

บทที่ 1

1. กระบวนการในการพัฒนาซอฟต์แวร์ ประกอบด้วยกิจกรรมหลักและกิจกรรมย่อยอะไรบ้าง

- 1) การจัดทำข้อกำหนดของซอฟต์แวร์ ประกอบด้วย 4 กิจกรรมย่อย ได้แก่ การศึกษาความเป็นไปได้ การวิเคราะห์ความต้องการ การกำหนดความต้องการ และการตรวจสอบความต้องการ
- 2) การออกแบบซอฟต์แวร์และการนำไปใช้ ประกอบด้วย 4 กิจกรรมย่อย ได้แก่ การออกแบบ สถาปัตยกรรม การออกแบบส่วนต่อประสานกับผู้ใช้ การออกแบบส่วนประกอบของระบบ และการออกแบบฐานข้อมูล
- 3) การตรวจสอบซอฟต์แวร์ ประกอบด้วย 3 กิจกรรมย่อย ได้แก่ การทดสอบส่วนประกอบ การทดสอบระบบ และการทดสอบเพื่อการยอมรับ
- 4) การพัฒนาต่อของซอฟต์แวร์ เป็นกิจกรรมที่ดำเนินการเพื่อให้ซอฟต์แวร์สามารถทำงานได้ตรงกับความต้องการของผู้ใช้ที่เปลี่ยนแปลงไป ซึ่งมีลักษณะใกล้เคียงกับการบำรุงรักษาซอฟต์แวร์

2. จงเปรียบเทียบแบบจำลองน้ำตกและไจล์ โดยสรุปเป็นตาราง พร้อมยกตัวอย่างระบบที่เหมาะสม

ด้าน	แบบจำลองน้ำตก	อไจล์
กรณีที่ต้องการของผู้ใช้ยังไม่ชัดเจน	น้อย	มาก
ใช้เทคโนโลยีที่ไม่คุ้นเคย	น้อย	น้อย
ความยุ่งยากซับซ้อน	ปานกลาง	น้อย
ความเชื่อถือได้	ปานกลาง	ปานกลาง
ใช้เวลาสั้น	น้อย	มาก
ตารางเวลาที่ชัดเจน	น้อย	น้อย

การพัฒนาซอฟต์แวร์ในแบบอไจล์จะเหมาะกับระบบที่มีความต้องการที่ยังไม่ชัดเจน มีการเปลี่ยนแปลงความต้องการอยู่ตลอดเวลา เช่น ระบบซอฟต์แวร์ที่ยังอาศัยข้อมูลจากการศึกษา และวิจัย ระบบซอฟต์แวร์ที่ผู้ใช้ยังไม่สามารถระบุความต้องการที่แท้จริงได้ในระยะเริ่มแรกของโครงการ

3. จงสรุปความเหมาะสมของแบบจำลองกระบวนการพัฒนาซอฟต์แวร์แต่ละแบบ โดยระบุข้อดีและข้อจำกัด

ด้าน	แบบน้ำตก	แบบจำลองวี	แบบวนซ้ำ	ต้นแบบระบบ	อไจล์
กรณีที่ต้องการของผู้ใช้ยังไม่ชัดเจน	น้อย	น้อย	ปานกลาง	มาก	มาก
ใช้เทคโนโลยีที่ไม่คุ้นเคย	น้อย	น้อย	ปานกลาง	น้อย	น้อย
ความยุ่งยากซับซ้อน	ปานกลาง	ปานกลาง	ปานกลาง	น้อย	น้อย
ความเชื่อถือได้	ปานกลาง	มาก	ปานกลาง	น้อย	ปานกลาง
ใช้เวลาสั้น	น้อย	น้อย	มาก	มาก	มาก
ตารางเวลาที่ชัดเจน	น้อย	น้อย	มาก	มาก	น้อย

4. หากท่านได้รับมอบหมายให้เป็นผู้จัดการโครงการพัฒนาซอฟต์แวร์การประเมินการสอนของอาจารย์

มหาวิทยาลัยแห่งหนึ่ง ซึ่งมีจำนวนอาจารย์ 1,000 คน และนักศึกษา 10,000 คน โดยระบบใหม่นี้จะต้องรองรับการใช้งานบน Smart Devices เช่น smart phone และ tablet ท่านจะเลือกแบบจำลองแบบใดในการพัฒนาระบบและมีปัจจัยใดในการเลือก จงอธิบายโดยละเอียด พร้อมยกตัวอย่างประกอบ

เนื่องจากความต้องการยังต้องการรายละเอียดอีกมาก ดังนั้นอาจจะเลือกใช้วิธีการพัฒนาแบบสกรัมซึ่งเป็นวิธีการพัฒนาซอฟต์แวร์แบบหนึ่งของอไจล์ โดยให้นักพัฒนาแบ่งงานออกเป็นงานย่อย ๆ ไว้ใน Product backlog และพัฒนางานชิ้นย่อย ให้เสร็จภายใน 2 สัปดาห์ โดยแต่ละวันจะมีการประชุมในทุกเช้าเพื่อให้ทีมได้รับทราบความคืบหน้าและปัญหา นอกจากนี้ ในการประชุมควรจะให้ผู้ใช้งานหรือผู้ที่รับผิดชอบเกี่ยวกับระบบซอฟต์แวร์เข้าร่วม โดยสมาชิกในทีมประกอบด้วยตำแหน่งงานทางด้านการพัฒนาซอฟต์แวร์ ตั้งแต่นักวิเคราะห์และออกแบบระบบ นักพัฒนา นักออกแบบ GUI นักทดสอบระบบ เป็นต้น ทุกตำแหน่งในทีมมีบทบาทและความสำคัญเท่ากัน สมาชิกภายในทีมสามารถแสดงความคิดเห็นได้ทุกเรื่อง โดยไม่ได้ให้ความสำคัญกับตำแหน่งงานในทีม นั่นคือสมาชิกในทีมต้องช่วยกันทำงานทุกอย่างให้ลุล่วงตามวัตถุประสงค์และเวลาที่กำหนด จากบทบาทและหน้าที่ของทีม หากสมาชิกคนหนึ่งในทีมมีปัญหา ทั้งทีมต้องช่วยกันแก้ไขปัญหาเพื่อให้งานสำเร็จ ซึ่งต่างจากขั้นตอนการพัฒนาแบบเดิมที่แบ่งหน้าที่กันอย่างชัดเจน และไม่ก้าวร้าวกันเมื่อคนใดคนหนึ่งประสบปัญหาต้องหาวิธีการแก้ไขปัญหาด้วยตนเอง ทำให้สมาชิกคนอื่นที่ทำงานในทีมเดียวกัน ไม่สามารถทราบปัญหาหรือวิธีการแก้ไขปัญหา ความรู้ทั้งทางด้านเทคนิคไม่ถ่ายทอดและแบ่งปันไปยังสมาชิกคนอื่นภายในทีม เป็นผลให้การทำงานของทีมล่าช้า

5. ระบุปัญหาและอุปสรรคที่ทำให้เกิดความล้มเหลวในการเลือกแบบจำลองที่ไม่เหมาะสม จงยกตัวอย่าง พร้อมอธิบายโดยละเอียด

การเลือกแบบจำลองที่ไม่เหมาะสมจะทำให้โครงการไม่ประสบความสำเร็จตามที่คาดหวัง อันเกิดจากสาเหตุต่อไปนี้

