

บทที่ 1

ความรู้เบื้องต้นเกี่ยวกับโปรแกรมเชิงวัตถุ

ปัจจุบันนี้ คอมพิวเตอร์มีส่วนอย่างมากที่มาเกี่ยวข้องกับชีวิตของมนุษย์แทบทุกคน จนแทบจะแยกจากกันไม่ได้ ซึ่งความเกี่ยวข้องนั้น อาจจะส่งผลกระทบโดยตรง หรือไม่ก็มีผลกระทบโดยทางอ้อม ซึ่งผลกระทบที่ได้อาจมีทั้งข้อดี และข้อเสีย ดังนั้นมนุษย์จะต้องเรียนรู้ และนำเอาเทคโนโลยีที่มีอยู่มาใช้ในการชีวิตของตนเองให้มีคุณค่าและประโยชน์มากที่สุด

ในเนื้อหาในบทนี้ จะกล่าวถึงวิวัฒนาการ และลักษณะของกระบวนการในการพัฒนาโปรแกรม (Coding) ซึ่งมีแนวคิดแต่ละวิวัฒนาการไม่เหมือนกัน ซึ่งสามารถแบ่งยุคของการพัฒนาโปรแกรม ได้เป็น 3 ยุค คือ การพัฒนาโปรแกรมแบบไม่มีโครงสร้าง การพัฒนาโปรแกรมแบบมีโครงสร้าง และการพัฒนาโปรแกรมแนวคิดแบบเชิงวัตถุ

1.1 การพัฒนาโปรแกรมแบบไม่มีโครงสร้าง (Non-Structural Programming)

ในการพัฒนาโปรแกรมแบบไม่มีโครงสร้าง เป็นยุคเริ่มแรกของการสร้างชุดคำสั่ง ในการสั่งงานคอมพิวเตอร์ การสั่งงานคอมพิวเตอร์จะใช้สั่งโดยตรง ซึ่งยุคเริ่มแรกจะมีการใช้คอมพิวเตอร์ยังไม่หลากหลาย ลักษณะของการใช้งานยังเป็นลักษณะของภาษาเครื่อง และภาษาสัญลักษณ์(มีใช้ในภาษาแอสเซมบลี)

	00000000	00000100	0000000000000000
01011110	00001100	11000010	0000000000000010
	11101111	00010110	0000000000000101
	11101111	10011110	0000000000001011
11111000	10101101	11011111	0000000000010010
	01100010	11011111	0000000000010101
11101111	00000010	11111011	0000000000010111
11110100	10101101	11011111	0000000000011110
00000011	10100010	11011111	0000000000100001
11101111	00000010	11111011	0000000000100100
01111110	11110100	10101101	
11111000	10101110	11000101	0000000000101011
00000110	10100010	11111011	0000000000110001
11101111	00000010	11111011	0000000000110100
	01010000	11010100	0000000000111011
		00000100	0000000000000001

รูปที่ 1.1 การสร้างชุดคำสั่งด้วยภาษาเครื่อง

จากรูปที่ 1.1 แสดงการสร้างชุดคำสั่งในการสั่งงานคอมพิวเตอร์ เป็นโปรแกรมในการคูณตัวเลข 2 ตัวเลข ซึ่งจะเห็นได้ว่าการสั่งงานคอมพิวเตอร์ จะใช้เลขฐานสองทั้งหมด โดยการสั่งงานจะใช้คำสั่งที่ต้องการลงไปทันที

entry	main, ^m<r2>
subl2	#12, sp
jsb	c\$MAIN_ARGS
movab	\$CHAR_STRING_CON
pushal	-8(fp)
pushal	(r2)
pushal	-12(fp)
pushal	3(r2)
calls	#2, SCANF
mull3	-8(fp), -12(fp), -
pusha	6(r2)
calls	#2, PRINTF
clrl	r0
ret	

รูปที่ 1.2 การสร้างชุดคำสั่งด้วยภาษาสัญลักษณ์

ภาษาสัญลักษณ์มีลักษณะของการใช้คำสั่งคล้ายกับภาษาเครื่อง โดยเมื่อต้องการสั่งงานคอมพิวเตอร์ก็สามารถสร้างชุดคำสั่ง สั่งงานได้โดยตรง แต่ลักษณะของคำสั่งนั้นจะแตกต่างจากภาษาเครื่องนั้นคือ จะใช้สัญลักษณ์แทนชุดของเลขฐานสอง ทั้งนี้ชุดคำสั่งด้วยด้วยภาษาสัญลักษณ์แทนเลขฐานสอง ก็เพื่อสะดวกในการจดจำ และการใช้งาน ซึ่งจะสามารถใช้งานคอมพิวเตอร์ได้ง่ายขึ้น แต่ว่าชุดคำสั่งเหล่านี้จะไม่สั่งงานคอมพิวเตอร์ได้โดยตรง จะต้องไปผ่านกระบวนการแปลเป็นภาษาเครื่องเสียก่อน ภาษาที่มีลักษณะการทำงานแบบนี้ คือ ภาษาแอสเซมบลี (Assembly) ซึ่งจะใช้ตัวแปลภาษาแอสเซมเบลอ ทำการแปลให้เป็นภาษาเครื่องเสียก่อน จากรูปที่ 1.2 แสดงให้เห็นการใช้คำสั่งให้ประมวลผลหาผลลัพธ์จากการคูณของตัวเลข 2 จำนวน

