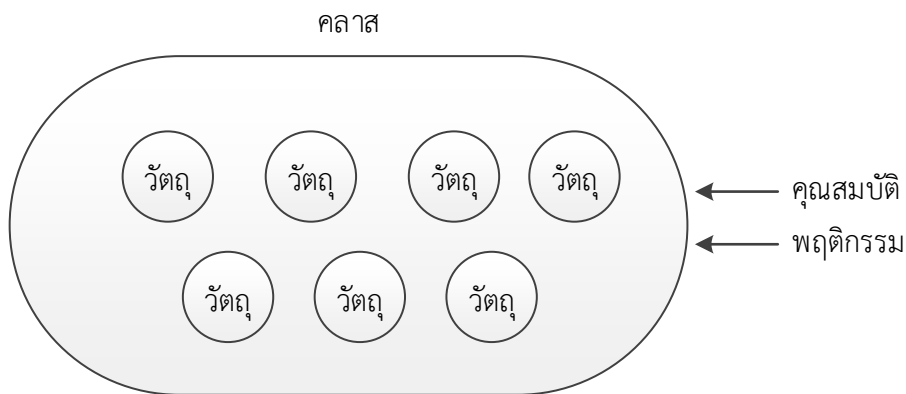


บทที่ 4

พื้นฐานการเขียนโปรแกรมเชิงวัตถุ

จากเนื้อหาบทที่ 1 ที่กล่าวมา ได้ทราบถึงวิวัฒนาการของโปรแกรมภาษาคอมพิวเตอร์ ซึ่งจะมีหลักการ และแนวคิดที่แตกต่างกันตามยุคสมัยเริ่มแรกมาถึงปัจจุบัน เนื้อหาในบทเรียนนี้จะกล่าวถึงหลักการทำงานพื้นฐานและแนวคิดเชิงวัตถุ และนำแนวคิดนี้ไปสู่การพัฒนาโปรแกรมแบบเชิงวัตถุ หรือที่เรียกว่า “OOP (Object Oriented Programing)” โดยแนวคิดแบบเชิงวัตถุ นั้นจะมีโครงสร้างที่สำคัญประกอบไปด้วย 4 ส่วน หลักๆ คือ คลาส วัตถุ คุณสมบัติ และพฤติกรรม ดังรูปที่ 4.1



รูปที่ 4.1 ส่วนประกอบของเชิงวัตถุ

จากรูปที่ อธิบายให้เห็นภาพส่วนประกอบของเชิงวัตถุ ประกอบไปด้วยคลาส ในตัวของคลาสนั้นจะมีองค์ประกอบภายในตัวอยู่ 2 องค์ประกอบ คือ คุณสมบัติ และพฤติกรรม ซึ่งวัตถุไหนมีคุณสมบัติ หรือพฤติกรรมเดียวกับคลาส ก็ถือว่าเป็นวัตถุที่อยู่ในคลาสดังกล่าว

4.1 คลาส (Class)

คลาส หมายถึง การให้นิยามความหมาย และคำจำกัดความกลุ่มของวัตถุกลุ่มใดกลุ่มหนึ่ง ในที่นี้ ยกตัวอย่างเช่น ถ้ากล่าวถึงคลาสมนุษย์ เมื่อมองในความหมายขอบเขตของมนุษย์ ก็จะทำให้ความหมายว่าเป็นลักษณะของสิ่งมีชีวิตที่เป็นรูปธรรม เช่น แขน ขา หู ตา จมูก เป็นต้นหรือคุณสมบัติ ด้านนามธรรมภายในตัวของมนุษย์ เช่น ชื่อ รหัสประจำตัวประชาชน ที่อยู่ อายุ เพศ น้ำหนัก ส่วนสูง เป็นต้น ในส่วนนี้จะมองในลักษณะของคลาสนั้นบ่งบอกลักษณะภายในตัวของคลาสดังกล่าว และส่วนที่สอง เป็นส่วนของการมองในส่วนของคลาสนั้นสามารถเกิด “พฤติกรรม (behavior)” ได้จากตัวอย่าง คลาสของมนุษย์ สามารถที่จะกระทำ หรือแสดงพฤติกรรมออกมาได้ เช่น วิ่ง พุด แก้วแขน กระโดด เป็นต้น ซึ่งเมื่อมองในความหมายภาพรวมเห็นได้ว่าเป็นลักษณะของมุมมองที่เป็นนามธรรม ที่กำหนดขอบเขตของกลุ่มใดกลุ่มหนึ่ง

