

หลักการพัฒนาโปรแกรม

พวงเพชร ฤทธิพรพันธุ์
ครูชำนาญการพิเศษ
วิทยาลัยชุมชนพังงา

ซอฟต์แวร์หรือโปรแกรมคอมพิวเตอร์ที่พัฒนาขึ้นในปัจจุบันมีความมุ่งหมายเพื่อสนองตอบต่อความต้องการของผู้ใช้งาน ไม่ว่าจะเป็นซอฟต์แวร์ระบบ หรือซอฟต์แวร์ประยุกต์ ล้วนแล้วแต่ต้องการให้ผู้ใช้เกิดความสะดวก รวดเร็ว และตรงตามวัตถุประสงค์การใช้งานแทบทั้งสิ้น การพัฒนาโปรแกรมตามความต้องการของผู้ใช้งานเพียงอย่างเดียวอาจไม่เพียงพอที่จะทำให้โปรแกรมนั้นสามารถทำงานได้ดี ยังต้องมีกระบวนการที่ดีในการพัฒนา นอกเหนือจากนี้ จะมีการวิเคราะห์ความต้องการของผู้ใช้งาน ตรวจสอบและทดสอบโปรแกรม การติดตั้งโปรแกรม การบำรุงรักษาให้ทำงานได้ไปตลอด ดังนั้น ในบทนี้จึงขอกล่าวถึงหลักการพัฒนาโปรแกรม การวิเคราะห์งานเพื่อให้ได้โปรแกรมตรงตามเป้าหมาย

1. ขั้นตอนการพัฒนาโปรแกรม

โปรแกรมคอมพิวเตอร์ หมายถึง กลุ่มคำสั่งที่เรียงเรียงตามไวยากรณ์เพื่อสั่งงานให้คอมพิวเตอร์ทำงานในสิ่งที่ต้องการ เช่น โปรแกรมคอมพิวเตอร์เพื่องานบัญชี จึงเป็นกลุ่มคำสั่งที่เรียงเรียงขึ้นเพื่อสั่งให้คอมพิวเตอร์ทำงานด้านการบัญชี หรือโปรแกรมคอมพิวเตอร์เพื่อการบริหารสถานศึกษา เป็นกลุ่มคำสั่งที่เรียงเรียงขึ้นเพื่อให้รองรับการทำงานในสถาบันการศึกษา เป็นต้น

ดังนั้น การพัฒนาโปรแกรมคอมพิวเตอร์ หมายถึง การเขียนกลุ่มคำสั่งอย่างเป็นกระบวนการ มีขั้นตอน และถูกต้องตรงตามไวยากรณ์ เพื่อสั่งให้คอมพิวเตอร์ทำงานในสิ่งที่ต้องการ มุ่งเน้นให้เกิดความน่าเชื่อถือ ถูกต้อง ทันเวลา เป็นปัจจุบัน โดยมีการดำเนินงาน 7 ขั้นตอน ดังนี้

1.1 การวิเคราะห์ปัญหา (Problem Definition)

การวิเคราะห์ปัญหา คือ การทำความเข้าใจปัญหา และค้นหาสิ่งที่ต้องการ นั่นคือการวิเคราะห์โดยละเอียดว่า ต้องการผลลัพธ์อะไร ต้องใช้ข้อมูลอะไรเพื่อให้ได้ผลลัพธ์ตามต้องการ และมีขั้นตอนการประมวลผลอย่างไร อาจเรียกขั้นตอนนี้ว่าเป็นขั้นตอนการวิเคราะห์ความต้องการ ซึ่งถือได้ว่าเป็นขั้นตอนแรกที่มีความสำคัญต่อการพัฒนาโปรแกรม มีขั้นตอนการวิเคราะห์ปัญหา ดังนี้

- | | |
|-------------------------|--|
| 1. ระบุสิ่งที่ต้องการ | เป็นการบอกให้ทราบว่าสิ่งที่ต้องการให้ทำ |
| 2. กำหนดรูปแบบผลลัพธ์ | การศึกษารูปแบบหรือลักษณะของการแสดงข้อมูล |
| 3. ระบุข้อมูลนำเข้า | ข้อมูลที่ใช้ในการประมวลผล |
| 4. กำหนดตัวแปรที่ใช้ | เป็นการบอกว่าจะใช้ตัวแปรใดบ้างแทนข้อมูลนำเข้า และตัวแปรใดบ้างแสดงผลลัพธ์ |
| 5. กำหนดวิธีการประมวลผล | เป็นขั้นตอนของคำสั่ง หรือวิธีการที่ใช้ในโปรแกรม โดยการเรียงลำดับก่อนหลัง |

ตัวอย่างการวิเคราะห์ปัญหา

ตัวอย่างที่ 1 จงแก้ปัญหาเพื่อหาพื้นที่สามเหลี่ยม โดยการรับค่าความกว้าง และความสูงจากผู้ใช้ ทางแป้นพิมพ์

วิเคราะห์ปัญหา

1. ระบุสิ่งที่ต้องการ
 - พื้นที่ของรูปสามเหลี่ยม
2. กำหนดรูปแบบผลลัพธ์
 - แสดงผลลัพธ์ของพื้นที่รูปสามเหลี่ยมเป็นตัวเลข
3. ระบุข้อมูลนำเข้า
 - ค่าความกว้างของรูปสามเหลี่ยม
 - ค่าความสูงของรูปสามเหลี่ยม
4. กำหนดตัวแปรที่ใช้
 - width แทน ความกว้างของรูปสามเหลี่ยม
 - height แทน ความสูงของรูปสามเหลี่ยม
 - area แทน พื้นที่ของรูปสามเหลี่ยม
5. กำหนดวิธีการประมวลผล
 1. เริ่มต้น
 2. รับค่า w และ h
 3. คำนวณค่าพื้นที่ของรูปสามเหลี่ยมจากสูตร $area = 0.5 \times width \times height$
 4. แสดงค่าพื้นที่ของรูปสามเหลี่ยม
 5. จบการทำงาน

