**Programación orientada a objetos**

Alberto Parera

09/enero/2024

**Paradigma orientado a objetos**

* Es totalmente modular.
* API (es como el #include) – Java
  + Clases (son como structs) – atributos y métodos (son como funciones/módulos)
* MVC (modelo, vista y controlador)
* Consiste en un conjunto de objetos y métodos relacionados entre sí.
  + Los objetos representan entidades del mundo real.
  + Los métodos son las acciones que pueden realizar los objetos.
  + En el proceso de abstracción se establecen los atributos y métodos.
* Ejemplo: Para un sistema de pagos
  + Persona sería el objeto y sus atributos serian edad, \*peso, nombre y \*tipo de sangre.
  + A la hora de descartar atributos de un objeto se hace un proceso de abstracción.
* **CamelCase** (buenas prácticas de programación):
  + Primera letra ce las clases con mayúscula, si son dos palabras la primera letra de cada palabra con mayúsculas y sin espacio.
  + Atributos y métodos empiezan con minúscula.

**Proceso de abstracción**: Solamente se consideran las características principales que puedan definir a un objeto para el problema o situación que se en cuestión.

Características principales de objetos:

* Los objetos tienen atributos. También se les conoce como estados.
  + Ejemplo: Libro tiene atributos: titulo, autor, editorial.
* Los objetos tienen comportamientos (métodos), los métodos me permiten acceder a los atributos de un objeto.
  + Ejemplo: setTitulo(); o getTitulo();.
* Los objetos pueden interactuar entre ellos para resolver una tarea específica.
* A partir de una clase se generan instancias.

A close up of a sign

Description automatically generated

Diagrama de clase:

A screenshot of a computer

Description automatically generated

* Diagrama UML (Lenguaje Unificado de Modelado).

**Herencia**

* Es una de las características más importantes de la programación orientada a objetos.
* Se pueden definir nuevas clases a partir de otras ya generadas. Es decir que heredan todos los atributos y métodos de la calase padre.
* Superclase:
  + Contiene los atributos y métodos comunes a las clases que heredan de ella.
* Subclase n:
  + Deben de contener los atributos y comportamientos específicos a ellas.
* En Java se utiliza la palabra “extends” para indicar la relación de herencia.
* Las subclases heredan los miembros de la superclase – métodos y atributos.