ในการเขียนคลาส ของภาษาโปรแกรมเชิงวัตถุ php เมื่อต้องการจะดำเนินการสร้างคลาสในโปรแกรม จะต้องใช้คำสั่ง “class” ตามด้วยชื่อของคลาส และภายในของคลาสนั้น จะประกอบด้วยกลุ่มของคำสั่งหรือประโยคคำสั่ง ซึ่งจะต้องอยู่ภายใต้เครื่องหมาย “{” และเครื่องหมาย “}” ดังรูปที่ 4.2

```
class ชื่อคลาส
{
    properties;
    behavior;
}
```

รูปที่ 4.2 โครงสร้างของการ class

ข้อควรระมัดระวังในการตั้งชื่อของคลาสนั้น สิ่งที่จะต้องพิจารณาในการตั้งชื่อคลาสนั้น นั่นคือตั้งชื่อให้สอดคล้องกับการทำงานของคลาสนั้นๆ เพื่อใช้ในการจดจำ และตรวจสอบได้ง่ายยิ่งขึ้น แต่การตั้งชื่อคลาสนั้น ต้องพิจารณาโดยมีข้อห้ามอย่าให้ไปซ้ำกับ “ชื่อสงวน (reserved word)” ที่มีใช้ในโปรแกรมภาษา php ดังตารางที่ 4.1

ตารางที่ 4.1 คำสงวนของภาษาโปรแกรม PHP

คำสงวน				
<code>_halt_compiler()</code>	<code>abstract</code>	<code>and</code>	<code>array()</code>	<code>as</code>
<code>break</code>	<code>callable (as of PHP 5.4)</code>	<code>case</code>	<code>catch</code>	<code>class</code>
<code>clone</code>	<code>const</code>	<code>continue</code>	<code>declare</code>	<code>default</code>
<code>die()</code>	<code>do</code>	<code>echo</code>	<code>else</code>	<code>elseif</code>
<code>empty()</code>	<code>enddeclare</code>	<code>endfor</code>	<code>endforeach</code>	<code>endif</code>
<code>endswitch</code>	<code>endwhile</code>	<code>eval()</code>	<code>exit()</code>	<code>extends</code>
<code>final</code>	<code>for</code>	<code>foreach</code>	<code>function</code>	<code>global</code>
<code>goto (PHP 5.3)</code>	<code>if</code>	<code>implements</code>	<code>include</code>	<code>include_once</code>
<code>instanceof</code>	<code>insteadof (มีอยู่ใน PHP 5.4)</code>	<code>interface</code>	<code>isset()</code>	<code>list()</code>
<code>namespace (PHP 5.3)</code>	<code>new</code>	<code>or</code>	<code>print</code>	<code>private</code>
<code>protected</code>	<code>public</code>	<code>require</code>	<code>require_once</code>	<code>return</code>
<code>static</code>	<code>switch</code>	<code>throw</code>	<code>trait (PHP 5.4)</code>	<code>try</code>
<code>unset()</code>	<code>use</code>	<code>var</code>	<code>while</code>	<code>xor</code>

หลักของการตั้งชื่อคลาสที่ถูกต้อง จะต้องใช้อักขระเริ่มต้นด้วยตัวอักษร หรือเครื่องหมาย underscore (_) และตามด้วย ตัวเลข ตัวอักษร หรือ underscores ดังตัวอย่างต่อไปนี้ _a-zA, a-zA-Z_, x7f-, xff)[a-zA-Z0-9_, x7f-, xff เป็นต้น แต่ในหลักของการตั้งชื่อคลาส ควรตั้งชื่อให้สอดคล้องกับคลาสที่ผู้ใช้งานต้องการกำหนดขึ้น

4.2 คุณสมบัติ (Properties)

คุณสมบัติ หมายถึง ส่วนที่เป็นคุณลักษณะ ข้อกำหนด ของคลาสนั้นมี ซึ่งจากที่ได้ยกตัวอย่าง เรื่องคลาสมนุษย์ มนุษย์จะมีคุณสมบัติภายในคลาสของมนุษย์นั้นคือ แขน ขา ชื่อ รหัสบัตร ที่อยู่ อายุ เพศ เป็นต้น

