INGENIERÍA MECATRÓNICA

DI_CERO

DIEGO CERVANTES RODRÍGUEZ

JAVA 8

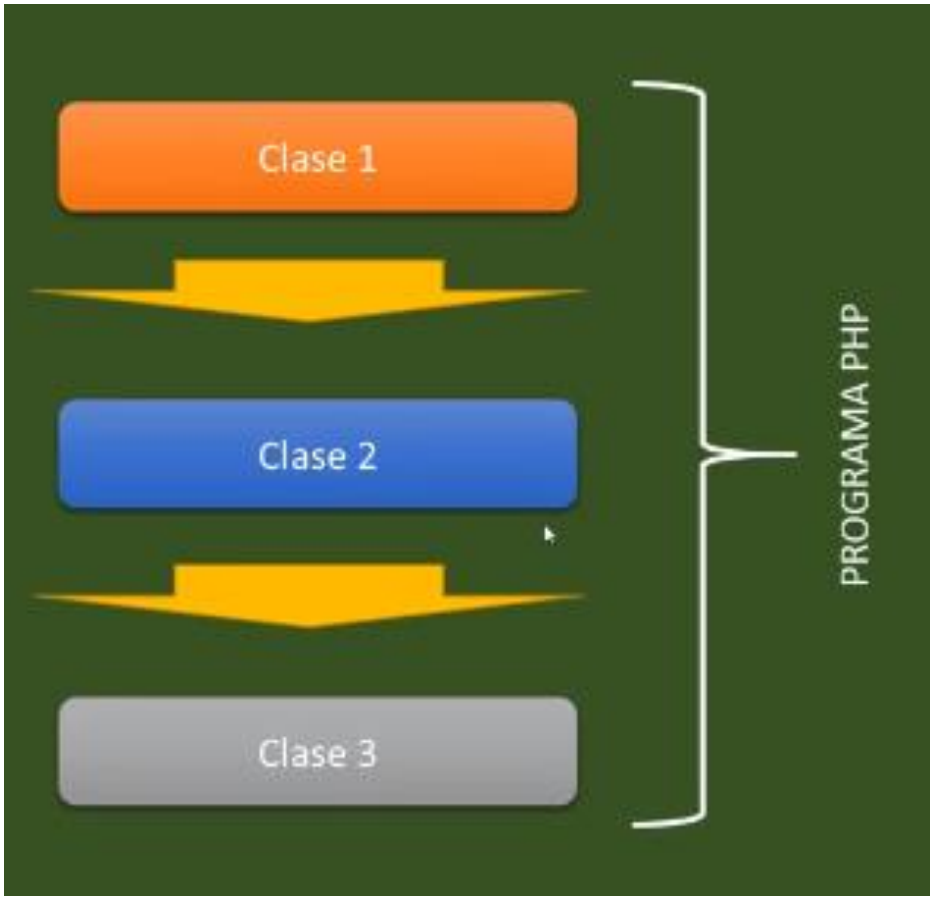
APACHE NETBEANS IDE 12.5

P00: Modularización, Encapsulación, Setter y Getter

Contenido

Modularización

Este concepto sirve para cuando nosotros queremos crear una aplicación compleja con un lenguaje POO, **lo que hace la modularización es dividir este gran programa en pequeñas partes o módulos (cada módulo es una clase)**, **todas las clases van estar interconectadas entre ellas y funcionarán como una unidad;** esto se hace para evitar que el programa entero falle porque **si uno de los módulos llegara a fallar o tenga algún error, solo fallaría esa parte del código y no el programa entero** de esta manera además podamos encontrar más fácilmente en dónde estuvo el error.



Esto se asemeja mucho a cómo funciona un equipo de audio de la vida real donde tenemos varias partes como las bocinas, el ecualizador, el lector de CD, el radio, etc. Donde todos están conectados para funcionar en conjunto.

Encapsulación

El concepto de modularización donde partimos nuestro código en varias partes (clases) viene en conjunto con el concepto de **encapsulación, esto se refiere a que debe haber ciertas características de cada clase en específico que no deben poder ser accedidas desde las otras clases**, ya sea por motivos de seguridad o para asegurarnos que ciertos valores no puedan ser alterados por otras clases.

Esto puede ser entendido mejor si regresamos al ejemplo del reproductor de música:

¿Tiene sentido que las canciones puedan ser retrocedidas o avanzadas desde la bocina? Pues no, entonces para que mi programa tenga sentido cada clase debe cumplir su cometido y nada más.

Todos estos aspectos pueden ser elegidos dependiendo del tipo de encapsulación que le asignemos a cada elemento, tanto a los métodos como a los atributos de cada clase, esto porque de igual manera debe haber propiedades y funcionalidades que puedan ser accedidas desde cualquier clase.

Modificadores de Acceso

Para poder encapsular algunas cosas sí y otras no y que algunos atributos y métodos puedan ser accedidos y otros no, es que se usan los modificadores de acceso, generalmente se usan sólo 3:

- **public:** Hace accesible a esa propiedad desde cualquier parte del programa (**desde cualquier clase**).
- **private:** Hace accesible a esa propiedad **solamente desde la propia clase**.
- **protected:** Hace accesible a esa propiedad **desde la propia clase o desde las clases que hereden de ella**.

Cuando no existe modificador en un método o atributo, el modificador por defecto es el **public**.

Métodos Getter y Setter

Cuando mis métodos o atributos están encapsulados, no podré acceder a ellos para nada, para ello se usan unos métodos especiales llamados **getter** (para poder acceder a las propiedades encapsuladas del objeto) y **setter** (para poder modificar las propiedades encapsuladas). Estos métodos como todos los demás se declaran dentro de las clases y siempre van después del constructor.

- Para declarar estos métodos se utiliza el operador **this** para que nos refiramos a los elementos de la misma clase en la que nos encontramos en conjunto con el símbolo **->** para poder acceder a los atributos o métodos de la clase.
- De igual manera se utiliza el concepto de **parámetro** porque es la manera en la que se le pasa el valor que queremos asignarle al atributo o método.
- Y también se usa la instrucción **return** porque es la manera en la que el método nos retorna el valor de un elemento encapsulado.

```
class NombreDeLaClase{

    //Atributos o propiedades encapsuladas.

    private $atributo_1;

    private $atributo_2;...

    private $atributo_n;


    //Métodos o funcionalidades encapsuladas.

    private function método_1(){

        Acción 1 que puede ejecutar nuestro objeto

    }

    private function método_2(){

        Acción 2 que puede ejecutar nuestro objeto

    }...

    private function método_n(){

        Acción n que puede ejecutar nuestro objeto

    }
```

```
    //MÉTODO CONSTRUCTOR que indica el estado inicial de los    //objetos    que
pertenezcan a esta clase
    function _ _construct(){
        $this ->atributo_1 = valor;
        $this ->atributo_2 = valor;
        $this ->atributo_n = valor;
    }

    //Métodos SETTER Y GETTER.
    function get_atributo(){
        return atributo_1;
    }
    function set_atributo($valor){
        $this -> atributo_1 = $valor;
    }

}
```

Referencias:

Pildoras Informáticas, “Curso Java”, 2023 [Online], Available: https://www.youtube.com/playlist?list=PLU8oAlHdN5BktAXdEVCLUYzvDyqRQJ2lk