

พ 31252

หลักการออกแบบเว็บไซต์



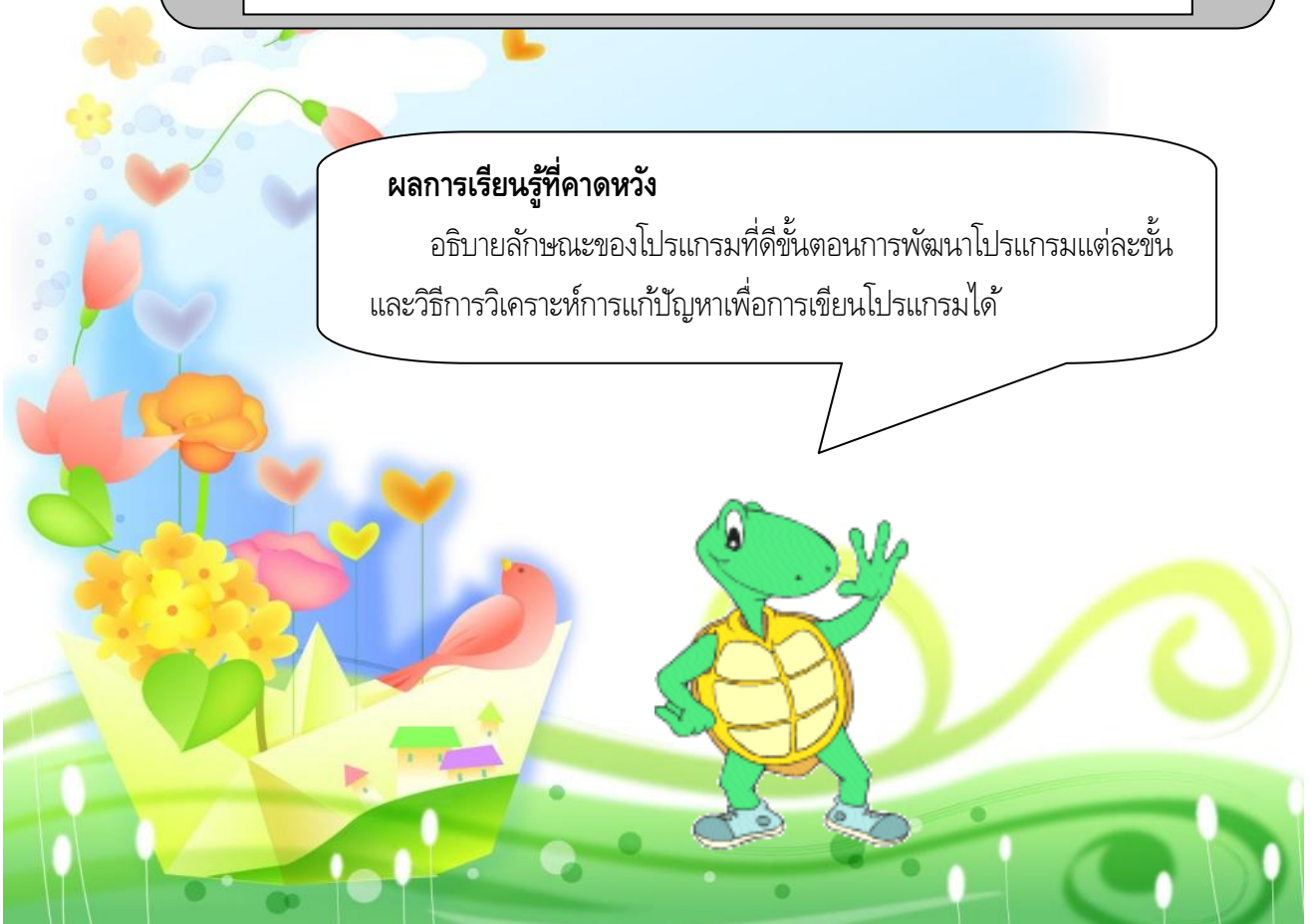
หน่วยการเรียนรู้ที่ 2 ขั้นตอนการพัฒนาโปรแกรม

สาระการเรียนรู้

- 1) คุณลักษณะโปรแกรมที่ดี
- 2) ขั้นตอนการพัฒนาโปรแกรม
- 3) ข้อพิจารณาในการออกแบบการทำงานสำหรับพัฒนาโปรแกรม