ในการเขียนโปรแกรมเชิงวัตถุ ด้วยภาษาโปรแกรม php สามารถสร้างคุณสมบัติของคลาสได้ ดังรูปที่ 4.3

```
class human
{
    public $name;
    public $sex= 'ชาย';
    private $age;
}
```

รูปที่ 4.3 การกำหนด properties ภายในคลาส

จากหลักการของคลาสที่ได้มาในข้างต้น เมื่อนำมาพัฒนาโปรแกรม มุมมองของคำสั่งที่เกิดขึ้น จะใช้คำศัพท์ที่แตกต่างกันออกไป ดังนี้คือ คลาสจะประกอบไปด้วย คุณสมบัติ ซึ่งในการพัฒนา โปรแกรมจะเรียกว่า “ตัวแปร Variable” คือ ค่าที่กำหนดขึ้นอ้างอิงกับหน่วยความจำ เพื่อใช้ในการ เก็บข้อมูล ซึ่งตัวแปรจะใช้เครื่องหมาย “\$” ไว้ข้างหน้าชื่อของตัวแปรนั้นๆ และค่าคงที่ (constants) หมายถึง ค่าที่มีค่าแน่นอน เช่นในตัวอย่างนี้ คุณสมบัติของมนุษย์ที่เป็นผู้ชาย ซึ่งจากตัวอย่างก็จะหมายถึง คลาสของมนุษย์ที่เป็นผู้ชายเท่านั้น

4.3 พฤติกรรม (Method)

นอกเหนือจากคุณสมบัติในข้างต้นแล้ว อีกส่วนหนึ่งมีอยู่ภายในคลาสนั้นก็คือ พฤติกรรม ในการเขียนโปรแกรมเชิงวัตถุ จะเรียกพฤติกรรมว่า “เมธอด” (Method) ซึ่งเมธอดนี้จะเก็บการทำงานของ คลาส หรือเรียกว่า “ฟังก์ชัน” (functions) ต่างๆ เอาไว้ดังรูปที่ 4.4

```
<?php
class human
{
```

```

    public $name;
    public $sex= 'ชาย';
    private $age;
    public function run()
    {
        echo "วิ่ง";
    }
    public function showName ()
    {
        echo $this->name;
    }
}
?>

```

รูปที่ 4.4 การกำหนด properties ภายในคลาส

คำสั่ง \$this จะถูกใช้งานในเมธอด ที่อยู่ภายในคลาส โดยเป็นการอ้างอิงไปยังค่าที่อยู่ใน properties ซึ่งจะสามารถใช้งานภายในคลาสได้เท่านั้น ไม่สามารถเรียกผ่านออบเจกต์ หรือภายนอกคลาสได้ ซึ่งถือว่าค่าต่างๆ ที่อยู่ในคลาสมีความปลอดภัย เพราะสามารถใช้งานได้เฉพาะภายในคลาสของตนเองเท่านั้น

4.4 วัตถุ (Object)

ออบเจกต์ เป็นสิ่งที่มีตัวตน อยู่ในคำจำกัดความของคลาส หรือจากที่ได้ทราบมาแล้วว่า ออบเจกต์ จะเกิดจากคลาส หลังจากที่ได้มีการสร้างคลาสเสร็จเรียบร้อยแล้ว เมื่อต้องการสร้างวัตถุ นั้น วิธีการในการเขียนโปรแกรมเชิงวัตถุจะใช้คำสั่ง “new” เป็นคำสั่งที่ใช้ในการสร้างวัตถุ ถ้าต้องการให้เกิดจากคลาสไหน ก็ต้องตามด้วยชื่อของคลาสนั้นๆ ดังตัวอย่างรูปที่ 4.3

```

<?php
class human
{
    public $name;
    public $sex= 'ชาย';
    private $age;

    public function run()
    {
        echo "วิ่ง";
    }
}

```

```

    }
    public function jump()
    {
        echo "กระโดด";
    }
    public function showName()
    {
        echo $this->name;
    }
}
$obj1=new human();
$obj1->sex;
$obj1->showName();

```

รูปที่ 4.5 การสร้างออบเจ็กต์

คำสั่ง “new” เป็นคำสั่งที่กำหนดให้มีการสร้างออบเจ็กต์ หรือเรียกอีกอย่างว่าตัว “อินสแตนท์ (Instance)” ซึ่งเมื่อออบเจ็กต์ก็จะได้รับการถ่ายทอด properties และ function มาด้วยพร้อมกัน ออบเจ็กต์ก็สามารถที่จะเรียกใช้ทั้งสองสิ่งนี้ได้ทันที

