

Herencia en PHP

Class extends

Tipos de Herencia teóricos

- Simple
- Multiple
- Multinivel
- Jeràrquica

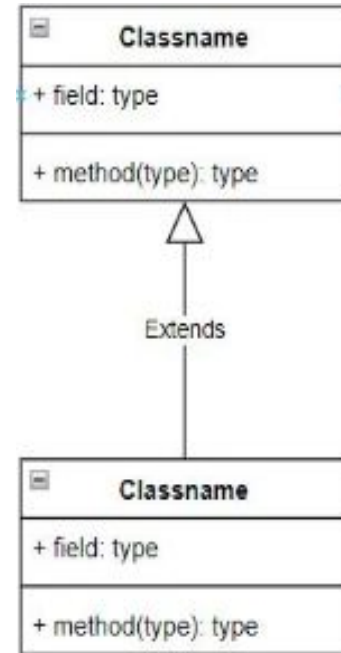


Simple

Propiedades:

- Un único padre
- Un único hijo

En PHP se implementa mediante “**extends**” al definir la clase heredera (hija).

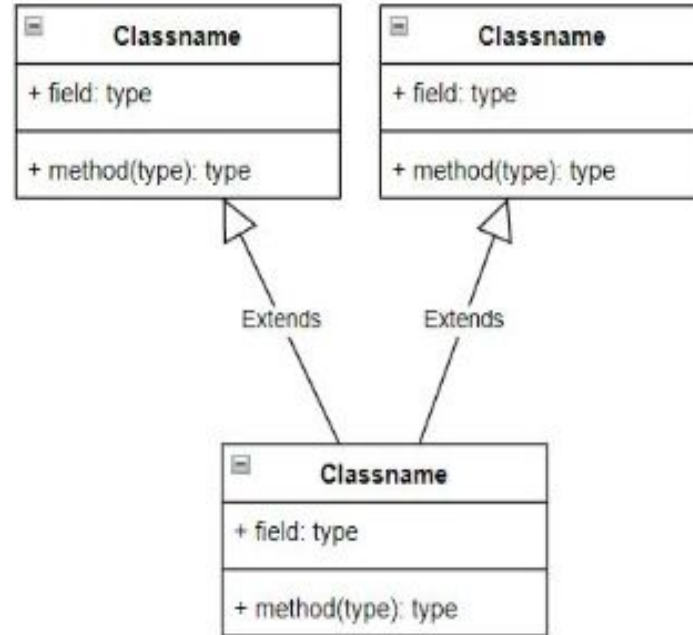


Multiple

Propiedades:

- Múltiples padres
- Un único hijo

Implementación **indirecta**
en PHP (spoiler: Interface &
Traits)

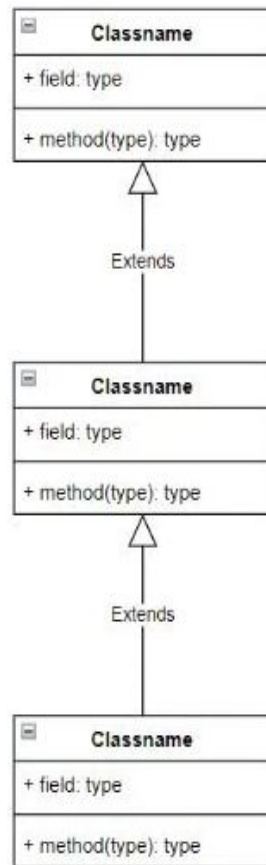


Multinivel

Propiedades:

- Un único padre
- Un único hijo
- Un hijo puede ser padre de otra clase

En PHP se implementa mediante “**extends**” al definir cada clase heredera (cada hijo).

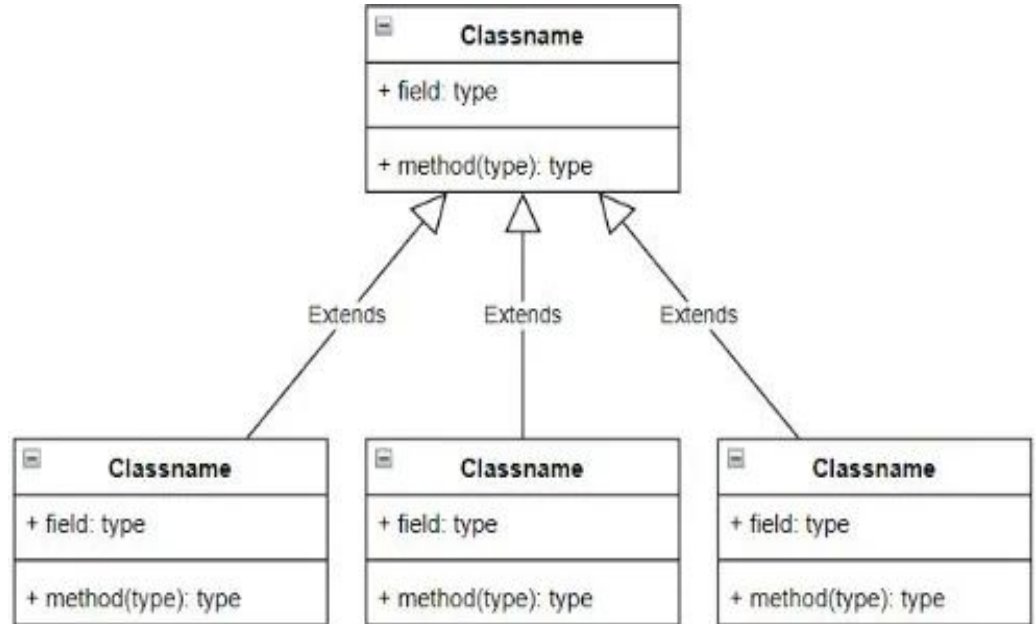


Jeràrquica

Propiedades:

- Un único padre
- Múltiples hijos

En PHP se implementa mediante “**extends**” al definir cada clase heredera (cada hijo).



Resumen de implementació

- **Simple:** Directamente mediante **extends**
- **Multiple:** Indirectamente mediante Interfaces y Traits
- **Multinivel:** Directamente mediante **extends** (herencia simple)
- **Jeràrquica:** Directamente mediante **extends** (herencia simple)

Otra denominación:

- Clase padre: superclase
- Clase hija: subclase



Traer atributos heredados

Los atributos de una clase padre se invocan igual que si hubieran sido declarados *dentro de la clase hija*:

```
$this->nom;
```

Esto solo funciona si están declaradas como **public** o como **protected**.

Ejemplo; con una instancia del objeto creada, si el atributo es public:

```
$variable_subclasse->nom;
```



Métodos constructores y herencia

Los métodos constructores NO se heredan, pero se pueden llamar.

Para acceder a un método constructor de una clase padre se debe emplear:

parent::__construct

Esta debe ser la primera instrucción del constructor de la clase hija.

Si no se declara, se realiza una llamada por defecto.



Llamar a métodos de la clase padre

Un hijo puede sobrescribir los métodos heredados.

Sin embargo, se pueden invocar los métodos por defecto de la clase padre mediante:

```
parent::nombre_metodo()
```

.

