

INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

JAVA 8

APACHE NETBEANS IDE 12.5

P00: Herencia

Contenido

Herencia	2
Superclases y Subclases	3
Sobreescritura de Métodos	4
Operador parent	4
Referencias:	5



Herencia

La herencia sirve para que cuando programemos con POO no escribamos código de más y si algunas clases usan los mismos atributos y/o métodos de otra clase, hagamos que herede de ella y de esa manera reutilice el código ya escrito. La herencia hace que **la clase padre herede los atributos** (que describen las características de las instancias de clase) **y métodos** (que describen las cosas que pueden hacer los objetos de la clase) **de la clase abuelo** y a su vez **la clase hijo hereda de la clase padre, por lo que posee los atributos tanto de la clase padre como de la clase abuelo.**



Por esta razón la clase que tendrá mayor número de atributos y métodos es la clase hijo.



Superclases y Subclases

El término de **súperclase** y **subclase** sirve para saber encontrar la jerarquía en la que se encuentra una clase en referencia con las demás cuando hablamos de herencia.

Cuando decimos **superclase** nos referimos a la clase que se encuentra **encima** de la clase actual en la que nos encontramos.

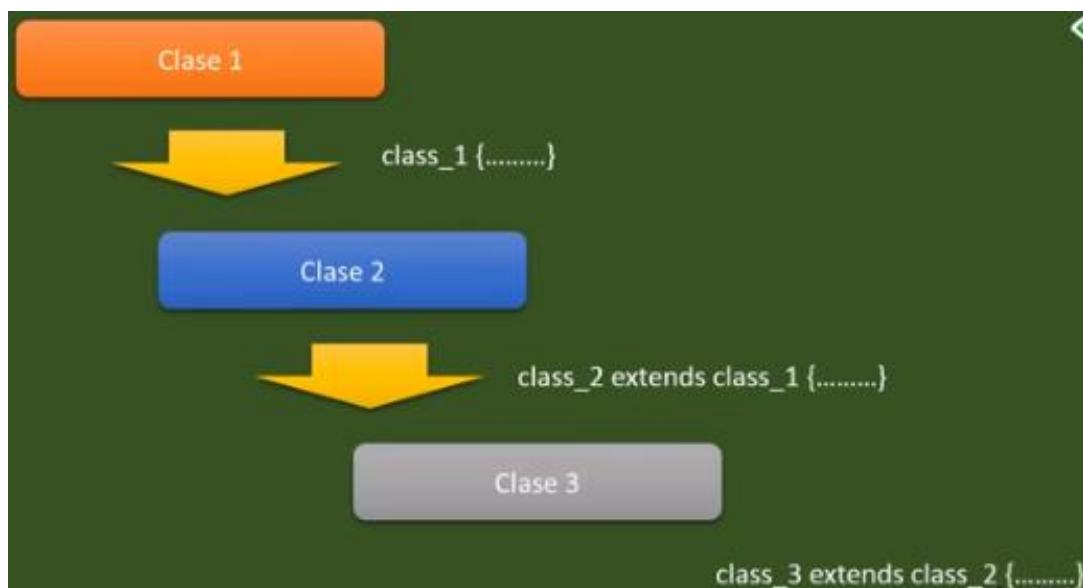
Cuando decimos **subclase** nos referimos a la clase que se encuentra **abajo** de la clase actual en la que nos encontramos.

Por ejemplo, la **superclase** de la Clase 2 es la Clase 1 y la **subclase** de la Clase 2 es la Clase 3.



Para que una clase pueda heredar de otra lo único que se debe hacer es agregar la instrucción **extends** después de que declaramos el nombre de la clase y poner el nombre de la clase de la que hereda.

PHP no permite la herencia múltiple, osea que una clase herede de varias.



Por convención siempre la primera letra de las clases se debe poner en mayúscula.

```
class NombreDeLaClaseHijo extends NombreDeLaClasePadre{
```

Esto se debe poner solo en la clase hijo, en la clase padre no se pone nada extra

```
}
```

Sobreescritura de Métodos

La sobre escritura de métodos sirve para cuando quiero modificar completamente algún método existente que la clase hijo está heredando de la clase padre, para que esto sea posible el método de la clase hija debe tener el mismo nombre que el de la clase padre.

```
class NombreDeLaClasePadre{
```

```
    function metodo_sobre-escrito(){
```

Código que quiero modificar para la clase hija que herede de esta clase

```
    }
```

```
}
```

```
class NombreDeLaClaseHijo extends NombreDeLaClasePadre{
```

```
    function metodo_sobre-escrito(){
```

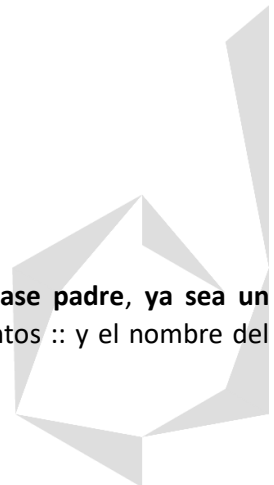
Código modificado para la clase hija

```
    }
```

```
}
```

Operador parent

El operador **parent** sirve para que la clase hijo ejecute algún código de la clase padre, ya sea un atributo o un método. Esto se hace con el operador **parent** seguido de dos puntos :: y el nombre del



atributo o método al que quiero acceder, esto sirve para cuando no quiero sobre escribir el método sino darle alguna funcionalidad o valor extra.

```
class NombreDeLaClasePadre{  
    var $atributo;  
    function metodo(){  
        Código al que le quiero añadir algo cuando se ejecute para la clase hija  
    }  
}
```

```
class NombreDeLaClaseHijo extends NombreDeLaClasePadre{  
    function metodo(){  
        parent::metodo();  
        parent::atributo;  
        Código agregado para el método de la clase hija  
    }  
}
```

Referencias:

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

