

บทที่ 2 การจำลองความคิด

เนื้อหา

2.1 ขั้นตอนการพัฒนาซอฟต์แวร์

- การวิเคราะห์และกำหนดรายละเอียดของปัญหา
- การเลือกเครื่องมือและออกแบบขั้นตอนวิธี
- การดำเนินการแก้ปัญหา
- การตรวจสอบและปรับปรุง

2.2 การจำลองความคิด

- ข้อความหรือคำบรรยาย (Pseudo code)
- สัญลักษณ์หรือแผนภาพ (Flowchart)
 - แบบลำดับ
 - แบบทางเลือก
 - แบบทำซ้ำ

2.3 การเขียนโปรแกรม

- โครงสร้างแบบลำดับ
- โครงสร้างแบบมีทางเลือก
- โครงสร้างแบบทำซ้ำ

วัตถุประสงค์

- 1. อธิบายขั้นตอนการพัฒนาซอฟต์แวร์และอธิบายถึงวิธีการในแต่ละขั้นตอนได้
- 2. อธิบายความหมายและเหตุผลในการเขียนแผนภาพที่ใช้ในการออกแบบโปรแกรมได้
- 3. บอกสัญลักษณ์และสามารถเขียนผังงานที่ใช้ในการออกแบบโปรแกรมได้

2.1 ขั้นตอนการพัฒนาซอฟต์แวร์

ในชีวิตประจำวันทุกคนต้องเคยพบกับปัญหาต่างๆ ไม่ว่าจะเป็นปัญหาด้านการเรียน การงาน การเงิน หรือแม้แต่การเล่นเกม เมื่อพบกับปัญหาแต่ละคนมีวิธีที่จะจัดการหรือแก้ปัญหาเหล่านั้นแตกต่างกันไป ซึ่งแต่ละวิธีการอาจให้ผลลัพธ์ที่เหมือนหรือแตกต่างกันเล็กน้อย ทั้งนี้ขึ้นอยู่กับความรู้ ความสามารถ และประสบการณ์ของบุคคลผู้นั้นอย่างไรก็ตาม หากเรานำวิธีการแก้ปัญหาต่างวิธีนั้นมาวิเคราะห์ให้ดี จะพบว่าสามารถสรุปวิธีการเหล่านั้นเป็น

ทฤษฎีซึ่งมีรูปแบบที่แน่นอนได้ และบางครั้งต้องอาศัยการเรียนรู้ในระดับสูงเพื่อแก้ปัญหาบางอย่างให้สมบูรณ์แบบแต่ก่อนที่เราจะศึกษาต่อไป ลองพิจารณาปัญหาต่อไปนี้

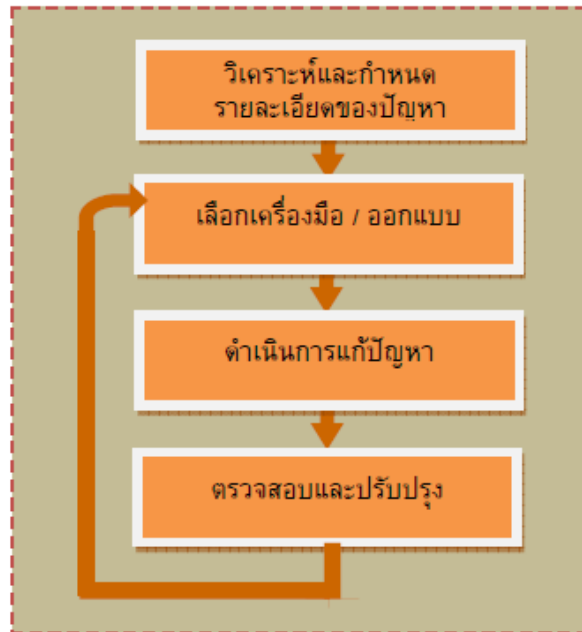
ตัวอย่างที่ 1 เกมทายใจ คือ เกมให้ผู้เล่นทายตัวเลข 3 ตัว ในการเล่นเกมต้องใช้ผู้เล่น 2 คน คนที่หนึ่ง คือ ผู้กำหนด เป็นคนกำหนดเลข 3 ตัว ที่ไม่ซ้ำกันโดยเลือกจากกลุ่มตัวเลข 1-9 และอีกคนหนึ่งคือ ผู้ทาย เป็นผู้ทายเลข 3 ตัว ที่ไม่ซ้ำกัน ที่ผู้กำหนดได้กำหนดเอาไว้แล้ว หลังจากทีผู้ทายทายเลขแต่ละครั้ง ผู้กำหนดต้องให้รายละเอียดว่าตัวเลขที่ทายมานั้นถูกต้องกี่ตัว และในกรณีที่ตัวเลขที่ทายมาถูกตำแหน่งด้วยก็ต้องบอกว่าถูกตำแหน่งกี่ตัว เช่น ถ้าตัวเลขที่กำหนดไว้เป็น 815 และ ผู้ทายทายว่า 123 ผู้กำหนดต้องแจ้งว่าตัวเลขที่ทายนั้นถูก 1 ตัว และไม่มีตัวใดถูกตำแหน่ง รูปที่ 1 เป็นตารางแสดงข้อมูลการเล่นเกมหาใจ

ตัวเลขที่ทาย	จำนวนตัวเลขที่ถูก	จำนวนตำแหน่งที่ถูก
123	1	-
415	2	2
425	1	1
416	1	1
715	2	2
815	3	3

รูปที่ 1 ข้อมูลการเล่นเกมหาใจ

จะเห็นว่าการแก้ปัญหาดังกล่าวข้างต้น นอกจากจะใช้วิธีลองผิดลองถูกในการทายครั้งแรกๆ แล้วยังมีการใช้เหตุผลประกอบการแก้ปัญหาซึ่งเราเรียกวิธีการดังกล่าวว่า "วิธีขจัด" (method of elimination) กล่าวคือ จะแยกข้อมูลออกเป็นกรณีที่เป็นไปไม่ได้ทิ้ง จนเหลือกรณีที่เป็นไปได้ วิธีการดังกล่าวสามารถอธิบายได้ว่าทำไมจึงคิดหรือทำเช่นนั้น รูปแบบของการใช้เหตุผลประกอบการแก้ปัญหอาจแตกต่างกัน ขึ้นอยู่กับเงื่อนไข ในปัญหาบางปัญหา อาจจะขจัดให้เหลือกรณีเดียวไม่ได้ ก็อาจจะทำให้เหลือกรณีน้อยที่สุด นอกจากเกมทายใจเรายังมีเกมลับสมองที่น่าสนใจและท้าทายความสามารถในการแก้ปัญหาอยู่อีกมากมายที่หาเล่นได้ตามเว็บไซต์