ผลการเรียนรู้ที่คาดหวัง

อธิบายลักษณะของโปรแกรมที่ดีขั้นตอนการพัฒนาโปรแกรมแต่ละขั้น
และวิธีการวิเคราะห์การแก้ปัญหาเพื่อการเขียนโปรแกรมได้



ใบความรู้ที่ 2 เรื่อง ขั้นตอนการพัฒนาโปรแกรม

1. ลักษณะโปรแกรมที่ดี

1. ให้ผลลัพธ์ที่ถูกต้องตามความต้องการ
2. ตรวจสอบความถูกต้องของรูปแบบของข้อมูลได้
3. รูปแบบของโปรแกรมสามารถทำความเข้าใจได้โดยงาน โดยโปรแกรมที่เขียนขึ้นควรมีรูปแบบการประมวลผลอย่างเป็นระบบ และมีการอธิบายที่ช่วยให้เข้าใจได้ดียิ่งขึ้น
4. ใช้อำนาจต่อการปรับปรุงแก้ไขที่อาจเกิดขึ้นในอนาคต และต้องมีเอกสารประกอบโปรแกรมที่พัฒนาด้วย

2. ขั้นตอนการพัฒนาโปรแกรม

การพัฒนาโปรแกรมประกอบด้วยขั้นตอน ดังต่อไปนี้ คือ

ขั้นตอนที่ 1 การวิเคราะห์ปัญหา (Problem Analysis)

ขั้นตอนที่ 2 การออกแบบโปรแกรม (Program Design)

ขั้นตอนที่ 3 การเขียนโปรแกรม (Program Coding)

ขั้นตอนที่ 4 การทดสอบและแก้ไขโปรแกรม (Program Testing & Verification)

ขั้นตอนที่ 5 การจัดทำเอกสารและคู่มือการใช้งาน (Program Documentation)

ขั้นตอนที่ 6 การใช้งานจริง (Program Implement)

ขั้นตอนที่ 7 การปรับปรุงและพัฒนาโปรแกรม (Program Maintenance)

1. การวิเคราะห์ปัญหา (Problem analysis)

1.1 กำหนดขอบเขตของปัญหา เพื่อให้ทราบขั้นตอนการทำงานต่างๆ ดังนี้

- กำหนดจุดประสงค์การทำงาน เพื่อให้ทราบว่าเขียนโปรแกรมเพื่อต้องการแก้ปัญหาอะไร เช่น ปัญหาเกี่ยวกับการคำนวณพื้นที่ของสามเหลี่ยม เป็นต้น

- กำหนดเป้าหมายให้ชัดเจน เพื่อให้ได้ผลลัพธ์ที่ต้องการ เช่น โปรแกรมคำนวณพื้นที่ของสามเหลี่ยม ต้องการคำนวณพื้นที่รูปสามเหลี่ยมได้หลายขนาดและแสดงผลเป็นตัวเลข



- ศึกษาวิธีการแก้ปัญหา เพื่อออกแบบขั้นตอนการทำงานได้อย่างเหมาะสม เช่น การใช้สูตรคำนวณทางคณิตศาสตร์ต่างๆ
 - กำหนดข้อจำกัดและศึกษาความเป็นไปได้เพื่อใช้ในการแก้ปัญหาตามวัตถุประสงค์ เช่น โปรแกรมคำนวณพื้นที่ของสามเหลี่ยม ต้องการคำนวณพื้นที่ได้เฉพาะรูปสามเหลี่ยม
- 1.2 กำหนดข้อมูลนำเข้า ซึ่งประกอบหัวข้อพิจารณา ดังนี้
- กำหนดลักษณะการรับข้อมูล เช่น รับข้อมูลจากแป้นพิมพ์ หรือ อ่านข้อมูลจากไฟล์
 - รูปแบบข้อมูลที่ได้รับเข้ามาเป็นอย่างไร เช่น ข้อมูลชื่อนิติเก็บเป็นตัวอักษรหรือสตริง ข้อมูลเงินเดือนพนักงานเก็บเป็นจำนวนทศนิยม เป็นต้น
 - ขอบเขตของข้อมูลมีช่วงค่าของข้อมูลได้เท่าไร เช่น รับข้อมูลเงินเดือนมีค่าอยู่ระหว่าง 0.00 ถึง 100,000.00 บาท เก็บข้อมูลเป็นจำนวนทศนิยม เป็นต้น
 - ข้อจำกัดในการรับข้อมูลอย่างไรบ้าง เช่น รับข้อมูลได้เฉพาะค่าตัวเลขที่มากกว่า 0 เป็นต้น
- 1.3. วิธีการประมวลผล เป็นขั้นตอนที่สำคัญในการแก้ปัญหา ประกอบด้วยข้อกำหนดดังนี้
- กำหนดวิธีการหรือเทคนิคที่ใช้แก้ปัญหา ปัญหาต่างๆจะมีวิธีการปัญหาแตกต่างกันไป ขึ้นอยู่กับสภาพของปัญหาและผู้แก้ปัญหา และปัญหาหนึ่งๆสามารถแก้ปัญหได้ด้วยหลายวิธีการ ดังนั้นให้เลือกวิธีการแก้ปัญหาที่เหมาะสมกับปัญหานั้นๆ
 - กำหนดขั้นตอนทำงานให้ชัดเจน เพื่อแก้ปัญหาตามลำดับการทำงานของวิธีการที่ได้เลือกใช้ และประมวลผลให้ได้ผลลัพธ์ที่ต้องการ
- 1.4 กำหนดผลลัพธ์
- กำหนดรูปแบบการแสดงผล เช่น แสดงผลลัพธ์เป็นภาพกราฟฟิกส์ทางจอภาพ หรือ พิมพ์ข้อมูลออกทางเครื่องพิมพ์ เป็นต้น
 - ตรวจสอบข้อผิดพลาดและความถูกต้องของผลลัพธ์ เช่น ตรวจสอบข้อผิดพลาดจากการคำนวณ ตรวจสอบผลลัพธ์ว่าถูกต้องตรงตามที่ต้องการหรือไม่

