

Programación orientada a objetos

Patricio Martínez

<2017-11-14 mar>

Introducción

A lo largo de la historia han habido varias evoluciones en la informática que la ha mejorada mucho hasta nuestros días:

Esta evolución ha sido gracias a:

- Avances tecnológicos
- Avances conceptuales
- Avances en cuanto al enfoque de la programación

Simula 67

Se desarrolló en el año 1967 pero no se reconoció hasta tiempo después.
Inspirados en el tenemos:

- Smalltak (1969- 1972)
- C++ (1983)
- Y muchos otros

Tipos de programación

- Programación lineal
- Programación estructurada
- Programación funcional
- Programación orientada a objetos

Qué es la POO (OOP)

Es un conjunto de técnicas que nos permiten incrementar enormemente nuestro proceso de producción de software aumentando drásticamente nuestra productividad y permitiéndonos además abordar proyectos de mucha mayor envergadura.

Usando estas técnicas, nos aseguramos la re-usabilidad de nuestro código.

Definición de Clase

Es una abstracción que hacemos de nuestra experiencia sensible. Una **clase define** las propiedades y métodos necesarios para **crear objetos**. También tenemos las **superclases y clases virtuales o abstractas**.

Definición de Objeto

Un objeto es un conjunto de datos y métodos. Lo importante es que ambos están intrínsecamente ligados y forman una única unidad conceptual y operacional.

Herencia

Esta es una de las características más importantes ya que nos permite reutilizar mucho código o incluso usar código creado por otra persona. Dos palabras claves son **this** y **super**

Encapsulación

Llamamos encapsulación al **intrínseco vínculo entre datos y métodos** y al modo de acceder y modificar las propiedades.

Polimorfismo

El polimorfismo es la capacidad de los objetos de ante el mismo mensaje responder de distinto modo

Es un tipo de polimorfismo donde varios métodos pueden tener el mismo nombre si el tipo de parámetro que reciben o el número de ellos es distinto.

<code>Write(int i);</code>	Escribe un entero
<code>Write(long l);</code>	Escribe un long
<code>Write(float f);</code>	Escribe un flotante
<code>Write(string s);</code>	Escribe una cadena de texto

Constructores y destructores

Para utilizar un objeto hay que construirlo o crearlo y para ello se usa el *Constructor de la Clase*. Para ello dependiendo del lenguaje existen dos procedimientos.

- 1 Utilizando un método especial usando una **palabra reservada**
- 2 Usando un operador especial normalmente **new**

Los **destructores** hacen justo lo contrario que los constructores.

La mayoría de los lenguajes tienen un constructor y un destructor **por defecto**.

Accesibilidad de Datos y Métodos

Indican la **visibilidad** que tiene una variable o método.

Públicos

Son visibles dentro y fuera de la clase

Usan la palabra reservada **public**

Protegidos

Son visibles dentro de su clase y las clases heredadas.

Usan la palabra reservada **protected**

Privados

Solo son accesibles desde dentro de la clase en la que existen

Usan la palabra reservada **private**

Lo normal es ver **Métodos públicos** y **propiedades protegidas o privadas**.