

OOP in PHP ตอนที่ 4

เก็บตก OOP in PHP 5

สวัสดีครับ สำหรับในตอนที่ 4 นี้เราจะพูดถึงเรื่องราวการเขียนโปรแกรมเชิงวัตถุของ PHP 5 กันบ้าง เพราะตอนที่ 1-3 นั้นจะเน้นไปในส่วนของ PHP 4 ที่มีความง่ายและไม่ซับซ้อนมากนัก เพื่อปูพื้นฐานการเขียนเชิงวัตถุได้อย่างไม่น่าเบื่อ

โดยตอนนี้เราจะเข้าส่วนของ PHP 5 ด้วยเหตุผลด้านการสนับสนุนในอนาคตที่ PHP Group นั้นจะเลิกสนับสนุนในวันที่ 5 กุมภาพันธ์ 2551 นี้ โดยถ้าเป็นไปได้เราควรศึกษาใน PHP 5 เลยเสียดีกว่า และ Web Hosting ต่าง ๆ ในไทยเริ่มสนับสนุนมากขึ้นแล้วเช่นกัน

ซึ่งเนื้อหาในตอนนี้จะพูดถึงเรื่อง

- การกำหนดการเข้าถึงคุณสมบัติและแบบแผนพฤติกรรม ซึ่งมีอยู่ใน PHP 5
- การสร้างพฤติกรรมพื้นฐานในแบบฉบับของ PHP5

โดยจริง ๆ แล้วยังมีอีกมากในรายละเอียดของการเขียนโปรแกรมเชิงวัตถุโดยอ้างอิงกับ PHP 5 ถ้ามีโอกาสในการเขียนบทความอีก ทางผู้เขียนจะเขียนส่วนอื่น ๆ ในครั้งต่อไป โดยในตอนที่ผ่านมาเรามีคำเฉพาะต่าง ๆ ดังนี้

- แม่แบบ (Class)
- ลูกแบบ (Sub-Class)
- วัตถุ (Object)
- การส่งข้อความ (Messaging)
- ข้อความ (Message)
- การห่อหุ้มข้อมูล (Encapsulation)
- การสืบทอด (Inheritance)
- การทำหลายรูปแบบ (Polymorphism)
- คุณสมบัติ (Properties)
- แบบแผนพฤติกรรม (Method Behavior)
- พฤติกรรมพื้นฐาน (Constructor Method Behavior)
- การสืบทอด (Inheritance)

โดยจากตอนที่แล้วนั้นก็เขียนสำหรับติดต่อกับฐานข้อมูลไปแล้ว ซึ่งเป็นชุดคำสั่งโปรแกรมที่ผู้เขียนใช้อยู่เป็นประจำ ซึ่งอ้างอิงตาม PHP 4 ทั้งหมด

เรื่องการกำหนดการเข้าถึงคุณสมบัติและแบบแผนพฤติกรรม

(Accessibility Modifiers or Visibility)

เรามีคำเฉพาะอยู่ 3 ตัวด้วยกันคือ

- **public**, เราสามารถเข้าถึง, เปลี่ยนแปลง และเรียกใช้งานได้ทั้งภายในและภายนอกแม่แบบ โดยจะไม่มีผลว่าจะมีการสืบทอดจากตัวแม่แบบมากี่ชั้นก็ตาม
- **protected**, เราสามารถเข้าถึง, เปลี่ยนแปลง และเรียกใช้งานได้แต่ภายในแม่แบบ และลูกแบบที่สืบทอดมาจากแม่แบบเหล่านั้น
- **private**, เราสามารถเข้าถึง, เปลี่ยนแปลง และเรียกใช้งานได้เพียงแต่ภายในแม่แบบเท่านั้น

โดยการกำหนดรูปแบบการเข้าถึงดังกล่าวนี้ช่วยให้เราควบคุมการใช้งานตัวแปรต่าง ๆ ได้ง่ายขึ้น ดังที่ได้บอกไปแล้วในตอนที่ 1 ว่าเราควรเขียนแบบแผนพฤติกรรมไว้เรียกใช้คุณสมบัติเสมอเพื่อรับรองความถูกต้องต่อการเข้าถึง, เปลี่ยนแปลง และเรียกใช้งานอยู่เสมอ ซึ่งเป็นเรื่องที่เราควรทำตามหลักการของการห่อหุ้มข้อมูล

การกำหนดการเข้าถึงคุณสมบัติ

ด้านล่างคือชุดคำสั่งอย่างง่ายที่อธิบายการขึ้นถึงภายในแม่แบบของคุณสมบัติได้เป็นอย่างดี

```
1. <?php
2. /** Define MyClass */
3. class MyClass {
4.     public $public = 'Public';
5.     protected $protected = 'Protected';
6.     private $private = 'Private';
7.     function printHello() {
8.         echo $this->public;
9.         echo $this->protected;
10.        echo $this->private;
11.    }
12. }
13. /** Define MyClass2 */
14. class MyClass2 extends MyClass {
15.     protected $protected2 = 'Protected2';
16.     function printHello() {
17.         echo $this->public;
18.         echo $this->protected;
19.         echo $this->protected2;
20.         echo $this->private;
21.     }
22. }
```