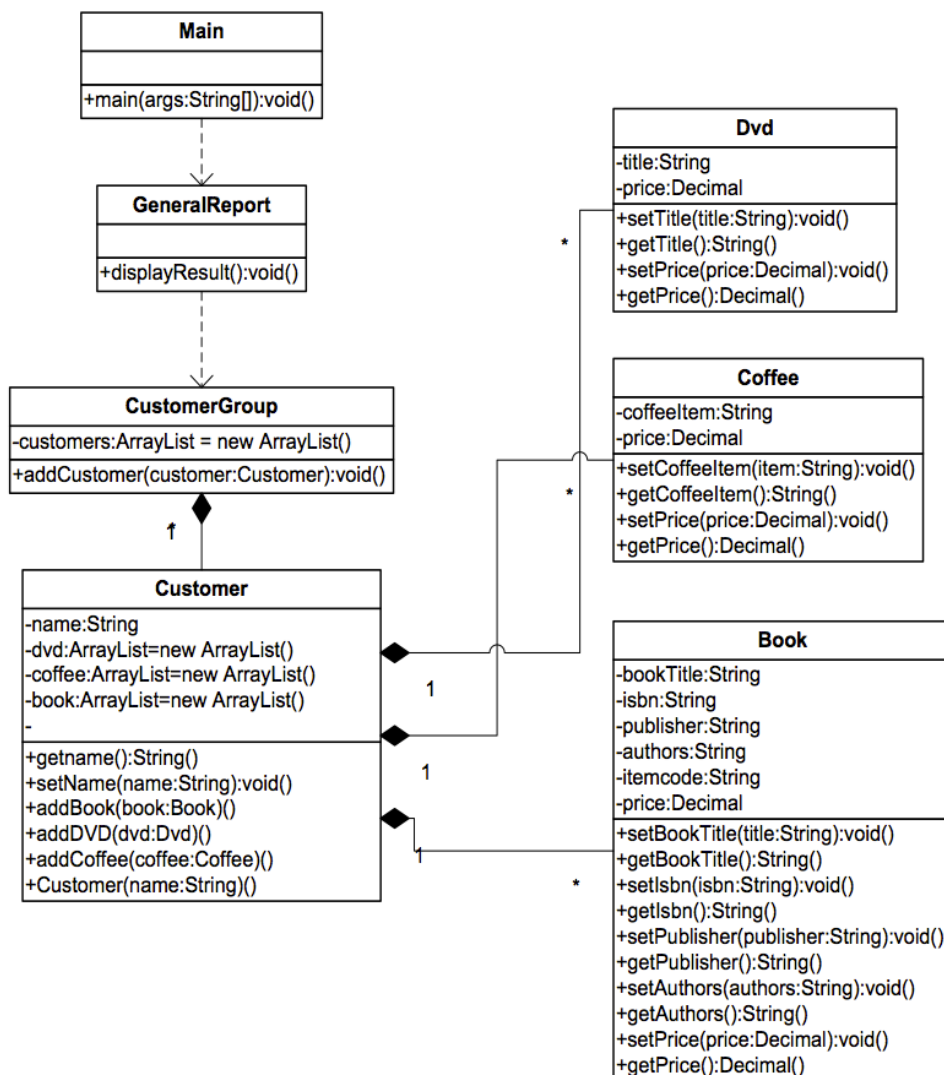
- ระยะเวลาในการดำเนินโครงการไม่เป็นไปตามกำหนด
- เกิดค่าใช้จ่ายมากกว่าที่กำหนด เนื่องจากระยะเวลาที่ช้ากว่ากำหนดจะทำให้เกิดค่าใช้จ่ายตามมา เช่น ค่าจ้างนักพัฒนา หรือการชดเชยตามสัญญาที่ได้ตกลงกับลูกค้า นอกจากนี้จะทำให้
- การเลือกแบบจำลองที่ไม่เหมาะสมอาจจะทำให้ทรัพยากรที่มากกว่าความจำเป็น ซึ่งจะทำให้การบริหารยากขึ้นและมีค่าใช้จ่ายที่สูงขึ้น
- การเลือกแบบจำลองบางแบบ เช่น แบบจำลองน้ำตก จะเหมาะสมกับโครงการที่มีความต้องการ และผู้ใช้กำหนดคุณลักษณะของซอฟต์แวร์ได้ชัดเจน หากนำไปใช้กับโครงการที่ยังขาดความต้องการ อาจส่งผลให้การแก้ไข ปรับปรุงคุณภาพของซอฟต์แวร์ทำได้ยากเนื่องจาก การย้อนกลับไปแก้ไข ขั้นตอนแรก ๆ ไม่สามารถทำได้
- มีปัญหากับผู้จ้าง หรือเจ้าของระบบเนื่องจากไม่ได้มีการสื่อสารกันที่ดี และขาดการมีส่วนร่วมระหว่างทีมพัฒนาและผู้จ้าง

บทที่ 2

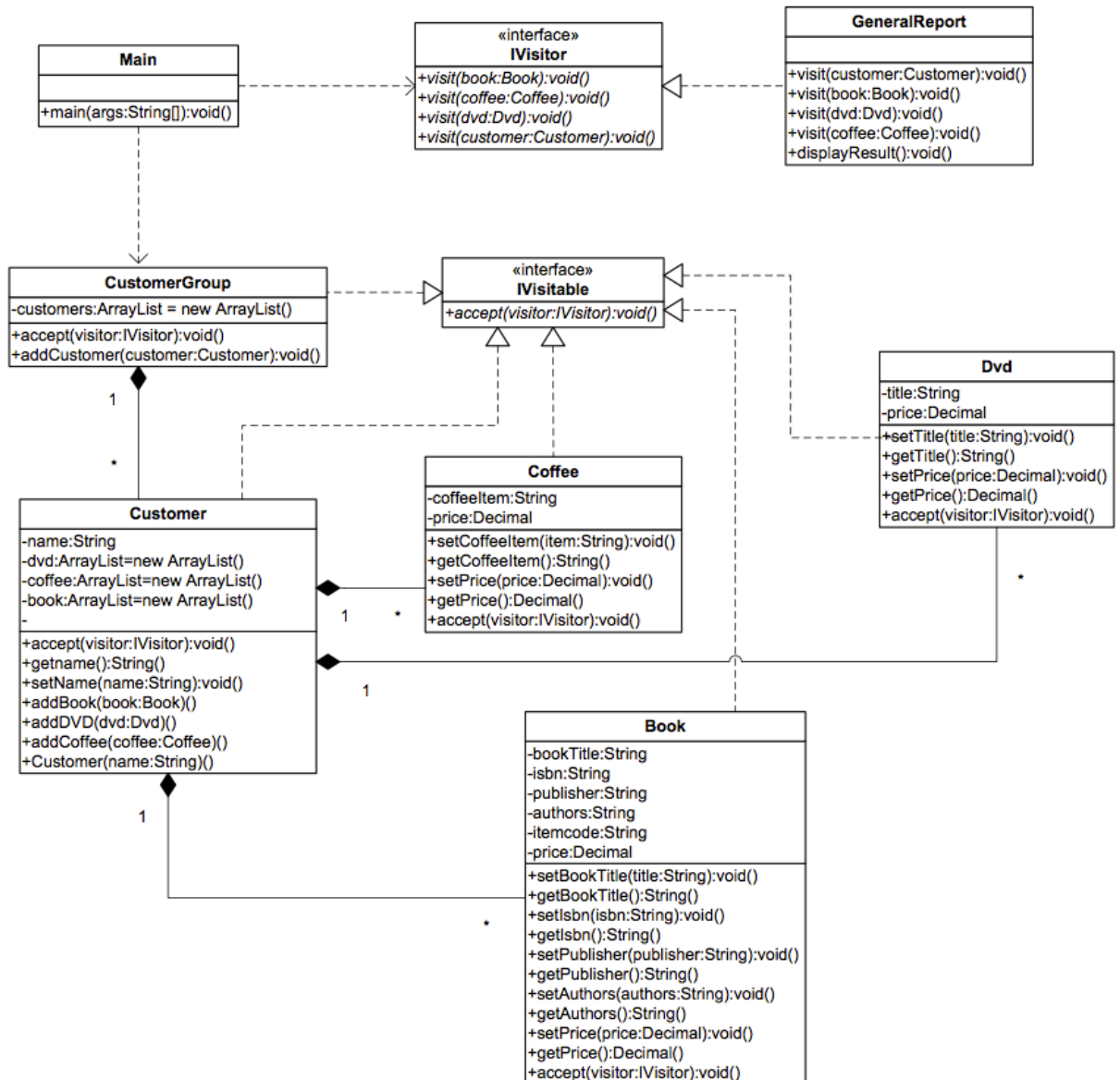
1. จงอธิบายความสำคัญของแบบแผนการออกแบบซอฟต์แวร์ (Design Pattern)

- ช่วยทำให้การสื่อสารภายในทีมพัฒนาซอฟต์แวร์ดีขึ้น
- ลดระยะเวลาในการออกแบบซอฟต์แวร์ เนื่องจากมีตัวอย่างให้ศึกษา ซึ่งได้ถูกใช้งานจริงมาแล้ว
- ช่วยเพิ่มการนำซอฟต์แวร์มาใช้ใหม่ได้ ทำให้ลดปัญหาของการออกแบบซอฟต์แวร์
- แก้ไขปัญหาได้เร็วขึ้นเนื่องจากในทีมพัฒนามีความเข้าใจที่ตรงกัน

2. จากแผนภาพคลาสของระบบรายงานสำหรับร้านค้าแห่งหนึ่ง ซึ่งมีการจำหน่ายสินค้าด้านความบันเทิง ได้แก่ DVD VCD ภาพยนตร์ และหนังสือ นอกจากนี้ ทางร้านยังมีการจำหน่ายกาแฟให้กับลูกค้าที่มาใช้บริการ โดยระบบรายงานนี้จะแสดงข้อมูลรายละเอียดของลูกค้าและสินค้าที่ลูกค้าเคยซื้อ ให้นักศึกษาทำการออกแบบระบบใหม่โดยนำแบบแผน Visitor มาประยุกต์ใช้



