

Basic Algorithms

With

Turbo C

อัลกอริทึมด้วยเทอร์โบซี

- ✓ การเขียนโปรแกรมคอมพิวเตอร์แบบโครงสร้าง
 - ✓ การออกแบบโปรแกรมด้วยผังงาน
- ✓ ความรู้เบื้องต้นเกี่ยวกับโปรแกรมภาษาซี
 - ✓ คำสั่งแสดงผล
 - ✓ คำสั่งรับข้อมูล
 - ✓ ตัวดำเนินการ
 - ✓ คำสั่งเงื่อนไข
 - ✓ คำสั่งวนซ้ำ
- ✓ คู่มือการใช้งานเทอร์โบซี

คำนำ

การเขียนเป็นการประยุกต์ของทักษะความรู้เชิงภาษาของโลกหนังสือ สำหรับคอมพิวเตอร์นั้น การเขียนโปรแกรมก็อาจเทียบเคียงได้เช่นเดียวกัน เพราะโปรแกรมคอมพิวเตอร์หนึ่งโปรแกรม เทียบได้กับการเขียนเรียงความหนึ่งเรื่อง ซึ่งหมายความว่า การเขียนโปรแกรมคอมพิวเตอร์ จะต้องวางโครงเรื่อง เรียนรู้และสรรหาคำสั่ง เรียบเรียงคำสั่งต่าง ๆ เข้าด้วยกัน เพื่อให้ได้โปรแกรมที่ต้องการ

หนังสือเล่มนี้เขียนขึ้น เพื่อจัดระเบียบแนวคิดของผู้ที่เริ่มเขียนโปรแกรมคอมพิวเตอร์ ให้สามารถวางโครงโปรแกรม โดยใช้อัลกอริทึม ซึ่งเป็นขั้นตอนการเขียนโปรแกรม และแสดงผังโปรแกรมเป็นรูปภาพ เพื่อให้เห็นอัลกอริทึมได้ง่ายขึ้น หลังจากนั้นจะนำโครงดังกล่าว ไปเขียนเป็นโปรแกรมให้ได้ผลตามที่ออกแบบไว้

การเขียนโปรแกรมภาษาซี ในเล่มนี้มุ่งเน้นให้ผู้เริ่มศึกษาการเขียนโปรแกรม และการศึกษาการออกแบบอัลกอริทึมเบื้องต้น ตามแนวทางของการเขียนโปรแกรมแบบโครงสร้าง ความรู้ในเชิงภาษาซีที่นำเสนอจะเริ่มจากความรู้เบื้องต้นจนถึงความรู้ระดับกลางเท่านั้น

ผู้ที่ควรจะศึกษาจากหนังสือเล่มนี้เป็นอย่างยิ่งคือ นักเรียน นักศึกษา ที่เป็นผู้เริ่มศึกษาการเขียนโปรแกรม และ ครู อาจารย์ ที่ต้องการสอนการเขียนโปรแกรม เนื่องจากเนื้อหาในหนังสือ จะแสดงแนวคิดเชิงทฤษฎีให้กระชับพอเข้าใจ และเน้นตัวอย่างที่มีอัลกอริทึม ผังโปรแกรม ตัวอย่างรหัสต้นฉบับ และผลการรันโปรแกรม ที่สามารถศึกษาได้โดยง่าย

ท้ายนี้ผู้เขียนหวังเป็นอย่างยิ่งว่า ผู้ที่ได้ศึกษาจากหนังสือเล่มนี้ จะมีทัศนคติที่ดีต่อการเขียนโปรแกรม จนพัฒนาตนเองให้เป็นโปรแกรมเมอร์ที่เก่ง สามารถคิดและวางแผนการเขียนโปรแกรมอย่างเป็นระบบ เพื่อพัฒนาประเทศไปสู่ความเป็นผู้นำทางด้านการพัฒนาซอฟต์แวร์ในอนาคต

เชาวลิต ชันคำ

4 พฤษภาคม 2549

สารบัญ

หน้า

คำนำ	(1)
สารบัญ	(3)
สารบัญภาพ	(7)
สารบัญตาราง	(17)
บทที่ 1 โปรแกรมภาษาและการเขียนโปรแกรมคอมพิวเตอร์	1
ความหมาย และยุคของภาษาคอมพิวเตอร์	1
การเขียนโปรแกรมคอมพิวเตอร์แบบโครงสร้าง	9
ขั้นตอนในการเขียนโปรแกรม	14
สรุป	16
แบบฝึกหัดท้ายบท	17
บทที่ 2 การออกแบบโปรแกรมด้วยผังงาน	19
สัญลักษณ์ของผังงาน	19
รูปแบบผังงานโครงสร้าง	23
สรุป	40
แบบฝึกหัดท้ายบท	41
บทที่ 3 ความรู้เบื้องต้นเกี่ยวกับโปรแกรมภาษาซี	43
ความเป็นมาของภาษาซี	43
ขั้นตอนในการเขียนโปรแกรมด้วยภาษาซี	46
รูปแบบของโปรแกรมที่เขียนด้วยภาษาซี	47
ตัวแปรในภาษาซี	63
คลังของฟังก์ชันในภาษาซี (Header files)	76
สรุป	79
แบบฝึกหัดท้ายบท	81

บทที่ 4	คำสั่งแสดงผล	83
	ฟังก์ชันการแสดงผลจาก CONIO.H	83
	ฟังก์ชันการแสดงผลจาก STDIO.H	101
	การแสดงรหัสแอสกี.....	116
	การแสดงผลจากฟังก์ชันพื้นฐาน.....	122
	สรุป.....	125
	แบบฝึกหัดท้ายบท.....	127
บทที่ 5	คำสั่งรับข้อมูล	129
	การนำเข้าข้อมูล	129
	ฟังก์ชันสำหรับการรับข้อมูลจาก STDIO.H	130
	ฟังก์ชันสำหรับการรับข้อมูล จาก CONIO.H	146
	สรุป.....	162
	แบบฝึกหัดท้ายบท.....	163
บทที่ 6	ตัวดำเนินการ	165
	ความหมายและประเภทของตัวดำเนินการ.....	165
	ตัวดำเนินการทางคณิตศาสตร์	166
	ตัวดำเนินการเปรียบเทียบความสัมพันธ์และตรรกะ	203
	ตัวดำเนินการบิตไวส์.....	216
	สรุป.....	224
	แบบฝึกหัดท้ายบท.....	225
บทที่ 7	คำสั่งเงื่อนไข	227
	ความหมายและความสำคัญ.....	227
	คำสั่ง if.....	228
	คำสั่ง switch.....	252
	สรุป.....	261
	แบบฝึกหัดท้ายบท	263

บทที่ 8	คำสั้งวนซั้	265
	ความหมายและประเภทคำสั้งวนซั้.....	265
	คำสั้ง for.....	266
	คำสั้ง while	285
	คำสั้ง do...while	303
	สรุป.....	321
	แบบฝึกหัดท้ายบท.....	323

บรรณานุกรม	325
-------------------------	-----

ภาคผนวก คู่มือการใช้งานเทอร์โบซี

บทที่ 1

โปรแกรมภาษาและการเขียนโปรแกรมคอมพิวเตอร์

คอมพิวเตอร์คืออุปกรณ์อิเล็กทรอนิกส์ที่สามารถประมวลผลให้ได้ตามที่มนุษย์ต้องการ องค์ประกอบที่จะทำให้สามารถประมวลผลได้ตามที่ต้องการนั้นประกอบด้วย ฮาร์ดแวร์และซอฟต์แวร์ เป็นหลัก ฮาร์ดแวร์ คืออุปกรณ์ของคอมพิวเตอร์ที่สามารถสัมผัสได้จับต้องได้ และซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์คือชุดคำสั่งที่สั่งให้คอมพิวเตอร์สามารถทำงานตามที่ต้องการ รายละเอียดของหนังสือเล่มนี้จะกล่าวถึงส่วนที่เป็นซอฟต์แวร์ ยุคของโปรแกรมภาษาคอมพิวเตอร์ การพัฒนาโปรแกรมโดยใช้โปรแกรมภาษาคอมพิวเตอร์แบบต่าง ๆ ขั้นตอนการพัฒนาโปรแกรมด้วยภาษาคอมพิวเตอร์และแสดงตัวอย่างการวิเคราะห์ปัญหาก่อนการเขียนโปรแกรม