ตัวอย่างที่ 2 จงแก้ปัญหาการคำนวณหาค่าเฉลี่ยของเลขจำนวนจริง 3 จำนวน โดยการรับค่าทางแป้นพิมพ์

วิเคราะห์ปัญหา

1. ระบุสิ่งที่ต้องการ
 - ค่าเฉลี่ยของเลขจำนวนจริง 3 จำนวน
2. กำหนดรูปแบบผลลัพธ์
 - แสดงค่าเฉลี่ยของเลขจำนวนจริง 3 จำนวน
3. ระบุข้อมูลนำเข้า
 - เลขจำนวนจริง 3 จำนวน

4. กำหนดตัวแปรที่ใช้

- num1 แทน เลขจำนวนจริงตัวที่ 1
- num2 แทน เลขจำนวนจริงตัวที่ 2
- num3 แทน เลขจำนวนจริงตัวที่ 3
- average แทน ค่าเฉลี่ยของเลขจำนวนจริง 3 จำนวน

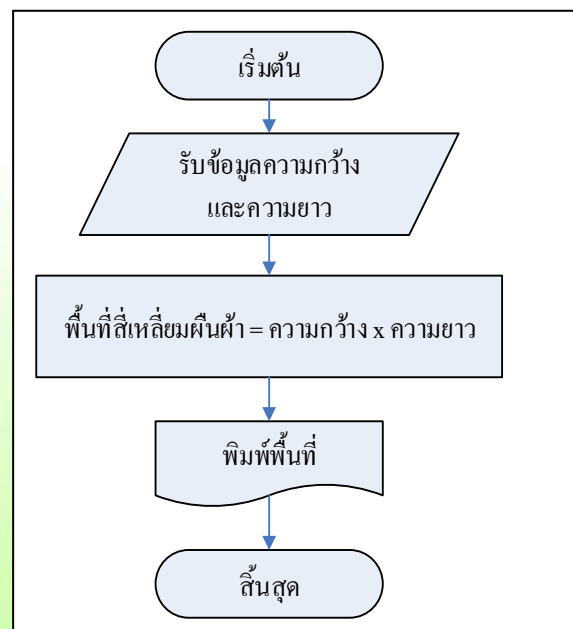
5. กำหนดวิธีการประมวลผล

1. เริ่มต้น
2. รับค่าข้อมูลเลขจำนวนจริง num1, num2, num3 ที่ละรายการ
3. คำนวณค่าเฉลี่ยจากสูตร $average = (num1+num2+num3) / 3.00$
4. แสดง average
5. จบการทำงาน

1.2 การออกแบบโปรแกรม (Program Design)

การออกแบบโปรแกรม เป็นขั้นตอนของการใช้เครื่องมือในการออกแบบลำดับการทำงานของโปรแกรมตามขั้นตอนการประมวลผลและสอดคล้องกับวิธีการทางคอมพิวเตอร์ โดยมีหลายวิธีที่นำมาช่วยในการออกแบบ ได้แก่

1. การเขียนผังงาน (Flow Chart) เป็นเครื่องมือที่นิยมใช้การนำเสนอขั้นตอนการทำงาน โดยเปลี่ยนจากข้อความ ในการวิเคราะห์งานมาเป็นรูปภาพหรือสัญลักษณ์ แล้วแสดงตามลำดับ ดังตัวอย่าง



รูปที่ 1.1 แสดงผังงานเพื่อการคำนวณพื้นที่สี่เหลี่ยมผืนผ้า

2. การเขียนรหัสจำลอง (Pseudo code) เป็นเครื่องมือที่ใช้กันมากในการออกแบบโปรแกรม ช่วยให้เขียนโปรแกรมเขียนโปรแกรมได้ง่ายขึ้น และใช้ภาษาอังกฤษในการเขียนรหัสจำลองดังตัวอย่าง

```
Area of Rectangle  
  
read wide and length  
  
compute area = wide * length  
  
display area  
  
end
```

3. การเขียนแผนภูมิโครงสร้าง (Structure Charts) เป็นการใช้รูปภาพ แสดงการแบ่งงาน จากงานใหญ่แตกออกจนเป็นงานย่อย จนกระทั่งสามารถเขียนโปรแกรมได้

4. การเขียนฮิปโปชาร์ต (HIPO chart: Hierarchy Input/Process/Output Chart) เป็นการแสดงภาพแบบไม่ละเอียด ถึงข้อมูลนำเข้าว่าประกอบด้วยอะไรบ้าง มีการประมวลผลอย่างไร และได้ผลลัพธ์อะไร

1.3 การเขียนโปรแกรม (Coding Program)

การเขียนโปรแกรม เป็นขั้นตอนของการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ตามผังงาน ที่เขียนไว้ โดยการแปลงผังงานให้เป็นโปรแกรมคอมพิวเตอร์ซึ่งผังงานที่เขียนไว้ต้องไม่ขึ้นอยู่กับภาษาคอมพิวเตอร์ใด การเขียนโปรแกรมคอมพิวเตอร์ อาจเขียนลงในกระดาษที่จัดทำขึ้นพิเศษ เรียกว่ากระดาษลงรหัส (Coding form) อาจเป็นกระดาษลงรหัสเฉพาะภาษาที่ใช้งานก็เป็นได้

