**Clase 02 – Programador Java**

**Introducción – Clase 02 Java**

La encapsulación sugiere que las variables de una clase deben ser privadas y solo deben ser llamadas por sus métodos.

* Set: Escritura, Get: Lectura
* Ejemplo: getEdad -> propiedad edad

|  |  |  |  |
| --- | --- | --- | --- |
|  | get | set | variable |
| Name | Si | No | name |
| birthdate | No | Si | birthdate |
| address | Si | Si | Address |
| age | Si | No | birthdate |

* <http://gcoronelc.blogspot.pe/>
* La encapsulación se define en dos métodos:
  + Get \_> Leer el valor de la variable
  + Set \_> Modificar o establecer el valor de la variable
* Propiedad, son los métodos get and set que identifica una variable.
  + getNombre -> nombre
  + setPrecio -> precio
* Tipos primitivos: Dato
  + Int, float: Cuando se declara una variable a nivel de clase, su valor por defecto es 0.
  + boolean: Cuando se declara una variable a nivel de clase, su valor por defecto es false.
* Las clases empiezan con letra Mayúscula.
  + Los apuntadores declarados a nivel de clase, su valor por defecto es null.
* Null, es ausencia de variables.
* EL constructor, se utiliza para inicializar el objeto.

Producto prod = new Producto();

Constructor por defecto

* Cuando tu clase no tiene un constructor de manera explícita, o sea no has definido, Java te agrega el constructor por defecto. Por ese motivo se puede crear instancias de una clase.
* El apuntador es una variable cuyo tipo de usuario es una clase.
* Ejemplo visual:

Objeto

prod

Apuntador

* Cuando dejas de usar el objeto, el objeto sigue en la memoria, cuando un objeto deja de tener variables, se retira de la memoria por el recolectador de basura de Java (Java recollector)
* Para eso el destructor es un método que ya existe en la clase Object
* Toda CLASE hereda de la clase Object, de manera explícita (OBJECT: CLASE RAíZ)
* En Object se encuentra el método finalize();
* Entonces para usarlo en tu CLASE hacemos un sobre-escritura de la función.
* EL DTO permite transportar datos de una manera más sencilla (como un carrito de compras)
* El DTO encapsula los datos para el transporte, es un objeto (Creado por Martin fowler)

Service

X

DTO

* Otra forma de encapsular las variables, click derecho -> refactor -> encapsulate field