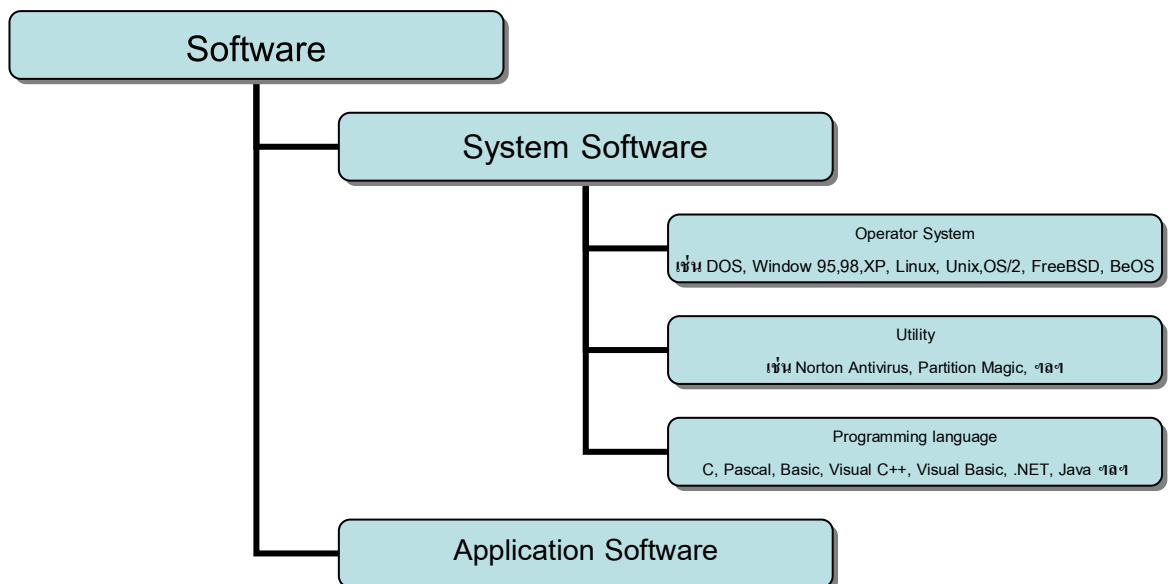
ความหมาย และยุคของภาษาคอมพิวเตอร์

เพ็ญศรี ปักกะสีนัง (2546, หน้า 92-97) ได้แบ่งโปรแกรมคอมพิวเตอร์ออกเป็น 2 ประเภทซอฟต์แวร์ระบบและซอฟต์แวร์ซอฟต์แวร์ประยุกต์ ซึ่งโปรแกรมระบบคือโปรแกรมที่ทำหน้าที่เกี่ยวกับการจัดการระบบของคอมพิวเตอร์ โปรแกรมเหล่านี้ได้แก่ ระบบปฏิบัติการ (Operator system) โปรแกรมอรรถประโยชน์ (Utility) โปรแกรมภาษาคอมพิวเตอร์ (Programming languages) เป็นต้น ส่วนโปรแกรมประยุกต์ ได้แก่ โปรแกรมที่พัฒนาขึ้นมาเฉพาะด้าน เช่น โปรแกรมที่ใช้ในการจัดการกราฟิก โปรแกรมเกม โปรแกรมสำหรับสำนักงาน เป็นต้น ผู้ที่มีบทบาทสำคัญที่สุดคือผู้ที่เขียนโปรแกรมขึ้นมา เรียกว่า โปรแกรมเมอร์ (Programmer) ซึ่งเขียนชุดคำสั่งต่าง ๆ ที่ทำให้คอมพิวเตอร์ทำงานได้ตามที่ผู้ใช้ต้องการ

ภาษาคอมพิวเตอร์คือภาษาที่ใช้สื่อสารกันระหว่างคอมพิวเตอร์กับมนุษย์ ชีรวัดน์ ประกอบผล (2545, หน้า 3) และ เพ็ญศรี ปักกะสีนัง (2546, หน้า 97) กล่าวว่าภาษาคอมพิวเตอร์เป็นการนำชุดคำสั่งแต่ละคำสั่งมาต่อกันให้คอมพิวเตอร์ทำงาน การ

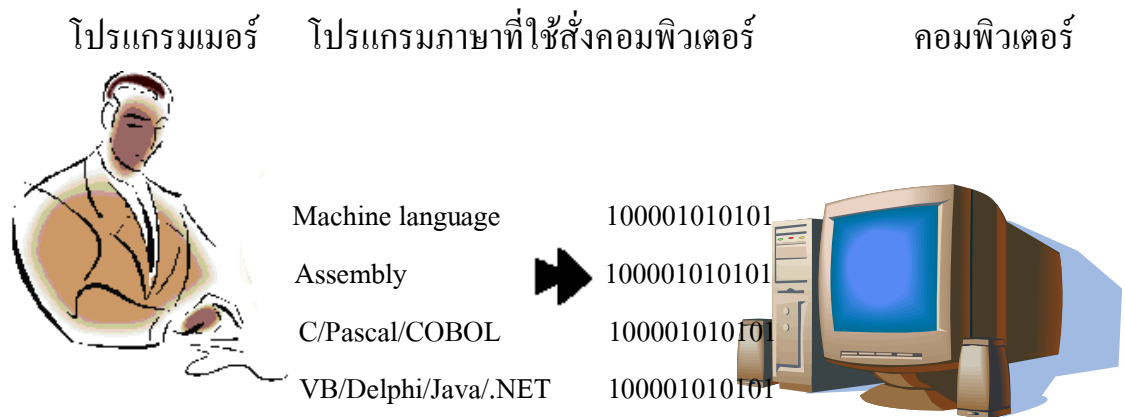
เขียนชุดคำสั่งนี้ไม่ว่าจะเขียนด้วยภาษาอะไรจะเรียกว่าโปรแกรมต้นฉบับ (source program) หรือ รหัสต้นฉบับ (source code)

โปรแกรมภาษาคอมพิวเตอร์ คือ โปรแกรมที่ใช้ในการพัฒนาโปรแกรมอื่น ๆ ตามรูปแบบของภาษาคอมพิวเตอร์แต่ละภาษา หน้าที่ของโปรแกรมภาษาคอมพิวเตอร์คือใช้สำหรับพัฒนาโปรแกรมให้ผู้อื่นใช้งาน โดยผู้ที่ใช้ภาษาคอมพิวเตอร์คือโปรแกรมเมอร์ พิจารณาตามภาพที่ 1.1 แสดงตำแหน่งของโปรแกรมภาษาคอมพิวเตอร์ในโครงสร้างของซอฟต์แวร์ ที่ระบุตำแหน่งของโปรแกรมที่จะศึกษาในหนังสือเล่มนี้



ภาพที่ 1.1 แสดงตำแหน่งของโปรแกรมภาษาคอมพิวเตอร์

ภาพที่ 1.2 แสดงแนวคิดของการเขียนโปรแกรม ซึ่งโปรแกรมเมอร์จะออกแบบและเขียนโปรแกรมด้วยโปรแกรมภาษาที่แตกต่างกันตามความถนัดและเหมาะสมกับแต่ละงาน หลังจากนั้นจะแปลคำสั่งของโปรแกรมที่เขียนขึ้นด้วยคอมไพเลอร์หรืออินเทอร์พรีเตอร์ให้เป็นคำสั่งที่ติดต่อกับคอมพิวเตอร์ได้ในลักษณะของภาษาเครื่องด้วยรหัสเลขฐานสอง เพื่อควบคุมให้คอมพิวเตอร์ทำงานได้ตามที่ต้องการ