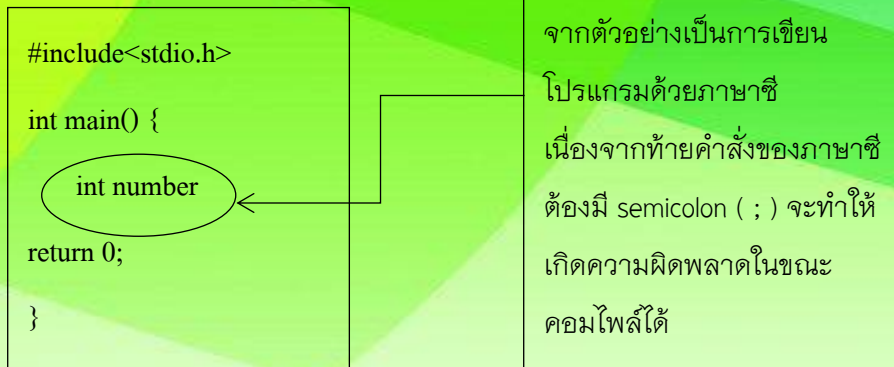
ดังนั้น ในขั้นตอนนี้จึงสามารถเลือกใช้ภาษาคอมพิวเตอร์ได้หลายภาษา เช่น ภาษาปาสคาล ภาษาเบสิก ภาษาโคบอล ภาษาซี ภาษาฟอร์แทรน เป็นต้น

1.4 การตรวจสอบความผิดพลาดของโปรแกรม (Debugging)

การตรวจสอบความผิดพลาดของโปรแกรม เป็นขั้นตอนที่ทำหลังจากที่เขียนโปรแกรมเสร็จเรียบร้อยแล้ว เป็นขั้นตอนของการแปลโปรแกรมเพื่อตรวจสอบว่าโปรแกรม มีข้อผิดพลาดหรือทำงานไม่ตรงตามวัตถุประสงค์ เรียกว่า เอเรอร์ (Error) ซึ่งในการตรวจสอบความผิดพลาดของโปรแกรมที่เขียนขึ้นนั้น ให้พิจารณาประเภทของความผิดพลาด และขั้นตอนของการตรวจสอบข้อผิดพลาด ดังนี้

1.4.1 ประเภทของความผิดพลาดของโปรแกรม สามารถจำแนกได้ 3 ประเภท ดังต่อไปนี้

1. ความผิดพลาดทางไวยากรณ์ (Syntax Error) เป็นข้อผิดพลาดที่เกิดขึ้นจากการเขียนรหัสคำสั่ง (source code) โดยการเขียนไวยากรณ์ของภาษาผิดพลาด ตรวจพบได้ขณะที่แปลโปรแกรม ตัวแปลภาษาคอมพิวเตอร์จะแจ้งให้ทราบในทันที สามารถแก้ไขได้ง่าย ตัวอย่างเช่น การกำหนดตัวแปรในภาษาซี ทำยคำสั่งต้องใส่เครื่องหมายเซมิโคลอน (;) แต่ในโปรแกรมตัวอย่างไม่มีเครื่องหมายดังกล่าว จึงเกิดข้อผิดพลาดของโปรแกรมขึ้นในขณะคอมไพล์



รูปที่ 1.2 แสดงตัวอย่างของข้อผิดพลาดที่เกิดขึ้นในโปรแกรมแบบความผิดพลาดทางไวยากรณ์

2. ความผิดพลาดขณะปฏิบัติงาน (Run-time Error) เป็นความผิดพลาดที่เกิดขึ้นระหว่างการปฏิบัติการ (Execution) ของคอมพิวเตอร์ ซึ่งความผิดพลาดนี้ไม่สามารถตรวจพบในขณะที่แปลผลโปรแกรม เช่น จากคำสั่ง x/y โดยกำหนดให้ $x=20, y=0$ โดยปกติแล้วการหารด้วย 0 จะทำให้เกิดความผิดพลาดได้ แต่การแปลโปรแกรมไม่มีข้อผิดพลาด ดังนั้น จึงตรวจพบได้ขณะปฏิบัติงาน

3. ความผิดพลาดของลำดับขั้นตอน (Logical Error) เป็นความผิดพลาดของโปรแกรมที่เกิดจากการตีความหมายของปัญหาผิดหรือออกแบบอัลกอริทึมผิดพลาด จึงทำให้กระบวนการทำงานของโปรแกรมผิดพลาดไปด้วย ซึ่งความผิดพลาดประเภทนี้เป็นความผิดพลาดที่แก้ไขได้ยาก เนื่องจากต้องวิเคราะห์ปัญหาใหม่ และยังตรวจสอบความผิดพลาดยากอีกด้วย การแก้ไขโปรแกรมประเภทนี้ ต้องตรวจโปรแกรมทีละคำสั่งเพื่อหาข้อผิดพลาด เช่น คำสั่งกำหนดให้ $x=10, y$ เกิดจากการคำนวณ ซึ่งหากคำนวณ แล้วผลลัพธ์ได้เท่ากับ 0 ให้ x/y ทำให้มีความผิดพลาดเกิดขึ้นได้

1.4.2 การตรวจสอบความผิดพลาดของโปรแกรม ได้มีการจำแนกขั้นตอนของการตรวจสอบความผิดพลาดไว้ 2 ประเภท ดังต่อไปนี้