นอกจากวิธีการแก้ปัญหาย่อยที่ยกตัวอย่างมาซึ่งได้แก่ วิธีการลองผิดลองถูก การใช้เหตุผล การใช้วิธีขจัด ยังมีวิธีการแก้ปัญหามากมายที่ผู้แก้ปัญหามารถเลือกใช้ให้เข้ากับตัวปัญหาและประสบการณ์ของผู้แก้ปัญหเอง แต่อย่างไรก็ตาม วิธีการเหล่านี้ล้วนมีขั้นตอนที่คล้ายคลึงกัน และจากการศึกษาพฤติกรรมในการเรียนรู้และแก้ปัญหาของมนุษย์พบว่าโดยปกติมนุษย์มีกระบวนการในการแก้ปัญหประกอบด้วย 4 ขั้นตอนดังรูปที่ 2



รูปที่ 2 แสดงกระบวนการแก้ปัญหา

1) การวิเคราะห์และกำหนดรายละเอียดของปัญหา (State the problem)

ขั้นตอนนี้เป็นขั้นตอนแรกสุดก่อนที่จะลงมือแก้ปัญหา แต่ผู้แก้ปัญหามักจะมองข้ามความสำคัญของขั้นตอนนี้อยู่เสมอ จุดประสงค์ของขั้นตอนนี้ คือ การทำความเข้าใจกับปัญหาเพื่อแยกให้ออกว่าข้อมูลที่กำหนดมาในปัญหาหรือเงื่อนไขของปัญหาคืออะไร และสิ่งที่ต้องการคืออะไร อีกทั้งวิธีการที่ใช้ประมวลผล ในการวิเคราะห์ปัญหาใด กล่าวโดยสรุปแล้วองค์ประกอบในการวิเคราะห์มีอยู่ 3 องค์ประกอบ

- 1.1) การระบุข้อมูลเข้า ได้แก่ การพิจารณาข้อมูลและเงื่อนไขที่กำหนดมาในปัญหา
- 1.2) การระบุข้อมูลออก ได้แก่ การพิจารณาเป้าหมายหรือสิ่งที่ต้องการคำตอบ
- 1.3) การกำหนดวิธีประมวลผล ได้แก่ การพิจารณาขั้นตอนวิธีการได้มาซึ่งคำตอบหรือข้อมูลออก

ตัวอย่างที่ 2 แสดงการวิเคราะห์และกำหนดรายละเอียดของการหาค่าเฉลี่ยของจำนวนเต็ม 5 จำนวน ได้แก่ 0 3 4 8 และ 12

1. การระบุข้อมูลเข้า ในที่นี้โจทย์กำหนดให้หาค่าเฉลี่ยของจำนวนเต็ม 5 จำนวน ดังนั้น ข้อมูลเข้า ได้แก่จำนวน 0 3 4 8 และ 12
2. การระบุข้อมูลออก จากโจทย์สิ่งที่เป็นคำตอบของปัญหา คือค่าเฉลี่ย (x) ของจำนวนทั้งห้า
3. การกำหนดวิธีประมวลผล จากสิ่งที่โจทย์ต้องการ "ค่าเฉลี่ย" หมายถึง ผลรวมของจำนวนทั้ง 5 หารด้วย 5 ดังนั้น ขั้นตอนของการประมวลผลประกอบด้วย

- 3.1 รับค่าจำนวนทั้ง 5 จำนวน
- 3.2 นำจำนวนเต็มทั้ง 5 มาบวกเข้าด้วยกัน

3.3 นำผลลัพธ์จากข้อ 3.2 มาหารด้วย 5

ตัวอย่างที่ 3 แสดงการวิเคราะห์และกำหนดรายละเอียดของการหาค่า x เมื่อ x คือจำนวนเต็มจำนวนหนึ่งในกลุ่มจำนวนเต็ม 5 จำนวนที่มีค่าเฉลี่ยเป็น 10 และจำนวนอีก 4 จำนวนได้แก่ 3 4 8 และ 12

1. การระบุข้อมูลเข้า จากโจทย์ข้อมูลเข้า ได้แก่ 2.1 จำนวนอีก 4 จำนวน คือ 3 4 8 12 ค่าเฉลี่ยของจำนวนทั้ง 5 จำนวน คือ 10
2. การระบุข้อมูลออก จากโจทย์สิ่งที่เป็ผลลัพธ์ คือ ค่า x
3. การกำหนดวิธีการประมวลผล จากโจทย์และความหมายของ "ค่าเฉลี่ย" เราสามารถสรุปขั้นตอนของการประมวลผลได้ดังนี้

3.1 หาค่าผลรวมของจำนวนเต็มทั้ง 5 โดยนำค่าเฉลี่ยคูณด้วยจำนวนของเลขจำนวนเต็ม นั่นคือ $10 * 5 = 50$

3.2 จากความหมายของ "ผลรวม" จะได้ $3+4+8+12+x = 50$

3.3 แก่สมการ $27 + x = 50$ (จะได้ $x = 23$ ซึ่งคือผลลัพธ์)

2) การเลือกเครื่องมือและออกแบบขั้นตอนวิธี (Tools and Algorithm development)

ขั้นตอนนี้เป็นขั้นตอนของการวางแผนในการแก้ปัญหาอย่างละเอียดถี่ถ้วน หลังจากที่เรารวบรวม เข้าใจกับปัญหา พิจารณาข้อมูลและเงื่อนไขที่มีอยู่ และสิ่งที่ต้องการหาแล้วในขั้นตอนที่ 1 เราสามารถคาดคะเน วิธีการที่เราจะใช้ในการแก้ปัญหาระบบการนี้จำเป็นอาศัยประสบการณ์ของผู้แก้ปัญหาเป็นหลักหากผู้แก้ปัญหา เคยพบกับปัญหานั้นมาแล้วก็สามารถดำเนินการตามแนวทางที่เคยปฏิบัติมา ขั้นตอนนี้จะเริ่มจากการเลือก เครื่องมือที่ใช้ในการแก้ปัญหา โดยพิจารณาความเหมาะสมระหว่างเครื่องมือกับเงื่อนไขต่างๆ ของปัญหา ซึ่งหมายรวมถึงความสามารถของเครื่องมือในการแก้ปัญหาดังกล่าว และสิ่งที่สำคัญคือความคุ้นเคยในการใช้งานเครื่องมือ นั้นๆ ของผู้แก้ปัญหา

อีกสิ่งหนึ่งที่สำคัญในการแก้ปัญหา คือ ยุทธวิธีที่ใช้ในการแก้ปัญหาหรือที่เราเรียกว่า ขั้นตอนวิธี (algorithm) ในการแก้ปัญหา หลังจากที่เรารู้เครื่องมือช่วยแก้ปัญหาแล้ว ผู้แก้ปัญหาก็ต้องวางแผนว่าจะใช้ เครื่องมือดังกล่าวอย่างไรเพื่อให้ได้ผลลัพธ์ที่ถูกต้องและดีที่สุด ในการออกแบบขั้นตอนวิธีในการแก้ปัญหา ผู้ แก้ปัญหาควรใช้แผนภาพหรือเครื่องมือในการแสดงขั้นตอนการทำงานเพื่อให้ถ่ายทอดความเข้าใจเช่น ผังงาน (flowchart) รหัสจำลอง (pseudo code) การใช้เครื่องมือช่วยออกแบบดังกล่าวนอกจากแสดงกระบวนการที่ ชัดเจนแล้ว ยังช่วยให้ผู้แก้ปัญหามีการค้นหาข้อผิดพลาดของวิธีการที่ใช้ได้ง่ายและแก้ไขได้อย่างรวดเร็ว

