Pedro Mauricio Rafael

Resumen

2016340190

Una clase es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se ocupan para representar entidades o conceptos como los sustantivos en el lenguaje. Cada clase es un modelo que define un conjunto de variables, el estado y métodos para operar con dichos datos, cada objeto creado a partir de la clase se denomina “instancia“ de la clase.

Las clases son un pilar fundamental de la [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos). Permiten [abstraer](https://es.wikipedia.org/wiki/Abstracci%C3%B3n_(inform%C3%A1tica)) los datos y sus operaciones asociadas al modo de una [caja negra](https://es.wikipedia.org/wiki/Caja_negra_(sistemas)). Los lenguajes de programación que soportan clases difieren sutilmente en su soporte para diversas características relacionadas con clases. La mayoría soportan diversas formas de [herencia](https://es.wikipedia.org/wiki/Herencia_(programaci%C3%B3n_orientada_a_objetos)). Muchos lenguajes también soportan características para proporcionar [encapsulación](https://es.wikipedia.org/wiki/Encapsulamiento_(inform%C3%A1tica)), como especificadores de acceso.

Una clase también puede tener una representación (metaobjeto) en tiempo de ejecución, que proporciona apoyo en tiempo de ejecución para la manipulación de los metadatos relacionados con la clase.

Las clases representan un tipo particular de objetos, los cuales tienen características y comportamiento similar. Cada clase tiene asociado un código (definición de la clase) que determina los atributos que tiene las clases y los métodos que pueden ejecutar y como lo hacen.

Programar orientado a objetos consiste en escribir código de clases de objetos.

Así de cada clase pueden crearse múltiples objetos.

Cada objeto tiene valores propios asignados a los atributos: estado del objeto

un objeto es una unidad dentro de un [programa de computadores](https://es.wikipedia.org/wiki/Programa_(computaci%C3%B3n)) que consta de un estado y de un comportamiento, que a su vez constan respectivamente de datos almacenados y de tareas realizables durante el tiempo de ejecución. Un objeto puede ser creado [instanciando](https://es.wikipedia.org/wiki/Instancia_(programaci%C3%B3n)) una [clase](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)), como ocurre en la programación orientada a objetos, o mediante escritura directa de código y la replicación de otros objetos, como ocurre en la [programación basada en prototipos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_basada_en_prototipos).

En el mundo de la [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) (POO), un objeto es el resultado de la instanciación de una [clase](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)).[1](https://es.wikipedia.org/wiki/Objeto_(programaci%C3%B3n)#cite_note-1)​ Una [clase](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)) es el anteproyecto que ofrece la funcionalidad en ella definida, pero ésta queda implementada sólo al crear una [instancia](https://es.wikipedia.org/wiki/Instancia_(programaci%C3%B3n)) de la [clase](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)), en la forma de un objeto.

Para instanciar (crear) objetos de una determinada clase (estándar de Java, creada por nosotros, o por un tercero) podemos encontrarnos con distintas formas que debemos aplicar. La más habitual de ellas es utilizar la palabra reservada new, siguiendo el siguiente formato:

New NombreDeLaCLase()

Esto sería una llamada al método constructor de la clase. Este tipo de llamada nos devuelve una instancia u objeto de dicha clase. Una vez creado el objeto, podemos guardarlo en una variable o utilizarlo dentro de cualquier expresión compatible con la clase del objeto creado.

Para declarar una variable que pueda contener un objeto de una determinada clase se actúa de forma similar a la declaración de las variables habituales, es decir, siguiendo la estructura:

NombreDeLaClase nombreDeLaVariable;

Es muy habitual que a la vez que se declara una variable, se cree un objeto de su misma clase y se le asigne a dicha variable, formando una sentencia completa de la siguiente forma:

NombreDeLaClase nombreDeLaVariable = new NombreDeLaClase();

Atributo. Los atributos son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades. Los atributos se guardan en variables denominadas de instancia, y cada objeto particular puede tener valores distintos para estas variables. Las [variables](https://www.ecured.cu/Variables) de [instancia](https://www.ecured.cu/index.php?title=Instancia&action=edit&redlink=1) también denominados miembros dato, son declaradas en la clase pero sus valores son fijados y cambiados en el [objeto](https://www.ecured.cu/Objeto). Además de las variables de instancia hay [variables](https://www.ecured.cu/Variables) de [clase](https://www.ecured.cu/Clase), las cuales se aplican a la [clase](https://www.ecured.cu/Clase) y a todas sus instancias.

 Existen dos formas de atributos:

* Atributos que se definen en [Common Language Runtime](https://www.ecured.cu/index.php?title=Common_Language_Runtime&action=edit&redlink=1) (CLR).
* Atributos personalizados que se pueden crear para agregar información adicional al código. Esta información se puede recuperar después mediante programación.

En la [programación](https://es.wikipedia.org/wiki/Programaci%C3%B3n), un método es una [subrutina](https://es.wikipedia.org/wiki/Subrutina) cuyo código es definido en una [clase](https://es.wikipedia.org/wiki/Clases_(programaci%C3%B3n_orientada_a_objetos)) y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un [objeto](https://es.wikipedia.org/wiki/Objetos_(programaci%C3%B3n_orientada_a_objetos)), como es el caso de los métodos de instancia. Análogamente a los procedimientos en [lenguajes imperativos](https://es.wikipedia.org/wiki/Lenguajes_imperativos), un método consiste generalmente de una serie de sentencias para llevar a cabo una acción, un juego de parámetros de entrada que regularán dicha acción o, posiblemente, un valor de salida (o valor de retorno) de algún tipo.

La diferencia entre un [procedimiento](https://es.wikipedia.org/wiki/Subrutina) (generalmente llamado función si devuelve un valor) y un método es que este último, al estar asociado con un objeto o clase en particular, puede acceder y modificar los datos privados del objeto correspondiente de forma tal que sea consistente con el comportamiento deseado para el mismo. Así, es recomendable entender a un método no como una secuencia de instrucciones sino como la forma en que el objeto es útil (el método para hacer su trabajo). Por lo tanto, podemos considerar al método como el pedido a un objeto para que realice una tarea determinada o como la vía para enviar un mensaje al objeto y que éste reaccione acorde a dicho mensaje