1.2 การพัฒนาโปรแกรมแบบมีโครงสร้าง (Structural Programming)

การพัฒนาโปรแกรมแบบไม่มีโครงสร้างในข้างต้น จะพบว่าการใช้คำสั่งนั้น จะเป็นคำสั่งที่ใกล้เคียงกับภาษาคอมพิวเตอร์ ซึ่งภาษาเหล่านี้จะสามารถสื่อสารกับคอมพิวเตอร์ได้อย่างรวดเร็ว ถึงแม้จะมีการดำเนินการแปลชุดคำสั่ง แต่การประมวลผลก็ยังคงมีความรวดเร็วในการทำงาน ดังที่กล่าว

มานี้ เป็นข้อดีของการพัฒนาชุดคำสั่งแบบไม่มีโครงสร้าง แต่ในทางกลับกัน ในการสร้างชุดคำสั่งเหล่านี้ การป้อนชุดคำสั่ง (Input) จะถูกดำเนินการด้วยมนุษย์ ซึ่งการที่จะจดจำคำสั่งเหล่านี้เป็นเรื่องยากมากในการที่จะจดจำ ดังนั้นการใช้งานคอมพิวเตอร์ทั่วไปในยุคเริ่มแรก ก็จะมีเฉพาะกลุ่มบุคคลที่มีความเกี่ยวข้องโดยตรงกับคอมพิวเตอร์เท่านั้นที่จะใช้งานได้ แต่เมื่อวิวัฒนาการของระบบคอมพิวเตอร์ได้พัฒนาเพิ่มมากขึ้น มนุษย์มีการใช้งานคอมพิวเตอร์เพิ่มมากขึ้น จึงมีการคิดค้นวิธีการที่จะสั่งงานคอมพิวเตอร์ให้ได้ง่ายขึ้น มีการสร้างชุดคำสั่งที่มีภาษาใกล้เคียงกับภาษามนุษย์ ให้สามารถสั่งงานคอมพิวเตอร์ได้ง่ายมากยิ่งขึ้น ลักษณะการใช้งานแบบนี้ซึ่งเรียกว่า “ภาษาระดับสูง”

ด้วยภาษาระดับสูงนั้น มีการพัฒนาขึ้นมาใช้งานอย่างมากมาย และต่อเนื่อง ซึ่งถูกพัฒนาให้มีการใช้งานในแต่ละด้าน แตกต่างกันไป เช่น ด้านธุรกิจ ด้านการทหาร ด้านนักวิทยาศาสตร์ ด้านงานวิจัย เป็นต้น ซึ่งชุดคำสั่งเหล่านี้ถูกสร้างขึ้นโดยเฉพาะกลุ่ม วิธีการพัฒนา หรือชุดคำสั่งต่างๆ จะไม่เหมือนกันเพราะผู้พัฒนาต่างกัน แต่ทว่าโครงสร้างในการดำเนินการพัฒนาทุกโปรแกรมภาษาระดับสูง ก็จะมีลักษณะเหมือนกันนั่นก็คือ จะมีโครงสร้าง ระเบียบ กฎเกณฑ์ในการสร้างชุดคำสั่งเหล่านั้น ซึ่งลักษณะแบบนี้ คือ การพัฒนาโปรแกรมแบบมีโครงสร้าง ดังตัวอย่าง ให้หาผลลัพธ์จากการคูณของตัวเลข 2 จำนวน ทั้งสองตัวอย่าง

บรรทัดที่	ชุดคำสั่ง
1	#include <stdio.h>
2	main()
3	{
4	int num1,num2,result;
5	printf("Enter first number :");
6	scanf ("%d", &num1);
7	printf("Enter second number :");
8	scanf ("%d", &num2);
9	result = num1*num2;
10	printf("Result is %d",result);
11	return;
12	}

รูปที่ 1.3 การสร้างชุดคำสั่งแบบมีโครงสร้าง ด้วยภาษาซี

จากตัวอย่างรูปที่ 1.3 แสดงการพัฒนาชุดคำสั่งโดยโปรแกรมภาษาซี ซึ่งเป็นลักษณะของโปรแกรมภาษาแบบมีโครงสร้าง ซึ่งมีวิธีการ และกฎระเบียบในการพัฒนาชุดคำสั่งดังต่อไปนี้

