

A thick dark blue vertical bar runs down the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. In the bottom left corner, several thin, curved lines in dark blue and light gray sweep upwards and to the right.

9-3-2020

# Programación Orientada a Objetos

Nayeli Jazmín Contreras Hernández

## La encapsulación

Es la característica de un lenguaje POO que permite que todo lo referente a un objeto quede aislado dentro de éste. Es decir, que todos los datos referentes a un objeto queden "encerrados" dentro de éste y sólo se puede acceder a ellos a través de los miembros que la clase proporcione (propiedades y métodos).

La encapsulación define los niveles de acceso para elementos de esa clase. Estos niveles de acceso definen los derechos de acceso para los datos, permitiéndonos el acceso a datos a través de un método de esa clase en particular, desde una clase heredada o incluso desde cualquier otra clase. Existen tres niveles de acceso: público, las funciones de todas las clases pueden acceder a los datos o métodos de una clase que se define con el nivel de acceso público. Este es el nivel de protección de datos más bajo; protegido, el acceso a los datos está restringido a las funciones de clases heredadas, es decir, las funciones miembros de esa clase y todas las subclasses; privado, el acceso a los datos está restringido a los métodos de esa clase en particular. Este es nivel más alto de protección de datos.

## La abstracción

Es que la clase debe representar las características de la entidad hacia el mundo exterior, pero ocultando la complejidad que llevan aparejada. O sea, nos abstrae de la complejidad que haya dentro dándonos una serie de atributos y comportamientos (propiedades y funciones) que podemos usar sin preocuparnos de qué pasa por dentro cuando lo hagamos. Así, una clase (y por lo tanto todos los objetos que se crean a partir de ella) debe exponer para su uso solo lo que sea necesario. Cómo se haga "por dentro" es irrelevante para los programas que hagan uso de los objetos de esa clase.

La abstracción encarada desde el punto de vista de la programación orientada a objetos expresa las características esenciales de un objeto, las cuales distinguen al objeto de los demás. Además de distinguir entre los objetos provee límites conceptuales. Entonces se puede decir que la encapsulación separa las características esenciales de las no esenciales dentro de un objeto. Si un objeto tiene más características de las necesarias los mismos resultarán difíciles de usar, modificar, construir y comprender sobre todo cuando es un método de número entero con terminación fraccionaria.

## El polimorfismo

En programación orientada a objetos el polimorfismo se refiere a la posibilidad de definir clases diferentes que tienen métodos o atributos denominados de forma idéntica, pero que se comportan de manera distinta. El concepto de polimorfismo se puede aplicar tanto a funciones como a tipos de datos. Así nacen los conceptos de funciones polimórficas y tipos polimórficos. Las primeras son aquellas funciones que pueden evaluarse o ser aplicadas a diferentes tipos de datos de forma indistinta; los tipos polimórficos, por su parte, son aquellos tipos de datos que contienen al menos un elemento cuyo tipo no está especificado.

El único requisito que deben cumplir los objetos que se utilizan de manera polimórfica es saber responder al mensaje que se les envía. La apariencia del código puede ser muy diferente dependiendo del lenguaje que se utilice.

## La herencia

es específica de la programación orientada a objetos, donde una clase nueva se crea a partir de una clase existente. La herencia (a la que habitualmente se denomina subclase) proviene del hecho de que la subclase (la nueva clase creada) contiene los atributos y métodos de la clase primaria. La principal ventaja de la herencia es la capacidad para definir atributos y métodos nuevos para la subclase, que luego se aplican a los atributos y métodos heredados.

Esta particularidad permite crear una estructura jerárquica de clases cada vez más especializada. La gran ventaja es que uno ya no debe comenzar desde cero cuando desea especializar una clase existente. Como resultado, se pueden adquirir bibliotecas de clases que ofrecen una base que puede especializarse a voluntad (la compañía que vende estas clases tiende a proteger los datos miembros usando la encapsulación).