

Programación web con PHP



Patrones: Factory Method

Temario

Tipo.....	4
Propósito.....	4
Utilización.....	4
Ventajas.....	4
Desventajas.....	4
Estructura.....	5
¿Cómo funciona?.....	5
Código.....	6
Interfaz Producto Abstracto.....	6
Clase Producto Concreto A.....	6
Clase Producto Concreto B.....	7
Clase abstracta Creador Abstracto.....	7
Clase Creador Concreto A.....	8
Clase Creador Concreto B.....	8
index.php.....	9

Tipo

Creación (Nivel clases)

Propósito

Define una interfaz para la creación de un objeto permitiendo a las subclases decidir de que Clase instanciarlo.

Utilización

- Cuando una clase no puede saber con anterioridad que objeto debe crear
- Cuando una clase quiere que sus subclases especifiquen los objetos que crea
- Cuando una clase delega responsabilidades a una de sus subclases y desea localizar la clase a la que se le delegó

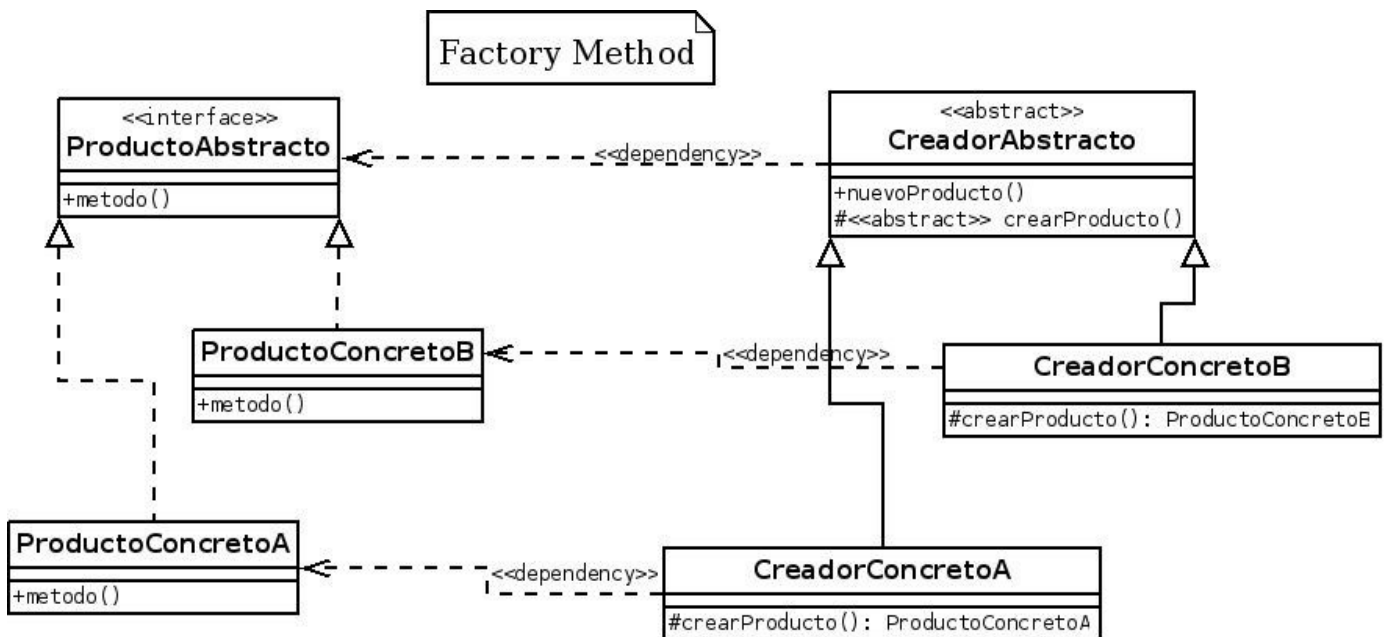
Ventajas

- Es un diseño flexible
- Facilita las ampliaciones pudiendo crear mas subclases

Desventajas

- Es obligatoria la definición de la clase creador.

Estructura



¿Cómo funciona?

- Definimos la interfaz Producto Abstracto que contendrá la definición de los métodos de Todos los Productos Concretos
- Definimos tantos Productos Concretos como necesitemos, estos implementaran la Interfaz Producto Abstracto y contendrán la implementación de sus métodos
- Definimos una clase abstracta Creador Abstracto que tendrá una función Publica nuevo Producto que a su vez llamará a la función crearProducto que será un método abstracto,
- Definimos tantos Creadores Concretos como Productos Concretos tengamos y definimos la implementación de la función crearProducto que devolverá una instancia de el Producto Concreto correspondiente al Creador Concreto.
- En nuestro programa cada vez que necesitemos instanciar un Producto Completo instanciaremos su Creador Concreto correspondiente, luego tomaremos al nuevo Producto correspondiente al Credor Abtracto, esta función llamará a la función crearProducto y dependiendo del Creador Concreto devolverá la instancia del Producto Concreto correspondiente.

Código

Interfaz Producto Abstracto

```
<?php
/*Producto Abstracto*/
interface ProductoAbstracto{
    public function metodo1();
}
```

Clase Producto Concreto A

```
<?php
/*Producto Concreto*/
class ProductoConcretoA implements ProductoAbstracto
{
    public function metodo1()
    {
        echo "Metdodo 1 Producto Concreto A....<br>";
    }

    function __construct()
    {
        echo "Creando Producto Concreto A...<br>";
    }
}
```

Clase *Producto Concreto B*

```
<?php
/*Producto Concreto*/
class ProductoConcretoB implements ProductoAbstracto
{
    public function metodo1()
    {
        echo "Metdodo 1 Producto Concreto B....<br>";
    }

    function __construct()
    {
        echo "Creando Producto Concreto B...<br>";
    }
}
```

Clase *abstracta Creador Abstracto*

```
<?php
/*Creador Abstracto*/
abstract class CreadorAbstracto
{
    public function nuevoProducto()
    {
        return $this>crearProducto();
    }

    protected abstract function crearProducto();
}
```

Clase Creador Concreto A

```
<?php
/*Creador Concreto*/
class CreadorConcretoA extends CreadorAbstracto
{
    protected function crearProducto()
    {
        return new ProductoConcretoA();
    }

    function __construct()
    {
        echo "Creando creador concreto A...<br>";
    }
}
```

Clase Creador Concreto B

```
<?php
/*Creador Concreto*/
class CreadorConcretoB extends CreadorAbstracto
{
    protected function crearProducto()
    {
        return new ProductoConcretoB();
    }

    function __construct()
    {
        echo "Creando creador concreto B...<br>";
    }
}
```


index.php

```
<?php
//Creo un creador concreto Aa
$creadorConcretoA=new CreadorConcretoA();//Creando creador concreto
A...
//El creador crea un producto
$productoA=$creadorConcretoA>nuevoProducto();//Creando Producto
Concreto A...
//Ejecuto el metodo de producto
$productoA>metodo1();//Metdodo 1 Producto Concreto A....

//Creo un creador concreto B
$creadorConcretoB=new CreadorConcretoB();//Creando creador concreto
B...
//El creador crea un producto
$productoB=$creadorConcretoB>nuevoProducto();//Creando Producto
Concreto B...
//Ejecuto el metodo de producto
$productoB>metodo1();//Metdodo 1 Producto Concreto B....
```