3) การดำเนินการแก้ปัญหา (Implementation)

หลังจากที่ได้ออกแบบขั้นตอนวิธีเรียบร้อยแล้ว ขั้นตอนนี้เป็นขั้นตอนที่ต้องลงมือแก้ปัญหาโดยใช้เครื่องมือที่ได้เลือกไว้ หากการแก้ปัญหาดังกล่าวใช้คอมพิวเตอร์เข้ามาช่วยงาน ขั้นตอนนี้ก็เป็นการใช้โปรแกรมสำเร็จ หรือใช้ภาษาคอมพิวเตอร์เขียนโปรแกรมแก้ปัญหา ขั้นตอนนี้ต้องอาศัยความรู้เกี่ยวกับเครื่องมือที่เลือกใช้ ซึ่งผู้แก้ปัญหาต้องศึกษาที่เข้าใจและเชี่ยวชาญ ในการดำเนินการอาจพบแนวทางที่ดีกว่าที่ออกแบบไว้ก็สามารถปรับเปลี่ยนได้

4) การตรวจสอบและปรับปรุง (Refinement)

หลังจากที่ลงมือแก้ปัญหาแล้ว ต้องตรวจสอบให้แน่ใจว่าวิธีการนี้ให้ผลลัพธ์ที่ถูกต้อง โดยผู้แก้ปัญหาต้องตรวจสอบว่าขั้นตอนวิธีที่สร้างขึ้นสอดคล้องกับรายละเอียดของปัญหา ซึ่งได้แก่ ข้อมูลเข้า และข้อมูลออก เพื่อให้มั่นใจว่าสามารถรองรับข้อมูลเข้าได้ในทุกกรณีอย่างถูกต้องและสมบูรณ์ ในขณะเดียวกันก็ต้องปรับปรุงวิธีการ เพื่อให้การแก้ปัญหานี้ได้ผลลัพธ์ที่ดีที่สุด

2.2 การจำลองความคิด

ขั้นตอนที่สำคัญในการแก้ปัญหาคือการวางแผน การวางแผนที่ดีจะช่วยให้การแก้ปัญหาเป็นไปได้อย่างง่ายดาย ผู้ที่สามารถวางแผนในการแก้ปัญหาได้ตืนอกจากจะต้องใช้ประสบการณ์ ความรู้ และความมีเหตุผลแล้ว ยังควรรู้จักวางแผนให้เป็นขั้นตอนอย่างเป็นระเบียบด้วย การจำลองความคิดเป็นส่วนหนึ่งในขั้นตอนที่สองของการแก้ปัญหา การจำลองความคิดออกมาในลักษณะเป็นข้อความ หรือเป็นแผนภาพจะช่วยให้สามารถแก้ปัญหาได้ดี โดยเฉพาะปัญหาที่ยุ่งยากซับซ้อน การวางแผนจะเป็นแนวทางในการดำเนินการแก้ปัญหาต่อไป อีกทั้งเป็นการแสดงแบบเพื่อให้ผู้ที่เกี่ยวข้องได้เข้าใจ และสามารถปฏิบัติตามในแนวทางเดียวกัน ทั้งนี้ก็ด้วยวัตถุประสงค์อย่างเดียวกับกลุ่มกิจการก่อสร้าง ซึ่งจำเป็นต้องมีแบบแปลนเป็นเครื่องมือติดต่อสื่อสารระหว่างผู้ออกแบบและผู้ก่อสร้าง

แบบแปลนเหล่านั้นจะอยู่ในรูปลักษณะของการวาดภาพหรือแสดงเครื่องหมายซึ่งเป็นที่เข้าใจกันระหว่างผู้เกี่ยวข้อง แบบแปลนจะต้องจัดทำให้เสร็จก่อนที่จะลงมือก่อสร้าง โดยผ่านการตรวจสอบทบทวนและพิจารณาจากผู้เกี่ยวข้องหลายฝ่าย เมื่อเห็นว่าเป็นที่ถูกต้องและพอใจของทุกฝ่ายแล้ว จึงก่อสร้างตามแบบนั้น แต่ถ้ายังไม่เป็นที่พอใจ ก็จะพิจารณาแก้ไขแบบแปลนส่วนนั้น ๆ เสียก่อนจะได้ไม่ต้องรื้อถอนหรือทุบทิ้งภายหลัง และเมื่อต้องการซ่อมแซมหรือต่อเติมก็นำเอาแบบแปลนเดิมมาตรวจสอบและเพิ่มแบบแปลนในส่วนนั้นได้โดยง่าย การใช้แบบแปลนจึงเป็นสิ่งที่จำเป็นระหว่างช่างก่อสร้าง ผู้ออกแบบและผู้เกี่ยวข้องอื่น ๆ เป็นอย่างมาก เพราะประหยัดเวลา ค่าใช้จ่ายและเข้าใจง่าย เมื่อสรุปรวมแล้วแบบแปลนเหล่านั้นก็คือข้อตกลงให้สร้างอาคารของผู้จ้างกับผู้รับจ้างที่อยู่ในรูปแบบกะทัดรัด แทนที่จะเขียนเป็นข้อความที่เป็นลายลักษณ์อักษรอย่างยืดยาว และยังเป็นเครื่องมือให้ช่างใช้ในการก่อสร้างอีกด้วย

เครื่องมือที่ใช้ในการจำลองความคิดมักจะประกอบขึ้นด้วยเครื่องหมายที่แตกต่างกันหลายอย่าง แต่พอสรุปได้ เป็น 2 ลักษณะ ได้แก่

1. ข้อความหรือคำบรรยาย
2. สัญลักษณ์

2.2.1) ข้อความหรือคำบรรยาย

เป็นการเขียนเค้าโครงด้วยการบรรยายเป็นภาษาที่มนุษย์ใช้สื่อสารกัน เพื่อให้ทราบถึงขั้นตอนการทำงานของโปรแกรมแต่ละตอน ในบางครั้งอาจใช้คำสั่งของภาษาที่ใช้เขียนโปรแกรมก็ได้การเขียนข้อความเพื่อการบรรยายขั้นตอนวิธีในการแก้ปัญหาทางคอมพิวเตอร์ สามารถเรียกอีกอย่างหนึ่งได้ว่า รหัสเทียม (pseudo code)

ตัวอย่างที่ 4 คำบรรยายแสดงขั้นตอนการเปลี่ยนยางรถเมื่อยางแตกขณะขับรถ

1. คลายสกรูยึดล้อ
2. จอดรถหลบข้างทาง
3. นำแม่แรงออกยกรถ
4. ถอดล้อออก นำยางอะไหล่มาเปลี่ยน
5. ขึ้นสกรูเข้า เก็บยางที่ชำรุดเพื่อไปซ่อม
6. คลายแม่แรง เก็บแม่แรง