ก่อนการสร้างชุดคำสั่งสั่งงานคอมพิวเตอร์ จะต้องมีการเรียกใช้ฟังก์ชันเพื่อใช้ต่างๆ ในโปรแกรมด้วย #include ตามด้วย ไลบรารี (Library) ที่เก็บคำสั่งต่างๆ เหล่านั้นไว้ เช่น คำสั่ง printf และ scanf จะถูกเก็บเอาไว้ในไลบรารี stdio.h

โปรแกรมหลัก (เมื่อมีการประมวลผลชุดคำสั่ง จะประมวลผลส่วนนี้เป็นส่วนแรก) จะต้องเขียนคำสั่ง main() และชุดคำสั่งที่สร้างขึ้นจะต้องอยู่ระหว่างสัญลักษณ์ “{“ และ “}” เท่านั้น

บรรทัดที่	ชุดคำสั่ง
1	Program multiply
2	uses crt;
3	var num1,num2,result : interger;
4	begin
5	clrscr;
6	result :=num1*num2;
7	writeln('Result is : ',result);
8	readln;
9	end.

รูปที่ 1.4 การสร้างชุดคำสั่งแบบมีโครงสร้าง ด้วยภาษาปาสคาล

เพื่อความเข้าใจที่ชัดเจนเพิ่มมากยิ่งขึ้น ในอีกตัวอย่างหนึ่งที่เป็นลักษณะของโปรแกรมแบบมีโครงสร้างนั่นก็คือ ภาษาปาสคาล ซึ่งการใช้งานภาษาก็จะมีลักษณะคล้ายๆ กับภาษาซี โดยการมีโครงสร้าง และกฎในการสร้างชุดคำสั่งดังนี้คือ

โปรแกรมที่สร้างขึ้นจะต้องมีชื่อของโปรแกรม โดยการตั้งชื่อจะต้องใส่ชื่อหลังคำสั่ง Program

ก่อนการสร้างชุดคำสั่งต่างๆ จะต้องมีการเรียกใช้คำสั่งในไลบรารีหลังคำสั่ง uses (บรรทัดที่ 2) ซึ่งโปรแกรมภาษาปาสคาล จะมีไลบรารี เหมือนกับภาษาซี ซึ่งเก็บฟังก์ชันการใช้งานต่างๆ ไว้ เมื่อมีการใช้งาน เช่น writeln (บรรทัดที่ 7) โปรแกรมภาษาก็จะไปเรียกคำสั่งนี้ที่ถูกเก็บไว้ในไลบรารี crt ขึ้นมาใช้งานได้ทันที

โปรแกรมหลัก จะต้องเริ่มต้นด้วยคำสั่ง begin และปิดชุดคำสั่ง ด้วยคำสั่ง end. ซึ่งคำสั่งนี้ในโปรแกรมภาษาจะต้องมีตัวเดียวเท่านั้น จะใช้ begin หลายๆ ครั้งไม่ได้ เพราะโปรแกรมหลักแต่ละโปรแกรมมีเพียงอันเดียวเท่านั้น (ภาษาซี ก็เหมือนกัน)

จากตัวอย่างโปรแกรมข้างต้น จะเห็นได้ว่าลักษณะของโปรแกรมภาษาแต่ละโปรแกรมภาษาไม่เหมือนกัน ขึ้นอยู่กับผู้พัฒนาซึ่งจะเป็นผู้คิดค้นต้นแบบขึ้นมา ผู้ใช้เมื่อจะใช้งานโปรแกรมภาษา

จะต้องศึกษา และทำความเข้าใจ โครงสร้าง กฎระเบียบและวิธีการในการสร้างชุดคำสั่งแต่ละโปรแกรมภาษา แต่ไม่ว่าจะเป็นโปรแกรมภาษาใด จะสังเกตได้ว่า คำสั่งที่ใช้ในแต่ละโปรแกรมภาษาจะเป็นคำสั่งที่ใกล้เคียงกับภาษามนุษย์ ทำให้มนุษย์สามารถใช้งานคอมพิวเตอร์ได้ง่ายขึ้น แต่ในทางกลับกัน ด้วยคอมพิวเตอร์ไม่ได้คุยกันเป็นภาษามนุษย์ ดังนั้นชุดคำสั่งเหล่านี้จะสั่งงานโดยตรงไม่ได้ จะต้องผ่านกระบวนการแปลภาษาให้เป็นภาษาเครื่องเสียก่อน ซึ่งในภาษาระดับสูงนั้น จะมีตัวแปลภาษา 2 ประเภทด้วยกัน คือ คอมไพเลอร์ (compiler) และอินเทอร์พรีเตอร์ (Interpreter) ดังนั้นเมื่อโปรแกรมภาษาระดับสูงต้องผ่านกระบวนการแปลภาษา จะทำให้เกิดการสูญเสียเวลาในการแปลผล ผลลัพธ์ที่ได้ที่จะแสดงให้เห็นกับผู้ใช้งานนั้นก็คือ เมื่อมีการสั่งงานคอมพิวเตอร์ การแสดงผลลัพธ์ที่ได้ จะช้าลง