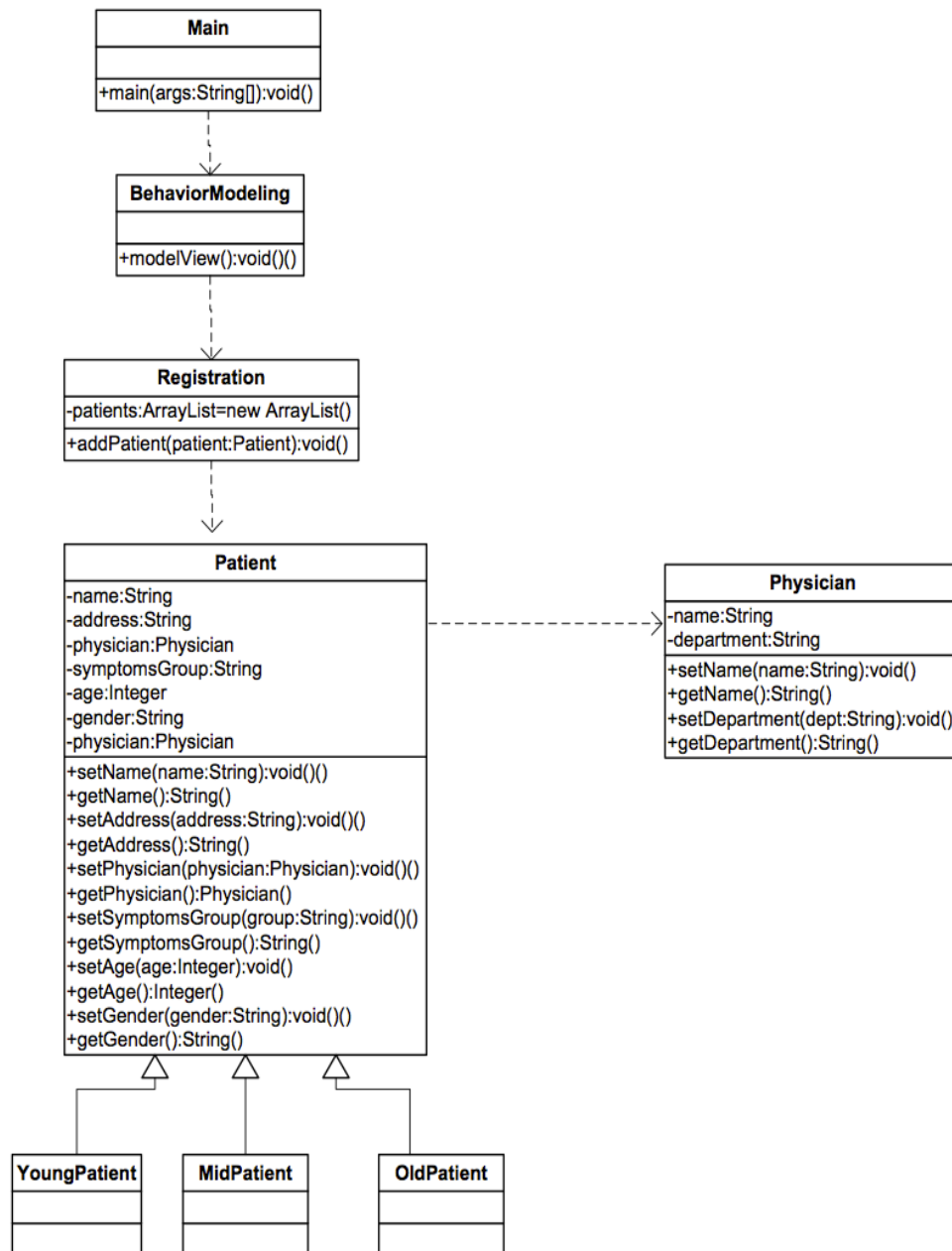
สามารถนำ Visitor pattern มาประยุกต์ใช้กับระบบได้ดังแผนภาพ UML class diagram ดังนี้



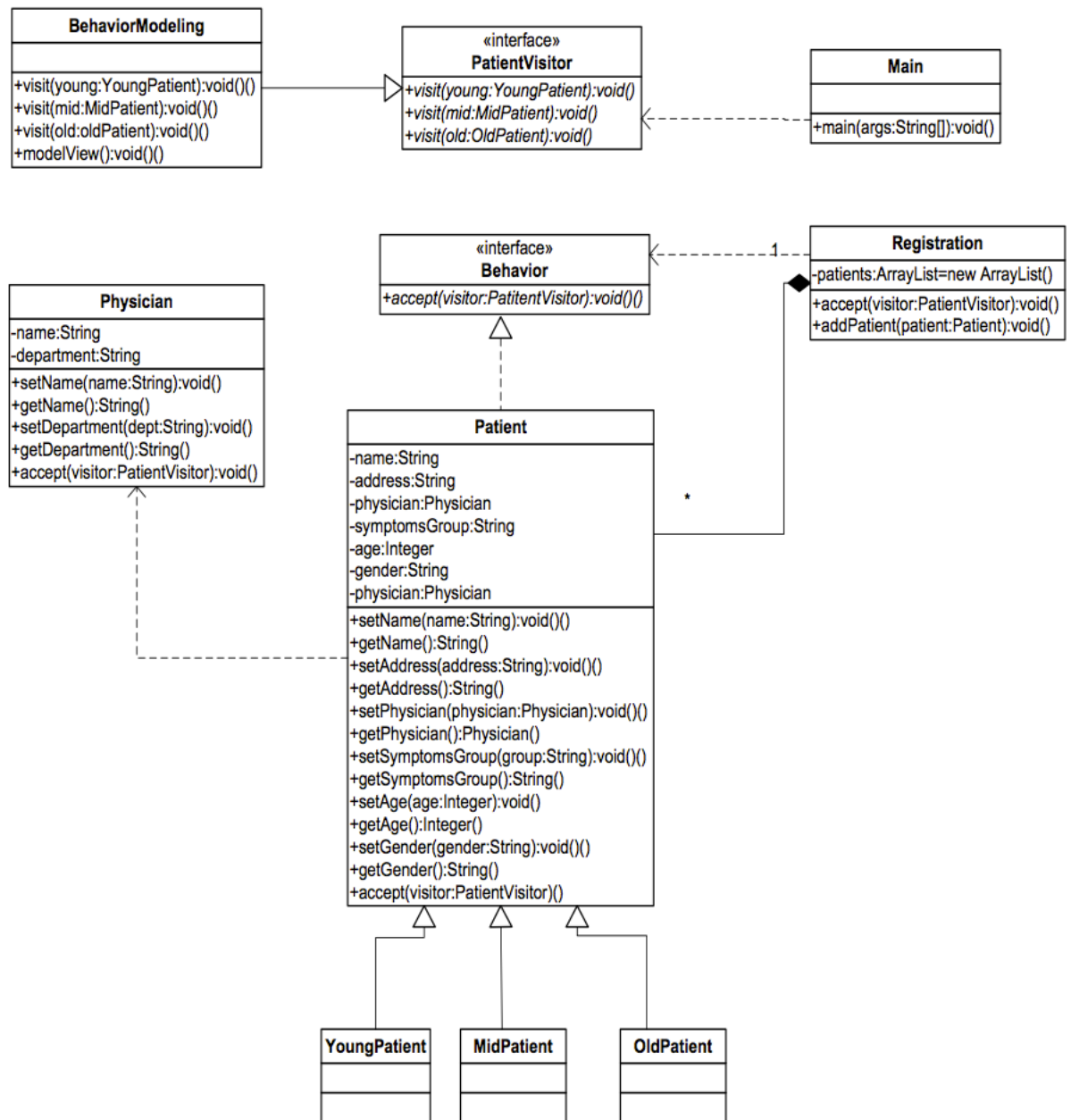
3. จากแผนภาพคลาสของระบบการจัดเก็บข้อมูลของคนไข้ (Patient) ที่เข้ามารับการรักษาพยาบาลของโรงพยาบาลแห่งหนึ่งเพื่อนำมาแสดงข้อมูลแบบจำลองพฤติกรรมของคนไข้ (Patient's Behavior Model) ซึ่งในระบบมีการแบ่งกลุ่มคนไข้ตามช่วงอายุเป็น 3 กลุ่ม ดังนี้

- Young Patient (0-15 ปี)
- Mid Patient (16-45 ปี)
- Old Patient (มากกว่า 45 ปี)

ให้นักศึกษาออกแบบระบบใหม่โดยประยุกต์ใช้แบบแผนการออกแบบซอฟต์แวร์อย่างน้อย 1 แบบแผน



สามารถนำ Visitor pattern มาประยุกต์ใช้กับระบบได้ดังแผนภาพ UML class diagram ดังนี้



4. จงอธิบายความหมายและประเภทของการเกี่ยวข้องกัน

การเกี่ยวข้องกันระหว่างโมดูล (Coupling) เป็นการวัดความสัมพันธ์ระหว่างโมดูล 2 โมดูลว่ามีความซับซ้อนหรือมีระดับการขึ้นต่อกันมากน้อยเพียงใด โครงสร้างของโมดูลที่ดีจะต้องมีระดับการขึ้นต่อกันของโมดูลน้อย (Loosely coupled) ประเภทของการเกี่ยวข้องกันระหว่างโมดูล ซึ่งมี 2 ประเภท คือ ความสัมพันธ์แบบสืบทอด (Inheritance coupling) และการเกี่ยวข้องกันแบบปฏิสัมพันธ์ (Interaction coupling)

