CONCEPTOS FUNDAMENTALES DE YUI 2

Ricardo Pérez López

IES Doñana, curso 2017-18

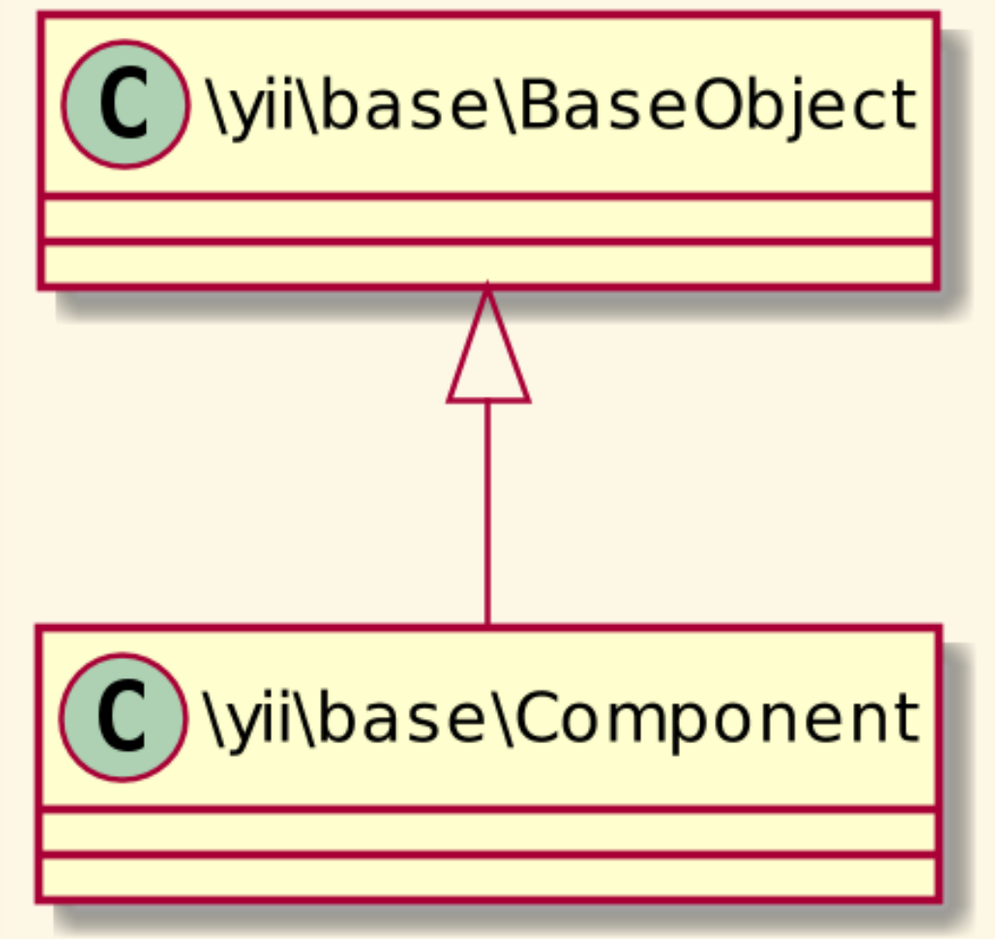
COMPONENTES

COMPONENTES

Se caracterizan por tener:

- Propiedades
- Configurabilidad
- Eventos
- Comportamientos (*behaviors*)

Las dos primeras características se heredan de `\yii\base\BaseObject`.


\yii\base\BaseObject
\yii\base\Component

\yii\base\BaseObject

Introduce las siguientes características:

- Propiedades
- Configurabilidad

PROPIEDADES

- En PHP, a las variables miembro de una clase (variables de instancia) se las denomina también *propiedades*.
- Esas variables son parte de la definición de la clase, y se usan para representar el estado de una instancia de dicha clase.
- La clase `\yii\base\BaseObject` de Yii 2 permite crear propiedades a partir de métodos *getter* y *setter*.
- Toda clase que herede (directa o indirectamente) de `\yii\base\BaseObject` podrá definir propiedades de esa manera.

Por ejemplo:

```php
namespace app\components;

class Foo extends \yii\base\BaseObject
{
    private $_label;

    // El método getter:
    public function getLabel()
    {
        return $this-> label;
    }

    // El método setter:
    public function setLabel($value)
    {
        $this->_label = trim($value);
    }
}
```

Crea la propiedad `label`, accesible mediante `$foo->label`.

```php
$foo = new Foo;

echo $foo->label;      // Llama internamente a $foo->getLabel()

$foo->label = 'hola'; // Llama internamente a $foo->setLabel('hola');
```

Como `setLabel($value)` está definido como:

```php
public function setLabel($value)
{
    $this->_label = trim($value);
}
```

al asignarle una cadena a la propiedad se *trimeará* automáticamente, eliminando los espacios sobrantes:

```php
$foo->label = '    hola    ';   // Se guarda sin espacios sobrantes

echo $foo->label;               // Devuelve "hola" (sin espacios)
```

PROPIEDADES DE SÓLO LECTURA

Si definimos sólo el *getter* y no el *setter*, crearemos una **propiedad de sólo lectura**, por lo que podremos consultar su valor pero no cambiarlo:

```php
class Prueba extends \yii\base\BaseObject
{
    public $_valor = 25;

    // Método getter (no hay setter):
    public function getValor()
    {
        return $this->_valor;
    }
}

$p = new Prueba;
echo $p->valor;  // Devuelve 25
$p->valor = 30;  // Da ERROR
```

CONFIGURABILIDAD

- Una instancia de la clase `\yii\base\BaseObject` (o de una subclase suya) permite ser *configurado*.
- Una **configuración** es simplemente un array que contiene parejas de `clave => valor`, donde la `clave` representa el nombre de una propiedad (una cadena), y el `valor` es el valor que queremos asignarle a dicha propiedad.
- Se pueden usar para:
  - Asignar valores de forma masiva a las propiedades de un objeto usando `Yii::configure($objeto, $config)`.
  - Crear una instancia asignándole valores iniciales a sus propiedades usando `Yii::createObject($config)`.
  - Más posibilidades que iremos viendo en su momento.

ASIGNACIÓN MASIVA

Supongamos la siguiente clase:

```php
use \yii\base\BaseObject;

class Prueba extends BaseObject
{
    public $uno;
    private $_dos;

    public function getDos()
    {
        return $this->_dos;
    }

    public function setDos($dos)
    {
        $this->_dos = $dos;
    }
}
```

Algunas posibles configuraciones:

```php
[ 'uno' => 5, 'dos' => 7 ]

[ 'dos' => 18 ]
```

Se pueden aplicar a un objeto ya existente:

```php
$p = new Prueba;

Yii::configure($p, [
    'uno' => 5,
    'dos' => 7,
]);

echo $p->uno; // Muestra "5"
echo $p->dos; // Muestra "7"
```

CREACIÓN DE NUEVAS INSTANCIAS

- Una configuración también se puede usar para crear nuevas instancias y asignarle valores iniciales *en la misma operación* usando el método `Yii::createObject($config)`.
- Para ello es necesario que la configuración indique el nombre de la clase que se desea instanciar mediante un elemento con clave `'class'`.
- Ejemplo:

```php
$p = Yii::createObject([
    'class' => 'Prueba',
    'uno' => 4,
    'dos' => 7,
]);
```

- Se crea en `$p` una nueva instancia de la clase `Prueba` con los valores

`$p->uno = 4` y `$p->dos = 7`.