1.3 การพัฒนาโปรแกรมแบบเชิงวัตถุ (Object Oriented Programming)

ในการพัฒนาโปรแกรมแบบไม่มีโครงสร้าง และแบบมีโครงสร้าง จะเห็นได้ว่ามีข้อแตกต่างกัน แต่ละโครงสร้างจะมีข้อดี และข้อเสียแตกต่างกัน แต่ด้วยวิวัฒนาการของการพัฒนาระบบคอมพิวเตอร์ทั้งในด้านฮาร์ดแวร์ (Hardware) ซอฟต์แวร์ (Software) และระบบเครือข่าย มีความเจริญก้าวหน้าไปอย่างรวดเร็วมากในปัจจุบัน เช่น มีความรวดเร็วในการประมวลผลสูง สามารถเก็บข้อมูลได้ปริมาณมากๆ ระบบเครือข่ายมีการส่งข้อมูลระหว่างอุปกรณ์ต่างๆ ได้อย่างรวดเร็ว เป็นต้น ปัญหาต่างๆ ในอดีตจึงลดน้อยลงไป

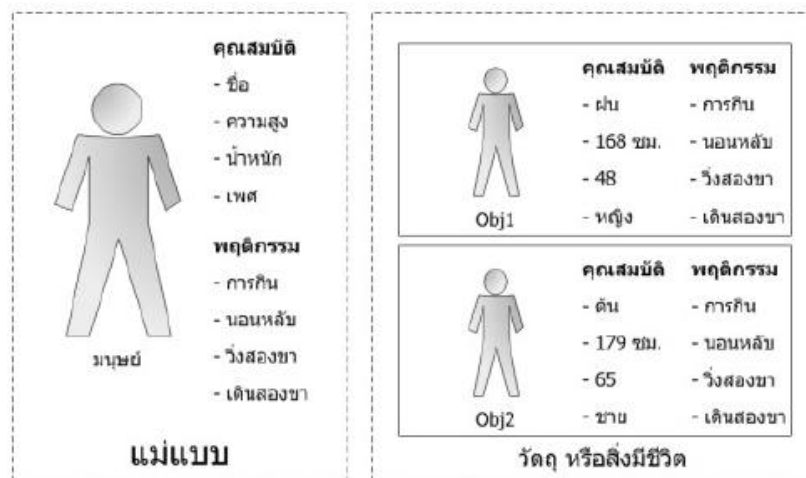
ในแนวคิดของด้านการพัฒนาโปรแกรม จากที่กล่าวมาในข้างต้น จะเห็นได้ว่าในการพัฒนาโปรแกรม แนวทางการพัฒนาโปรแกรมภาษา จะถูกจัดทำเฉพาะกลุ่มเท่านั้น หลักการพัฒนา หรือโครงสร้างของโปรแกรมภาษาจะถูกออกแบบตามเฉพาะกลุ่มเท่านั้น ดังนั้นเมื่อมีการเปลี่ยนโปรแกรมภาษา จะต้องทำการศึกษาจากโปรแกรมภาษาใหม่ เช่น จากตัวอย่างรูปที่ 1.3 และรูปที่ 1.4 จะมีโครงสร้าง หรือการใช้คำสั่งต่างๆ ไม่เหมือนกัน เมื่อพัฒนาโปรแกรมภาษาซี และจะชุดคำสั่งในภาษาซี และจะนำมาใช้งานภาษาปาสคาลจะไม่สามารถทำได้

จากปัญหาดังกล่าวในข้างต้น นักพัฒนาโปรแกรมภาษาในปัจจุบัน จึงมีแนวคิดของการที่จะพัฒนาโปรแกรมภาษาเป็นลักษณะของการใช้ทรัพยากรร่วมกัน เพื่อให้เกิดประโยชน์สูงสุด นั่นคือ เมื่อพัฒนาโปรแกรมภาษาหนึ่ง ก็จะสามารถนำชุดคำสั่งในภาษานั้นนำมาใช้ในอีกภาษาหนึ่งได้ จึงเกิดเป็นลักษณะของแนวคิดที่เรียกว่า “เชิงวัตถุ”