5. จงอธิบายความหมายและประเภทของการเกี่ยวข้องกันแบบปฏิสัมพันธ์

การเกี่ยวข้องกันระหว่างเมธอดและอ็อบเจกต์ต่าง ๆ ผ่านการส่งผ่านข้อความ (Message passing) เพื่อเรียกให้เมธอดหรือคลาสอื่นทำงานตามต้องการ การเกี่ยวข้องกันแบบปฏิสัมพันธ์มี 6 ประเภท

- 1) การเกี่ยวข้องกันด้วยเนื้อหา (Content coupling)

- 2) การเกี่ยวข้องกันแบบทั่วไป (Common coupling)
- 3) การเกี่ยวข้องกันด้วยการควบคุม (Control coupling)
- 4) การเกี่ยวข้องกันด้วยข้อมูลบางส่วน (Stamp coupling)
- 5) การเกี่ยวข้องกันด้วยข้อมูลทั้งหมด (Data coupling)
- 6) ไม่มีการเกี่ยวข้องกันโดยตรง (No direct coupling)

6. ข้อตกลงของเมอร์ทอดคืออะไร ประกอบด้วยอะไรบ้าง

ข้อตกลงของเมอร์ทอดประกอบด้วยข้อมูลที่จำเป็นสำหรับนักพัฒนาอ็อบเจกต์ผู้ให้บริการ เพื่อให้ทราบว่าข้อความใดสามารถส่งไปยังอ็อบเจกต์ผู้ให้บริการและสิ่งที่อ็อบเจกต์ผู้ให้บริการคาดว่าจะได้รับการคืนค่ากลับมาจากอ็อบเจกต์ผู้ให้บริการ โดยข้อตกลงประกอบด้วยส่วนต่าง ๆ ดังนี้

- 1) ชื่อเมอร์ทอด และชื่อคลาส
- 2) รหัสข้อตกลง
- 3) อ็อบเจกต์ผู้ให้บริการ
- 4) ยูสเคสที่สัมพันธ์กัน
- 5) คำอธิบายความรับผิดชอบ
- 6) การรับอาร์กิวเมนต์
- 7) ชนิดข้อมูลที่คืนค่า
- 8) พรีคอนดิชัน (Precondition) และโพสต์คอนดิชัน (Postcondition)

7. คุณลักษณะของการออกแบบที่ดีมีอะไรบ้าง

- 1) มีความยืดหยุ่นในการปรับตัว (Portability)
- 2) มีความถูกต้องและเชื่อถือได้ (Correctness & reliability)
- 3) มีประสิทธิภาพ (Efficiency)
- 4) มีความยืดหยุ่นต่อการใช้งาน (Flexibility)

8. จงอธิบายความแตกต่างระหว่าง High-level design และ Detailed design

การออกแบบระดับสูง (High-level design) เป็นการออกแบบระบบที่มีการแบ่งระบบออกเป็นหลายองค์ประกอบ ตามแนวคิดของการออกแบบสถาปัตยกรรม กล่าวคือ มีระบบย่อยและโมดูลที่แสดงให้เห็นถึงการมีปฏิสัมพันธ์กับส่วนอื่น ๆ การออกแบบระดับสูงมุ่งเน้นไปที่วิธีการที่ระบบสามารถทำงานพร้อมกันทุกองค์ประกอบในรูปแบบของโมดูล และตระหนักถึงโครงสร้างแบบโมดูลของแต่ละระบบย่อย รวมถึงความสัมพันธ์และการมีปฏิสัมพันธ์ระหว่างกัน

การออกแบบรายละเอียด (Detailed design) เรียกอีกอย่างหนึ่งว่า Implementation design เป็นการออกแบบรายละเอียดในส่วนการดำเนินการ ทั้งในการออกแบบสถาปัตยกรรมและการออกแบบระดับสูง ซึ่งมีรายละเอียดมากขึ้นต่อโมดูลและการใช้งาน ถูกกำหนดโครงสร้างเชิงตรรกะของแต่ละโมดูล และการเชื่อมต่อการ

สื่อสารกับโมดูลอื่น ๆ ซึ่งเป็นการอธิบายรายละเอียดของแต่ละส่วนประกอบของซอฟต์แวร์ เพื่อให้อ่านง่ายต่อการเขียนโปรแกรมให้มากที่สุด

9. การหาค่าแฟกทอเรียล (Factorial) ต้องหาจากจำนวนเต็มใด ๆ ที่มากกว่าหรือเท่ากับ 0 เท่านั้น ดังนั้น เมธอดที่คำนวณหาค่าแฟกทอเรียลของจำนวนใด ๆ ควรมีพรีคอนดิชันเป็นอย่างไร

precondition: $n \geq 0$

10. เมธอดสำหรับการหารากที่ 2 ของจำนวนใด ๆ ต้องมีโพสท์คอนดิชันว่าอย่างไร

postcondition: $\text{this.sqrt}(x) \geq 0.0$

11. จากภาพที่ 2.36 จงเขียนข้อตกลงของเมธอด middleListInsert โดยให้สมมติรหัส ID ตามต้องการ

Class Name: OrderList
Method Name: middleListInsert
ID: 150
Clients (consumers): Customer
Associated Use Cases: addCustomerOrder
Description of Responsibilities: แทรกอีอบเจกต์ OrderNode ไปยังตำแหน่งที่ถูกต้องในลิงค์ลิสต์เดี่ยวแบบเรียงลำดับของอีอบเจกต์ Order
Arguments Received: newOrderNode:OrderNode
Type of Value Returned: void
Precondition: ไม่มี
Postcondition: ไม่มี

บทที่ 3

1. ในการพัฒนาโปรแกรมมีข้อควรพิจารณาที่สำคัญอะไรบ้าง

- 1) ส่วนโครงสร้างควบคุมการทำงานของโปรแกรม
- 2) อัลกอริทึม
- 3) โครงสร้างข้อมูล

2. หลักการเขียนโปรแกรมเชิงโครงสร้างแตกต่างจากหลักการเขียนโปรแกรมเชิงวัตถุอย่างไร จงอธิบายโดยละเอียด พร้อมยกตัวอย่าง

การเขียนโปรแกรมเชิงโครงสร้าง (Structured Programming) คือการกำหนดขั้นตอนให้เครื่องคอมพิวเตอร์ทำงาน มีการจัดการคำสั่งต่าง ๆ ให้มีรูปแบบมาตรฐานที่เขียนโปรแกรมและตรวจสอบได้ง่าย จะต่างกับการเขียนโปรแกรมเชิงวัตถุ เนื่องจากไม่ได้มีการพิจารณาประเภทของวัตถุ กล่าวคือ วัตถุหนึ่ง ๆ จะรวบรวมข้อมูลและกระบวนการเข้าด้วยกัน โดยจะมีคลาสเป็นตัวกำหนดคุณสมบัติ และสามารถสืบทอดคุณสมบัติได้ อีกทั้งจะนำไปใช้โปรแกรมอื่นที่ต้องการการทำงานเหมือนกันในลักษณะของคลังโปรแกรม (Library) ซึ่งมีการนำกลับมาใช้ใหม่ (Reusable) ทำให้ลดขั้นตอนการพัฒนาโปรแกรมลงได้

3. จงเขียนคำสั่งเทียม (Pseudocode) ของเมธอดที่ตรวจสอบว่าค่าที่เป็นพารามิเตอร์ของเมธอดเป็นจำนวนเฉพาะหรือไม่

```
procedure prime_number : number
  FOR loop = 2 to number - 1
    check if number is divisible by loop
    IF divisible
      RETURN "NOT PRIME"
    END IF
  END FOR
  RETURN "PRIME"
end procedure
```

4. จงเขียนเมธอดในภาษาจาวาที่สอดคล้องกับคำสั่งเทียม (Pseudocode) ที่ให้มาต่อไปนี้

```
Declare an integer variable arr
Declare an integer variable loopcounter

Set arr to size 10
for loopcounter = 0 to (size of arr)-1
  arr[loopcounter]=random()
  loopcounter = loopcounter + 1
  print arr[loopcounter]
endfor
```

```
public void question4(){
    int arr;
    arr = new int[10];
    for(int i = 0; i < arr.size(); i++){
        arr[i] = random();
        print arr[i];
    }
}
```

5. จงยกตัวอย่างการตั้งชื่อตัวแปรที่ดีและไม่ดีในภาษาจาวา

ตัวอย่างที่ดี	ตัวอย่างที่ไม่ดี
String staffName int empld double expectedResult	String sname int id double test

6. จงยกตัวอย่างการตั้งชื่อเมธอดที่ดีและไม่ดีในภาษาจาวา

ตัวอย่างที่ดี	ตัวอย่างที่ไม่ดี
public void setEmpName(String name) public int getEmpAge(int empld) public double calProfit()	public void setXX(String name) public int getYY() public double CalProfit()

