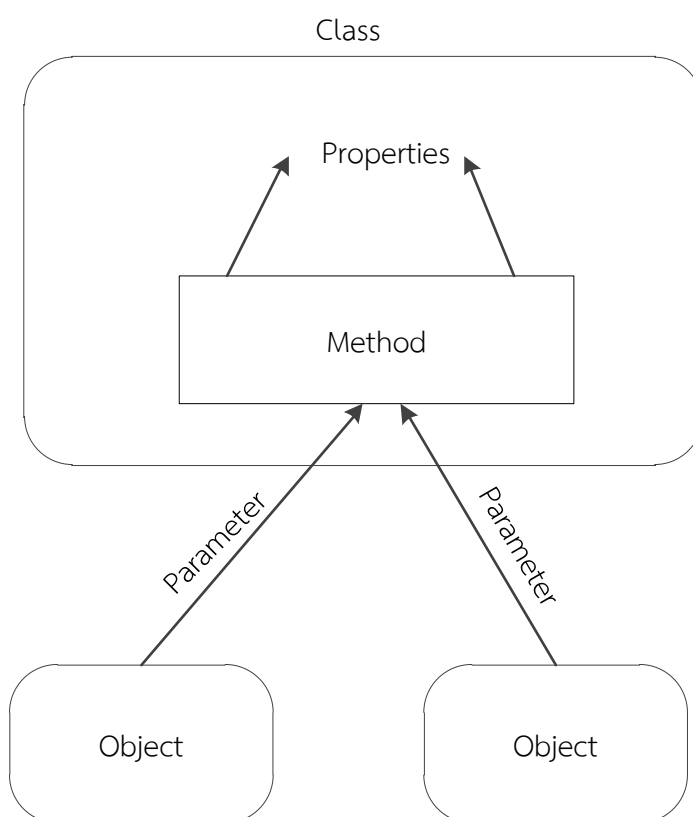


## บทที่ 7

### เมธอด (Method) แบบรับค่า พารามิเตอร์ (Parameter)

ในการทำงานของโปรแกรมแบบเชิงวัตถุอีกแบบหนึ่งก็คือ การส่งผ่านค่าระหว่าง  
ออบเจ็กต์กับคลาส ผ่านเมธอด ซึ่งจะแบ่งลักษณะการทำงานได้ดังนี้



#### 7.1 หลักการส่งผ่านค่าพารามิเตอร์ในเมธอด

ในการส่งผ่านค่าพารามิเตอร์นั้น จะกระทำในส่วนของเมธอดของคลาส ซึ่งการส่งผ่านค่าแบบนี้จะสามารถส่งผ่านค่าได้ต้องเป็นเมธอดที่เป็นแบบ Public เท่านั้น ดังตัวอย่าง ตารางที่ 7.1

ตารางที่ 7.1 การส่งผ่านค่าพารามิเตอร์

1	<?
2	class Paramet

3	{
4	public function setValue(\$name, \$surname)
5	{
6	echo \$name." ".\$surname;
7	}
8	}
9	\$obj = new Paramet();
10	\$obj->setValue("กอเอี๋ย","กอไ้");
11	?>

### ผลลัพธ์

กอเอี๋ย กอไ้

ในการส่งผ่านค่านี้้น เมื่อมีการกำหนดพารามิเตอร์ในการส่งผ่านค่า (บรรทัดที่ 4) มีจำนวนเท่าใด เมื่อส่งผ่านค่าจากออบเจ็กต์จะต้องมีจำนวนเท่ากัน (บรรทัดที่ 10) ถ้ามีจำนวนไม่เท่ากัน จะได้ผลดังตัวอย่างตารางที่ 7.2

ตารางที่ 7.2 การส่งผ่านค่าพารามิเตอร์ไม่เท่ากัน

1	<?
2	class Paramet
3	{
4	public function setValue(\$name,\$surname)
5	{
6	echo \$name." ".\$surname;
7	}
8	}
9	\$obj = new Paramet();
10	\$obj->setValue("กอเอี๋ย");
11	?>

## ผลลัพธ์ของโปรแกรม

**Warning:** Missing argument 2 for Paramet::setValue(), called in C:\AppServ\www\oop\Paramet.php on line 18 and defined in C:\AppServ\www\oop\Paramet.php on line 12  
 กอเอ๋ย

ดังนั้น เพื่อแก้ไขปัญหาที่เกิดขึ้น โดยให้กำหนดพารามิเตอร์เริ่มต้นเอาไว้ (เพราะบางที่ อาจมี ออบเจ็กต์หลายตัวเข้ามาใช้งาน ดังนั้นควรกำหนดค่าเริ่มต้นให้กับพารามิเตอร์ดีกว่า) ดังตัวอย่าง ตารางที่ 7.3

ตารางที่ 7.3 การกำหนดพารามิเตอร์เริ่มต้นให้กับเมธอด

1	<?
2	class Paramet
3	{
4	public function setValue(\$name="", \$surname="")
5	{
6	echo \$name." ".\$surname;
7	}
8	}
9	\$obj = new Paramet();
10	\$obj->setValue("กอเอ๋ย");
11	\$obj2 = new Paramet();
12	\$obj2->setValue("ขอไข่", "ในเล้า");
13	\$obj3 = new Paramet();
14	\$obj3->setValue();
15	?>