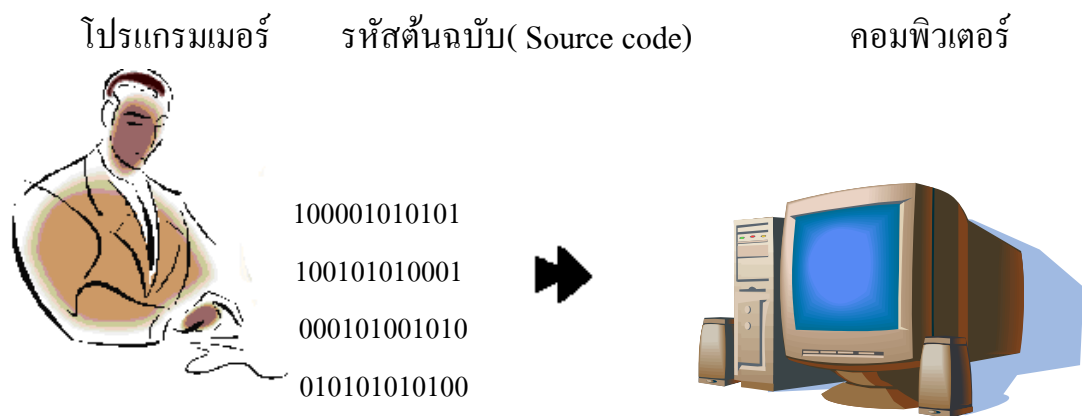
ภาพที่ 1.2 แสดงความสัมพันธ์ระหว่างโปรแกรมภาษากับการเขียนโปรแกรม

วิวัฒนาการของภาษาคอมพิวเตอร์จะถูกแบ่งออกเป็นยุค ๆ และถูกแบ่งออกเป็นระดับต่าง ๆ กัน เพ็ญศรี ปักกะสีนัง (2546, หน้า 98-100) ได้แบ่งระดับของภาษาคอมพิวเตอร์ออกเป็น 5 ระดับ ได้แก่ ภาษาเครื่อง ภาษาแอสเซมบลี ภาษาระดับสูง ภาษาระดับสูงมาก และภาษาธรรมชาติ สำหรับในหนังสือเล่มนี้จะแบ่งโดยเน้นด้านการพัฒนาโปรแกรมภาษาตามการเขียนโปรแกรมโดยแบ่งดังเป็นยุคที่ 1 ถึงยุคที่ 5 โดยเน้นยุคที่ 4 (Fourth Generation Language : 4GL) ที่มีบทบาทในปัจจุบันและยุคที่ 5 (Fifth Generation Language : 5GL) ซึ่งจะมีความสำคัญมากในอนาคต รวมถึงแนวทางการพัฒนาโปรแกรมคอมพิวเตอร์ในยุคที่ N ซึ่งการเขียนโปรแกรมอาจมีใช้เพียงโปรแกรมเมอร์เท่านั้น แต่มนุษย์ทุกคนจำเป็นต้องเขียนโปรแกรมด้วย

1. ยุคที่ 1

การพัฒนาโปรแกรมคอมพิวเตอร์ในยุคแรก เรียกว่า ภาษาเครื่อง หรือรหัสเครื่อง (Machine language or Machine code) การเขียนโปรแกรมใช้การป้อนชุดคำสั่งโดยใช้รหัสเลขฐานสองคือ 0 กับ 1 ให้คอมพิวเตอร์โดยตรง ซึ่งทำให้การเขียนโปรแกรมเป็นภาระหนักของโปรแกรมเมอร์มาก เนื่องจากไม่มีเครื่องมือที่ช่วยพัฒนาโปรแกรม ภาพที่ 1.3 แสดงแนวคิดของการเขียนโปรแกรมในยุคดังกล่าวนี้

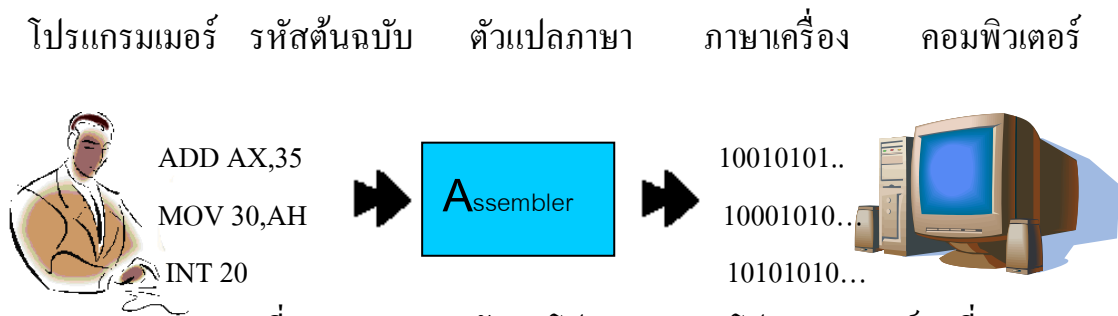
ข้อดีคือ เครื่องคอมพิวเตอร์ไม่ต้องแปลรหัสต้นฉบับ (Source code) คอมพิวเตอร์สามารถนำไปใช้งานได้ทันที ทำให้การทำงานเร็วมาก แต่ข้อเสียคือ การเขียนโปรแกรมทำได้ยาก เพราะต้องจดจำคำสั่งและข้อมูลต่าง ๆ เป็นรหัสเลขฐานสอง จึงทำให้ในยุคต่อมาต้องพัฒนาเป็นโปรแกรมภาษาในระดับที่สูง



ภาพที่ 1.3 แสดงการพัฒนาโปรแกรมของโปรแกรมเมอร์ยุคที่ 1

2. ยุคที่ 2

ยุคที่ 2 เริ่มเมื่อประมาณต้นของช่วง 1950 ยุคนี้มีแนวคิดที่จะทำให้ภาษาคอมพิวเตอร์มีความใกล้เคียงภาษามนุษย์มากขึ้น โดยนำเอาภาษาอังกฤษมาสร้างเป็นคำสั่ง ภาษาในยุคนี้คือภาษาแอสเซมบลี (Assembly language) มีคำสั่งซึ่งเป็นคำภาษาอังกฤษง่าย ๆ เช่น ADD, SUB, MUL, DIV แทนการบวก ลบ คูณ และหาร เป็นต้น การเขียนโปรแกรมในยุคนี้ตัวแปลภาษาจะแปลจากภาษาอังกฤษมาเป็นภาษาเครื่องโดยใช้ตัวแปลภาษาที่เรียกว่า แอสเซมเบลอร์ (Assembler) โปรแกรมเมอร์จะต้องมีความรู้ทั้งรูปแบบการใช้คำสั่งและระบบของฮาร์ดแวร์เป็นอย่างดี จึงจะสามารถเขียนโปรแกรมได้ดี แนวคิดในการเขียนโปรแกรมในยุคนี้แสดงดังภาพที่ 1.4



ภาพที่ 1.4 แสดงการพัฒนาโปรแกรมของโปรแกรมเมอร์ยุคที่ 2

เครื่องมือที่ใช้ในการพัฒนาโปรแกรมภาษาในยุคนี้มีให้เลือกมากขึ้น เช่น โปรแกรม Turbo Assembly , Macro Assembly เป็นต้น

ข้อดีคือ สามารถพัฒนาโปรแกรมได้รวดเร็วขึ้น การทำงานของโปรแกรมทำงานเร็ว แต่ข้อเสียคือ โปรแกรมเมอร์ต้องเรียนรู้สถาปัตยกรรมของคอมพิวเตอร์ โดยเฉพาะเครื่องคอมพิวเตอร์แต่ละยี่ห้อ แต่ละรุ่นต่างมีสถาปัตยกรรมที่แตกต่างกันไป ทำให้แทนที่โปรแกรมเมอร์จะทุ่มเทให้การเขียนโปรแกรมอย่างเดียวกลับต้องศึกษาสถาปัตยกรรมคอมพิวเตอร์อีก จึงจำเป็นต้องพัฒนาโปรแกรมภาษาให้ง่ายต่อการเขียนโปรแกรมให้มากขึ้น

3. ยุคที่ 3

ยุคที่ 3 เริ่มเมื่อประมาณกลางช่วง 1950 ยุคนี้ถือว่าเป็นยุคแรกของภาษาระดับสูง (High level language) เพราะยุคที่ 1 และ 2 นั้นถือว่าเป็นยุคของภาษาระดับต่ำ (Low level language) ที่ยังไม่เหมือนภาษามนุษย์เท่าใดนักแต่ยุคนี้ภาษาคอมพิวเตอร์จะใกล้เคียงกับภาษาของมนุษย์มากขึ้นอีก โดยใช้ประโยคหรือวลีในภาษาอังกฤษเพื่อเขียนโปรแกรม ข้อดีของภาษาในยุคนี้คือ