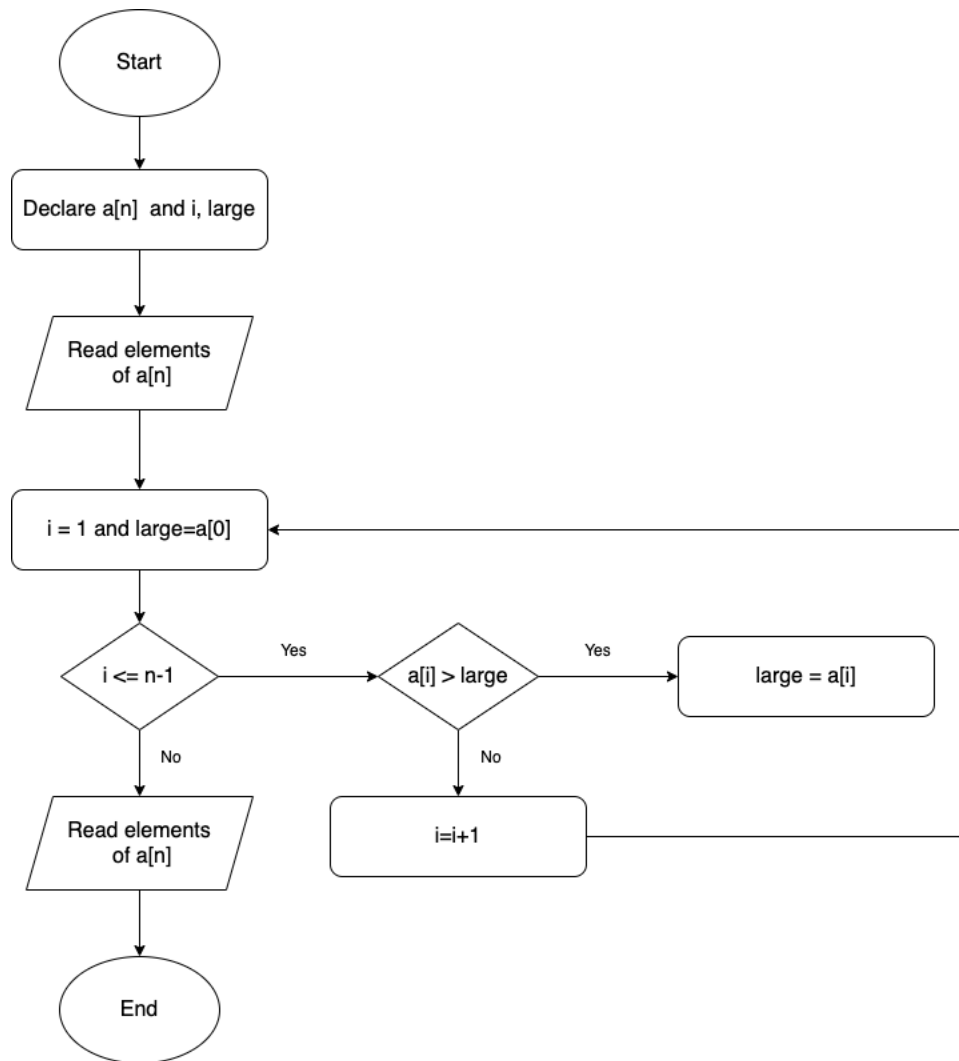
7. จงยกตัวอย่างความสัมพันธ์ของคลาสในลักษณะของ use-a ในภาษาจาวา

ในเมธอด main เป็นเรื่องเกี่ยวกับการบวกเวกเตอร์เพื่อสร้างอ็อบเจกต์ เมื่อเมธอด main จบการทำงาน วัตถุก็จะถูกทำลายไป

8. จงอธิบายประโยชน์ของการสืบทอดคุณสมบัติในการเขียนโปรแกรมเชิงวัตถุ

โปรแกรมที่เคยเขียนไว้มีโอกาสที่จะถูกนำกลับมาใช้ใหม่และเอามาเพียงคลาสนั้นๆ ทั้งนี้ คุณสมบัติต่าง ๆ หรือตัวแปรได้เก็บรวบรวม เป็นคลาสดียวกัน เมื่อประกาศอ็อบเจกต์ของคลาสนั้นมา คุณสมบัติต่าง ๆ รวมถึงฟังก์ชันที่ใช้งานของคลาสนี้ก็จะได้รับการประกาศให้ใช้งานด้วย คุณสมบัติของโปรแกรมอ็อบเจกต์ในลักษณะนี้เรียกว่า การนำกลับมาใช้ใหม่ (Reusability)

9. จงเขียนไดอะแกรมที่แสดงอัลกอริทึมการหาจำนวนค่าที่มากที่สุดของอาร์เรย์ตัวเลข



10. เอกสารโปรแกรมแบ่งเป็นกี่ชนิด อะไรบ้าง

- 1) การจัดทำเอกสารภายใน
- 2) การจัดทำเอกสารภายนอก

บทที่ 4

1. จงอธิบายขั้นตอนการทดสอบระบบซอฟต์แวร์โดยทั่วไป

- 1) การวางแผนการทดสอบ
- 2) การวิเคราะห์และการออกแบบ
- 3) การสร้างและการดำเนินการทดสอบ
- 4) การรายงานและประเมินผล
- 5) การสิ้นสุดกระบวนการทดสอบ

2. จงอธิบายความแตกต่างระหว่าง Error และ Bug

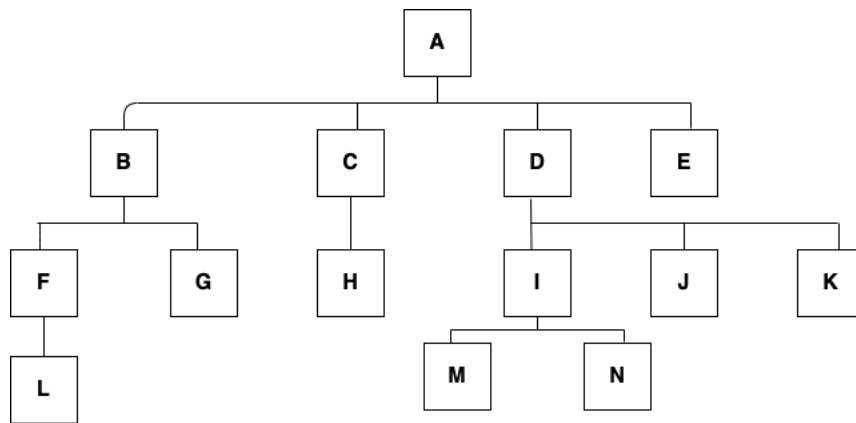
- ความผิดพลาด (Error) คือ ความคลาดเคลื่อนที่เกิดขึ้นระหว่างการคำนวณ การสังเกตการวัดค่าหรือเงื่อนไขตามความเป็นจริงที่ถูกกำหนด หรือความแตกต่างระหว่างผลลัพธ์และผลที่แท้จริง นอกจากนี้ยังหมายถึงข้อผิดพลาดที่เกิดขึ้นจากมนุษย์
- ข้อบกพร่อง (Defect หรือ Bug) คือ ข้อบกพร่องภายในส่วนประกอบหรือระบบที่ส่งผลให้ระบบเกิดความล้มเหลวในการทำงาน

3. จงอธิบายประโยชน์ของสแต็บ (Stub) และไดรเวอร์ (Driver) พร้อมยกตัวอย่างประกอบ

- ไดรเวอร์ (Driver) คือ โปรแกรมหลักที่ทำหน้าที่รับข้อมูลกรณีทดสอบเพื่อจำลองการเรียกใช้งานส่วนต่อประสานหรือฟังก์ชันการทำงานต่าง ๆ ของโมดูลที่ต้องการทดสอบ และทำการรายงานผลการทดสอบ
- สแต็บ (Stub) คือ โปรแกรมที่ทำหน้าที่เหมือนส่วนต่อประสานหรือฟังก์ชันการทำงานที่ต้องการทดสอบการเรียกใช้งาน แต่มักไม่มีการประมวลผลข้อมูล โดยมีวัตถุประสงค์เพื่อตรวจสอบและควบคุมความถูกต้องในการเข้าและออกของส่วนที่ทำการทดสอบเท่านั้น

สำหรับประโยชน์ของไดรเวอร์ และสแต็บคือ ทำให้นักพัฒนาสามารถทดสอบซอฟต์แวร์ในส่วนย่อย ได้โดยไม่จำเป็นต้องรอให้ระบบเสร็จสมบูรณ์ก่อน เช่น ในการทดสอบ ฟังก์ชันการ Login อาจจะมีการทำ สแต็บ และไดรเวอร์ เพื่อทำการป้อนข้อมูลและตรวจสอบโดยไม่จำเป็นต้องทำในส่วนที่ติดต่อกับฐานข้อมูลเสร็จสิ้นก่อน

4. จากภาพเป็นตัวอย่างของระบบประกอบด้วยโมดูลในลำดับชั้นต่าง ๆ จงอธิบายการทดสอบโดยใช้วิธี Bottom-up, Top-down, Big bang และ Sandwich