ในการสร้างออบเจ็กต์ สามารถสร้างได้จำนวนมากมาตามต้องการที่จะใช้งาน ซึ่งเมื่อสร้างออบเจ็กต์มาแล้วนั้น คุณลักษณะที่บ่งบอกถึงความแตกต่างระหว่างออบเจ็กต์ นั่นก็คือข้อมูลที่อยู่ในคุณสมบัติในแต่ละออบเจ็กต์นั้นๆ เช่น ในคลาสของนักศึกษา วัตถุก็คือคนที่เป็นนักศึกษา ซึ่งนักศึกษาที่เป็นวัตถุนั้นๆ จะมีคุณสมบัติของชื่อที่แตกต่างกันออกไป นักศึกษาคนที่ 1 ชื่อนายสมบัติ นักศึกษาคนที่ 2 ชื่อ สุภาพร นักศึกษาคนที่ 3 ชื่อวิชัย เป็นต้น ดังตัวอย่างรูปภาพที่ 4.6

```

<?php
class human
{
    public $name;
    public function run()
    {
        echo "สามารถวิ่งได้";
    }
}
$obj1 = new human();
$obj1 -> name="สมบัติ";
$obj2 = new human();
$obj2 -> name="สุภาพร";

```

```
$obj3 = new human();
$obj3 -> name="วิชัย";
?>
```

รูปที่ 4.6 การเรียกใช้ properties และ function

สำหรับออบเจ็กต์ที่ต้องการอ้างอิงไปยังคุณสมบัติ หรือการกระทำนั้น จะให้สัญลักษณ์ “->” (ลักษณะคล้ายหัวลูกศร) ในการอ้างอิงจากออบเจ็กต์

เพื่อให้เข้าใจหลักการทำงานให้มากกว่านี้ จะขอยกตัวอย่างการเขียนโปรแกรมตามหลักการเชิงวัตถุดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 4.1 สัตว์ประกอบไปด้วยคุณสมบัติ ชื่อของสัตว์ ประเภทของสัตว์ และสถานที่อยู่อาศัย และสัตว์นั้นสามารถอาศัยอยู่ได้ทั้งบก ในน้ำ และทั้งบนบกและในน้ำ จงเขียนโปรแกรมแบบเชิงวัตถุตามนิยามดังกล่าว โดยให้ยกตัวอย่างออบเจ็กต์มา 3 ออบเจ็กต์

จากนิยามดังกล่าวสามารถแยกออกเป็น คลาส คุณสมบัติ พฤติกรรม และออบเจ็กต์ ดังต่อไปนี้

ตารางที่ 4.2 นิยามความหมายจากโจทย์ตัวอย่างที่ 4.1

ชื่อ	ความหมาย
animal	คลาส (Class)
name group live	คุณสมบัติ (Properties)
land amphibian aquatic	พฤติกรรม (Method)
elephant crocodile shark	ออบเจ็กต์ (ช้าง) ออบเจ็กต์ (จระเข้) ออบเจ็กต์ (ฉลาม)

รูปที่ 4.7 นิยามความหมายจากโจทย์ที่ 4.1

คำจำกัดความจากตารางดังกล่าว สามารถนำมาเขียนโปรแกรมเชิงวัตถุตามความหมายจากโจทย์ที่กำหนดให้ดังต่อไปนี้

ตารางที่ 4.3 การสร้างชุดคำสั่งจากโจทย์ตัวอย่างที่ 4.1

แถวที่	ชุดคำสั่ง
1	<?php
2	class animal

3	{
4	public \$name;
5	public \$group;
6	public \$live;
7	public function land()
8	{
9	echo "อาศัยอยู่บนบกเท่านั้น";
10	}
11	public function amphibian()
12	{
13	echo "อาศัยได้ทั้งในบกและในน้ำ";
14	}
15	public function aquatic()
16	{
17	echo "อาศัยอยู่ในน้ำ";
18	}
19	}
20	\$elephant = new animal();
21	echo \$elephant->name="ช้าง ";
22	echo \$elephant->group=" สัตว์บก";
23	echo \$elephant->live=" ไทย, อินเดีย ";
24	\$elephant->land();
25	echo " ----- ";
26	\$crocodile = new animal();
27	echo \$crocodile->name="จระเข้ ";
28	echo \$crocodile->group=" สัตว์เลื้อยคลาน";
29	echo \$crocodile->live=" อินเดีย, อเมริกา, บราซิล ";
30	\$crocodile->amphibian();
31	echo " ----- ";
32	\$shark = new animal();
33	echo \$shark->name="ฉลาม";
34	echo \$shark->group=" สัตว์น้ำ";
35	echo \$shark->live=" ออสเตรเลีย, แอฟริกาใต้ ";
36	\$shark->aquatic();
37	?>