- 1) การเขียนโปรแกรมใช้เวลาน้อยและต้นทุนต่ำลง
- 2) โปรแกรมเมอร์ไม่จำเป็นต้องเรียนรู้สถาปัตยกรรมของคอมพิวเตอร์ ทำให้การเขียนโปรแกรมทำได้กว้างขวางมากขึ้น

แนวการเขียนโปรแกรมมีชื่อเรียกหลากหลาย เช่น โปรแกรมแบบโครงสร้าง (Structure program) โปรแกรมแบบโมดูลหรือแบบโพรซีเจอร์อล (Modular program or procedural program) มีหลายภาษาที่เกิดขึ้นในยุคนี้ คำสั่งต่าง ๆ จะถูกกำหนดโดย

ไวยากรณ์ของแต่ละภาษา โปรแกรมเมอร์ต้องเขียนรหัสต้นฉบับของคำสั่งทั้งหมดด้วยตนเอง

เครื่องมือช่วยพัฒนา ส่วนมากจะเป็นอิดิเตอร์ (Editors) ที่ติดมากับตัวภาษาแต่ละภาษา โปรแกรมเมอร์จะเขียนรหัสต้นฉบับทั้งหมด จากนั้นใช้ตัวแปลภาษาซึ่งอาจเป็นคอมไพเลอร์ (Compiler) หรืออินเตอร์พรีเตอร์ (Interpreter) แปลรหัสต้นฉบับให้เป็นภาษาเครื่องสั่งให้คอมพิวเตอร์ทำงาน ภาษาคอมพิวเตอร์ยุคนี้มีหลากหลายเหมาะสมกับงานที่แตกต่างกัน เช่น FORTRAN เหมาะกับงานคำนวณทางด้านวิทยาศาสตร์ C , Pascal , BASIC เหมาะกับงานทั่วไป ภาษา COBOL เหมาะกับงานทางด้านธุรกิจ เป็นต้น ภาพที่ 1.5 แสดงแนวคิดของการพัฒนาโปรแกรมภาษาในยุคนี้



ภาพที่ 1.5 แสดงการพัฒนาโปรแกรมของโปรแกรมเมอร์ยุคที่ 3

แม้ว่าภาษายุคนี้จะใกล้เคียงกับภาษามนุษย์มากก็ตาม แต่ปัจจัยทางด้านธุรกิจและความต้องการให้โปรแกรมทำงานยืดหยุ่นรองรับการทำงานได้กว้างขวางและมีความเข้ากันได้กับโปรแกรมอื่นๆ ได้ดีมากขึ้น จำเป็นต้องพัฒนาโปรแกรมภาษาให้ก้าวหน้าไปอีกยุคที่ 4 จึงเกิดขึ้นในเวลาต่อมา

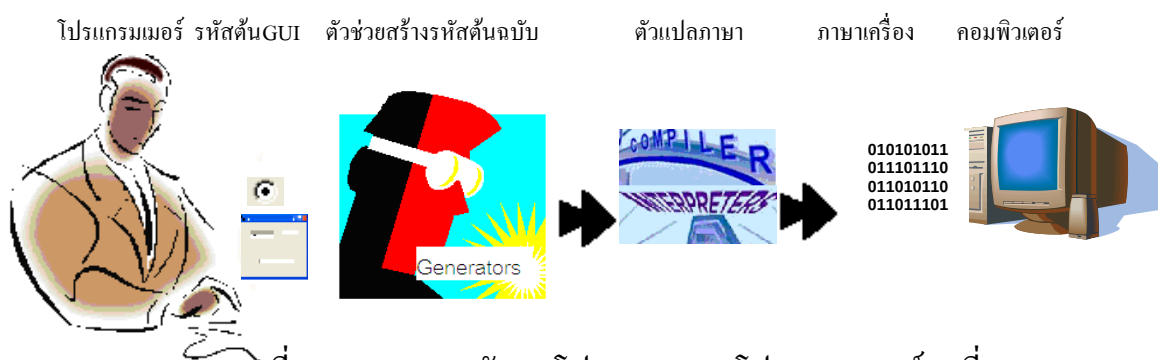
4. ยุคที่ 4

จากความจำเป็นในยุคที่ 3 ที่กล่าวมาแล้วทำให้ยุคที่ 4 ใช้โปรแกรมภาษาแนวใหม่เพื่อให้สนองความต้องการด้านต่าง ๆ เหล่านั้น ต้นยุคเริ่มเมื่อประมาณช่วง 1970 ยุคนี้อาจกล่าวได้ว่าเป็นยุค non-procedural การพัฒนานี้เน้นให้ได้ผลลัพธ์ตามความต้องการอย่างรวดเร็วมากกว่าที่จะเน้นว่าจะต้องเขียนอย่างไร ในยุคก่อนหน้านี้ โปรแกรมเมอร์ต้องรู้คำสั่งและวิธีการเขียนทุกสิ่งทุกอย่างด้วยตนเอง แต่ยุคนี้มีเครื่องมือช่วยพัฒนาได้รวดเร็วขึ้นกว่าเดิมมาก หากเปรียบเทียบการเขียนโปรแกรมเป็นการประกอบ

รถยนต์ 1 คัน ในยุคที่ผ่านมาผู้ประกอบต้องสร้างชิ้นส่วนทุกอย่างด้วยตนเองก่อนจึงประกอบเป็นรถยนต์ได้ แต่ยุคนี้การประกอบรถยนต์ทำได้ในเวลาอันรวดเร็วเนื่องจากมีผู้ผลิตชิ้นส่วนทั้งหมดไว้ให้แล้ว ผู้ประกอบเพียงแค่เลือกมาประกอบให้เหมาะสมเท่านั้น

การเขียนโปรแกรมยุคนี้เรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming :OOP) ซึ่งมองการเขียนโปรแกรมเหมือนการสร้างวัตถุเป็นชิ้น ๆ มีเครื่องมือช่วยเขียนโปรแกรม เรียกว่า ตัวสร้าง (Generators) ที่ติดมากับแต่ละภาษา ทำหน้าที่ติดต่อกับโปรแกรมเมอร์แบบกราฟิกอินเทอร์เฟซ (graphic interface) บางทีอาจเรียกว่าเป็นการเขียนโปรแกรมแบบวิสวล (Visual programming) โปรแกรมเมอร์ใช้วิธีแดร็กแอนด์ดรอป (Drag and Drop) คือการคลิกเอาวัตถุที่ใช้เป็นส่วนประกอบต่าง ๆ ของโปรแกรม เช่น ปุ่ม ฟอนต์ ไอคอน มาวางในพื้นที่เขียนโปรแกรม หลังจากนั้นกำหนดคุณสมบัติและหน้าที่การทำงานที่ควบคุมด้วยเหตุการณ์ที่เข้ามากระทำต่อวัตถุที่สร้างขึ้น หลังจากนั้นตัวสร้างจะแปลงให้เป็นรหัสต้นฉบับแล้วแปลด้วยคอมไพเลอร์หรืออินเตอร์พรีเตอร์ที่ติดมาด้วยเสมอ ตัวอย่างภาษายุคนี้ เช่น Visual Basic ,Visual C++, Borland Delphi, Borland C++ Builder, JBuilder ,VisualStudio.NET เป็นต้น ภาพที่ 1.6 แสดงแนวคิดของโปรแกรมภาษาในยุคนี้

ข้อดีของโปรแกรมภาษาในยุคนี้คือ การเขียนโปรแกรมทำได้รวดเร็วมาก ส่วนการติดต่อกับโปรแกรมเมอร์นำใช้มากขึ้น ใช้งานง่ายกว่าภาษายุคที่ 3 มาก แต่ข้อเสียคือโปรแกรมเมอร์ไม่สามารถรู้รายละเอียดบางอย่างที่ซ่อนเอาไว้โดยเครื่องมือช่วยพัฒนาเหล่านั้น โปรแกรมจะมีขนาดใหญ่มากขึ้น ฮาร์ดแวร์ที่ใช้จะต้องมีความเร็วสูง เนื่องจากจะต้องมีการแปลคำสั่งหลายขั้นตอน