## ผลลัพธ์

กอเอ๋ย ขอไข่ ในเล้า

จากตัวอย่างจะเห็นได้ว่า เมื่อมีการกำหนดค่าเริ่มต้นให้กับพารามิเตอร์ (บรรทัดที่ 4) เมื่อมีการสร้างออบเจ็กต์ขึ้นมาหลายๆ ตัว ก็สามารถเรียกใช้ผ่านพารามิเตอร์ได้เลย ซึ่งอาจจะมีทั้งที่ส่ง

พารามิเตอร์ไม่ครบ (บรรทัดที่ 10) และครบ (บรรทัดที่ 12) หรือไม่มีการส่งเลย (บรรทัดที่ 14) ก็จะไม่แสดงผลต่อเมธอดนั้นๆ

ในการเรียกใช้พารามิเตอร์ผ่านเมธอดต่างๆ นั้น จะมีการเรียกใช้ผ่านออบเจกต์หลายๆ ตัว ดังนั้นออบเจกต์แต่ละตัวจะต้องมีการส่งข้อมูลผ่านพารามิเตอร์ของตนเอง เพราะถ้าไม่มีข้อมูลส่งผ่านพารามิเตอร์ ผลที่ได้เมื่อมีการเรียกใช้เมธอด ข้อมูลที่ได้จะเป็นของค่าพารามิเตอร์ของออบเจกต์ก่อนหน้าที่มีการส่งผ่านเข้าไปในเมธอด ดังตัวอย่างตารางที่ 7.4

ตารางที่ 7.4 ออบเจกต์ที่ไม่ส่งผ่านพารามิเตอร์

1	<?
2	class Paramet
3	{
4	public function setValue(\$name="", \$surname="")
5	{
6	echo \$name." ".\$surname;
7	}
8	}
9	\$obj = new Paramet();
10	\$obj->setValue("กอเอี๋ย");
11	\$obj2 = new Paramet();
12	\$obj2->setValue();
13	?>

ผลลัพธ์

กอเอี๋ย

## 7.2 การส่งผ่านค่าในพรอพเพอร์ตี้ของคลาส

การส่งผ่านแบบนี้ จะต้องเข้าใจการเข้าถึงของคลาส (กล่าวมาแล้วในบทที่ผ่านมา) ซึ่งการเข้าถึงจะแบ่งสิทธิ์เข้าไปใช้งานภายในคลาสดังตัวอย่างตารางที่ 7.5

ตารางที่ 7.5 แสดงการเข้าถึงคลาส

1	<?
2	class Paramet

```

3  {
4      public $g_name;
5      private $g_surname;
6
7      public function setValue($name="", $surname="")
8      {
9          $this->g_name=$name;
10         $this->g_surname=$surname;
11     }
12     public function getValue()
13     {
14         return $this->g_name." ".$this->g_surname;
15     }
16 }
17 $obj = new Paramet();
18 $obj->setValue("กอเอ๋อ", "กอไก่");
19 echo $obj->g_name;
20 echo $obj->g_surname;
21 ?>

```

ผลลัพธ์

กอเอ๋อ

**Fatal error:** Cannot access private property Paramet::\$g\_surname  
in C:\AppServ\www\oop\Paramet.php on line 20

จากตัวอย่าง ตารางที่ 7.5 แสดงให้เห็นผลลัพธ์ของโปรแกรม การเข้าถึง Properties แบบ private ซึ่งปกติแล้วจะไม่สามารถเรียกใช้ หรือเปลี่ยนค่าโดยตรงได้ แต่ถ้าต้องการเรียกดู หรือเปลี่ยนแปลงค่า จะต้องให้เมธอดที่อยู่ในคลาสเป็นตัวกระทำเท่านั้น จึงจะทำได้

ตารางที่ 7.5 การส่งผ่านค่าพารามิเตอร์ผ่านพรอพเพอร์ตี้ของคลาส

1	<?
2	class Paramet
3	{
4	private \$g_name;
5	private \$g_surname;
6	
7	public function setValue(\$name="", \$surname="")
8	{
9	\$this->g_name=\$name;
10	\$this->g_surname=\$surname;
11	}
12	public function getValue()
13	{
14	return \$this->g_name." ".\$this->g_surname;
15	}
16	}
17	\$obj = new Paramet();
18	\$obj->setValue("กอเอ๋อ", "กอไก่");
19	echo \$obj->getValue();
20	?>

**ผลลัพธ์**

กอเอ๋อ กอไก่

จากตารางที่ 7.5 จะเห็นได้ว่าการส่งผ่านค่าพารามิเตอร์ไปยังพรอพเพอร์ตี้ที่มีอยู่ในคลาส แต่การส่งผ่านค่าจากออบเจกต์ไปยังพรอพเพอร์ตี้โดยตรงไม่ได้ เพราะจากตัวอย่างพรอพเพอร์ตี้ของคลาสเป็นแบบ private (บรรทัดที่ 4-5) ซึ่งจะไม่ยอมให้ออบเจกต์ หรือคลาสอื่นเข้ามาใช้งานได้ โดยอนุญาตใช้งานภายในคลาสของตนเองเท่านั้น แต่ถ้าต้องการส่งผ่านพารามิเตอร์จากออบเจกต์ไปยังพรอพเพอร์ตี้ จะต้องดำเนินการผ่านเมธอดที่อยู่ภายในคลาสเท่านั้น โดยการส่งผ่านพารามิเตอร์ (บรรทัดที่ 7) แล้วให้ return ค่าที่ได้ออกมาจากพรอพเพอร์ตี้