หลักการทั่วไปในการเขียนรหัสเทียม

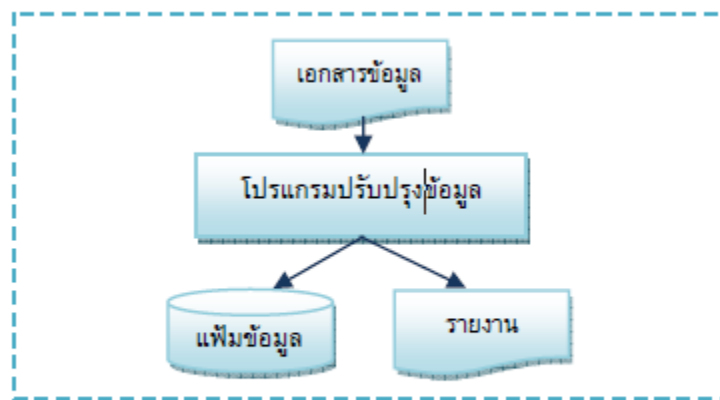
1. สัญลักษณ์ที่ใช้ในการดำเนินการทางคณิตศาสตร์ต่างๆ จะถูกใช้งานตามปกติ คือ “+” สำหรับการบวก “-” สำหรับการลบ “*” สำหรับการคูณ และ “/” สำหรับการหาร
2. ชื่อข้อมูลแทนจำนวนที่จะถูกดำเนินการ
3. การกำหนดค่าให้กับชื่อข้อมูล เช่น เมื่อเราต้องการกำหนดให้ ชื่อข้อมูล pi มีค่าเท่ากับ 3.14 สามารถเขียนได้ด้วยข้อความ $pi \leftarrow 3.14$ หรือ $pi := 3.14$ หรือ $pi = 3.14$ ในการกำหนดค่าทางคอมพิวเตอร์ด้านซ้ายของเครื่องหมายมักใช้แทน *ที่เก็บข้อมูล* และด้านขวาแทน *ข้อมูลที่ต้องการนำไปเก็บ* (ดังนั้นหากใช้ข้อความว่า $3.14 = pi$ ถือว่าไม่ถูกต้องตามความหมายนี้)
4. คำสงวนบางคำที่ใช้ในภาษาระดับสูงทั่วไปอาจถูกนำมาใช้ เช่น Read หรือ Enter สำหรับการรับข้อมูลเข้า และ Write หรือ Print สำหรับการแสดงข้อมูลออก
5. การเพิ่มหรือลดระยะย่อหน้าอย่างเหมาะสม เพื่อแสดงระดับของขั้นตอนการทำงานในโครงสร้างควบคุมการทำงานในกลุ่มเดียวกัน

2.2.2) สัญลักษณ์หรือแผนภาพ

เครื่องหมายรูปแบบต่างๆ ซึ่งใช้สำหรับสื่อสารความหมายให้เข้าใจตรงกัน สถาบันมาตรฐานแห่งชาติอเมริกัน (The American National Standard Institute, ANSI) ได้กำหนดสัญลักษณ์ไว้เป็นมาตรฐานแล้วสมควรนำไปใช้ได้ตามความเหมาะสมต่อไป ซึ่งมีรายละเอียดรูปแบบและความหมายที่ควรทราบตามตารางต่อไปนี้

การนำสัญลักษณ์ไปใช้เพื่อแสดงขั้นตอนการทำงานต่างๆ ของงานหรือโปรแกรม รวมถึงแสดงการไหลของข้อมูลในระบบตั้งแต่แรกจนได้ผลลัพธ์ตามต้องการเรียกว่า การเขียนผังงาน(Flowchart) ซึ่งสามารถแบ่งออกได้เป็น 2 แบบคือ

1. **ผังงานระบบ(System Flowchart)** หมายถึง ผังงานที่แสดงขั้นตอนต่างๆ ในการทำงานของระบบ ช่วยอธิบายลำดับการทำงานของส่วนต่างๆ ในระบบ เช่น การนำข้อมูลเข้า(Input) ถูกเก็บอยู่ที่ใดบ้าง ใช้สื่อบันทึกข้อมูลแบบใด ลักษณะของการประมวลผล ตลอดจนลักษณะของผลลัพธ์(Output) ผังงานระบบจะช่วยอำนวยความสะดวกให้แก่ผู้เขียนโปรแกรมและผู้ที่เกี่ยวข้องที่ต้องการทำความเข้าใจการทำงานของระบบ ตัวอย่างดังรูปที่ 3



รูปที่ 3 ผังงานระบบ

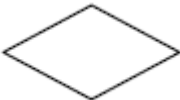
2. **ผังงานโปรแกรม(Program Flowchart)** หมายถึง ผังงานที่ใช้ในการแสดงการทำงานของโปรแกรมโดยละเอียดในแต่ละขั้นตอน ผังงานโปรแกรมเป็นสิ่งจำเป็นสำหรับผู้เขียนโปรแกรม เพราะต้องใช้เป็นแนวทางในการเขียนโปรแกรมและเมื่อโปรแกรมเกิดข้อผิดพลาดการเข้าไปวิเคราะห์ผังงานโปรแกรมจะทำได้ง่ายกว่าการเข้าไปวิเคราะห์ตัวโปรแกรมโดยตรง

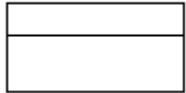
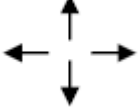
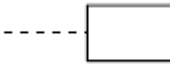
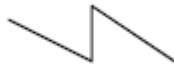
ประโยชน์ของผังงาน

1. ใช้แทนการจำลองความคิด ช่วยให้เข้าใจลำดับและความสัมพันธ์ระหว่างขั้นตอนในการทำงานต่างๆ