2. การออกแบบโปรแกรม (Program Design)

การออกแบบโปรแกรม (Program Design) ประกอบด้วยวิธีการดังนี้ คือ

1. การออกแบบโปรแกรมโดยใช้อัลกอริทึม (Algorithm) เป็นการอธิบายถึงลำดับขั้นตอนการทำงานของการแก้ปัญหาโดยใช้ประโยคข้อความที่ชัดเจนไม่คลุมเครือ สามารถบอกลำดับการทำงานได้ซึ่งมีประโยชน์ดังนี้

- ทำให้เห็นลำดับของการทำงานและวิธีการทำงานแต่ละขั้นตอนได้อย่างละเอียด
- ทำให้เห็นภาพรวมของการทำงานของขั้นตอนทั้งหมด
- เป็นการวางแผนการทำงานไว้ล่วงหน้า ทำให้สามารถนำไปเขียนเป็นโปรแกรมได้อย่างถูกต้องและรวดเร็ว

ตัวอย่าง อัลกอริทึมการหาพื้นที่รูปสามเหลี่ยมมุมฉาก

1. เริ่มต้น
2. รับค่าความยาวของฐานของรูปสามเหลี่ยม
3. รับค่าความยาวของส่วนสูงของรูปสามเหลี่ยม
4. คำนวณหาพื้นที่รูปสามเหลี่ยมมุมฉาก จากสูตร

$$\text{พื้นที่} = \frac{1}{2} \times \text{ฐาน} \times \text{สูง}$$

5. แสดงผลลัพธ์ ที่ได้จากการคำนวณ คือ พื้นที่ของรูปสามเหลี่ยม
6. จบการทำงาน

2. การออกแบบโปรแกรมโดยใช้รหัสจำลอง (Pseudo Code) เป็นการออกแบบขั้นตอนการทำงานของโปรแกรมโดยใช้ข้อความภาษาอังกฤษที่ใกล้เคียงกับภาษาคอมพิวเตอร์ ซึ่งมีหลักการทำงานและประโยชน์เหมือนกับการใช้อัลกอริทึม แต่มีข้อดี ดังนี้คือ

- สามารถนำรหัสจำลองไปใช้เขียนโปรแกรมได้ง่ายกว่าการใช้อัลกอริทึมเพราะมีความใกล้เคียงกับคำสั่งคอมพิวเตอร์
- ผู้ออกแบบโปรแกรมต้องมีความรู้ทางด้านภาษาคอมพิวเตอร์บ้าง เพื่อให้สามารถนำไปประยุกต์ใช้และเขียนโปรแกรมเป็นภาษาอื่น ๆ ได้หลายภาษา