1. การตรวจสอบความผิดพลาดด้วยตนเอง (Self-Checking) เป็นการตรวจสอบความผิดพลาดการทำงานที่ละขั้นตอนของโปรแกรมด้วยตนเอง โดยการทดลองเขียนโปรแกรมลงบนกระดาษแล้วใส่ตรวจสอบการทำงานของโปรแกรมทีละขั้นตอนด้วยตนเอง ว่าโปรแกรมมีการทำงานที่ถูกต้องได้ผลลัพธ์ตรงตามความเป็นจริงหรือไม่ ซึ่งอาจตรวจพบความผิดพลาดทางการเขียนไวยากรณ์ หรือความผิดพลาดของลำดับขั้นตอน

2. ตรวจสอบความผิดพลาดด้วยการแปลโปรแกรม (Translating) สามารถตรวจสอบความผิดพลาดทางการเขียนไวยากรณ์ แต่ไม่สามารถตรวจสอบความผิดพลาดของลำดับขั้นตอนและความผิดพลาดขณะการปฏิบัติงาน โดยหลังจากที่เขียนโปรแกรมเสร็จและมีการตรวจสอบด้วยตนเองเรียบร้อยแล้วก็จะติดตั้งโปรแกรมเข้าสู่เครื่องคอมพิวเตอร์เพื่อทำการแปลโปรแกรม โดยจะต้องเรียกใช้ตัวแปลภาษาโปรแกรม ที่เรียกว่า คอมไพเลอร์ หรืออินเตอร์พรีเตอร์ ทำการแปลภาษาโปรแกรมให้เป็นภาษาเครื่อง การแปลนี้จะเป็นการตรวจสอบความผิดพลาดของโปรแกรม หากมีข้อผิดพลาดเกิดขึ้นเครื่องคอมพิวเตอร์จะแจ้งให้ทราบทางหน้าจอ โดยทั่วไปพบว่า หลังจากเขียนโปรแกรมเสร็จแล้ว

เวลาส่วนใหญ่ของเวลาในการพัฒนาโปรแกรม จะถูกใช้ไปในการหาข้อผิดพลาดของโปรแกรมและการแก้ไขข้อผิดพลาดนั้น

1.5 การทดสอบความถูกต้องของโปรแกรม (Testing and Validating)

การทดสอบความถูกต้องของโปรแกรมเป็นการทดสอบว่าโปรแกรมให้ผลลัพธ์ถูกต้องหรือไม่สามารถทำได้โดยการป้อนข้อมูลสมมติหลายกรณีทั้งที่เป็นไปได้ และที่เป็นไปไม่ได้แล้ว จึงตรวจสอบผลลัพธ์ที่ได้ว่ามีค่าถูกต้องหรือไม่ หากถูกต้องแสดงว่าโปรแกรมนั้นมีความถูกต้องสมบูรณ์ สามารถนำไปใช้ได้ต่อไป โดยมีวิธีการทดสอบโปรแกรมคอมพิวเตอร์ 2 แบบ ได้แก่

1.5.1 การทดสอบโครงสร้างของโปรแกรม หรือที่เรียกว่า White Box Testing เป็นวิธีการทดสอบโปรแกรมเพื่อดูโครงสร้างของโปรแกรม หรือทางเดินในโปรแกรม (Structured Testing) ว่าโปรแกรมสามารถทำงานได้ตามโครงสร้างหรือไม่ โดยต้องสร้างชุดทดสอบเฉพาะสำหรับการทดสอบในเงื่อนไขที่แตกต่างกัน ซึ่งชุดทดสอบจะต้องประกอบด้วยชุดที่สามารถประมวลผลอย่างปกติและไม่ปกติ และพยายามให้ทุกคำสั่งที่สร้างขึ้น ถูกทดสอบอย่างน้อย 1 ครั้ง

1.5.2 การทดสอบฟังก์ชันการทำงานของโปรแกรม หรือที่เรียกว่า Black Box Testing เป็นการทดสอบโดยไม่คำนึงถึงคำสั่งภายในโปรแกรม โดยจะมุ่งทดสอบการทำงานได้ตามฟังก์ชัน (Functional Testing) ที่ได้บอกความต้องการไว้ตั้งแต่เริ่มศึกษาความต้องการ พิจารณาผลลัพธ์ที่ได้ให้สอดคล้องกับข้อมูลเริ่มต้นเท่านั้น

1.6 การทำเอกสารประกอบโปรแกรม (Documentation)

การทำเอกสารประกอบโปรแกรมที่เขียนขึ้นเป็นสิ่งจำเป็นและสำคัญอย่างยิ่ง โดยเฉพาะในการเขียนโปรแกรมแต่ละขั้นตอนจะต้องจัดทำเอกสารประกอบอย่างสม่ำเสมอ ทุกขั้นตอนโดยวัตถุประสงค์ของการจัดทำเอกสารนั้นเพื่ออธิบายรายละเอียดการทำงานของโปรแกรม ทำให้อ่านโปรแกรมและแก้ไขปัญหานั้นได้ง่ายในภายหลัง ซึ่งประกอบด้วยเอกสาร 2 ประเภทคือ

1.6.1 เอกสารประกอบโปรแกรมสำหรับผู้ใช้งาน (User Documentation) อธิบายการใช้งานโปรแกรมสำหรับผู้ใช้งาน เช่น การติดตั้ง การเปิดโปรแกรมใช้งาน การกำหนดค่าเริ่มต้น การปิดโปรแกรม การพิมพ์เอกสาร การขอความช่วยเหลือ เป็นต้น