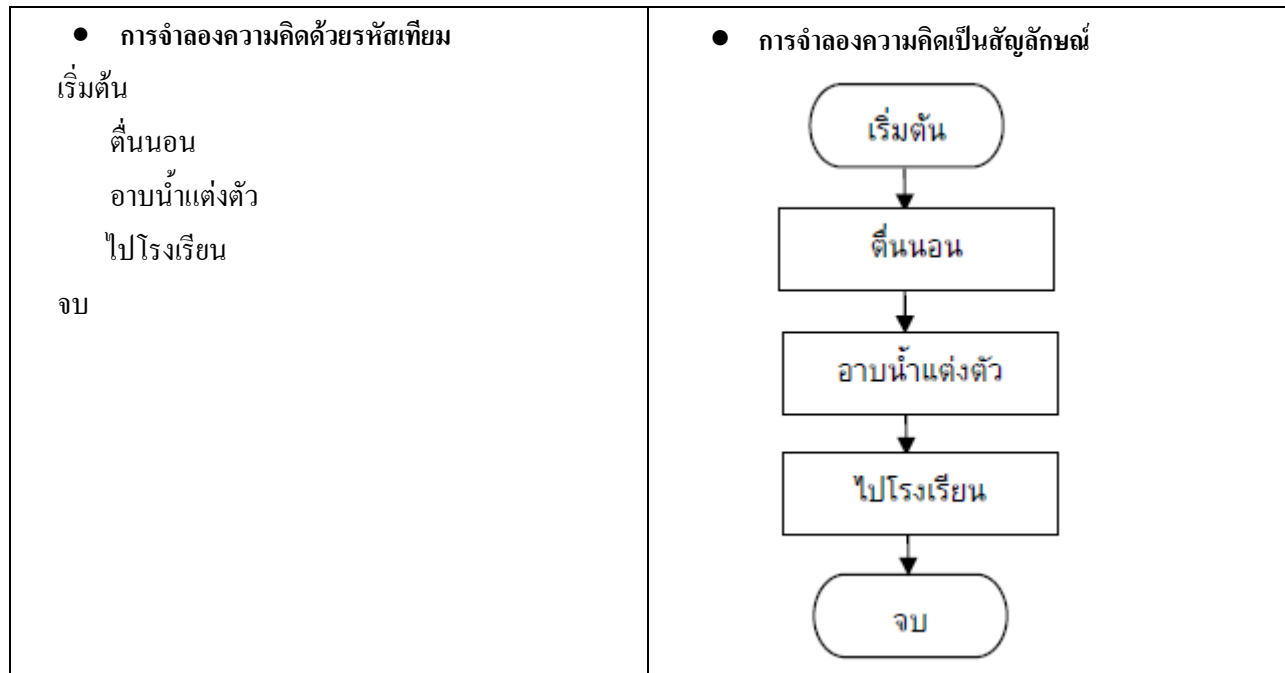
2. ใช้เป็นสื่อกลางในการติดต่อประสานความคิดระหว่างผู้ที่เกี่ยวข้อง เช่น นักวิเคราะห์ระบบ(systems analyst) นักเขียนโปรแกรม(programmer)
3. ช่วยในการทดสอบหรือทบทวนขั้นตอนการทำงาน เพื่อหาข้อผิดพลาด

ตารางที่ 1 แสดงสัญลักษณ์และความหมายของสัญลักษณ์

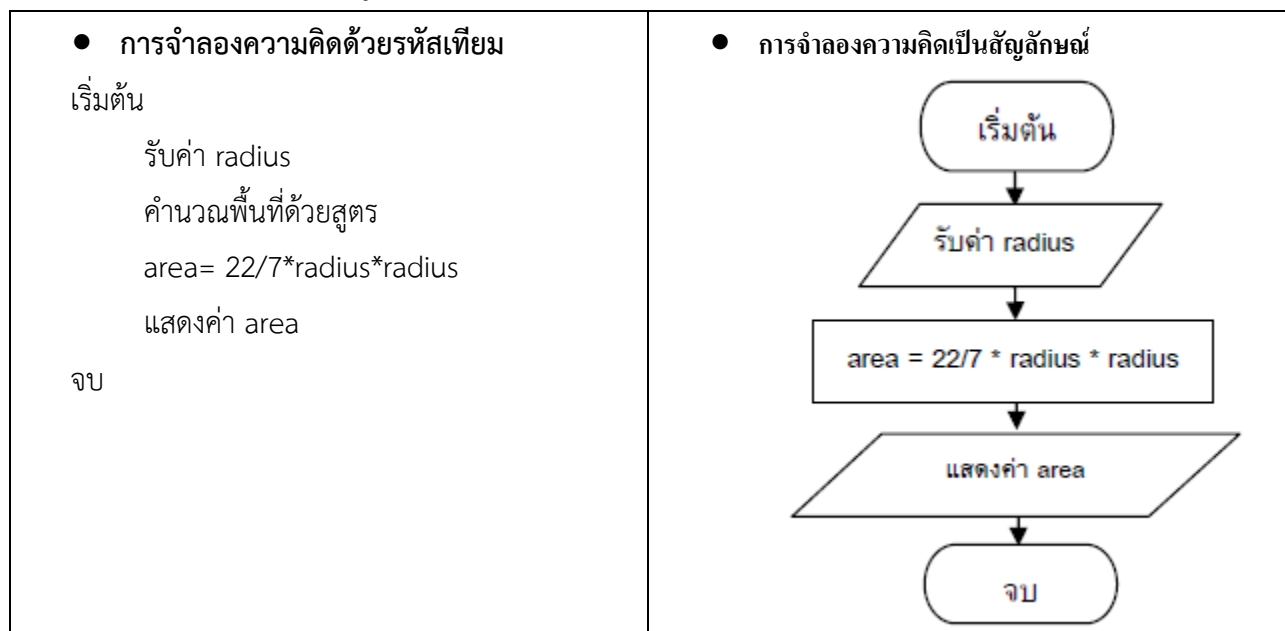
สัญลักษณ์	ความหมาย
	การทำงานด้วยมือ (Manual Operation) แทนจุดที่มีการทำงานด้วยแรงคน
	การนำข้อมูลเข้า-ออกโดยทั่วไป(General input/output) แทนจุดที่จะนำข้อมูลเข้าหรือออกจากระบบคอมพิวเตอร์โดยไม่ระบุชนิดอุปกรณ์
	แถบบันทึกข้อมูล(Magnetic tape) แทนจุดที่จะนำข้อมูลเข้าหรือออกจากโปรแกรมด้วยแถบบันทึกข้อมูล
	จานบันทึกข้อมูล(Magnetic disk) แทนจุดที่จะนำข้อมูลเข้าหรือออกจากโปรแกรมด้วยจานบันทึกข้อมูล
	การป้อนข้อมูลเข้าด้วยมือ (Manual Input) แทนจุดที่จะนำข้อมูลเข้าด้วยมือ
	การแสดงผลข้อมูล (Display) แทนจุดที่จะนำข้อมูลเข้าด้วยจอภาพ
	การทำเอกสาร (Document) แทนจุดที่มีข้อมูลเป็นเอกสารหรือแสดงผลข้อมูลด้วยเครื่องพิมพ์
	การตัดสินใจ (Decision) แทนจุดที่จะต้องเลือกปฏิบัติอย่างใดอย่างหนึ่ง
	การปฏิบัติงาน (Process) แทนจุดที่มีการปฏิบัติงานอย่างใดอย่างหนึ่ง