Bottom-up

$L + F \rightarrow G + F \rightarrow B + F + G$

$H + C$

$M + N + I \rightarrow I + J + K + D \rightarrow D + E$

$A + B + C + D + E$

Top-down

$A + B + C + D + E$

$B + F + G$

$C + H$

$D + E$

$L + F$

$D + I + J + K$

$M + N + I$

Big bang

A

Sandwich

$L + F + G$

$H + C$

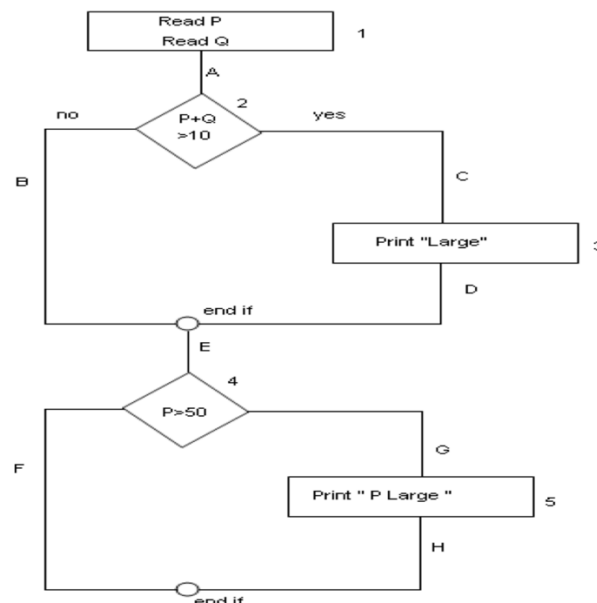
$M + N + I \rightarrow I + J + K$

$A + B + C + D + E$

$B + F + G$

$D + I + J + K$

5. จากแผนผัง Flow chart ให้แสดง Statement Coverage, Branch Coverage, และ Path Coverage



Statement coverage:

หาจำนวนเส้นทางที่ผ่านทุกโหนด ในที่นี้คือ 1, 2, 3, 4 และ 5 เส้นทางดังกล่าว คือ 1A-2C-3D-E-4G-5H เส้นทางเดียว ดังนั้น Statement coverage เท่ากับ 1

Branch coverage:

หาจำนวนเส้นทางที่ผ่านเส้นทางที่เดินไปยังโหนด เส้นทางดังกล่าว คือ 1A-2C-3D-E-4G-5H และ 1A-2B-E-4F ดังนั้น Branch coverage เท่ากับ 2

Path coverage:

หาจำนวนเส้นทางที่ครอบคลุมเส้นทางทั้งหมด เส้นทางดังกล่าว ได้แก่ 1A-2B-E-4F, 1A-2B-E-4G-5H, 1A-2C-3D-E-4G-5H และ 1A-2C-3D-E-4F ดังนั้น Path coverage เท่ากับ 4

6. จงอธิบายความแตกต่างระหว่างการทดสอบแบบ Black box และ White box

- การทดสอบแบบกล่องดำ ผู้ที่ทำการทดสอบในลักษณะนี้ไม่จำเป็นต้องมีความรู้เรื่องระบบว่ามีการออกแบบหรือเขียนโปรแกรมอย่างไร สนใจเฉพาะว่าระบบมีฟังก์ชันการทำงานอย่างไร มีความต้องการอะไรในกระบวนการทำงาน เพื่อจะทำการป้อนข้อมูลนำเข้าและดูผลลัพธ์ที่ได้ว่าเป็นอะไรบ้าง ถูกต้องตามความต้องการหรือไม่
- การทดสอบแบบกล่องขาว (White box testing) หรือในบางครั้งอาจถูกเรียกว่า การทดสอบแบบกล่องแก้ว (Glass box) เนื่องจากผู้ทำการทดสอบต้องมีความรู้ในส่วนหนึ่งของโปรแกรมหรือเข้าถึงโครงสร้างการทำงานในส่วนต่าง ๆ (ภาพที่ 4.8) ที่อยู่ภายใน เพื่อทำการเตรียมข้อมูลและสร้างกรณีทดสอบในทุก ๆ ส่วนการทำงานของโปรแกรมที่เป็นไปได้ ซึ่งในการเตรียมข้อมูลที่ใช้เป็นข้อมูลนำเข้าของการทดสอบนี้ไม่เพียงเป็นข้อมูลที่ถูกต้องตรงตามเงื่อนไขที่ระบุเท่านั้น แต่รวมถึงข้อมูลที่ไม่ถูกต้อง

หรือไม่เป็นไปตามเงื่อนไขที่ระบุด้วย เพื่อตรวจสอบการดำเนินการของระบบว่ามีการตอบสนองการทำงานกับข้อมูลในแต่ละกรณีเป็นอย่างไร

7. การทดสอบหน่วย (Unit testing) มีวิธีการทดสอบรูปแบบใดบ้าง

- 1) การทดสอบแบบกล่องดำ
- 2) การทดสอบแบบกล่องขาว
- 3) การทดสอบแบบกล่องเทา

8. จงอธิบายกระบวนการติดตาม แก้ไข และป้องกันข้อบกพร่อง

หลังจากที่ได้มีการตรวจวิเคราะห์และยืนยันข้อมูลของปัญหาแล้ว หัวหน้าทีมทดสอบหรือผู้ทดสอบจะทำหน้าที่บันทึกข้อมูลข้อบกพร่องต่าง ๆ ที่ตรวจพบ รวมถึงระดับความสำคัญและความรุนแรงของปัญหา จากนั้นจะเข้าสู่กระบวนการติดตามและแก้ไขข้อบกพร่อง โดยมีขั้นตอนการทำงานดังนี้

- สร้างใหม่ (New) จะถูกใช้เมื่อพบข้อบกพร่องนั้นในครั้งแรกหรือเป็นการแจ้งถึงการพบข้อบกพร่องใหม่ที่มีการนำเข้าสู่ระบบ
- เปิด (Open) จะมีการเปลี่ยนสถานะเป็น “Open” เมื่อผู้พัฒนาหรือผู้ที่มีส่วนเกี่ยวข้องในการแก้ไขเริ่มกระบวนการค้นหาวิธีการแก้ไขปัญหาข้อบกพร่องที่ได้ตรวจพบ จากนั้นจะเข้าสู่กระบวนการกำหนดส่วนที่มีหน้าที่เกี่ยวข้องในการดำเนินการกับข้อบกพร่องที่ตรวจสอบ “Assigned” ซึ่งอาจแบ่งได้เป็น 3 ทางเลือก คือ
 - 1) ไม่ถือเป็นข้อบกพร่อง (Not a Defect) ในกรณีที่ตรวจสอบแล้วพบว่าไม่ใช่ข้อบกพร่องของระบบ จะทำการเปลี่ยนสถานะเป็น “Not a Defect”
 - 2) แก้ไข (Fixed) ในกรณีที่พบว่าเป็นข้อบกพร่องของระบบและมีข้อมูลต่าง ๆ เพียงพอจะมีการดำเนินการแก้ไขปัญหาข้อบกพร่องในทันทีซึ่งหลังจากที่ได้ทำการแก้ไขข้อบกพร่องที่พบเป็นที่เรียบร้อยแล้วจะเข้าสู่กระบวนการ “Re-Tested” หรือในบางครั้งอาจเรียกว่าเป็นการ “Regression test” ซึ่งเป็นกระบวนการทดสอบซ้ำว่าข้อบกพร่องนั้นได้รับการแก้ไขไปแล้วหรือไม่ ซึ่งอาจพบปัญหาใหม่หรือปัญหาที่ยังไม่ได้รับการแก้ไขจะมีการทำซ้ำกลับเข้าสู่กระบวนการวิเคราะห์เพื่อแก้ไขปัญหาลงต่อไป แต่หากการแก้ไขปัญหาหรือข้อบกพร่องที่พบเสร็จสิ้นจะเข้าสู่สถานะ “Close” ซึ่งถือว่าเป็นการสิ้นสุดกระบวนการ
 - 3) ชะลอการแก้ไข (Postponed) จะถูกใช้เมื่อปัญหาหรือข้อบกพร่องที่ตรวจพบอาจต้องรอกำหนดช่วงเวลาในการแก้ไข ซึ่งอาจเนื่องมาจากหลายสาเหตุ เช่น ยังไม่มีข้อมูลที่ใช้ในการทดสอบ ไม่มีข้อมูลรายละเอียดเกี่ยวกับฟังก์ชันการทำงานที่เกิดข้อผิดพลาด ขาดทรัพยากร บุคลากร

9. จงอธิบายประโยชน์ของเครื่องมือสำหรับการวิเคราะห์แบบพลวัต พร้อมยกตัวอย่างประกอบ

ช่วยให้นักพัฒนาสามารถเข้าใจเส้นทางการทำงานของโปรแกรม และตรวจสอบข้อมูลในขณะโปรแกรมกำลังทำงาน ตัวอย่างของเครื่องมือประเภทนี้ คือ Debugger ใน IDE ที่ช่วยให้นักพัฒนาซอฟต์แวร์สามารถวิเคราะห์ค่าของตัวแปรในแต่ละขั้นตอนของการทำงานเพื่อหาสาเหตุของข้อบกพร่อง

10. ระบบคำนวณเงินภาษีออนไลน์เปิดทำงานตั้งแต่เวลา 06.00 น. ถึง 23.00 น. ของแต่ละวัน เพื่อทดสอบว่าระบบทำงานได้ตามช่วงเวลาที่กำหนดไว้อย่างถูกต้อง ให้ระบุช่วงเวลาเพื่อใช้แทนกลุ่มของข้อมูลที่จะต้องนำมาใช้เป็นข้อมูลสำหรับการทดสอบแบบสมมูล (Equivalence partitioning) และการวิเคราะห์ค่าข้อมูลจากขอบเขต (Boundary Value Analysis)

การทดสอบแบบสมมูล แบ่งข้อมูลที่ใช้ทดสอบได้เป็น 3 กลุ่มดังนี้

- กรณีระบบทำงานก่อนเวลา 06.00 น.
- กรณีระบบทำงานในช่วงระยะเวลา 06.00 – 23.00 น.
- กรณีระบบทำงานเกินกว่าเวลา 23.00 น.