การพัฒนาโปรแกรมแบบเชิงวัตถุ นั้น เป็นการพัฒนาโปรแกรมให้เข้าใกล้การดำเนินไปตามสภาพแวดล้อมที่เป็นจริงของโลก กล่าวคือ เป็นการมองทุกอย่างในการพัฒนาโปรแกรมเป็นดังการดำรงอยู่ของสิ่งต่างๆ บนโลกนี้นั่นเอง โดยทุกๆ ซึ่งกล่าวได้ว่าสิ่งบนโลกนี้มักเกิดจากแม่แบบ (Class) แทบทั้งสิ้น ซึ่งแม่แบบเหล่านี้ จะประกอบไปด้วยสิ่งที่จำเป็นอยู่ 2 ส่วน คือ คุณสมบัติ (Properties) และแบบแผนพฤติกรรม (Behavior หรือ Method) อาจจะมีความเหมือน หรือต่างกันทั้งใน

คุณสมบัติ และพฤติกรรม โดยผลของแบบแผนพฤติกรรมนั้นจะได้รับเป็นแบบใด ขึ้นอยู่กับคุณสมบัติ ในช่วงเวลาสั้นๆ ซึ่งจะอธิบายให้เห็นภาพชัดเจนได้ง่ายขึ้นดังตัวอย่าง

ถ้ามองถึง “มนุษย์” เป็นแม่แบบ ทุกๆ คนบนโลกนี้เกิดจากแม่แบบมนุษย์ ซึ่งวัตถุ (Object) ที่เกิดจากแม่แบบมนุษย์นั้น จะมีคุณสมบัติ ที่มีชื่อเรียกแตกต่างกันออกไป เช่น ความสูง น้ำหนัก เพศ หรือส่วนประกอบทางร่างกายที่แตกต่างกันต่างๆ ซึ่งสิ่งเหล่านี้คือคุณสมบัติของมนุษย์ โดยมีพฤติกรรม ที่เหมือนๆ กันโดยพื้นฐานของมนุษย์ ซึ่งได้แก่ การกิน การนอน วิ่ง หรือเดิน ซึ่งถือเป็นพฤติกรรมแม่แบบ ซึ่งสามารถอธิบายได้ดังรูปที่ 1.5



รูปที่ 1.5 แม่แบบและวัตถุ

จากรูปที่ 1.5 จะเห็นว่า Obj1 และ Obj2 มีรูปแบบพฤติกรรมและคุณสมบัติเหมือนกับแม่แบบ มนุษย์ แต่คุณสมบัติของทั้งสองวัตถุนั้น ได้ถูกกำหนดต่อมาให้แตกต่างกัน ทำให้ผลของพฤติกรรมนั้นแตกต่างกันไปตามข้อมูลของคุณสมบัติที่เปลี่ยนแปลงไป เช่น Obj1 นั้นมีความสูง 168 ซม. น้ำหนัก 48 กก. และเป็นผู้หญิง ซึ่งมีคุณสมบัติแตกต่างกับ Obj2 ที่มีความสูง 179 ซม. น้ำหนัก 65 กก. และเป็นเพศชาย กล่าวคือ ในสภาวะเป็นจริง ความแตกต่างของทั้งสองวัตถุนั้นถูกกำหนดจากคุณสมบัติ ซึ่งทำผลของพฤติกรรมนั้นแตกต่างกัน แม้จะเกิดจากแม่แบบเดียวกันก็ตาม

จากรายละเอียดเบื้องต้น สามารถสรุปได้ว่า แม่แบบ หรือคลาส มีความเป็นนามธรรม (Abstract) และวัตถุนั้น มีความเป็นรูปธรรม (Concept)”

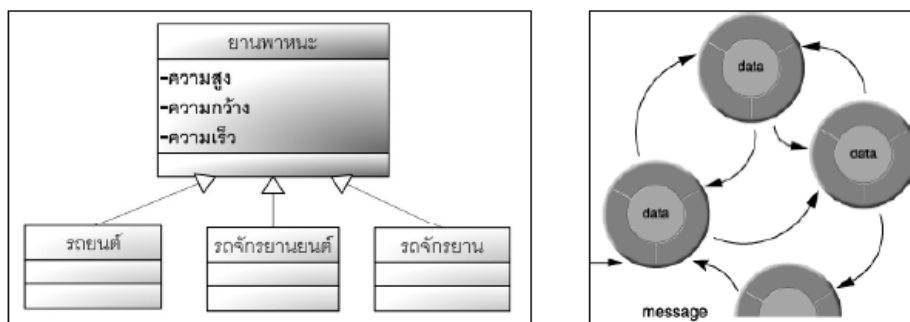
โดยทั่วไป คุณสมบัติต่างๆ ของวัตถุนั้นต้องมีการห่อหุ้มอยู่ใน (Encapsulation) ซึ่งทำให้คุณสมบัติของข้อมูลนั้นถูกต้องเสมอ โดยจากตัวอย่างที่ผ่านมา ในกรณีของคุณสมบัติของเพศนั้น จะต้องอยู่ภายใต้เงื่อนไขของเพศชาย และหญิงเท่านั้น จึงจำเป็นต้องมีตัวควบคุมคุณสมบัติ หรือแบบแผนพฤติกรรม เข้ามาเกี่ยวข้อง โดยความถูกต้องของข้อมูลภายในนั้นสำคัญมาก เพราะนั่นหมายถึง