	การเตรียมการ (Preparation) แทนจุดกำหนดชื่อข้อมูลหรือค่าเริ่มต้นต่างๆ
	การเรียกโปรแกรมภายนอก (External subroutine) แทนจุดเรียกใช้โปรแกรมย่อยที่ไม่ได้อยู่ในโปรแกรมนั้น
	การเรียกโปรแกรมภายใน (Internal subroutine) แทนจุดเรียกใช้โปรแกรมย่อยที่อยู่ในโปรแกรมนั้น
	การเรียงข้อมูล (Sort) แทนจุดที่มีการเรียงข้อมูลใหม่ตามข้อกำหนด
	ทิศทาง (Flow line) แทนทิศทางขั้นตอนการดำเนินงานซึ่งจะปฏิบัติต่อเนื่องกันตามหัวลูกศรชี้
	หมายเหตุ (Annotation) แทนจุดที่แสดงรายละเอียดเพิ่มเติมหรือหมายเหตุของจุดต่างๆ ที่แสดงในผังงานด้วยสัญลักษณ์ไม่ชัดเจน
	การติดต่อทางไกล (Communication Link) แทนช่วงที่มีการติดต่อหรือการย้ายข้อมูลด้วยระบบติดต่อทางไกล
	จุดเชื่อมต่อ (Connector) แทนจุดเชื่อมต่อของผังงานเมื่อใช้สัญลักษณ์เพื่อให้ดูง่าย
	จุดเชื่อมต่อหน้ากระดาษ (Off page Connector) แทนจุดเชื่อมต่อของผังงานที่อยู่คนละหน้ากระดาษ
	เริ่มต้นและลงท้าย (Terminal) แทนจุดเริ่มต้นและลงท้ายของผังงานของโปรแกรมหลักและโปรแกรมย่อย

ตัวอย่างที่ 5 การวางแผนไปโรงเรียน



ตัวอย่างที่ 6 การคำนวณพื้นที่รูปวงกลม



ตัวอย่างที่ 7 การจำลองความคิดในการหาผลบวก 1,2,3,4,5,... จนถึง 20

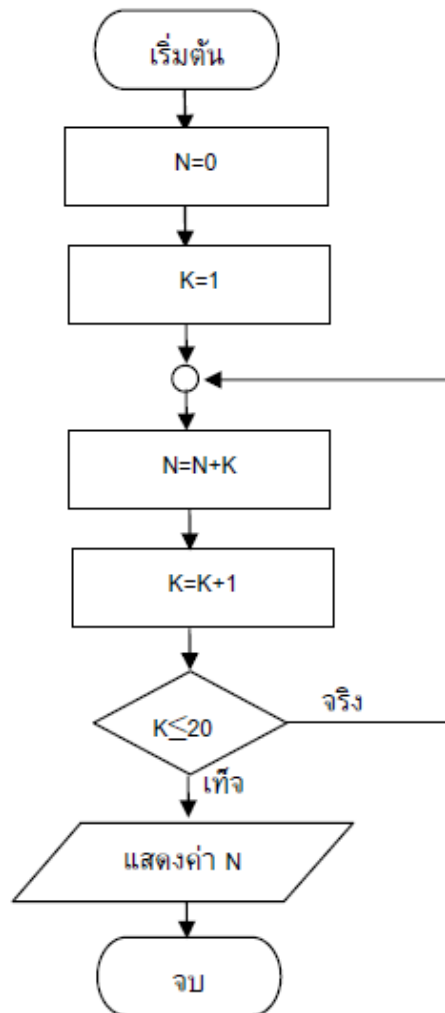
● การจำลองความคิดด้วยรหัสเทียม

เริ่มต้น

1. กำหนดให้ N มีค่าเริ่มต้นเป็น 0
2. กำหนดให้ K มีค่าเริ่มต้นเป็น 1
3. นำค่า K มารวมกับค่า N เดิม ได้ผลลัพธ์เท่าไรเก็บไว้ที่ N
4. นำค่า 1 มารวมกับค่า K เดิม ได้ผลลัพธ์เท่าไรเก็บไว้ที่ K
5. เปรียบเทียบค่า K กับ 20 ถ้า K น้อยกว่าหรือเท่ากับ 20 ให้วนกลับไปทำในขั้น 3 และทำคำสั่งถัดลงมาตามลำดับ แต่ถ้า K มากกว่า 20 ให้แสดงคำตอบ

จบ

● การจำลองความคิดเป็นสัญลักษณ์



2.3 การเขียนโปรแกรม

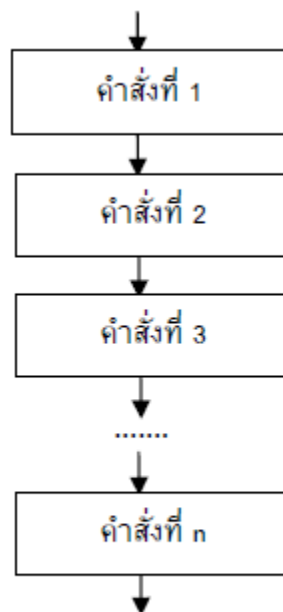
จากการศึกษาหลักการขั้นตอนการแก้ปัญหา ในหัวข้อ 5.1 และ 5.2 หลังจากที่เราสามารถวิเคราะห์ปัญหาและสร้างแบบจำลองความคิดเพื่อแสดงขั้นตอนในการแก้ปัญหาแล้ว ขั้นตอนต่อไปคือ การลงมือแก้ปัญหาตามขั้นตอนที่ได้ออกแบบไว้ โดยใช้เครื่องมือช่วยในการแก้ปัญหา ในที่นี้หากเครื่องมือที่นักเรียนเลือก คือ ภาษาคอมพิวเตอร์ซึ่งถือได้ว่าเป็นขั้นตอนหนึ่งที่สำคัญในการแก้ปัญหาด้วยคอมพิวเตอร์

การเขียนโปรแกรม (programming) หมายถึง กระบวนการใช้ภาษาคอมพิวเตอร์เพื่อกำหนดโครงสร้างของข้อมูล และกำหนดขั้นตอนวิธีเพื่อใช้แก้ปัญหาตามที่ได้ออกแบบไว้โดยอาศัยหลักเกณฑ์การเขียนโปรแกรมคอมพิวเตอร์ของแต่ละภาษา

ก่อนการเขียนโปรแกรม ผู้พัฒนาโปรแกรมจะต้องเลือกภาษาที่จะนำมาช่วยใช้งานโดยพิจารณาจากปัจจัยต่างๆ ในการทำงาน เช่น ลักษณะของปัญหา ความถนัดของผู้เขียนโปรแกรม สภาพแวดล้อมในการทำงานของระบบคอมพิวเตอร์ เป็นต้น เนื่องจากในปัจจุบันมีภาษาคอมพิวเตอร์ให้เลือกได้หลายภาษา เช่น ภาษาปาสคาล ภาษาซี ภาษาจาวา ภาษาเดลฟาย เป็นต้น ถึงแม้แต่ละภาษาจะมีรูปแบบและหลักการในการสร้างงานที่แตกต่างกัน แต่ทุกภาษาจะต้องมีโครงสร้างควบคุมหลักทั้ง 3 แบบได้แก่ โครงสร้างแบบลำดับ(sequential structure) โครงสร้างแบบมีทางเลือก (selection structure) และโครงสร้างแบบทำซ้ำ (repetition structure)