การวิเคราะห์ข้อมูลจากขอบเขต แบ่งข้อมูลทดสอบได้เป็น 6 ค่าดังนี้

- กรณีระบบทำงานเวลา 05.59 น.
- กรณีระบบทำงานเวลา 06.00 น.
- กรณีระบบทำงานเวลา 06.01 น.
- กรณีระบบทำงานเวลา 22.59 น.
- กรณีระบบทำงานเวลา 23.00 น.
- กรณีระบบทำงานเวลา 23.01 น.

บทที่ 5

1. จงอธิบายกระบวนการทบทวนซอฟต์แวร์

การทบทวนซอฟต์แวร์มีรายละเอียดที่หลากหลายแตกต่างกันในแต่ละประเภทของระบบงาน ประกอบด้วย 3 กิจกรรมหลัก คือ กิจกรรมก่อนการทบทวน กิจกรรมการประชุมเพื่อทบทวน และกิจกรรมหลังการทบทวน

2. จงอธิบายความแตกต่างของคำว่า ความคลาดเคลื่อน (Fault) และ ความผิดพลาด (Error)

ความคลาดเคลื่อน (Fault) เป็นขั้นตอน กระบวนการ หรือนิยามข้อมูลในโปรแกรมที่ไม่ถูกต้อง ความผิดพลาด (Error) เป็นการกระทำของมนุษย์ที่ทำให้เกิดผลลัพธ์ที่ไม่ถูกต้อง

3. จงยกตัวอย่างการสร้างความทนทานให้กับซอฟต์แวร์

การใช้ RAID เพื่อทำการสำรองข้อมูลของเครื่องแม่ข่าย

4. จงบอกวิธีการป้องกันข้อบกพร่องของซอฟต์แวร์อย่างน้อย 1 วิธี

การทดสอบในระดับ Unit testing

5. ร่องรอยโค้ดที่ไม่ดีจะส่งผลเสียต่อระบบซอฟต์แวร์อย่างไร

มีโอกาที่จะทำให้เกิดข้อบกพร่องในซอฟต์แวร์

6. จงอธิบายความหมายของรีแฟคตอริง

การเปลี่ยนแปลงโครงสร้างภายในของซอฟต์แวร์ โดยไม่เปลี่ยนแปลงการทำงานของซอฟต์แวร์ มีวัตถุประสงค์เพื่อให้ซอร์สโค้ดง่ายต่อการทำความเข้าใจ ซึ่งจะเพิ่มความสามารถในการบำรุงรักษาซอฟต์แวร์

7. จงอธิบายการทำ Extract methods พร้อมยกตัวอย่างประกอบ

เมื่อในคลาสมีคำสั่งเพื่อบรรลุวัตถุประสงค์เดียวกัน ควรสร้างเมธอดขึ้นมาใหม่โดยนำคำสั่งที่มีวัตถุประสงค์เดียวกันมาอยู่ด้วยกัน

8. จงอธิบายการทำ Pull up filed พร้อมยกตัวอย่างประกอบ

เมื่อมีคลาสมากกว่าหนึ่งคลาสซึ่งมีแอตทริบิวต์ที่เหมือนกัน ควรย้ายแอตทริบิวต์ดังกล่าวไปไว้ในคลาสแม่

9. จงยกตัวอย่างปัญหาด้านบุคลากรที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์

นักพัฒนาซอฟต์แวร์บางคนไม่ต้องการให้มีวิจารณ์งานที่ทำ เนื่องจากเข้าใจเจตนาที่ผิดคิดว่าจ้องจับผิด เป็นต้น

10. การเปลี่ยนแปลงซอฟต์แวร์มีผลอย่างไรกับเรื่องคุณภาพซอฟต์แวร์

อาจทำให้คุณภาพซอฟต์แวร์ลดต่ำลง เช่น เมื่อมีการเปลี่ยนแปลงความต้องการ จะทำให้นักพัฒนาอาจไม่ได้ทำการทดสอบอย่างเพียงพอกับสิ่งที่เปลี่ยนแปลงไป

บทที่ 6

1. จงอธิบายความแตกต่างของการฝึกหัด (Training) แต่ละประเภท

การฝึกอบรมสามารถแบ่งเป็นผู้ใช้ออกเป็น 2 กลุ่ม คือ

- 1) ผู้ใช้ (User) ที่มีหน้าที่ปฏิบัติงานที่กระทำซ้ำ ๆ กันเป็นงานประจำ
- 2) พนักงานที่ทำงานเกี่ยวกับการปฏิบัติการ เป็นบุคลากรที่สนับสนุนหรือแก้ปัญหาในเชิงเทคนิค

2. เอกสารคู่มือผู้ใช้ที่ดีควรมีส่วนประกอบหรือองค์ประกอบเป็นอย่างไร

- 1) การอธิบายวัตถุประสงค์ของระบบ
- 2) คุณลักษณะ จุดเด่น อุปกรณ์ต่าง ๆ ที่สามารถใช้ในระบบ
- 3) ประโยชน์ของระบบ
- 4) ความสามารถและการทำงานของระบบทั้งหมดที่ปรากฏขึ้นในการปฏิบัติงาน

3. จงอธิบายแผนการติดตั้งระบบซอฟต์แวร์

ก่อนที่จะทำการติดตั้งใช้งานระบบซอฟต์แวร์จะต้องวางแผนการปรับเปลี่ยนหรือโอนย้ายระบบ ในแผนการปรับเปลี่ยนระบบ ประกอบด้วย แผนการติดตั้งฮาร์ดแวร์ แผนการติดตั้งซอฟต์แวร์ และแผนการแปลงข้อมูล นอกจากนี้ยังต้องวางแผนการจัดการการเปลี่ยนแปลงด้วย

4. กิจกรรมสำคัญที่เกิดขึ้นภายหลังจากการติดตั้งระบบแล้ว มีอะไรบ้าง

- 1) การสนับสนุนระบบ เป็นการเตรียมทีมงานเพื่อให้ความช่วยเหลือแก่ผู้ใช้งานระบบให้สามารถใช้งานระบบได้อย่างสมบูรณ์
- 2) การบำรุงรักษาระบบ เป็นการกระทำอย่างต่อเนื่องเพื่อซ่อมแซมข้อบกพร่องให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น
- 3) การประเมินระบบ เป็นกระบวนการวิเคราะห์เพื่อให้เข้าใจว่า กิจกรรมต่าง ๆ ถูกดำเนินการอย่างดี และกิจกรรมไหนจำเป็นต้องปรับปรุงให้ดีขึ้นในอนาคต

5. จงยกตัวอย่างการแสดงข้อความที่สามารถช่วยเหลือผู้ใช้งานระบบทั้งที่ดีและไม่ดี

ตัวอย่างที่ดี ได้แก่ “ไม่สามารถบันทึกไฟล์ได้ เนื่องจากไม่มีสิทธิ”

ตัวอย่างที่ไม่ดี ได้แก่ “Error”

6. จงอธิบายหน้าที่ของผู้ใช้งานระบบและผู้ปฏิบัติ

- 1) หน้าที่ของผู้ใช้
 - การจัดการแฟ้มข้อมูล
 - การจำลองกิจกรรม
 - การวิเคราะห์ข้อมูล

- การติดต่อสื่อสารข้อมูล
 - การวาดกราฟและแผนผัง
- 2) หน้าที่ของผู้ปฏิบัติ
- การให้สิทธิการเข้าถึงแก่ผู้ใช้
 - การให้สิทธิการเข้าถึงแฟ้มข้อมูล
 - การสำรองข้อมูล
 - การติดตั้งอุปกรณ์ใหม่
 - การติดตั้งซอฟต์แวร์ใหม่
 - การค้นคืนแฟ้มข้อมูลที่เสียหาย