ตัวอย่าง รหัสจำลอง การหาพื้นที่รูปสามเหลี่ยมมุมฉาก

START

READ BASE

READ HIGH

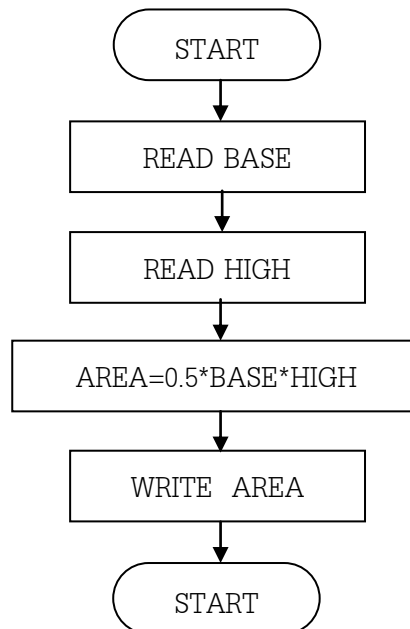
AREA = 0.5 * BASE * HIGH

WRITE AREA

END

3. การออกแบบโปรแกรมโดยใช้ผังงาน (Flowchart) คือ การใช้สัญลักษณ์รูปภาพ หรือกล่องข้อความบรรยายรายละเอียดการทำงาน และใช้ลูกศรบอกทิศทางลำดับของการทำงาน ซึ่งมีข้อดีดังนี้ คือ

- สามารถอ่านและเข้าใจการทำงานได้ง่าย เพราะมองเห็นภาพรวมขั้นตอนการทำงานทั้งหมดได้ชัดเจน
- สามารถออกแบบโครงสร้างการทำงานได้หลากหลายโดยใช้ลูกศรแสดงทิศทางการทำงาน ทำให้แก้ปัญหาที่มีหลายเลือกและซับซ้อนได้
- คำสั่งหรือคำบรรยายรายละเอียดในกล่องข้อความสามารถนำไปเขียนเป็นคำสั่งของโปรแกรมได้



3. การเขียนโปรแกรม (Program Coding)

เป็นขั้นตอนสำคัญหลังจากได้ผ่านการออกแบบโปรแกรมแล้ว โดยการนำแนวคิดจากอัลกอริทึม หรือผังงานมาแปลงให้อยู่ในรูปคำสั่งคอมพิวเตอร์ โดยอาศัยความรู้และทักษะการเขียนโปรแกรมและใช้ภาษาคอมพิวเตอร์รวมทั้งเครื่องมือช่วยในการเขียนโปรแกรมต่างๆ เพื่อให้เกิดผลลัพธ์ถูกต้องและทำงานตามที่เราต้องการ สรุปการเขียนโปรแกรม ต้องพิจารณาองค์ประกอบดังนี้

1. เลือกภาษาที่เหมาะสม
2. ลงมือเขียนโปรแกรม โดยการแปลงขั้นตอนการทำงาน (ประมวลผล) ที่ได้จากการออกแบบ ให้อยู่ในรูปของคำสั่งที่ถูกต้อง ตรงตามรูปแบบของภาษาที่เลือกนั้น

4. การทดสอบและแก้ไขโปรแกรม (Program Testing & Verification)

การทดสอบและแก้ไขโปรแกรม เป็นขั้นตอนการตรวจสอบโปรแกรมที่เขียนได้ ว่าทำงานถูกต้องตรงตามความต้องการของผู้ใช้ หรือตรงตามลักษณะงานของโปรแกรมนั้นหรือไม่

ความผิดพลาด (Errors) ที่สามารถเกิดขึ้นได้จากการเขียนโปรแกรม มีดังนี้

1. Syntax Error ความผิดพลาดที่เกิดจากการใช้คำสั่งผิดรูปแบบที่ภาษานั้นกำหนด เช่น การลืมประกาศตัวแปร การเขียนคำสั่งผิด เช่น คำสั่ง `while()` เป็น `WHILE()`

2. Logic Error ความผิดพลาดที่เกิดจากการที่โปรแกรมทำงานผิดไปจากขั้นตอนที่ควรจะเป็น เช่น การตรวจสอบเงื่อนไขผิดไม่ตรงตามวัตถุประสงค์ คำนวณค่าได้คำตอบไม่ถูกต้อง หรือ ทำงานผิดลำดับขั้นตอน เป็นต้น