2.3.1) โครงสร้างแบบลำดับ

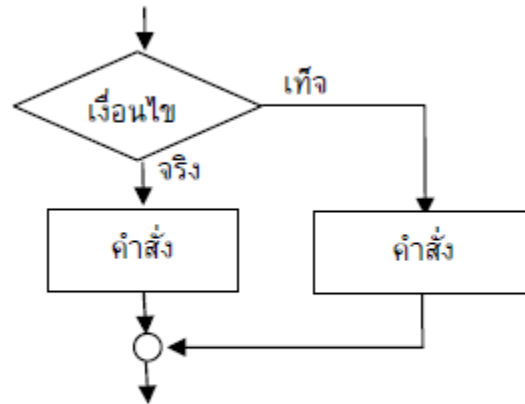
โครงสร้างแบบลำดับ คือโครงสร้างแสดงขั้นตอนการทำงานที่เป็นไปตามลำดับก่อนหลัง และแต่ละขั้นตอนจะถูกประมวลผลเพียงครั้งเดียวเท่านั้น สามารถแสดงการทำงานของโครงสร้างนี้โดยใช้ผังงานได้ดังรูป



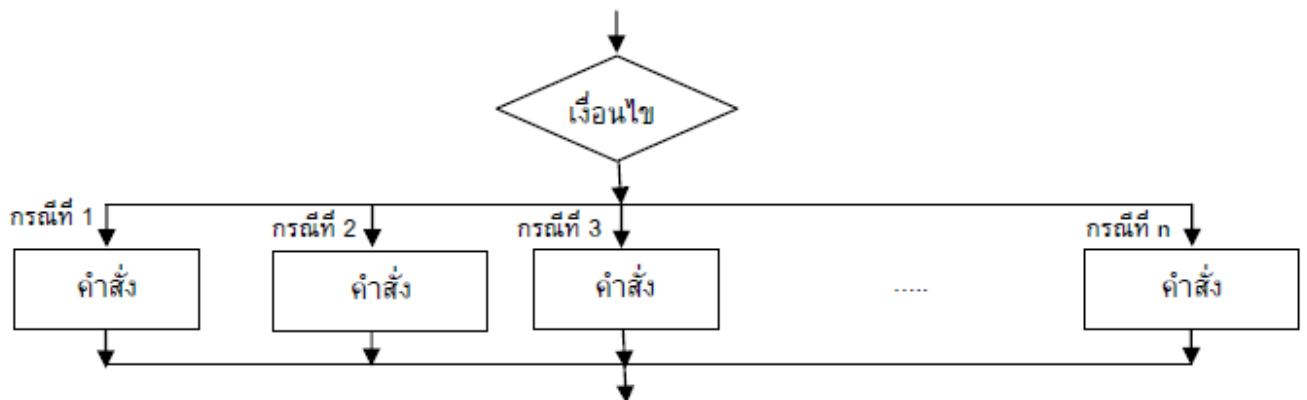
รูปที่ 4 แสดงการทำงานของโครงสร้างแบบลำดับ

2.3.2) โครงสร้างแบบมีทางเลือก

โครงสร้างแบบมีทางเลือก คือโครงสร้างที่มีเงื่อนไข ขั้นตอนการทำงานบางขั้นตอนต้องมีการตัดสินใจ เพื่อเลือกวิธีการประมวลผลขั้นตอนต่อไป และจะมีบางขั้นตอนที่ไม่ได้รับการประมวลผล การตัดสินใจอาจมี 2 ทางหรือมากกว่าก็ได้ โครงสร้างที่มีทางเลือกเพียง 2 ทางเราเรียกชื่อว่า โครงสร้างแบบ if...then...else... และโครงสร้างที่มีทางเลือกมากกว่า 2 ทาง เราเรียกชื่อว่าโครงสร้างแบบ case ซึ่งสามารถแสดงการทำงานของโครงสร้างนี้โดยใช้ผังงานดังรูป

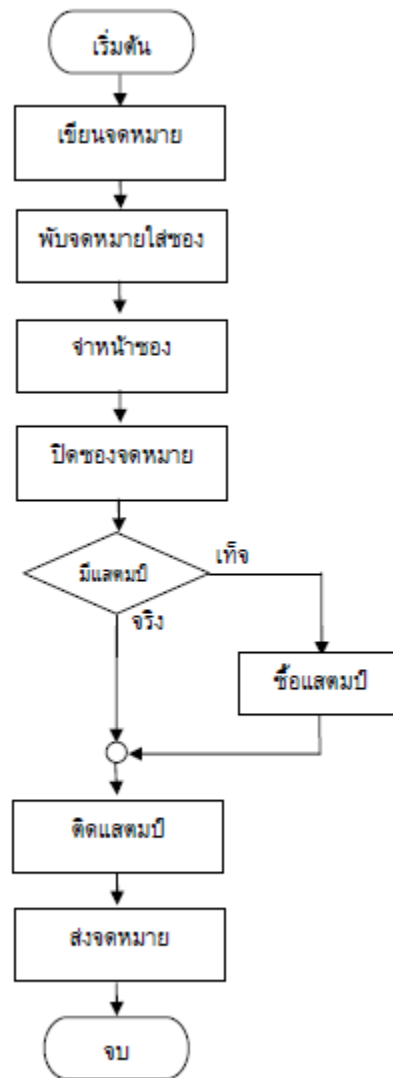


รูปที่ 5 แสดงการทำงานของโครงสร้าง if...then...else...