1.6.2 เอกสารประกอบสำหรับผู้เขียนโปรแกรม (Technical Documentation) เป็นเอกสารอธิบายรายละเอียดขั้นตอนการทำงานของโปรแกรม ชื่อโปรแกรมหลัก ชื่อโปรแกรมย่อย และแต่ละโปรแกรมทำหน้าที่อะไร สำหรับผู้เขียนโปรแกรมหรือเพื่อการปรับปรุงโปรแกรมในอนาคต

ทั้งนี้ ไม่ว่าจะเป็เอกสารประกอบโปรแกรมสำหรับผู้ใช้งาน และเอกสารสำหรับผู้เขียนโปรแกรม จะต้องมีการปรับปรุงแก้ไข เพิ่มเติมเมื่อเวลาผ่านไป เนื่องจากอาจมีการตรวจสอบ พบข้อผิดพลาดเพิ่มเติมในภายหลังก็เป็นได้

1.7 การติดตั้งและการบำรุงรักษาโปรแกรม (Implementation and Program Maintenance)

การติดตั้งโปรแกรม เป็นขั้นตอนที่จะต้องทำการติดตั้งโปรแกรมให้กับผู้ใช้งานจริง การกำหนดสภาพแวดล้อมของระบบ การอธิบายถึงแหล่งที่จะให้ความช่วยเหลือเบื้องต้น หรือการจัดการในเบื้องต้นเมื่อเกิดข้อผิดพลาด จัดการฮาร์ดแวร์หรืออุปกรณ์ที่เกี่ยวข้อง เช่น ระบบไฟฟ้า ระบบเครือข่าย เป็นต้น

การบำรุงรักษาโปรแกรม เป็นขั้นตอนที่ผู้เขียนโปรแกรมต้องคอยเฝ้าระวังหาความผิดพลาดของโปรแกรมระหว่างการใช้งาน หรือในการใช้งานได้นาน ผู้ใช้อาจต้องการปรับเปลี่ยนเพิ่มเติมโปรแกรมเพื่อให้ทันต่อเหตุการณ์ ซึ่งผู้เขียนโปรแกรมอาจต้องปรับปรุงให้ เป็นไปตามความต้องการของผู้ใช้ที่เปลี่ยนแปลงไปนั่นเอง

ดังนั้น ในการพัฒนาโปรแกรมนั้น ผู้เขียนโปรแกรมจะต้องมีการเตรียมงานที่เกี่ยวกับการเขียนโปรแกรมอย่างเป็นขั้นตอน เรียกขั้นตอนเหล่านั้นว่า วงจรการพัฒนาโปรแกรม ประกอบด้วย 7 ขั้นตอน ได้แก่ การวิเคราะห์ปัญหา การออกแบบโปรแกรม การเขียนโปรแกรม การตรวจสอบข้อผิดพลาดของโปรแกรม การทดสอบความถูกต้องของโปรแกรม การจัดทำเอกสารประกอบโปรแกรม การติดตั้งและการบำรุงรักษาโปรแกรม

2. ภาษาคอมพิวเตอร์ (Computer Language)

มนุษย์ใช้ภาษาที่แตกต่างกันในการสื่อสาร เช่น ภาษาไทย ภาษาจีน ภาษาอังกฤษ เป็นต้น การติดต่อสื่อสารซึ่งกันและกันเพื่อถ่ายทอดความคิดและความต้องการระหว่างผู้ส่งสารกับผู้รับสาร ให้ความเข้าใจและสามารถทำงานได้บรรลุวัตถุประสงค์ ในทำนองเดียวกันหากมนุษย์ต้องการสื่อสารเพื่อให้คอมพิวเตอร์สามารถทำงานได้ตรงตามวัตถุประสงค์ จึงต้องอาศัยสื่อกลางในการติดต่อสื่อสารระหว่างมนุษย์กับคอมพิวเตอร์ สิ่งนั้นคือ ภาษาคอมพิวเตอร์

ภาษาคอมพิวเตอร์มีมากมายหลายภาษา โดยแต่ละภาษาจะประกอบไปด้วยกลุ่มคำ (Word) สัญลักษณ์ (Symbol) และกฎเกณฑ์หรือไวยากรณ์ (Syntax) ของแต่ละภาษาที่มีความแตกต่างกันไป โดยแต่ละภาษาจะมีกลุ่มคำสั่งและไวยากรณ์เฉพาะของภาษา ซึ่งไม่สามารถนำไปใช้ปะปนกันได้ ดังนั้น ผู้เขียนโปรแกรมจะต้องเรียนรู้กลุ่มคำสั่ง และกฎเกณฑ์หรือไวยากรณ์ ของภาษาที่ต้องการใช้งานเสียก่อน เพื่อจะได้เขียนโปรแกรมได้อย่างถูกต้อง

ปัจจุบันภาษาคอมพิวเตอร์มีมากมายหลายภาษาให้เลือกใช้งาน มีจุดเด่น ประสิทธิภาพ แตกต่างกันไป ดังนั้นผู้เขียนโปรแกรมจึงต้องศึกษาว่าภาษาใดมีคำสั่งที่มีประสิทธิภาพควบคุมการทำงานตามที่ต้องการ เพื่อให้สามารถนำไปเขียนโปรแกรมที่ตรงตามวัตถุประสงค์ของระบบงาน