คุณสมบัติที่ถูกต้องของวัตถุนั้นๆ ด้วย รวมถึงถ้ามองในด้านการพัฒนาโปรแกรม การห่อหุ้มภายในนั้น จะช่วยในการดูแลรักษา และพัฒนาชุดคำสั่งนั้น ทำให้ง่ายขึ้น เพราะชุดคำสั่งได้ถูกแยกออกเป็น ส่วน และยังบอกถึงพฤติกรรมของสิ่งที่ห่อหุ้มด้วยว่าควรเป็นพฤติกรรมอะไร และเมื่อพบพฤติกรรมนั้นๆ แล้ว ผลของพฤติกรรมนั้นจะให้ข้อมูลและผลการการทำงานออกมาเป็นอะไรด้วย



รูปที่ 1.6 การห่อหุ้มภายในคุณสมบัติ

คุณสมบัติที่ทำให้การพัฒนาโปรแกรมเชิงวัตถุได้รับความนิยมมาก คือ การสืบทอด (Inheritance) ของแม่แบบ ที่ทำให้ลูกแบบ (Sub-Class) นั้นมีความเฉพาะเจาะจงมากขึ้นไปอีก และ ยังช่วยให้การสร้างสรรค์งานนั้นทำได้ง่าย เพราะไม่จำเป็นต้องเขียนโปรแกรมหลายๆ อย่างซ้ำๆ กันใน แม่แบบที่มีคุณสมบัติเหมือนๆ กัน เช่น แม่แบบ “ยานพาหนะ” มีลูกแบบ ซึ่งได้แก่ รถยนต์ รถจักรยานยนต์ และจักรยาน เป็นต้น ซึ่งทั้ง 3 ลูกแบบที่กล่าวมานั้นมีคุณสมบัติบางอย่างที่เหมือนกัน จนได้เป็นยานพาหนะ เช่น ความสูง ความกว้าง และความเร็วในการเคลื่อนที่ เป็นต้น ซึ่งการแบ่งเป็น ลูกแบบนี้ทำให้ความเฉพาะเจาะจงของลูกแบบมีมากขึ้นทำให้การสร้างวัตถุใดๆ จากลูกแบบเหล่านี้ นั้น มีคุณสมบัติที่เที่ยงตรง ต่อการแสดงพฤติกรรมได้มากขึ้นด้วย และลดการซ้ำซ้อนของโครงสร้าง แม่แบบลงได้มาก เพราะได้ถูกกำหนดคุณสมบัติหลักๆ ไว้แล้วในแม่แบบ ทำให้ลูกแบบมีความซับซ้อน น้อยลง



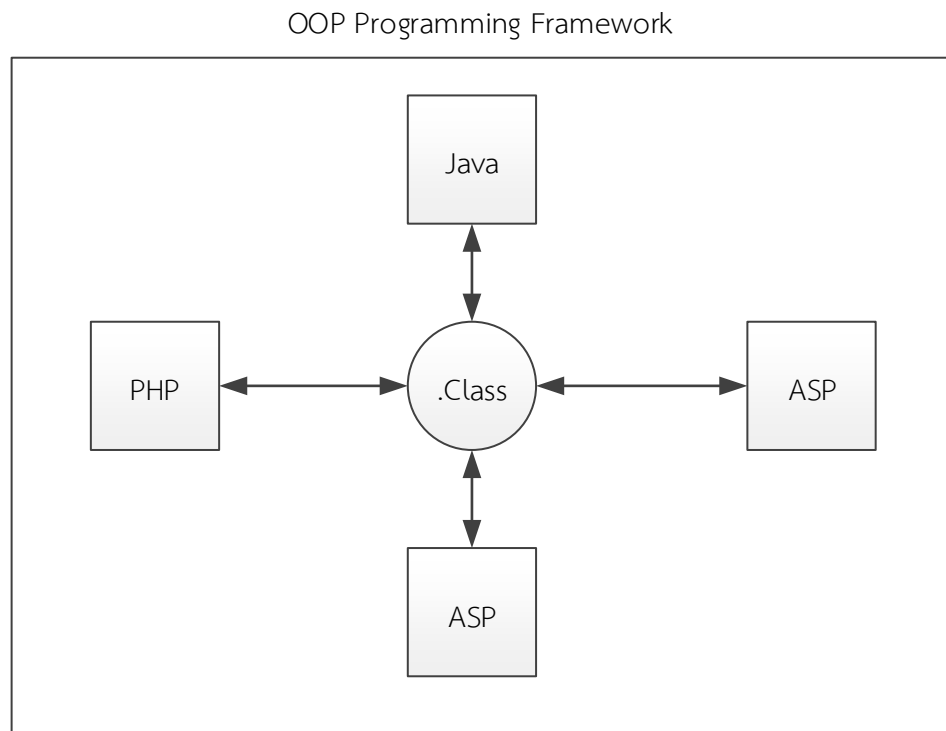
รูปที่ 1.7 การสืบทอดคุณสมบัติ และการสื่อสารกันของวัตถุ

