PHP的继承方法如何获取子类名？get\_class() 和 get\_called\_class

PHP里的\_\_CLASS\_\_这类东西是静态绑定的，如果不在子类里重载的话，那么继承父类方法所得到的依旧是父类的名称，而不是子类的名称，比如：

**<?php**

**class A**

**{**

**function** **\_\_construct**()

{

**echo** \_\_CLASS\_\_;

}

**static** **function** **name**()

{

**echo** \_\_CLASS\_\_;

}

}

**class B extends A**

**{**

}

$objB = **new** B(); *// 输出 A*

B::name(); *// 输出 A*

此时，无论将B实例化还是直接调用静态方法，echo出来的都会是A。

而实际上我想要得到的是 子类B 的名称！那如何实现呢？

PHP自带两个函数 **get\_class()**和 **get\_called\_class()**可以解决这个问题。

get\_class() 用于实例调用，加入参数($this)可解决子类继承调用的问题，而get\_called\_class() 则是用于静态方法调用。

需要注意的是： get\_called\_class() 需要 PHP>=5.3.0  才支持，官方手册：***[http://php.net/manual/en/function.get-called-class.php](http://php.net/manual/en/function.get-called-class.php" \t "_blank)*** ，对于 PHP5.3.0以下的版本，有人给出了如下实现方式：

**<?php**

**if** (!function\_exists('get\_called\_class'))

{

**class classTools**

**{**

**private** **static** $i = 0;

**private** **static** $file = **null**;

**public** **static** **function** **get\_called\_class**()

{

$bt = debug\_backtrace();

*// 使用 call\_user\_func 或 call\_user\_func\_array 函数调用类方法，处理如下*

**if** (array\_key\_exists(3, $bt) && array\_key\_exists('function', $bt[3]) && in\_array($bt[3]['function'], **array**('call\_user\_func', 'call\_user\_func\_array'))

)

{

*// 如果参数是数组*

**if** (is\_array($bt[3]['args'][0]))

{

$toret = $bt[3]['args'][0][0];

**return** $toret;

}

**else** **if** (is\_string($bt[3]['args'][0]))

{

*// 如果参数是字符串*

*// 如果是字符串且字符串中包含::符号，则认为是正确的参数类型，计算并返回类名*

**if** (**false** !== strpos($bt[3]['args'][0], '::'))

{

$toret = explode('::', $bt[3]['args'][0]);

**return** $toret[0];

}

}

}

*// 使用正常途径调用类方法，如:A::make()*

**if** (**self**::$file == $bt[2]['file'] . $bt[2]['line'])

{

**self**::$i++;

}

**else**

{

**self**::$i = 0;

**self**::$file = $bt[2]['file'] . $bt[2]['line'];

}

$lines = file($bt[2]['file']);

preg\_match\_all('/([a-zA-Z0-9\\_]+)::' . $bt[2]['function'] . '/', $lines[$bt[2]['line'] - 1], $matches);

**return** $matches[1][**self**::$i];

}

}

**function** **get\_called\_class**()

{

**return** classTools::get\_called\_class();

}

}

现在，把例子修改下：

**<?php**

**class A**

**{**

**function** **\_\_construct**()

{

**echo** get\_class($this);

}

**static** **function** **name**()

{

**echo** get\_called\_class();

}

}

**class B extends A**

**{**

}

$objB = **new** B(); *// 输出 B*

B::name(); *// 输出 B*