จากโปรแกรมตัวอย่าง สามารถอธิบายส่วนของการทำงานต่างๆ ตามโจทย์ที่กำหนดให้ ได้ดังต่อไปนี้

ตารางที่ 4.4 อธิบายชุดคำสั่งจากโจทย์ตัวอย่างที่ 4.1

บรรทัดที่	คำอธิบาย
2	สร้างคลาสชื่อ animal
4-6	สร้างพรีอเพอร์เตอร์ 3 ตัว ได้แก่ \$name \$group \$live
7-18	สร้างเมธอด 3 ตัว ได้แก่ land() amphibian() และ aquatic()
20, 26, 32	สร้างออบเจ็ค 3 ตัว ได้แก่ \$elephant \$crocodile และ \$shark (พรีอเพอร์เตอร์ และออบเจ็ค จะใช้ \$ ช้างหน้าเหมือนกัน เพราะบ่งบอกว่าตัวนั้นมีข้อมูลในตัวเอง)
21-23	การเรียกใช้พรีอเพอร์เตอร์ ที่อยู่ในคลาส และนำข้อมูลเข้าไปในพรีอเพอร์เตอร์ของออบเจ็ค \$elephant
24	การเรียกใช้เมธอด land() ของออบเจ็ค \$elephant
26-29	การเรียกใช้พรีอเพอร์เตอร์ ที่อยู่ในคลาส และนำข้อมูลเข้าไปในพรีอเพอร์เตอร์ของออบเจ็ค \$crocodile
30	การเรียกใช้เมธอด amphibian() ของออบเจ็ค \$crocodile
33-35	การเรียกใช้พรีอเพอร์เตอร์ ที่อยู่ในคลาส และนำข้อมูลเข้าไปในพรีอเพอร์เตอร์ของออบเจ็ค \$shark
36	การเรียกใช้เมธอด aquatic () ของออบเจ็ค \$shark

เมื่อทำการประมวลผลชุดคำสั่งจากโจทย์ตัวอย่างที่ 4.1 ด้วยการเขียนแบบโครงสร้างเชิงวัตถุ จะได้ผลลัพธ์ดังต่อไปนี้

ช้าง สัตว์บก ไทย,อินเดีย อาศัยอยู่บนบกเท่านั้น ----- จระเข้ สัตว์เลื้อยคลาน อินเดีย, อเมริกา, บราซิล อาศัยได้ทั้งในบกและในน้ำ ----- ฉลาม สัตว์น้ำ ออสเตรเลีย, แอฟริกาใต้ อาศัยอยู่ในน้ำ
--

รูปที่ 4.6 ผลลัพธ์จากชุดคำสั่งโจทย์ตัวอย่างที่ 4.1

ตัวอย่างที่ 4.2 รถเก๋งประกอบไปด้วยคุณสมบัติ ทะเบียนรถ สี จำนวนประตู ล้อ และรถยนต์นั้นสามารถบรรทุกของหนักได้และไม่ได้ จงเขียนโปรแกรมแบบเชิงวัตถุนำนิยามดังกล่าว โดยให้ยกตัวอย่างออบเจ็กต์มา 2 ออบเจ็กต์

จากคำนิยามดังกล่าวสามารถแยกออกเป็น คลาส คุณสมบัติ พฤติกรรม และออบเจ็กต์ดังต่อไปนี้

ตารางที่ 4.5 นิยามความหมายจากโจทย์ตัวอย่างที่ 4.2

ชื่อ	ความหมาย
Car	คลาส (Class)
Kind door wheel	คุณสมบัติ (Properties)
no_carry carry	พฤติกรรม (Method)
car1 car2	ออบเจ็กต์ (รถเก๋ง) ออบเจ็กต์ (รถกระบะ)