ภาพที่ 1.6 แสดงการพัฒนาโปรแกรมของโปรแกรมเมอร์ยุคที่ 4

5. ยุคที่ 5

ปัจจุบันเริ่มเข้าสู่ต้นยุคที่ 5 แล้ว ยุคนี้จะเน้นการนำปัญญาประดิษฐ์ (Artificial Intelligent :AI) มาใช้เป็นแนวคิดหลักของการเขียนโปรแกรม โดยเริ่มใช้ภาษาธรรมชาติของมนุษย์ (Natural language) ในแนวการเขียนโปรแกรมแบบตรรกะและโมเดล ภายใต้สภาวะแวดล้อมของโลกจริง ๆ สนับสนุนและยืดหยุ่นต่อพฤติกรรมที่เปลี่ยนไปภายใต้การจัดการของคอมพิวเตอร์ ปัจจุบันเราเริ่มเห็นหุ่นยนต์และซอฟต์แวร์ที่สามารถโต้ตอบกับมนุษย์ได้ ตัวอย่างภาษาช่วงเริ่มต้นของยุคนี้เช่น ภาษาโปรล็อก 2 (Prolog 2) เป็นโปรแกรมที่สามารถสอบถามข้อมูลด้วยประโยคที่เราใช้สื่อสารกัน ระบบผู้เชี่ยวชาญ (Expert System) ที่สามารถวินิจฉัยในเรื่องที่ระบบผู้เชี่ยวชาญได้ ระบบฐานความรู้ (Knowledge Base System) ที่เก็บรวบรวมและประมวลผลข้อมูลอย่างชาญฉลาด เมื่อถึงยุคปลายของยุคนี้ทั้งฮาร์ดแวร์และซอฟต์แวร์จะเป็นยุคเอไอ (AI) อย่างสมบูรณ์ การเขียนโปรแกรมจะใช้ภาษาธรรมชาติ มนุษย์สามารถเขียนโปรแกรมคอมพิวเตอร์โดยภาษาที่เราใช้สื่อสารกัน

ข้อดีของภาษายุคนี้คือ การเขียนโปรแกรมทำได้โดยไม่ต้องเรียนรู้ภาษาคอมพิวเตอร์ให้ยุ่งยาก แต่ข้อเสียคือโปรแกรมเมอร์อาจถูกจำกัดความสามารถเนื่องจากมีเครื่องมือช่วยในการเขียนโปรแกรมที่ฉลาดมากเกินไป ทำให้ไม่สามารถจัดการบางสิ่งบางอย่างตามที่ต้องการได้ กล่าวคือในระบบต่าง ๆ เหล่านั้นจะต้องทำการแปลคำสั่งเช่นเดียวกับยุคที่ผ่าน ๆ มา แต่ผู้เขียนโปรแกรมไม่รู้สึกรู้ว่าต้องทำอะไรเหมือนที่ผ่านมาเพราะระบบเหล่านั้นมีความฉลาดเปรียบเสมือนรวมการแปลและการสั่งการจากภาษาธรรมชาติแล้วแปลจนกระทั่งสามารถเป็นภาษาเครื่องได้ ภาพที่ 1.7 แสดงแนวคิดดังกล่าวโดยที่ไม่แยกให้เห็นความซับซ้อนของระบบต่าง ๆ ที่เกิดจากการพัฒนาโปรแกรมภาษาในยุคนี้



ภาพที่ 1.7 แสดงการพัฒนาโปรแกรมยุคที่ 5

ดังนั้นหากแบ่งระดับของโปรแกรมภาษาคอมพิวเตอร์แบ่งได้ดังนี้ ภาษาเครื่อง ภาษาระดับต่ำภาษาระดับสูงและภาษาธรรมชาติ ซึ่งต่างก็เป็นเครื่องมือสำหรับการเขียนโปรแกรมคอมพิวเตอร์ทั้งสิ้น ในหัวข้อถัดไปจะกล่าวแนวคิดในการเขียนโปรแกรมแบบโครงสร้างที่จะใช้งานหนังสือเล่มนี้

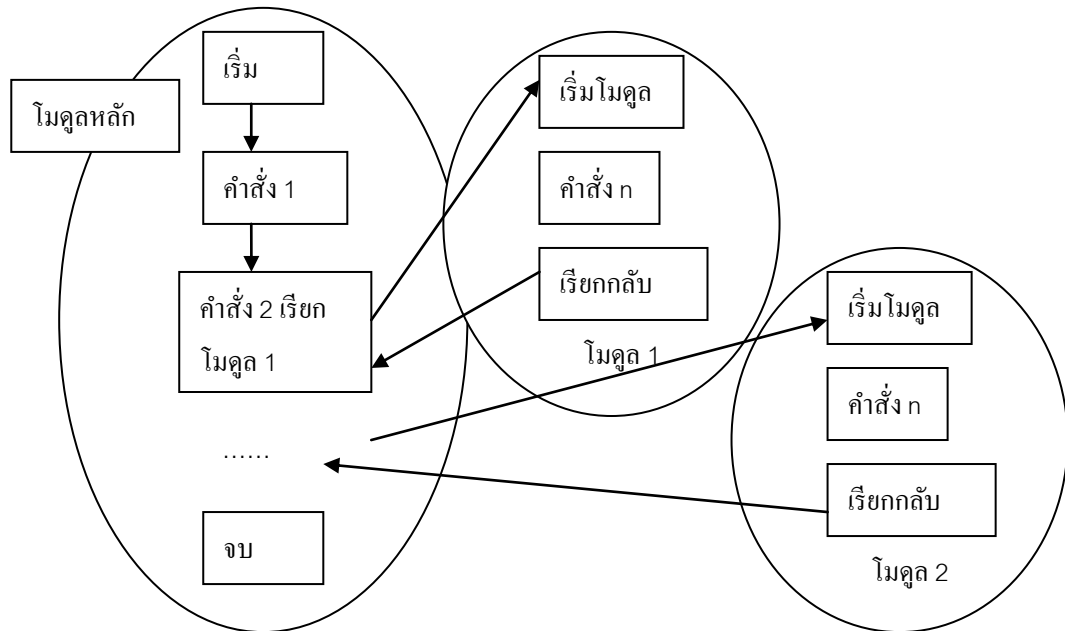
การเขียนโปรแกรมคอมพิวเตอร์แบบโครงสร้าง

การพัฒนาโปรแกรมคอมพิวเตอร์ของโปรแกรมเมอร์จะต้องเลือกโปรแกรมภาษาที่จะใช้พัฒนาก่อนเป็นลำดับแรก หลังจากนั้นจะต้องศึกษารูปแบบการใช้งานของโปรแกรมภาษา ซึ่งแต่ละภาษาจะมีรูปแบบการเขียนโปรแกรมที่แตกต่างกันในแต่ละคำสั่ง ถึงแม้แต่ละภาษาจะมีรูปแบบของคำสั่งที่แตกต่างกันแต่จะมีรูปแบบการเขียนที่เป็นรูปแบบเดียวกันที่พอจะจำแนกได้คือ การเขียนโปรแกรมแบบภาษาเครื่อง การเขียนโปรแกรมแบบภาษาโครงสร้าง การเขียนโปรแกรมแบบภาษาเชิงวัตถุ และการเขียนโปรแกรมด้วยภาษาธรรมชาติ แต่ในหนังสือเล่มนี้จะใช้การเขียนโปรแกรมแบบโครงสร้าง ดังนั้นต่อไปนี้จะนำเสนอเกี่ยวกับแนวคิดการเขียนโปรแกรมด้วยภาษาเชิงโครงสร้างและรูปแบบการเขียนโปรแกรมแบบนี้

1. แนวคิดการเขียนโปรแกรมด้วยภาษาเชิงโครงสร้าง

การพัฒนาโปรแกรมด้วยภาษาเชิงโครงสร้างนี้ถือเป็นยุคที่สามของโปรแกรมภาษาคอมพิวเตอร์ และได้รับความนิยมเป็นอย่างมาก ปัจจุบันเองก็ยังนิยมใช้กันอย่างแพร่หลายอยู่ในทุก ๆ วงการ ลักษณะของการพัฒนาโปรแกรมก็คือ จะพัฒนาเป็นโมดูล