3. System Design Error ความผิดพลาดที่เกิดจากการที่โปรแกรมทำงานได้ไม่ตรงตามความต้องการของลูกค้า

ขั้นตอนการทดสอบและแก้ไขโปรแกรม

1. Desk-Checking ผู้เขียนโปรแกรมตรวจสอบโปรแกรมด้วยตนเอง ถ้าให้ผู้อื่นช่วยดูจะเรียกว่า Structured-Walkthrough

2. Translating ตรวจสอบรูปแบบคำสั่งต่างๆที่ใช้ในโปรแกรมโดยตัวแปลภาษา (Translator) เป็นผู้ตรวจ

3. Debugging เป็นการทดลองใช้โปรแกรมจริง เพื่อค้นหาข้อบกพร่อง เช่น ผลลัพธ์ที่ไม่ตรงตามความต้องการ ซึ่งอาจมีสาเหตุจาก Logic Errors และถ้าได้ทดสอบกับผู้ใช้จริงก็จะสามารถตรวจสอบ System Design Errors ได้

5. การจัดทำเอกสารและคู่มือการใช้งาน (Program Documentation)

การจัดทำเอกสารและคู่มือการใช้งานจัดทำเอกสารต่างๆ ที่เกี่ยวข้องกับระบบหรือการเขียนโปรแกรม ได้แก่

1. คู่มือสำหรับผู้ใช้โปรแกรม (User's Manual or User's Guide) คือเอกสารที่อธิบายวิธีการใช้ระบบหรือโปรแกรม เรียกว่า User Manual ใช้สำหรับผู้ใช้งานโปรแกรม แนะนำวิธีการใช้งานโปรแกรม แนะนำคุณสมบัติและองค์ประกอบของโปรแกรมต่างๆ วิธีการติดตั้งโปรแกรม สามารถทำควบคู่ไปกับการเขียนโปรแกรม อาจทำเป็นคู่มือเอกสารที่อยู่ในรูปแบบโปรแกรมออนไลน์ก็ได้ (Online Manual)

2. คู่มือสำหรับผู้เขียนโปรแกรม (Programmer's Manual or Programmer's Guide) เป็นคู่มือที่จัดทำขึ้นเพื่อให้ผู้พัฒนาโปรแกรม รวมทั้งเทคนิคพิเศษต่างๆของโปรแกรม เพื่อให้สะดวกต่อการปรับปรุงแก้ไขโปรแกรมที่มีอยู่เดิม โดยทั่วไปจะเป็นเอกสารแสดงการวิเคราะห์และออกแบบระบบ เรียกว่า System Manual ใช้สำหรับผู้พัฒนาระบบหรือโปรแกรม เท่านั้น

1.6 การใช้งานจริง (Program Implement)

การใช้งานจริง เป็นขั้นตอนสำคัญหลังจากทำการทดสอบและแก้ไขโปรแกรมให้มีความถูกต้อง
เรียบร้อยแล้ว โดยการนำโปรแกรมไปใช้งานจริงด้วยการป้อนข้อมูลต่างๆ สภาพแวดล้อมและสถานการณ์ต่างๆโดยผู้ใช้งานโปรแกรมสามารถทำงานตามฟังก์ชันและทำตามจุดประสงค์ของโปรแกรมที่เขียนไว้ ขั้นตอนการใช้งานจริงของโปรแกรม หากพบข้อผิดพลาดก็สามารถปรับปรุง แก้ไข โปรแกรมให้ถูกต้องได้

1.7 การปรับปรุงและพัฒนาโปรแกรม (Program Maintenance)

การเขียนโปรแกรมที่ดีต้องมีขั้นตอนการปรับปรุงและพัฒนาโปรแกรมให้มีความถูกต้องทันสมัย และตรงกับความต้องการของผู้ใช้มากที่สุด โดยทั่วไปโปรแกรมที่ใช้งานจะประกอบด้วยหลายๆรุ่น เช่นรุ่นทดสอบ (Beta Version) และ รุ่นที่ใช้งานจริง (Release Version) และต้องมีการปรับเปลี่ยนโปรแกรมให้ดีขึ้น ดังตัวอย่างเช่นโปรแกรมเวอร์ชัน 1 มีการเพิ่มเติม ปรับปรุง แก้ไขโปรแกรมเป็นเวอร์ชัน 1.2 เป็นต้น การพัฒนาโปรแกรมให้ดีขึ้นมีประโยชน์ต่อผู้ใช้งานและผู้เขียนโปรแกรมต้องอาศัยคู่มือการใช้งานและเอกสารประกอบของโปรแกรมเพื่อเป็นแนวทางในการแก้ไขและให้ผู้อื่นๆสามารถพัฒนาต่อได้