ชุดคำสั่งที่ 1 (จาก PHP Manual, Chapter 19 Classes and Objects (PHP 5))

จากชุดคำสั่งที่ 1 นั้นเรากำหนดตัวแปรไว้ทั้งหมด 3 ตัวที่แตกต่างกันในรูปแบบการเข้าถึงในแม่แบบ MyClass โดยในแม่แบบ MyClass2 นั้นสืบทอดจาก MyClass โดยมีการกำหนดตัวแปร 1 ตัวที่เป็นชนิด protected อีกตัวด้วยกัน

```
1. $obj = new MyClass();
2. echo $obj->public; // works
3. echo $obj->protected; // Fatal Error
4. echo $obj->private; // Fatal Error
5. $obj->printHello(); // Shows Public, Protected and Private
6.
7. $obj2 = new MyClass2();
8. echo $obj2->public; // works
9. echo $obj2->private; // Undefined
10. echo $obj2->protected; // Fatal Error
11. $obj2->printHello(); // Shows Public, Protected, Protected2, not Private
12. ?>
```

ชุดคำสั่งที่ 2 (จาก PHP Manual, Chapter 19 Classes and Objects (PHP 5))

จากชุดคำสั่งที่ 2 นี้ท่านผู้เขียนจะอธิบายที่ละบรรทัด โดยเมื่อสั่งทำงานจะเกิดข้อผิดพลาดจะมีข้อผิดพลาดดังนี้

1. บรรทัดที่ 1 สร้างวัตถุจากแม่แบบ MyClass
2. บรรทัดที่ 2, แสดงค่าของ public ออกมา โดยแสดงค่าออกมาได้ปกติ
3. บรรทัดที่ 3, แสดงค่า protected แต่ตัวแปร protected ไม่สามารถเข้าถึงได้จากภายนอกแม่แบบได้
4. บรรทัดที่ 4, แสดงค่า private แต่ตัวแปร private ไม่สามารถเข้าถึงได้จากภายนอกแม่แบบได้
5. บรรทัดที่ 5, แสดงค่าทั้งหมดของทั้ง 3 ตัวแปรโดยตัวแปรสามารถทำงานได้ปกติทั้งหมด
6. บรรทัดที่ 7, สร้างวัตถุจากแม่แบบ MyClass2 โดยที่แม่แบบนี้สืบทอดมาจากแม่แบบ MyClass
7. บรรทัดที่ 8, แสดงค่าของ public ออกมา โดยแสดงค่าออกมาได้ปกติ
8. บรรทัดที่ 9, แสดงค่า private แต่ตัวแปร private ไม่ได้กำหนดไว้ในแม่แบบ MyClass2 และไม่ได้สืบทอดคุณสมบัตินี้มาจาก MyClass เพราะ MyClass นั้นกำหนดไว้เป็น private
9. บรรทัดที่ 10, แสดงค่า protected แต่ตัวแปร protected ไม่สามารถเข้าถึงได้จากภายนอกแม่แบบที่สืบทอดมาจาก MyClass
10. บรรทัดที่ 11, แสดงค่าทั้งหมดของทั้ง 4 ตัวแปรโดยตัวแปรสามารถทำงานได้ปกติทั้งหมด โดยที่ตัวแปร private จะไม่มีอะไรแสดงขึ้นมา เพราะเราไม่ได้กำหนดค่าตัวแปรนี้ไว้ในแม่แบบ MyClass2 และตัวแปร protected นั้นสามารถเรียกใช้ได้จากแม่แบบ MyClass2 ได้ด้วย

จากตัวอย่างทั้ง 2 ชุดคำสั่งนั้นถ้าเราไม่ได้กำหนดการเข้าถึงตัว PHP จะทำการกำหนดให้เป็น public โดยค่าเริ่มต้นเสมอ

การสร้างพฤติกรรมพื้นฐานในแบบฉบับของ PHP5

ความหมายของการสร้างพฤติกรรมพื้นฐานนั้นทางผู้เขียนจะไม่ย้อนกลับไปอธิบายซ้ำ โดยการสร้างพฤติกรรมพื้นฐานใน PHP 5 นั้นจะแตกต่างกันตรงที่ชื่อของแบบแผนพฤติกรรมที่จะใช้คำเฉพาะแทนการใช้ชื่อเดียวกับแม่แบบ เพื่อง่ายและไม่ซับซ้อนต่อการนำไปประยุกต์ใช้งานในอนาคต