1.3.1 คุณลักษณะพิเศษของแนวความคิดแบบเชิงวัตถุ

แนวความคิดแบบเชิงวัตถุ เป็นแนวคิดที่นำแก้ไขปัญหา จากแนวคิดอันเดิม ซึ่งสามารถจำแนกคุณลักษณะพิเศษ ได้ดังต่อไปนี้

1.3.1.2 คุณสมบัติด้านการใช้ทรัพยากรร่วมกัน (Sharing Resource)

หมายถึง ในการพัฒนาโปรแกรมภาษาแบบเชิงวัตถุ นั้น ไม่ว่าจะพัฒนาด้วยโปรแกรมภาษาใดก็ตาม ผู้พัฒนาสามารถที่จะนำเอา ชุดคำสั่งที่พัฒนาต่างโปรแกรมภาษาอื่น สามารถนำมาใช้งานร่วมกันได้ ซึ่งสามารถอธิบายได้ดังรูปที่ 1.8



รูปที่ 1.8 การใช้ทรัพยากรร่วมกันของกลุ่มโปรแกรมภาษาเชิงวัตถุ

การใช้ทรัพยากรร่วมกันของกลุ่มโปรแกรมภาษาเชิงวัตถุ นั้นคือ เมื่อมีการสร้างชุดคำสั่งจากโปรแกรมภาษาใดก็ตามที่เป็นโปรแกรมภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ เมื่อมีการแปลภาษาจากตัวแปลภาษา ไฟล์ที่ได้จากการแปลภาษาจะเป็นไฟล์แบบเดียวกันนั่นคือไฟล์นามสกุล class โปรแกรมภาษาต่างๆ ก็สามารถนำไฟล์นี้ไปเรียกใช้งานได้ทันที ซึ่งถือว่าเป็นประโยชน์มาก เพราะเมื่อมีการเรียกใช้งานที่เป็นแบบเดิมทุกๆ ครั้ง ไม่ต้องจำเป็นต้องมาสร้างชุดคำสั่งใหม่ๆ

1.3.1.2 คุณสมบัติปรับปรุงแก้ไข (Validation Data) หมายถึง

ในการปรับปรุงแก้ไข ข้อมูลต่างๆ ที่อยู่ภายในคลาส สามารถที่จะปรับปรุงแก้ไขได้ ซึ่งจะมีผลกระทบต่อค่าต่างๆ ที่อยู่ภายในคลาส ซึ่งการเปลี่ยนแปลงต่างๆ จะมีผลกระทบเฉพาะในคลาสของตนเอง ดังนั้น

เมื่อมีข้อผิดพลาดต่างๆ ที่เกิดขึ้นภายในคลาส ก็จะสามารถดำเนินการตรวจสอบข้อผิดพลาดได้ง่ายเพิ่มมากยิ่งขึ้น

1.3.2 ลักษณะของโปรแกรมเชิงวัตถุ

ลักษณะของการนำหลักการแนวคิดเชิงวัตถุไปใช้ สามารถแบ่งลักษณะการใช้งานได้ดังต่อไปนี้

1.3.2.1 การใช้โปรแกรมเชิงวัตถุ

การใช้โปรแกรมเชิงวัตถุ จะเป็นลักษณะของโปรแกรมภาษาที่มีการแบ่งกลุ่มเครื่องมือต่างๆ ไว้ใช้งาน ซึ่งแต่ละกลุ่ม ก็คือแต่ละคลาส ซึ่งเวลาที่จะนำมาใช้งาน ก็ต้องเลือกคลาสก่อน แล้วนำไปการจัดตามหน้าต่างของโปรแกรมที่ออกแบบไว้ หรือที่เรียกว่า GUI (Graphic User Interface) โปรแกรมภาษาในปัจจุบันจะมีการออกแบบเป็นลักษณะนี้แทบทั้งสิ้น ยกตัวอย่าง เช่น โปรแกรม Visual Studio .Net, Dreamweaver, Delphi เป็นต้น ดังตัวอย่างโปรแกรม