ย่อย ๆ และให้โปรแกรมหลักได้เรียกใช้งาน โดยการเรียกใช้งานจากโปรแกรมหลักนั้น จะทำงานจากบนลงล่าง คือเริ่มต้นไปจนถึงสิ้นสุด แสดงแนวคิดดังภาพที่ 1.8



ภาพที่ 1.8 แสดงการใช้งานแบบเชิงโครงสร้าง

จากแนวคิดนี้จึงเป็นพื้นฐานการพัฒนาโปรแกรมในรูปแบบอื่น ๆ ด้วย เช่น รูปแบบเชิงวัตถุเองก็จะมีส่วนที่เป็นโครงสร้างนี้ไปฝังเป็นส่วนหนึ่งของเนื้อโปรแกรม อยู่เสมอ ซึ่งในหนังสือเล่มนี้จะใช้การเขียนโปรแกรมแบบโครงสร้างนี้

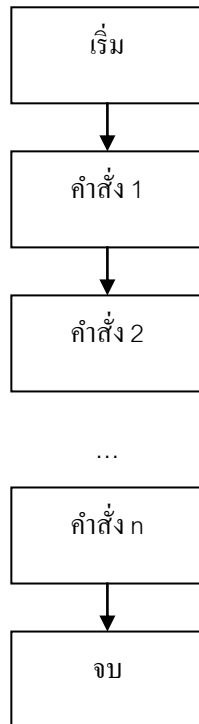
ในการเขียนโปรแกรมใดๆ ของการเขียนโปรแกรมแบบโครงสร้างนั้น อาจไม่ได้เป็นข้อบังคับว่าจะต้องมีโมดูลมากกว่าหนึ่งเสมอไป ซึ่งอาจมีเพียงโมดูลหลักเพียงโมดูล เดียวก็สามารถทำงานได้ และลักษณะของโมดูลเองก็ไม่จำเป็นต้องอยู่ภายนอกโมดูล หลักก็ได้ อาจอยู่เป็นโมดูลย่อย ๆ ภายในโมดูลหลักก็เป็นได้

2. รูปแบบการเขียนโปรแกรมแบบโครงสร้าง

กระทรวงศึกษาธิการ (2541, หน้า 9) ได้แบ่งรูปแบบการเขียนโปรแกรมออกเป็น เรียงลำดับ แบบทางเลือก และแบบวนซ้ำ ซึ่งรูปแบบการเขียนโปรแกรมแบบโครงสร้าง เริ่มจากบนลงล่าง จากซ้ายไปขวาเสมอ หากแบ่งรูปแบบการพัฒนาอาจแบ่งรูปแบบการ เขียนโปรแกรมเชิงโครงสร้างได้ดังนี้

2.1 โปรแกรมแบบเรียงลำดับ

การเขียนโปรแกรมแบบนี้จะไม่มี ความซับซ้อนใด ๆ โดยจะนำคำสั่งของโปรแกรมภาษาที่ใช้ในการเขียนมาเรียงต่อกันเป็นลำดับ ต่อเนื่องกันไปจากต้นจนจบ แสดงตัวอย่างแนวคิดการเขียนโปรแกรมดังกล่าวได้ดังภาพที่ 1.9

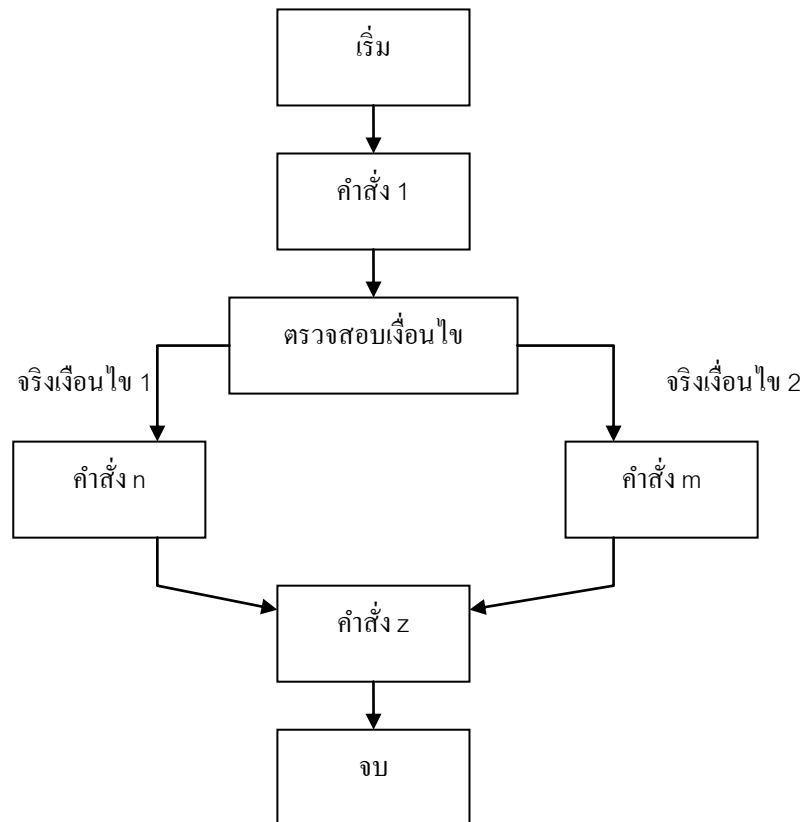


ภาพที่ 1.9 แสดงตัวอย่างแนวคิดการเขียนโปรแกรมแบบตามลำดับ

การพัฒนาโปรแกรมลักษณะนี้จะไม่มี ความซับซ้อนและเหมาะสำหรับผู้ที่เริ่มหัดเขียนโปรแกรมใหม่ ๆ และผู้ที่ต้องการฝึกหัดใช้คำสั่งของภาษาคอมพิวเตอร์ภาษาใด ๆ ใหม่ ๆ ซึ่งในหนังสือเล่มนี้บทแรก ๆ จะเป็นการเขียนโปรแกรมตามแนวทางนี้

2.2 การเขียนโปรแกรมแบบมีเงื่อนไข

การเขียนโปรแกรมแบบนี้ โดยหลักการทั่วไปจะเริ่มจากบนลงล่างตามแนวทางของโปรแกรมเชิงโครงสร้าง แต่ระหว่างทางจากบนลงล่างจะมีการตัดสินใจ โดยมีทางเลือกให้กับโปรแกรมในการเลือกทำงานอย่างใดอย่างหนึ่งตามเงื่อนไขที่ตรวจสอบได้ ตัวอย่างแนวคิดแสดงดังภาพที่ 1.10

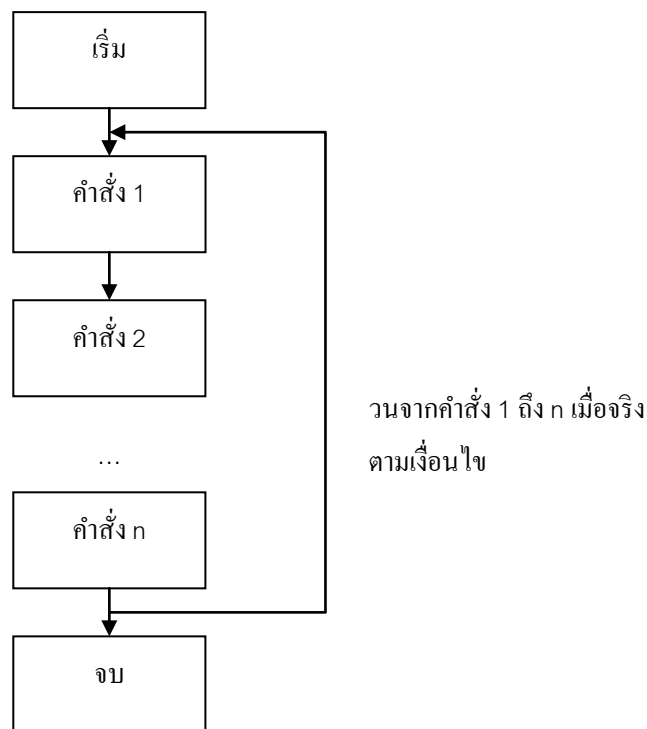


ภาพที่ 1.10 แสดงแนวคิดของการเขียนโปรแกรมแบบมีเงื่อนไข

ลักษณะของโปรแกรมหากล่าวนี้จะตรวจสอบเงื่อนไขว่าตรงตามเงื่อนไขใดแล้วจะไปทำคำสั่งตามที่กำหนดไว้ ซึ่งเงื่อนไขอาจมีมากกว่าหนึ่งเงื่อนไขก็ได้ และลักษณะของการตรวจสอบเงื่อนไขก็จะมีหลายรูปแบบเช่นกัน

