edad, tamaño, peso, nombre y casta así como los métodos comer y correr.



La herencia es una consecuencia de la especialización. Sin embargo, los informáticos emplean mucho más a menudo el término *hereda* que *especializa* para designar la relación entre una subclase y su superclase.

7. CLASES ABSTRACTAS Y CONCRETAS

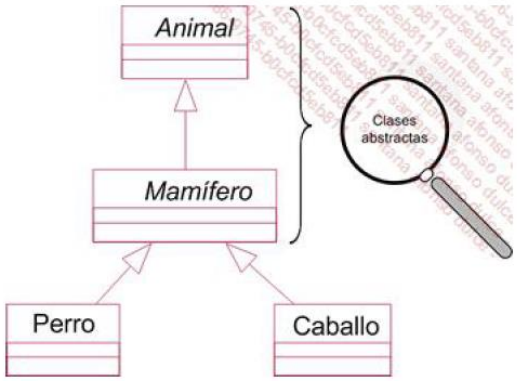
Si examinamos la jerarquía presentada en la figura anterior vemos que en ella existen dos tipos de clases:

- Las clases que poseen instancias, es decir, las clases Caballo y Perro, llamadas clases concretas.
- Las clases que no poseen directamente instancias, como la clase Animal. En efecto, si bien en el mundo real existen caballos, perros, etc., el concepto de animal propiamente dicho continúa siendo abstracto. No basta para definir completamente un animal. La clase Animal se llama clase abstracta.

La finalidad de las clases abstractas es poseer subclases concretas y sirven para factorizar atributos y métodos comunes a las subclases.

Ejemplo

La siguiente figura retoma la jerarquía detallando las clases abstractas y las clases concretas. En UML, el nombre de las clases abstractas se escribe en cursiva.



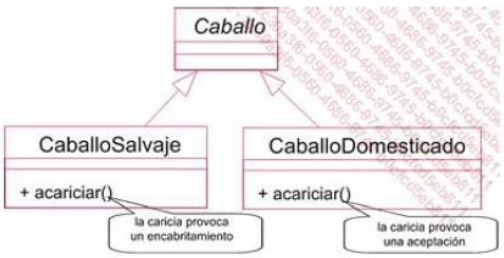
8. POLIMORFISMO

El polimorfismo significa que una clase (generalmente abstracta) representa un conjunto formado por objetos diferentes, ya que éstos son instancias de subclases diferentes. Cuando se llama a un método del mismo nombre, esta diferencia se traduce en comportamientos distintos (excepto en los casos en los que el método es común y las subclases lo han heredado de la superclase).

Ejemplo

El método acariciar tiene un comportamiento diferente según si el caballo es una instancia de CaballoSalvaje o de CaballoDomesticado. En el primer caso, el comportamiento será un rechazo (que se traducirá en un encabritamiento) mientras que, en el segundo, el comportamiento será una aceptación.

Si consideramos la clase Caballo en su totalidad, tenemos un conjunto de caballos que reaccionan de distinta manera al activarse el método acariciar.

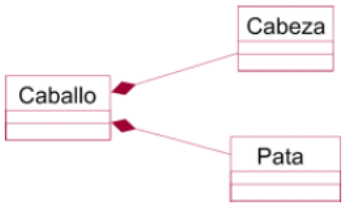


9. COMPOSICIÓN

Un objeto puede ser complejo y estar compuesto por otros objetos. La asociación que une a estos objetos es la composición, que se define a nivel de sus clases, pero cuyos vínculos se establecen entre las instancias de las clases. Los objetos que forman el objeto compuesto se denominan *componentes*.

Ejemplo

Un caballo es un ejemplo de objeto complejo. Está formado por diferentes órganos (patas, cabeza, etc.).



La composición puede adoptar dos formas:

- composición débil o agregación;
- composición fuerte.

En la composición débil, los componentes pueden ser compartidos por varios objetos complejos. En la composición fuerte, los componentes no pueden compartirse y la destrucción del objeto compuesto conlleva la destrucción de sus componentes.

Ejemplo

Si retomamos el ejemplo precedente con el supuesto de un caballo de carreras enjaezado y añadimos a sus componentes una silla, obtenemos:

- *Una composición fuerte para las patas y la cabeza. Efectivamente, las patas y la cabeza no pueden compartirse y la desaparición del caballo conlleva la desaparición de sus órganos.*
- *Una agregación o composición débil para la silla.*