2.1 ประเภทของภาษาคอมพิวเตอร์

ภาษาคอมพิวเตอร์สามารถจำแนกได้ 3 ระดับ จำแนกตามลักษณะที่ติดต่อกันระหว่างภาษากับเครื่องคอมพิวเตอร์ ได้แก่ ภาษาเครื่อง ภาษาระดับต่ำ และภาษาระดับสูง

2.1.1 ภาษาเครื่อง เป็นภาษาพื้นฐานที่คอมพิวเตอร์สามารถเข้าใจได้และปฏิบัติตามคำสั่งได้ทันที แต่ละคำสั่งประกอบขึ้นจากเลขฐานสอง คือ 0 และ 1 ซึ่งเป็นตัวเลขฐานสองเขียนเรียงติดต่อกัน โดยที่โครงสร้างคำสั่งในภาษาเครื่องจะประกอบด้วย 2 ส่วนคือ

1. ส่วนที่ 1 คือ โอเปอเรชันโคด (Operation Code) เป็นคำสั่งที่สั่งให้เครื่องคอมพิวเตอร์ปฏิบัติการ ทำหน้าที่บอกให้คอมพิวเตอร์ทราบว่าต้องทำอะไร เช่น การบวก (Addition) การลบ (Subtraction) เป็นต้น

2. ส่วนที่ 2 คือ โอเปอเรนด์ (Operands) เป็นส่วนที่บอกตำแหน่งของหน่วยความจำหลักที่ใช้เก็บข้อมูล ซึ่งคอมพิวเตอร์ต้องนำมาใช้การทำงาน

ตัวอย่างเช่น ต้องการเขียนคำสั่งเพื่อบวกข้อมูลที่เก็บอยู่ในหน่วยความจำหลัก 2 ตำแหน่งเข้าด้วยกัน กำหนดให้

01011010 แทน คำสั่งบวกในภาษาเครื่อง

00001000 แทน ตำแหน่งของหน่วยความจำที่เก็บข้อมูลจำนวนแรกอยู่ในตำแหน่ง 00001000

00001010 แทน ตำแหน่งของหน่วยความจำที่เก็บข้อมูลจำนวนที่สองอยู่ในตำแหน่ง 00001000

รูปแบบคำสั่งในภาษาเครื่อง คือ 01011010 0000 1000 0000 1010

จากตัวอย่างจะเห็นว่า คำสั่งภาษาเครื่องจะมีแต่ตัวเลข 0 กับ 1 เท่านั้น ทำให้ยากต่อการทำความเข้าใจ ดังนั้นจึงทำให้มีโอกาสในการเขียนโปรแกรมผิดพลาดได้ง่าย

ข้อจำกัดของภาษาเครื่อง คือ ยากต่อการเขียน การจดจำและเรียนรู้คำสั่งผู้เขียนโปรแกรมจำเป็นต้องมีความรู้เกี่ยวกับฮาร์ดแวร์เป็นอย่างดี นอกจากนี้ยังเป็นภาษาที่ยึดติดกับเครื่องคอมพิวเตอร์ (Hardware Dependent) กล่าวคือ ไม่สามารถนำโปรแกรมเดียวกันไปใช้บนเครื่องต่างชนิดกันได้

2.1.2 ภาษาระดับต่ำ เป็นภาษาที่คอมพิวเตอร์ รับรู้คำสั่งโดยตรงหรือเข้าใจง่าย แต่เป็นภาษาที่มนุษย์อ่านแล้วเข้าใจยาก ภาษาระดับต่ำเป็นภาษาที่ขึ้นกับฮาร์ดแวร์ของคอมพิวเตอร์แต่ละระบบ นั่นคือไม่สามารถนำโปรแกรมจากคอมพิวเตอร์หนึ่งไปทำงานกับอีกคอมพิวเตอร์หนึ่งได้ โดยภาษาระดับต่ำได้แก่ ภาษาแอสแซมบลี

ภาษาแอสแซมบลี จัดเป็นภาษาที่ใช้สัญลักษณ์ข้อความ (Mnemonic code) แทนกลุ่มตัวเลขฐานสอง เพื่อให้ง่ายต่อการเขียนและการจดจำมากขึ้น การทำงานของโปรแกรมจะต้องแปลภาษาแอสแซมบลีให้เป็นภาษาเครื่องด้วยตัวแปลภาษาที่เรียกว่า แอสแซมเบลอร์

การทำงานของภาษาแอสแซมบลี ยังคงมีความสัมพันธ์กับภาษาเครื่องอยู่ นั่นคือ ภาษาแอสแซมบลี 1 คำสั่ง จะถูกแปลเป็นภาษาเครื่อง 1 คำสั่งเท่านั้น และยังคงเป็นภาษาที่ยึดติดกับเครื่อง

คอมพิวเตอร์เช่นเดียวกับภาษาเครื่อง ผู้เขียนโปรแกรมจึงต้องมีความรู้เกี่ยวกับฮาร์ดแวร์เป็นอย่างดี ดังตัวอย่างต่อไปนี้

LD HL, MEMLOC

จากตัวอย่างคำสั่งข้างต้นเป็นการอ่านค่าตัวแปร (ที่เก็บในหน่วยความจำ) ชื่อ MEMLOC ไปเก็บไว้ในรีจิสเตอร์ ซึ่งสามารถเขียนเทียบเคียงเป็นภาษาเครื่องคือ

00100001 10001111 00001000

ข้อจำกัดของภาษาแอสแซมบลี คือ เป็นภาษาที่ยังขึ้นอยู่กับฮาร์ดแวร์ หรือคอมพิวเตอร์ เฉพาะระบบเท่านั้น