2.3 การเขียนโปรแกรมแบบวนซ้ำ

การเขียนโปรแกรมแบบนี้จะใช้จุดเด่นของคอมพิวเตอร์คือสามารถทำงานเดิมซ้ำๆ กันโดยไม่เบื่อเหมือนคน หลักการเขียนแบบนี้ยังคงเขียนจากบนลงล่างและเมื่อมีกลุ่มคำสั่งใดหรือโมดูลใดต้องทำงานซ้ำ ก็จะวนซ้ำตรงตามที่ตรงตามเงื่อนไขที่กำหนดไว้ เช่นกำหนดให้วนจนครบหนึ่งร้อยรอบ เป็นต้น แนวคิดดังกล่าวแสดงได้ดังภาพที่ 1.11



ภาพที่ 1.11 แสดงตัวอย่างการเขียนโปรแกรมแบบวนซ้ำ

ลักษณะของการวนซ้ำหรือเงื่อนไขของการวนนั้นจะแตกต่างกันออกไป ขึ้นอยู่กับโปรแกรมภาษาที่ใช้ในการเขียนซึ่งจะทำให้รูปแบบการเขียนนั้นซับซ้อนแตกต่างกันออกไป

ในการเขียนโปรแกรมหนึ่ง ๆ นั้นจะผสมผสานรูปแบบการเขียนทั้งสามอยู่เสมอ เพราะว่ามีบางช่วงจะเป็นเรียงลำดับและบางช่วงจะเป็นแบบเงื่อนไข หรือแบบวนซ้ำ ซึ่งในส่วนเงื่อนไขอาจมีแบบเรียงลำดับซ้อนอยู่ หรือในแบบวนซ้ำเองก็มีแบบเงื่อนไขอยู่ด้วย ในการที่จะพิจารณาความเหมาะสมว่าจะใช้การเขียนโปรแกรมแบบใด ในช่วงใดนั้นจะขึ้นอยู่กับลักษณะของงานที่วิเคราะห์ได้เป็นตัวกำหนด

ขั้นตอนในการเขียนโปรแกรม

ธีรวัฒน์ ประกอบผล (2545, หน้า 5) ได้กล่าวว่าขั้นตอนการเขียนโปรแกรมมีดังนี้

- (1) การกำหนดและวิเคราะห์ปัญหา
- (2) การเขียนผังงานและซูดโค้ด (pseudo coding)
- (3) การเขียนโปรแกรม
- (4) การทดสอบและแก้ไขโปรแกรม
- (5) การทำเอกสารและการบำรุงรักษา

ขณะที่ กระทรวงศึกษาธิการ ได้กล่าวไว้ใน หลักการเขียนโปรแกรม คพ. ทป. 014 (2533, หน้า 46) ระบุขั้นตอนการเขียนโปรแกรมดังนี้

- (1) การวิเคราะห์ปัญหา
- (2) การออกแบบขั้นตอนการทำงานของโปรแกรม
- (3) การลงรหัสโปรแกรม
- (4) การทดสอบและแก้ไขโปรแกรม
- (5) การทำเอกสารประกอบโปรแกรม

ดังนั้นขั้นตอนของการเขียนโปรแกรมมีดังนี้

1. ศึกษาปัญหาและทำความเข้าใจสิ่งที่ต้องการของผู้ใช้

ขั้นตอนนี้เป็นขั้นตอนแรกสุดเพราะหากไม่สามารถมองปัญหาออกก็ไม่สามารถที่จะเขียนโปรแกรมหรือสร้างงานออกมาได้อย่างถูกต้องตามความต้องการอย่างแน่นอน ซึ่งในขั้นตอนนี้สิ่งที่ต้องพิจารณาและทำความเข้าใจดังต่อไปนี้

1.1 สิ่งที่โจทย์หรือผู้ใช้งานต้องการ (Output) ของงาน ซึ่งผู้เขียนจะต้องรู้อย่างชัดเจนว่าผู้ใช้งานต้องการผลการทำงานอย่างไร แนวปฏิบัติของขั้นตอนนี้คือศึกษาจากโจทย์ที่ได้รับ หรือสอบถามรวบรวมข้อมูลจากผู้ปฏิบัติงานของผู้ใช้งานจริงให้ละเอียดที่สุด

1.2 สิ่งที่โจทย์ให้มาหรือปัจจัยต่าง ๆ ที่ทำให้ได้มาซึ่งผลลัพธ์ ซึ่งก็คือสิ่งที่ต้องมีเพื่อที่จะให้เกิดผลลัพธ์เรียกว่าข้อมูลนำเข้า (Input) ดังนั้นผู้เขียนโปรแกรมจะต้องพิจารณาประเด็นนี้เป็นสิ่งสำคัญอันดับที่สอง โดยดูจากโจทย์กำหนดให้ว่ามีปัจจัยใด

ประกอบอยู่แล้วบ้าง และสิ่งใดจะต้องกำหนดหรือหาเพิ่มเติมเพื่อที่จะช่วยในการแก้ไขปัญหาดังกล่าวให้ได้ผลลัพธ์ตามต้องการ

1.3 วิธีการดำเนินงาน (Process) ซึ่งจะเป็นลำดับขั้นตอนการทำงาน เป็นกระบวนการเพื่อจะนำข้อมูลนำเข้ามาดำเนินการเพื่อให้ได้ผลลัพธ์

2. การกำหนดวิธีการทำงาน

การกำหนดวิธีการทำงานนั้นสำหรับผู้ที่จะเริ่มเขียนโปรแกรมคอมพิวเตอร์นั้นจำเป็นอย่างย้งต้องมีแนวคิดเหมือนการแก้โจทย์หรือการแก้ปัญหาทางคณิตศาสตร์ กล่าวคือจะต้องคิดเป็นขั้นตอนและกำหนดลำดับการแก้ปัญหาให้ชัดเจนดังนี้คือ

2.1 ให้เขียนลำดับขั้นตอนการแก้ปัญหาเป็นขั้นตอนให้ชัดเจนที่สุดเท่าที่สามารถทำได้

2.2 แทนลำดับขั้นตอนการแก้ปัญหาดังกล่าวด้วยแผนภาพหรือผังงาน (Flowchart)

2.3 ลงรหัสโปรแกรม (Pseudo Code) หรือรหัสเทียม ของโปรแกรม โดยรหัสเทียมดังกล่าวจะต้องดูจากผังการทำงานเป็นหลัก การลงรหัสเทียมนี้จะเขียนไว้เป็นกลาง ๆ อาจจะอิงโปรแกรมภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่งก็ได้หรือเขียนเป็นภาษาอังกฤษล้วน ๆ ก็ได้ หรือหากเป็นภาษาไทยก็ยังไม่ยอมให้ใช้ได้เช่นกัน

3. ลงมือเขียนโปรแกรม

ในการเริ่มลงมือเขียนโปรแกรมคอมพิวเตอร์นั้น ผู้เขียนจะต้องดำเนินการดังต่อไปนี้

3.1 ศึกษา รูปแบบไวยากรณ์ (syntax) ของโปรแกรมภาษานั้น ๆ ซึ่งจะเป็นกฎเกณฑ์ ข้อบังคับ หรือรูปแบบบังคับต่าง ๆ ของโปรแกรมภาษาที่ใช้ในการเขียนของโปรแกรมภาษานั้น ๆ ให้เข้าใจและแม่นยำจึงจะสามารถเขียนโปรแกรมได้อย่างมีประสิทธิภาพ