โดยคำเฉพาะนั้นคือ “**__construct**” โดยนำไปแทนที่ชื่อแบบแผนพฤติกรรมในแบบเดิม ๆ ใน PHP 4 ซึ่งใน PHP นั้นไม่สามารถที่จะทำการเรียกพฤติกรรมพื้นฐานส่งต่อเป็นทอด ๆ โดยอัตโนมัติ (implicitly call) ได้ ถ้าเราจะใช้การเรียกแบบดังกล่าวนั้น ต้องทำด้วยการลงชุดคำสั่งด้วยตัวผู้พัฒนาเอง โดยใน PHP 4 จะใช้การเรียกผ่านชื่อแบบแผนพฤติกรรมในแบบการเรียกใช้ปกติ แต่ใน PHP 5 นั้นสามารถใช้ “**parent::__construct()**” แทน เพื่อลดการจำชื่อของแบบแผนพฤติกรรมที่สืบทอดมา

โดยพฤติกรรมพื้นฐานนั้นจะถูกตั้งให้เป็น **public** เสมอ และไม่สามารถกำหนดการเข้าถึงให้เป็นแบบอื่น ๆ ได้

```
<?php
class BaseClassPHP4 {/** PHP 4 */
    function BaseClassPHP4() {print "In BaseClass constructor\n";}
}
class SubClassPHP4 extends BaseClassPHP4 {
    function SubClassPHP4() {
        $this->BaseClassPHP4();
        print "In SubClass constructor\n";
    }
}
class BaseClassPHP5 {/** PHP 5 */
    function __construct() {print "In BaseClass constructor\n";}
}
class SubClassPHP5 extends BaseClassPHP5 {
    function __construct() {
        parent::__construct();
        print "In SubClass constructor\n";
    }
}
$obj = new BaseClassPHP4();
$obj = new SubClassPHP4();
$obj = new BaseClassPHP5();
$obj = new SubClassPHP5();
?>
```

ชุดคำสั่งที่ 3 (จาก PHP Manual, Chapter 19 Classes and Objects (PHP 5))

การกำหนดการเข้าถึงแบบแผนพฤติกรรม

การเข้าถึงในลักษณะของแบบแผนพฤติกรรมนั้นไม่แตกต่างจากของคุณสมบัติ เพียงแต่มีข้อยกเว้นว่า ตัวพฤติกรรมพื้นฐานจะต้องเป็น **public** เสมอ

```
1. <?php
2. /** Define MyClass */
3. class MyClass{
4.     // Constructors must be public
5.     public function __construct() { }
6.     // Declare a public method
7.     public function MyPublic() { }
8.     // Declare a protected method
9.     protected function MyProtected() { }
10.    // Declare a private method
11.    private function MyPrivate() { }
12.    // This is public
13.    function Foo(){
14.        $this->MyPublic();
15.        $this->MyProtected();
16.        $this->MyPrivate();
17.    }
18. }
19. $myclass = new MyClass;
20. $myclass->MyPublic(); // works
21. $myclass->MyProtected(); // Fatal Error
22. $myclass->MyPrivate(); // Fatal Error
23. $myclass->Foo(); // Public, Protected and Private work
24. /** Define MyClass2 */
25. class MyClass2 extends MyClass {
26.     // This is public
27.     function Foo2(){
28.         $this->MyPublic();
29.         $this->MyProtected();
30.         $this->MyPrivate(); // Fatal Error
31.     }
32. }
33. $myclass2 = new MyClass2;
34. $myclass2->MyPublic(); // works
35. $myclass2->Foo2(); // Public and Protected work, not Private
36. ?>
```

ชุดคำสั่งที่ 4 (จาก PHP Manual, Chapter 19 Classes and Objects (PHP 5))

จากตัวอย่างชุดคำสั่งที่ 4 นั้นจะเห็นว่าตัวอย่างนั้นไม่แตกต่างจากของคุณสมบัติแต่ประการใด

ทิ้งท้าย

สำหรับฉบับนี้คงเป็นฉบับสุดท้ายในรอบปี สำหรับบทความของ OOP in PHP นั้นผมจะนำส่วนที่ปรับปรุงและชุดคำสั่งต่าง ๆ ไปไว้ใน <http://www.thaicyperpoint.com/ford/blog/projects/> ไว้ครับ (จริง ๆ ตอนที่แล้วชุดคำสั่งทางผู้เขียนก็ยังไม่ได้นำไปใส่ เพราะด้วยหน้าที่การงานเลยไม่มีเวลาจริง ๆ) สำหรับเนื้อหาตอนต่อไป ถ้าสามารถติดตามต่อได้ที่ blog ของผู้เขียนที่ <http://www.thaicyperpoint.com/ford/blog/> ซึ่งจะยังคงเขียนต่อไปเรื่อย ๆ ครับ

เอกสารอ้างอิง

- **PHP Manual**, <http://www.php.net>, January 2007
- **Zend PHP Certification Study Guide**, Zend Technologies, August 2004
- **Concepts of Programming Languages (7th Edition)**, Robert W. Sebesta, April 2005
- **The Object Model**, <http://developer.apple.com/>, December 2006