2.1.3 ภาษาระดับสูง เนื่องจากภาษาระดับต่ำเป็นภาษาที่ยากต่อการเข้าใจและยากต่อการพัฒนา ทั้งยังเป็นภาษาที่ขึ้นกับเครื่องคอมพิวเตอร์เฉพาะระบบใดระบบหนึ่ง เมื่อการสื่อสารพัฒนาขึ้น ทำให้มนุษย์ไม่จำกัดการใช้งานอยู่เพียงเครื่องเดียว ทำให้โปรแกรมที่ถูกเขียนขึ้นด้วยภาษาเครื่องและภาษาระดับต่ำไม่สามารถแลกเปลี่ยนข้อมูลระหว่างกันได้ จึงได้มีการพัฒนาภาษาคอมพิวเตอร์กลุ่มใหม่ขึ้น โดยมีจุดมุ่งหมาย คือให้ใช้งานและเรียนรู้คำสั่งง่าย เรียกภาษาที่พัฒนาขึ้นนี้ว่า ภาษาระดับสูง

ภาษาระดับสูง เป็นภาษาที่สร้างขึ้นเพื่อช่วยอำนวยความสะดวกในการเขียนโปรแกรม ลักษณะของคำสั่งจะประกอบด้วยคำในภาษาอังกฤษ โปรแกรมที่ได้จะไม่ขึ้นกับฮาร์ดแวร์ ผู้อ่านสามารถเข้าใจความหมายได้ทันที ภาษาระดับสูงมีมากมายหลายภาษา อาทิเช่น ภาษาฟอร์แทรน ภาษาโคบอล ภาษาปาสคาล ภาษาเบสิก ภาษาวิซวลเบสิก ภาษาซี และภาษาจาวา เป็นต้น

2.2 การเลือกใช้ภาษาคอมพิวเตอร์

การเลือกใช้ภาษาคอมพิวเตอร์ในการพัฒนาโปรแกรมเพื่อประยุกต์ใช้ในการแต่ละด้าน จำเป็นอย่างยิ่งที่จะต้องเลือกใช้ภาษาที่เหมาะสม และเนื่องจากภาษาคอมพิวเตอร์มีเป็นจำนวนมาก ดังนั้น จึงมีข้อควรพิจารณาการเลือกใช้ภาษาคอมพิวเตอร์ ดังนี้

1. ภาษามาตรฐานที่ใช้ในองค์กร การพัฒนาโปรแกรมเพื่อใช้งานในองค์กรหนึ่ง โดยส่วนใหญ่จะยึดภาษาใดภาษาหนึ่งเป็นหลัก เนื่องจาก ทีมงานมีความเชี่ยวชาญในภาษานั้นแล้ว ไม่ต้องเสียเวลาศึกษาภาษาใหม่ ทำให้สะดวกต่อการพัฒนาขึ้นใหม่หรือปรับปรุงเปลี่ยนแปลงแก้ไข

2. คุณสมบัติและความเหมาะสมกับงาน ควรเลือกภาษาคอมพิวเตอร์ที่มีคุณสมบัติและความเหมาะสมกับงาน ซึ่งควรพิจารณาเกี่ยวกับความยืดหยุ่น จุดเด่นและประสิทธิภาพของภาษาลักษณะของการประมวลผล เช่น งานด้านธุรกิจ งานด้านบัญชี นิยมใช้ภาษาโคบอลในงานด้านวิทยาศาสตร์ และงานด้านวิศวกรรม ควรเลือกใช้ภาษาซี เป็นต้น

3. การทำงานร่วมกับโปรแกรมอื่น การเขียนโปรแกรมเพื่อใช้งานในองค์กร อาจมีการทำงานร่วมกันกับโปรแกรมอื่น เช่น ระบบการลงทะเบียน จะต้องมีการทำงานทั้งส่วนที่ติดต่อกับผู้ใช้ และส่วน

ของฐานข้อมูล ดังนั้น จึงควรพิจารณาเลือกใช้ภาษาที่สามารถรองรับกัน จะช่วยให้ระบบทำงานอย่างมีประสิทธิภาพ

4. การทำงานร่วมกับระบบอื่น โปรแกรมประยุกต์บางตัว มีเป้าหมายในการทำงานข้ามระบบ เช่นโปรแกรมรับรายการอาหารจากลูกค้า และการออกใบเสร็จเก็บเงินลูกค้า ต้องทำงานร่วมกันระหว่างโปรแกรมรับรายการสินค้าของพนักงานขาย และส่งข้อมูลไปยังโปรแกรมคำนวณเงินค่าอาหาร เป็นต้น

2.3 ตัวแปลภาษา

การเขียนรหัสคำสั่งควบคุมการทำงานระบบด้วยภาษาคอมพิวเตอร์ที่มีใช้ภาษาเครื่อง ระบบจะไม่สามารถประมวลผลได้ในทันที เนื่องจากการทำงานของระบบเป็นรหัสเลขฐานสอง คือ 0 กับ 1 ดังนั้นจึงต้องอาศัยตัวแปลภาษาคอมพิวเตอร์เพื่อแปลรหัสคำสั่งไปเป็นรหัสเลขฐานสอง โดยสามารถจำแนกตัวแปลภาษาได้ 3 ประเภท ได้แก่

2.3.1 ตัวแปลภาษาแบบแอสเซมเบลอร์ (Assembler)

ตัวแปลภาษาแอสเซมเบลอร์ ใช้แปลรหัสคำสั่งเฉพาะภาษาแอสเซมบลีให้เป็นภาษาเครื่อง

2.3.2 ตัวแปลภาษาแบบคอมไพเลอร์ (Compiler)