The screenshot shows a software application window titled "Products File". The main area is a form for entering product details. The form is organized into several sections:

- Product Information:** Fields for Product ID (JBA-016), Barcode (No125241), Supplier Code, Price (92,500.00 Bht/Pc), and Price 1 (1,850.00 Us\$/Pc).
- Product Details:** Fields for Product Name (JBA-016), Brand (JK), Weight (2.00 gms), Color (Cl), and Size (Size).
- Product Specifications:** Fields for Product Type (BANGLES), Product Status (Set), and Product Material (18KYG 23.66 G. Dia 17/1 517 CT).
- Product Images:** Two preview images of the product (a bangle and a ring).
- Product Price:** Fields for Price 2 (4,375.59 Bht/gm) and Price 3 (87.51 Us\$/gm).
- Product Weight:** Fields for Weight (21.14 gms) and Weight (21.14 gms).
- Product Material:** Fields for Metal (21.14 gms), Diamond (Cts), and Stones (Cts).
- Buttons:** Buttons for "Preview", "Print", "Add", "Edit", "Delete", "Save", "Cancel", "Report", and "Quit".

รูปที่ 1.9 การใช้โปรแกรมเชิงวัตถุ

จากตัวอย่างรูปที่ 1.9 แสดงให้เห็นลักษณะที่มีการออกแบบโดยใช้เครื่องมือที่เป็นแบบเชิงวัตถุ เช่น ในภาพจะประกอบไปด้วยกลุ่มคลาสที่เป็น textbox ซึ่งมีหน้าที่ในการรับข้อมูล คลาส button จะเป็นคลาสปุ่มที่ทำหน้าที่ในการยอมรับคำสั่งแล้วนำไปประมวลผลต่าง ๆ เป็นต้น

13.1.2 การเขียนโปรแกรมเชิงวัตถุ

ในลักษณะมุมมองอีกแบบหนึ่ง ที่ใช้เทคนิควิธีการตามแนวความคิดของเชิงวัตถุ นั่นก็คือ ใช้ในพัฒนาโปรแกรม หรือการเขียนโปรแกรม ซึ่งในปัจจุบันนี้เป็นที่นิยมกันมาก และ

แทบทุกโปรแกรมภาษาจะดำเนินไปสู่กระบวนการการพัฒนาโปรแกรมแบบนี้แทบทั้งสิ้น ยกตัวอย่างเช่นในการพัฒนาโปรแกรมภาษา php ที่มีการพัฒนาในรูปแบบเชิงวัตถุ ดังรูปที่ 1.10

บรรทัด	ชุดคำสั่ง
1	<?php
2	class human
3	{
4	public \$name;
5	var \$sex;
6	public \$height;
7	public \$weight;
8	public \$age;
9	
10	public function run()
11	{
12	echo "you can runnig";
13	}
14	public function jump()
15	{ echo "you can jumping"; }
16	}
17	
18	\$fon = new human();
19	\$fon -> name="ฝน";
20	\$fon -> sex="หญิง";
21	\$fon -> height=168;
22	\$fon -> weight=48;
23	\$fon -> age=28;
24	\$fon -> run();
25	\$fon -> jump();
26	echo "ชื่อ : ".\$fon->name." ";
27	echo "เพศ : ".\$fon->sex." ";
28	echo "ความสูง : ".\$fon->height." ";
29	echo "น้ำหนัก : ".\$fon->weight." ";

30	echo "น้ำหนัก : ".\$fon->age." ";
31	
32	\$ton = new human();
33	\$ton -> name="ตัน";
34	\$ton -> sex="ชาย";
35	\$ton -> height=179;
36	\$ton -> weight=65;
37	\$ton -> age=30;
38	
39	echo "ชื่อ : ".\$ton->name." ";
40	echo "เพศ : ".\$ton->sex." ";
41	echo "ความสูง : ".\$ton->height." ";
42	echo "น้ำหนัก : ".\$ton->weight." ";
43	echo "น้ำหนัก : ".\$ton->age." ";
44	\$fon -> run();
45	\$fon -> jump();
46	
47	?>

รูปที่ 1.10 การเขียนโปรแกรมเชิงวัตถุด้วยภาษา php

จากตัวอย่างภาพที่ 1.10 แม่แบบของคลาสก็คือ คลาส human ประกอบไปด้วยออบเจกต์ 2 ออบเจกต์คือ \$fon และ \$ton ในคลาสนี้ประกอบไปด้วยคุณสมบัติ \$name, \$sex, \$height, \$weight และ \$age และประกอบไปด้วยการกระทำ run() และ jump()