รูปที่ 4.7 นิยามความหมายจากโจทย์ที่ 4.2

คำจำกัดความจากตารางดังกล่าว สามารถนำมาเขียนโปรแกรมเชิงวัตถุตามความหมายจากโจทย์ที่กำหนดให้ได้ดังต่อไปนี้

ตารางที่ 4.6 การสร้างชุดคำสั่งจากโจทย์ตัวอย่างที่ 4.2

แถวที่	ชุดคำสั่ง
1	<?php
2	class animal
3	{
4	public \$name;
5	public \$group;
6	public \$live;
7	public function land()
8	{
9	echo "อาศัยอยู่บนบกเท่านั้น";
10	}
11	public function amphibian()

12	{
13	echo "อาศัยได้ทั้งในบกและในน้ำ";
14	}
15	public function aquatic()
16	{
17	echo "อาศัยอยู่ในน้ำ";
18	}
19	}
20	\$elephant = new animal();
21	echo \$elephant->name="ช้าง";
22	echo \$elephant->group="สัตว์บก";
23	echo \$elephant->live="ไทย, อินเดีย ";
24	\$elephant->land();
25	echo " ----- ";
26	\$crocodile = new animal();
27	echo \$crocodile->name="จระเข้";
28	echo \$crocodile->group="สัตว์เลื้อยคลาน";
29	echo \$crocodile->live="อินเดี๋ย, อเมริกา, บราซิล ";
30	\$crocodile->amphibian();
31	echo " ----- ";
32	\$shark = new animal();
33	echo \$shark->name="ฉลาม";
34	echo \$shark->group="สัตว์น้ำ";
35	echo \$shark->live="ออสเตรเลีย, แอฟริกาใต้ ";
36	\$shark->aquatic();
37	?>

จากโปรแกรมตัวอย่าง สามารถอธิบายส่วนของการทำงานต่างๆ ตามโจทย์ที่กำหนดให้ ได้ดังต่อไปนี้

ตารางที่ 4.7 อธิบายชุดคำสั่งจากโจทย์ตัวอย่างที่ 4.1

บรรทัดที่	คำอธิบาย
2	สร้างคลาสชื่อ animal
4-6	สร้างพร็อพเพอร์ตี้ 3 ตัว ได้แก่ \$name \$group \$live
7-18	สร้างเมธอด 3 ตัว ได้แก่ land() amphibian() และ aquatic()
20, 26, 32	สร้างออบเจ็ค 3 ตัว ได้แก่ \$elephant \$crocodile และ \$shark (พร็อพเพอร์ตี้ และออบเจ็ค จะใช้ \$ ช้างหน้าเหมือนกัน เพราะบ่งบอกว่าตัวนั้น

	มีข้อมูลในตัวเอง)
21-23	การเรียกใช้พร็อพเพอร์ตี้ ที่อยู่ในคลาส และนำข้อมูลเข้าไปในพร็อพเพอร์ตี้ของ ออบเจ็ค \$elephant
24	การเรียกใช้เมธอด land() ของออบเจ็ค \$elephant
26-29	การเรียกใช้พร็อพเพอร์ตี้ ที่อยู่ในคลาส และนำข้อมูลเข้าไปในพร็อพเพอร์ตี้ของ ออบเจ็ค \$crocodile
30	การเรียกใช้เมธอด amphibian() ของออบเจ็ค \$crocodile
33-35	การเรียกใช้พร็อพเพอร์ตี้ ที่อยู่ในคลาส และนำข้อมูลเข้าไปในพร็อพเพอร์ตี้ของ ออบเจ็ค \$shark
36	การเรียกใช้เมธอด aquatic () ของออบเจ็ค \$shark

เมื่อทำการประมวลผลชุดคำสั่งจากโจทย์ตัวอย่างที่ 4.1 ด้วยการเขียนแบบโครงสร้างเชิงวัตถุ
จะได้ผลลัพธ์ดังต่อไปนี้

ช้าง
สัตว์บก
ไทย,อินเดีย
อาศัยอยู่บนบกเท่านั้น

จระเข้
สัตว์เลื้อยคลาน
อินเดีย, อเมริกา, บราซิล
อาศัยได้ทั้งในบกและในน้ำ

ฉลาม
สัตว์น้ำ
ออสเตรเลีย, แอฟริกาใต้
อาศัยอยู่ในน้ำ

รูปที่ 4.6 ผลลัพธ์จากชุดคำสั่งโจทย์ตัวอย่างที่ 4.2