บทที่ 6
การสืบทอดคลาส (Inheritance)

จากที่ได้กล่าวจากบทที่ผ่านมา จะเห็นได้ว่าการใช้งานของคลาส มีด้วยกัน 3 ประเภท ได้แก่ Public Protect และ Protected ซึ่งจะมีความสำคัญเกี่ยวข้องในบทนี้

คลาสนั้นสามารถที่จะแยกออกมาเป็นคลาสย่อยๆ ได้อีกหลายคลาส เช่น มนุษย์ สามารถแยกเป็นคลาสย่อยได้แก่ มนุษย์ที่เป็นเพศหญิง และเพศชาย เป็นต้น แต่ว่าไม่ว่าจะแยกคลาสออกมาเท่าไร คลายย่อยจะได้รับคุณสมบัติ และการกระทำมาจากคลาสหลักเสมอ เช่น มนุษย์จะเป็นผู้หญิงหรือผู้ชาย ก็จะมีคุณสมบัติ เช่น ชื่อ เพศ มือ แขน ขา เป็นต้น หรือก็จะมีการกระทำที่เหมือนกัน เช่น วิ่ง กระโดด พูด เป็นต้น เหมือนกัน ซึ่งเรียกการสืบทอดลักษณะคลาส การได้มาซึ่งแอททริบิวต์ และเมธอดของคลาสว่า “Inheritance” ซึ่งในการเขียนโปรแกรม เมื่อต้องการใช้งานบางอย่าง ซึ่งถ้าหากมีคลาสใด ที่มีคุณสมบัติที่ต้องการใช้งานดังกล่าวอยู่แล้ว ก็สามารถที่จะอ้างอิงเข้าใช้งานได้เลย ถ้าเปรียบเทียบทั่วๆ ไป นั่นก็คือ ไม่ต้องทำการประกาศค่าตัวแปรในการเก็บข้อมูลหลายๆ ครั้ง ซึ่งเป็นการอ้างอิงที่ซ้ำซ้อนกันมากจนเกินไป

การสืบทอดคลาสจะใช้คำสั่ง extends ในคลาสย่อย ตามด้วยชื่อคลาสหลักที่แยกออกมา ซึ่งในการอ้างอิงคลาสระหว่างกัน สามารถสร้างในไฟล์เดียวกัน หรือแยกไฟล์กันก็ได้ ซึ่งมีรายละเอียดดังต่อไปนี้

6.1 การสืบทอดคลาสแบบไฟล์เดียว

ในการสร้างคลาสนั้น สามารถสร้างไฟล์เดียวเก็บคลาสหลายๆ คลาสได้ ซึ่งจากตัวอย่างที่ 6.1 ประกอบไปด้วยคลาสที่เป็นคลาสหลัก และคลาสย่อย รวมกันอยู่ในไฟล์เดียวกัน

ตารางที่ 6.1 คลาส Human

```
1  <?php
2  class Human
3  {
4          public $name="สมชาย";
5          private $age=25;
6          protected $salary="10000";
```

```
7
8           public function say()
9           {
10                  return "ชื่อ ".$this->name." อายุ = ".$this->age." เงินเดือน =
11  ".$this->salary;
12          }
13
14          public function walk()
15          {
16                  return "Walking";
17          }
18  }
19
20  class Male extends Human
21  {
22          public function show_text()
23          {
24                  $this->name="สมหมาย";
25                  $this->age="40";
26                  $this->salary="10";
27                  return $this->say()."<br>ชื่อ ".$this->name." อายุ = ".$this-
28  >age." เงินเดือน = ".$this->salary;
29          }
30  }
31  $obj = new Male();
32  echo $obj->show_text();
33  echo "  ".$obj->walk();
34
35  ?>
```

ผลลัพธ์ของโปรแกรม

```
ชื่อ สมหมาย อายุ = 25 เงินเดือน = 10
ชื่อ สมหมาย อายุ = 40 เงินเดือน = 10 Walking
```

จากตัวอย่างโปรแกรม 6.1 คลาส Human เป็นคลาสแม่ หรือคลาสหลัก และคลาสย่อยชื่อ Male ซึ่งถูกสร้างมาจากคลาสหลัก Human อีกครั้งหนึ่งซึ่งสามารถอธิบายการทำงานได้ดังต่ไปนี้

6.1.1 บรรทัดที่ 2 คลาส human คือ คลาสหลัก มีแอททริบิวต์ ประกอบไปด้วย 4 แอทริบิวต์ประกอบไปด้วย $name $age และ $salary และมีเมธอด say() และ walk()

6.2.1 บรรทัดที่ 20 คลาสย่อย Male ที่ถูกสร้างมาจากคลาสหลัก Human จะใช้คำสั่ง extends

6.2.1 บรรทัดที่ 31 สร้างอ๊อบเจ๊กชื่อ obj ที่เกิดจากคลาสย่อย ดังนั้นจะสามารถเรียกใช้งานจากคลาสหลักที่เป็นแบบ public ได้

6.1 การสืบทอดคลาสแบบแยกไฟล์

ในตัวอย่างจะเป็นการแยกเก็บไฟล์หลักเอาไว้ และแยกเก็บคลาสย่อยในอีกไฟล์เอาไว้ ดังตารางที่ 6.2

ตารางที่ 6.2 คลาส Human ตั้งชื่อไฟล์ Human.php

```
<?php
class Human
{
	public $name="สมชาย";
	private $age=25;
	protected $salary="10000";

	public function say()
	{
		return "ชื่อ ".$this->name." อายุ = ".$this->age." เงินเดือน = ".$this-
>salary;
	}
	public function walk()
	{	return "Walking";	}
}
?>
```

จากตารางที่ 6.2 สร้างคลาสหลักชื่อ Human และบันทึกชื่อไฟล์ Human.php เพื่อใช้ในการอ้างอิงในการสร้างคลาสย่อยต่อไป

ตารางที่ 6.3 คลาส Male ตั้งชื่อไฟล์ Male.php

```
1   <?
2   require_once("Human.php");
3   class Male extends Human
4   {
5           public function show_text()
6           {
7                   $this->name="สมหมาย";
8                   $this->age="40";
9                   $this->salary="10";
10                  return $this->say()."<br>ชื่อ ".$this->name." อายุ = ".$this-
11  >age." เงินเดือน = ".$this->salary;
12          }
13  }
14  $obj = new Male();
15  echo $obj->show_text();
16  echo "  ".$obj->walk();
17  ?>
```

ในขั้นตอนการทำงานของชุดคำสั่งนั้น เหมือนกับข้างต้นที่ได้กล่าวมา เพียงแต่แยกไฟล์ในการเก็บคลาสหลักและคลาสย่อยเท่านั้น จะมีสิ่งที่เพิ่มขึ้นมานั่นคือ ในบรรทัดที่ 2 ของตารางที่ 6.3 require_once("Human.php"); ซึ่งจะขาดคำสั่งนี้ไม่ได้ เพราะเป็นคำสั่งในการอ้างอิงหรือเชื่อมโยงระหว่างไฟล์ที่เป็นคลาสหลักและคลาสย่อย