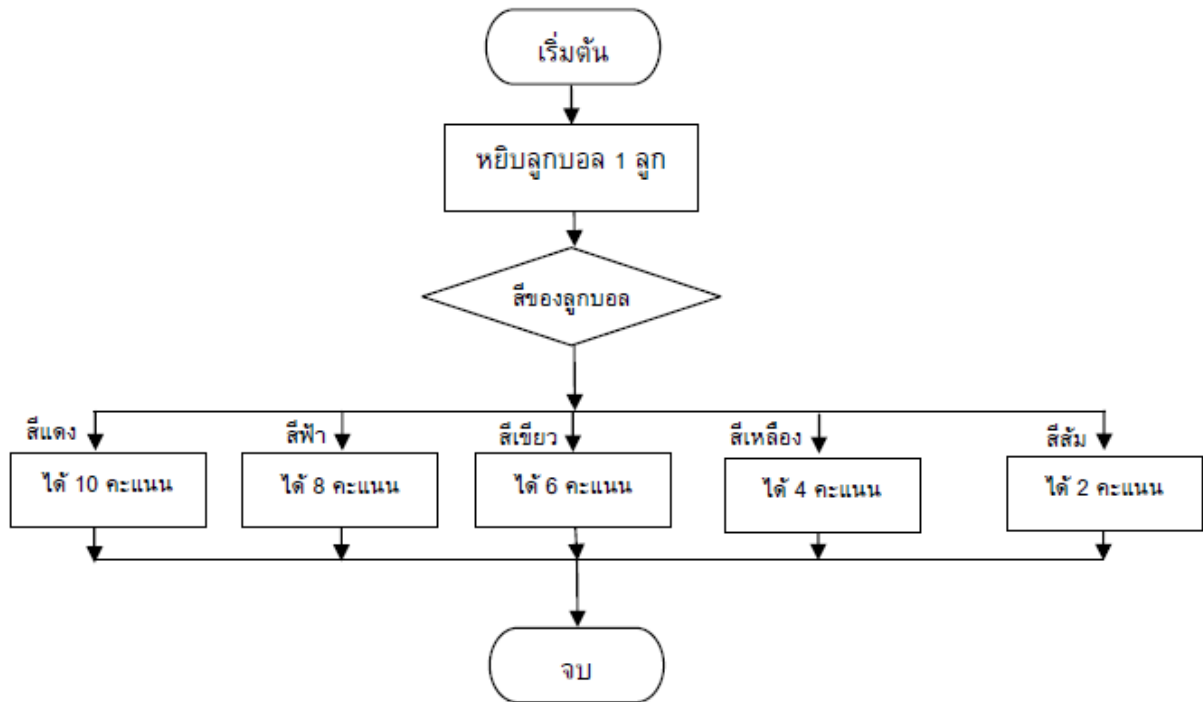
รูปที่ 6 แสดงการทำงานของโครงสร้าง case

ตัวอย่างที่ 8 แสดงผังงานที่จำลองขั้นตอนวิธีการเขียนและส่งจดหมายให้อยู่ในรูปของสัญลักษณ์



ตัวอย่างที่ 9 แสดงผังงานที่จำลองขั้นตอนวิธีของเกมหยิบลูกบอลให้อยู่ในรูปของสัญลักษณ์ โดยมีเงื่อนไขว่า มีลูกบอลซึ่ง มี 5 สี

กรณีหยิบได้สีแดง	จะได้ 10 คะแนน
กรณีหยิบได้สีฟ้า	จะได้ 8 คะแนน
กรณีหยิบได้สีเขียว	จะได้ 6 คะแนน
กรณีหยิบได้สีเหลือง	จะได้ 4 คะแนน
กรณีหยิบได้สีส้ม	จะได้ 2 คะแนน

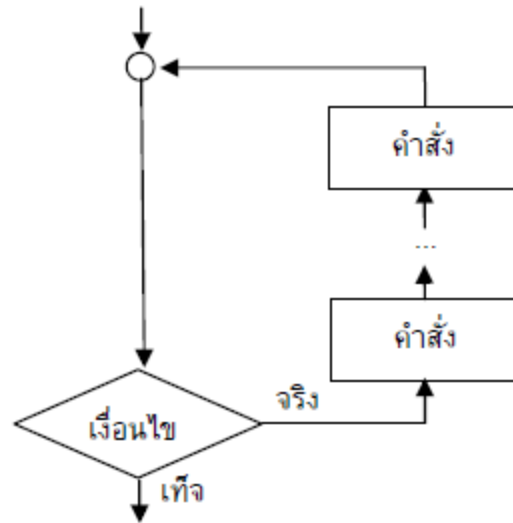


2.3.3) โครงสร้างแบบทำซ้ำ

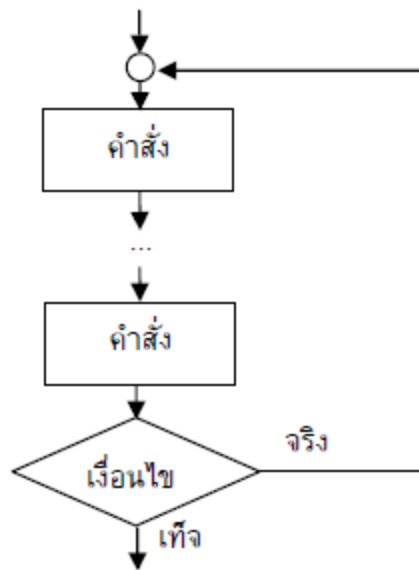
โครงสร้างแบบทำซ้ำ คือโครงสร้างที่ขั้นตอนการทำงานบางขั้นตอนได้รับการประมวลผลมากกว่า 1 ครั้ง ทั้งนี้ขึ้นอยู่กับเงื่อนไขบางประการ โครงสร้างแบบทำซ้ำนี้ต้องมีการตัดสินใจในการทำงานซ้ำ และลักษณะการทำงานของโครงสร้างแบบนี้มี 2 ลักษณะ ได้แก่

แบบที่มีการตรวจสอบเงื่อนไขในการทำซ้ำทุกครั้งก่อนดำเนินการกิจกรรมใดๆ ถ้าเงื่อนไขเป็นจริงจะทำงานซ้ำไปเรื่อยๆ และหยุดเมื่อเงื่อนไขเป็นเท็จ การทำงานลักษณะนี้แบ่งได้เป็น 2 แบบย่อย ได้แก่ การทำซ้ำแบบ for และแบบ while ลักษณะการทำงานของทั้งสองแบบนี้จะเหมือนกัน โดยสำหรับแบบ for นั้นมักใช้กรณีที่ต้องการกำหนดจำนวนรอบการทำงานที่ชัดเจน

แบบที่มีการดำเนินการกิจกรรมใดๆ ก่อนจำนวนหนึ่งรอบ แล้วจึงค่อยตรวจสอบเงื่อนไขในการทำซ้ำ ถ้าเงื่อนไขเป็นจริงจะทำงานซ้ำไปเรื่อยๆ และหยุดเมื่อเงื่อนไขเป็นเท็จ เรียกการทำงานแบบนี้ว่า การทำซ้ำแบบ do...while ผังงานแสดงขั้นตอนการทำงานซ้ำทั้งสองแบบแสดงดังรูปที่ 7 และ 8



รูปที่ 7 แสดงการทำงานของการทำงานซ้ำแบบ for และแบบ while



รูปที่ 8 แสดงการทำงานของการทำงานซ้ำแบบ do...while

โครงสร้างควบคุมทั้ง 2 แบบที่กล่าวมาข้างต้นก็คือ ขั้นตอนที่เราใช้ในการแก้ปัญหานั้นเอง พิจารณาตัวอย่างที่ 10 เป็นขั้นตอนการเลือกเครื่องมือและการออกแบบขั้นตอนวิธี คือ ขั้นตอนที่ 2 ในหัวข้อ 5.1 เราเลือกสร้างผังงานมาจำลองขั้นตอนวิธีในการหาค่าเฉลี่ยของจำนวน 5 จำนวน จากตัวอย่างที่ 2 และในตัวอย่างที่ 10 มีการแสดงโครงสร้างควบคุมแบบลำดับและแบบทำซ้ำที่ใช้ในการแก้ปัญหาด้วย

ตัวอย่างที่ 10 แสดงผังงานที่จำลองขั้นตอนวิธีการหาค่าเฉลี่ยของจำนวนเต็ม 5 จำนวน ให้อยู่ในรูปของสัญลักษณ์