ตัวแปลภาษาแบบคอมไพเลอร์ มีลักษณะของการแปลคือ แปลคำสั่งทั้งโครงสร้างโปรแกรมแล้วจึงแจ้งความผิดพลาดทั้งหมดเพื่อให้แก้ไข หากไม่มีความผิดพลาดเกิดขึ้นจะสร้างโปรแกรมใหม่เพื่อเก็บรหัสคำสั่ง ทำให้การทำงานครั้งต่อไปสามารถเรียกหรืออ่านโปรแกรมได้เลยโดยไม่ต้องแปลรหัสใหม่ หากไม่มีการแก้ไขโปรแกรม

2.3.3 ตัวแปลภาษาแบบอินเทอร์พรีเตอร์ (Interpreter)

ตัวแปลภาษาแบบอินเทอร์พรีเตอร์ มีลักษณะของการแปลคือ แปลรหัสทีละคำสั่ง เมื่อพบความผิดพลาดจะหยุดทำงาน แล้วแจ้งความผิดพลาดนั้นให้ทราบทันที จากนั้นจึงทำการประมวลผลใหม่ จนกว่าจะไม่มีความผิดพลาดเกิดขึ้นอีก ทั้งนี้จะไม่มีการสร้างโปรแกรมใหม่เพื่อเก็บรหัสคำสั่ง

กล่าวได้ว่า ภาษาคอมพิวเตอร์มีความสำคัญอย่างยิ่งต่อผู้เขียนโปรแกรม หรือนักพัฒนาระบบ โดยภาษาคอมพิวเตอร์ มีวิวัฒนาการตั้งแต่ ภาษาเครื่อง ภาษาระดับต่ำ ภาษาระดับสูง และการเลือกใช้ภาษาคอมพิวเตอร์จะต้องพิจารณาจาก ภาษาที่ใช้ในองค์กร คุณสมบัติและความเหมาะสมของภาษาในด้านความยืดหยุ่น จุดเด่นและประสิทธิภาพ และการประมวลผล เป็นต้น ทั้งยังต้องพิจารณาจาก การทำงานร่วมกับภาษาและระบบสารสนเทศที่มีอยู่แล้วอีกด้วย นอกเหนือจากนี้ ตัวแปลภาษายังมีความสำคัญต่อการแปลภาษาให้เป็นภาษาเครื่อง ได้แก่ ตัวแปลภาษาแอสเซมเบลอร์ ตัวแปลภาษาแบบคอมไพเลอร์ และตัวแปลภาษาแบบอินเทอร์พรีเตอร์

สรุป

โปรแกรมคอมพิวเตอร์ หมายถึง กลุ่มคำสั่งที่เรียงเรียงตามไวยากรณ์เพื่อสั่งงานให้คอมพิวเตอร์ทำงานในสิ่งที่ต้องการ โดยการเขียนกลุ่มคำสั่งให้ถูกต้องตามหลักไวยากรณ์ เพื่อสั่งให้คอมพิวเตอร์ทำงานในสิ่งที่ต้องการ นั่นคือ การพัฒนาโปรแกรมคอมพิวเตอร์ โดยมีรูปแบบของการพัฒนาโปรแกรมที่ดี คือ อำนวยง่าย ง่ายต่อการบำรุงรักษา การเขียนคอมเมนต์ การตั้งชื่อ การจัดย่อหน้าของโปรแกรม การกำหนดค่าตัวเลข การเว้นช่องว่าง การใช้งานฟังก์ชัน ผลลัพธ์ของโปรแกรม โดยมีขั้นตอนการพัฒนาโปรแกรม ที่เรียกว่า วงจรการพัฒนาโปรแกรม ประกอบด้วย 7 ขั้นตอน ได้แก่ การวิเคราะห์ปัญหา การออกแบบโปรแกรม การเขียนโปรแกรม การตรวจสอบข้อผิดพลาดของโปรแกรม การทดสอบความถูกต้องของโปรแกรม การจัดทำเอกสารประกอบโปรแกรม การติดตั้งโปรแกรมและการบำรุงรักษา นอกจากนี้ขั้นตอนการพัฒนาโปรแกรมแล้ว ยังต้องคำนึงถึงภาษาคอมพิวเตอร์ โดยภาษาคอมพิวเตอร์ คือ รหัสคำสั่งที่ใช้ในการควบคุมการทำงานของอุปกรณ์ในระบบคอมพิวเตอร์ มีความสำคัญต่อการพัฒนาโปรแกรม เพื่อให้บรรลุวัตถุประสงค์ที่ผู้ใช้งานโปรแกรมกำหนดไว้ ซึ่งมีพัฒนาการจากภาษาเครื่อง ภาษาระดับต่ำ และภาษาระดับสูง ผู้พัฒนาโปรแกรมจำเป็นต้องพิจารณาเลือกใช้โปรแกรมภาษาที่เป็นภาษามาตรฐานที่องค์กรเลือกใช้และมีความยืดหยุ่นสูง ตัวแปลภาษาคอมพิวเตอร์ คือ ตัวกลางที่ทำหน้าที่ในการแปลรหัสคำสั่งให้เป็นรหัสเลขฐานสอง ซึ่งตัวแปลภาษาคอมพิวเตอร์มี 3 ลักษณะ คือ ตัวแปลภาษาแบบแอสแซมบลอร์ ตัวแปลภาษาแบบคอมไพเลอร์ และตัวแปลภาษาแบบอินเทอร์พรีเตอร์