3.2 ศึกษา คำสั่งปฏิบัติการ ตัวดำเนินการ ตัวแปร หรือฟังก์ชัน (Functions) ต่าง ๆ ที่มีในภาษาที่เลือก ซึ่งในขั้นตอนนี้ผู้ใช้จะต้องดำเนินการเขียนโปรแกรมจริง ๆ จึงจะสามารถเข้าใจและใช้งานได้อย่างถูกต้อง

4. แปลจากระหัสต้นฉบับเป็นรหัสที่คอมพิวเตอร์เข้าใจ

การแปลรหัสต้นฉบับนั้นแต่ละโปรแกรมภาษาที่เลือกมาใช้เขียนอาจมีความแตกต่างกัน ใช้คำสั่งแตกต่างกัน แต่โดยทั่วไปจะขึ้นอยู่กับตัวแปลภาษาซึ่งถ้าเป็นคอมไพเลอร์ จะทำการแปลรหัสดังกล่าวเป็นอ็อบเจกต์โค้ดที่เป็นภาษาเครื่องและบันทึกเอาไว้เพื่อการเรียกใช้งานภายหลัง โดยไม่ต้องทำการแปลคำสั่งใหม่ แต่ถ้าเป็นตัวแปลภาษาที่เป็นอินเตอร์พรีเตอร์จะต้องแปลทุกครั้งเมื่อต้องการรันโปรแกรมเพื่อใช้งาน

6. การรันโปรแกรม

การรันโปรแกรมหรือการเรียกใช้งานโปรแกรมแต่ละโปรแกรมจะมีการเรียกใช้ต่างกันบ้างแต่โดยทั่วไปจะทำได้หลังจากที่คอมไพล์แล้วไม่มีข้อผิดพลาดซึ่งจะเป็นการเรียกใช้งานตามที่ต้องการผ่านระบบปฏิบัติการต่าง ๆ

สรุป

โปรแกรมภาษาคอมพิวเตอร์ คือ โปรแกรมที่ใช้เพื่อการเขียนโปรแกรมต่าง ๆ ซึ่งแตกต่างกันไปตามยุคและระดับของภาษาที่ใช้ แนวทางการเขียนโปรแกรมที่นิยมใช้คือการเขียนโปรแกรมแบบโครงสร้าง การเขียนโปรแกรมเชิงวัตถุ และในปัจจุบันยังมีการเขียนโปรแกรมแบบภาษาธรรมชาติที่กำลังเข้าสู่ยุคเริ่มต้น การออกแบบโปรแกรมเชิงโครงสร้างจะออกแบบจากบนลงล่างเสมอ โดยทั่วไปถ้างานมีความซับซ้อนจะออกแบบโดยใช้โมดูลย่อย ๆ ของแต่ละงาน แล้วนำมาประกอบกัน ซึ่งในการออกแบบจะมีรูปแบบเงื่อนไข และการวนซ้ำเข้ามาใช้งานร่วมด้วย การเขียนโปรแกรมคอมพิวเตอร์จะต้องออกแบบโปรแกรมก่อน ด้วยการวิเคราะห์ความต้องการ ออกแบบและวางแผนการเขียนโปรแกรมก่อน จึงลงมือเขียนตามความต้องการที่วิเคราะห์ไว้แล้ว คอมไพล์และทดลองรันโปรแกรม เพื่อปรับปรุงที่ผิด และแก้ไขให้ถูกต้องก่อนการนำไปใช้งานจริง

แบบฝึกหัดท้ายบท

1. จงอธิบายความหมายของคำต่อไปนี้พอสังเขป
 - คอมพิวเตอร์
 - โปรแกรมคอมพิวเตอร์
 - โปรแกรมภาษาคอมพิวเตอร์
 - การเขียนโปรแกรมคอมพิวเตอร์
2. ยุคของโปรแกรมภาษาคอมพิวเตอร์แบ่งได้อย่างไรบ้าง จงอธิบาย
3. จงอธิบายและยกตัวอย่างระดับของโปรแกรมภาษาคอมพิวเตอร์
4. จงยกตัวอย่างโปรแกรมภาษาคอมพิวเตอร์ในยุคต่าง ๆ ตั้งแต่ยุคที่ 1-5 ยุคละ 3 โปรแกรมภาษา
5. จงวิเคราะห์ว่าทำไมภาษาคอมพิวเตอร์จึงใกล้เคียงกับภาษาของมนุษย์มากขึ้นทุกขณะ
6. จงคาดคะเนเกี่ยวกับแนวทางการเขียนโปรแกรมคอมพิวเตอร์ในอนาคตอีก 50 ข้างหน้า พร้อมอธิบายเหตุผล
7. จงอธิบายว่าการเขียนโปรแกรมแบบโครงสร้างว่ามีจุดดีจุดด้อยอย่างไร
8. จงอธิบายขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์
9. จงอธิบายขั้นตอนว่าหากท่านได้รับการกิจให้เขียนโปรแกรมระบบบุคลากรให้กับหน่วยงานท่านจะเริ่มต้น และดำเนินการอย่างไร
10. ท่านคิดว่าในอนาคตที่ก้าวผ่านขั้นยุคที่ N ไปแล้วการเขียนโปรแกรมคอมพิวเตอร์จะเป็นอย่างไร

บทที่ 2

การออกแบบโปรแกรมด้วยผังงาน

การเขียนโปรแกรมแบบโครงสร้างมีเครื่องมือที่ช่วยในการออกแบบ เพื่อสื่อสารกับผู้เขียนโปรแกรมให้เข้าใจตรงกันในการเขียนโปรแกรม และยังช่วยในการตรวจสอบโปรแกรมให้ถูกต้องตรงตามความต้องการของผู้ใช้อีกด้วย เครื่องดังกล่าวเรียกว่าผังงานแบบโครงสร้าง (Structure flowchart) ในบทนี้จะกล่าวถึงสัญลักษณ์ต่าง ๆ ของผังงานเพื่อใช้ในการออกแบบ โดยการนำเสนอตามแนวทางการเขียนโปรแกรมแบบโครงสร้าง ได้แก่ การออกแบบตามลำดับ การออกแบบเงื่อนไข การออกแบบวนซ้ำ และยกตัวอย่างในการวิเคราะห์และออกแบบอีกด้วย

สัญลักษณ์ของผังงาน

ครรชิต มาลัยวงศ์ และ วิจิต คุ้มวัตร (2536 หน้า 39-40) กล่าวว่า ผังงานเป็นวิธีการแสดงแบบซอฟต์แวร์ที่เก่าแก่ที่สุด ซึ่งประกอบด้วย ผังระบบที่แสดงขั้นตอนการทำงานของระบบทั้งหมด และผังโปรแกรมที่แสดงรายละเอียดของโปรแกรม

กระทรวงศึกษาธิการ (2533 หน้า 90) เขียนไว้ในหนังสือเรียน “หลักการเขียนโปรแกรม คพ.ทป. 014 ระดับม.ปลาย” ว่า ผังงาน คือเครื่องมือที่ใช้ช่วยในการเขียนโปรแกรมแบบหนึ่ง โดยใช้เครื่องหมายสัญลักษณ์ต่าง ๆ มาประกอบต่อเนื่องกัน

Wikipedia (2007) กล่าวว่า ผังงานเป็นหนึ่งในเจ็ดของเครื่องมือควบคุมคุณภาพ โดยทั่วไปจะใช้ในการนำเสนองานทางด้านธุรกิจ ซึ่งจะอธิบายการทำงานของฟังก์ชันงานต่าง ๆ และรวมไปถึงหน้าที่ความรับผิดชอบของแต่ละส่วน

ดังนั้น ผังงานคือ เครื่องมือสำหรับออกแบบระบบหรือโปรแกรม มีอยู่ 2 ประเภท คือผังงานระบบ (System flowchart) ที่ใช้ในการออกแบบระบบโดยรวมให้ทราบว่าระบบที่จะพัฒนาขึ้นประกอบไปด้วยสิ่งใดบ้าง มีข้อมูลใดเข้ามาเกี่ยวข้องบ้าง และผังงานอีกประเภทหนึ่ง คือผังโปรแกรม (program flowchart) ที่จะอธิบายว่าโปรแกรมที่จะ