7. เฮลป์เดสก์ (Help desk) คืออะไร

เป็นหน่วยให้ความช่วยเหลือแก่ผู้ใช้ระบบขององค์กร ทำหน้าที่ตอบคำถามรับฟังปัญหาต่าง ๆ ที่เกิดขึ้นจากการดำเนินงานของระบบ

8. จงอธิบายวิธีการประเมินระบบหลังจากการติดตั้งใช้งาน

- 1) การศึกษาผลกระทบแบบเป็นทางการ
- 2) การตรวจสอบอย่างสม่ำเสมอ
- 3) การตรวจสอบสมรรถนะ

9. จงอธิบายกระดานข่าว (Bulleting board) มาพอสังเขป

เป็นวิธีการสนับสนุนแหล่งความรู้หรือข่าวสารที่เป็นประโยชน์แก่ผู้ใช้งาน โดยองค์กรสามารถฝากข้อความหรือข่าวสารไว้ที่กระดานข่าวได้

10. ฟอรัมสนับสนุนออนไลน์คืออะไร จงอธิบาย

เป็นส่วนที่องค์กรตั้งกระทู้ไว้เพื่อให้ผู้ใช้เปิดประเด็นคำถามโดยผ่านเครือข่าย องค์กรสามารถตอบคำถาม แนะนำวิธีการหรือเทคนิคในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้น

บทที่ 7

1. การบำรุงรักษาซอฟต์แวร์คืออะไร

กระบวนการบริหารจัดการความเปลี่ยนแปลงต่าง ๆ ที่เกิดขึ้นกับซอฟต์แวร์ที่ถูกพัฒนาหรือจัดหาเข้าใช้งานในองค์กร

2. เหตุใดจึงต้องมีการบำรุงรักษาซอฟต์แวร์หลังจากพัฒนาหรือติดตั้งซอฟต์แวร์เสร็จเรียบร้อยแล้ว

หลังจากพัฒนาหรือติดตั้งซอฟต์แวร์เสร็จเรียบร้อยแล้ว การบำรุงรักษาซอฟต์แวร์มีความจำเป็นอย่างยิ่งในการแก้ไขข้อบกพร่องต่าง ๆ ตลอดจนการปรับปรุงคุณภาพ และเพิ่มเติมความสามารถของซอฟต์แวร์ให้สอดคล้องกับความต้องการที่อาจมีการเปลี่ยนแปลงไปตามสภาพการใช้งาน

3. ปัญหาที่เกิดขึ้นในการบำรุงรักษาซอฟต์แวร์มีอะไรบ้าง

ข้อจำกัดของความเข้าใจ ปัญหาทางด้านเทคนิค ปัญหาในการทดสอบระบบ ปัญหาในการพัฒนาแล้วผลผลิตตกต่ำ ความยากง่ายของความผิดพลาดที่เกิดขึ้น

4. ทีมงานบำรุงรักษาซอฟต์แวร์ประกอบด้วยใครบ้าง ทำหน้าที่อย่างไร จงอธิบาย

- 1) นักพัฒนาซอฟต์แวร์
- 2) นักวิเคราะห์ระบบ

5. ปัจจัยใดบ้างที่ส่งผลต่อค่าใช้จ่ายในการบำรุงรักษาซอฟต์แวร์

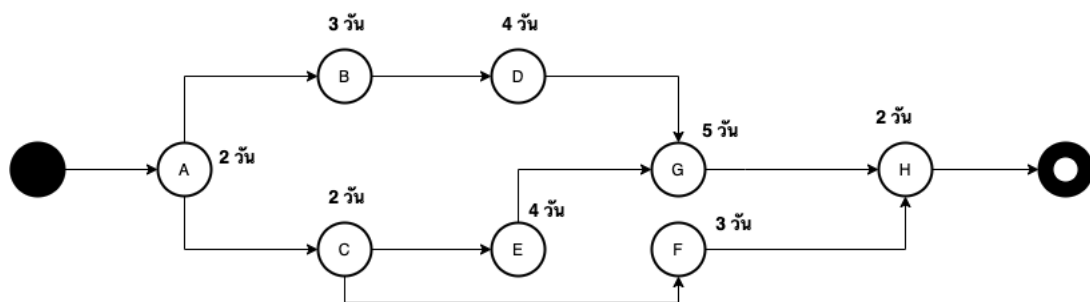
- 1) ชนิดของซอฟต์แวร์
- 2) ความมั่นคงของทีมงาน
- 3) ช่วงชีวิตของระบบซอฟต์แวร์
- 4) คุณลักษณะของฮาร์ดแวร์ที่ใช้ในระบบ
- 5) คุณภาพของการออกแบบ
- 6) คุณภาพในการเขียนโปรแกรม
- 7) คุณภาพเอกสาร
- 8) คุณภาพการทดสอบระบบ

6. จงอธิบายความหมายของค่าดัชนีการบริหารการบำรุงรักษา (Backlog management index : BMI)

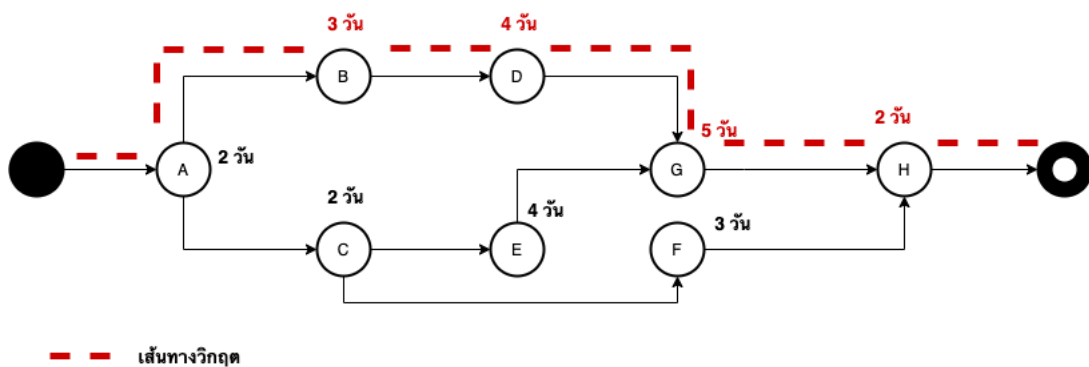
ค่าร้อยละของปริมาณงานบำรุงรักษาระบบที่ทำเสร็จภายในแต่ละเดือน เปรียบเทียบจากจำนวนงานบำรุงรักษาระบบที่เข้ามาใหม่ทั้งหมดในแต่ละเดือน โดยจำแนกตามประเภทของการบำรุงรักษาระบบ

7. จากข้อมูลในตาราง ให้สร้างแผนภาพ PERT

กิจกรรม	งานก่อนหน้า	จำนวนวัน
A	-	2
B	A	3
C	A	2
D	B	4
E	C	4
F	C	3
G	D, E	5
H	F, G	2



8. จากตารางในข้อ 7 ให้ระบุเส้นทางวิกฤต



9. จงอธิบายการประเมินค่าใช้จ่ายในการบำรุงรักษาระบบซอฟต์แวร์ โดยใช้แบบจำลองโคโคโม (COCOMO) พร้อมยกตัวอย่างประกอบ

การพยากรณ์ค่าใช้จ่ายในการพัฒนาซอฟต์แวร์ ซึ่งเป็นวิธีที่นิยมใช้อย่างแพร่หลายมาใช้ในการประเมินเวลาและค่าใช้จ่ายสำหรับการบำรุงรักษาระบบได้ โดยมีสูตรการคำนวณดังสมการที่ (1)

$$MA = EAF * F * Dev \quad (1)$$

โดยที่

- ค่า MA (Maintenance adjustment) คือจำนวนเวลาที่ใช้ในการบำรุงรักษาระบบซอฟต์แวร์ต่อปี มีหน่วยเป็นเดือน (Man-month)
- ค่า EAF (Effort adjustment factor) คือ ผลคูณค่าน้ำหนักปัจจัยความซับซ้อนในการบำรุงรักษาระบบทั้งหมด 15 ปัจจัย ซึ่งองค์กรต้องพิจารณาและเลือกค่าน้ำหนักที่เหมาะสมสำหรับแต่ละปัจจัย
- ค่า F (Frequency) คือ ความถี่ของการขอเปลี่ยนแปลงแก้ไขต่อปี
- ค่า Dev (Development) คือ จำนวนเวลาที่ใช้ในการปรับปรุงแก้ไขและพัฒนาระบบซอฟต์แวร์ มีหน่วยเป็นเดือน (Man-month)

การคำนวณค่า Dev สามารถคำนวณได้ดังสมการที่ (2)

$$\text{Dev} = a (\text{KLOC})^b \quad (2)$$

โดยที่

- ค่า KLOC (Thousands of line of code) คือ จำนวนบรรทัดโปรแกรมโค้ดในหน่วยหลักพัน เช่น ระบบนี้ถูกเขียนด้วยภาษาจาวาจำนวน 3,500 บรรทัด ดังนั้น KLOC จะมีค่าเท่ากับ 3.5
- ค่าคงที่ a และ b จะพิจารณาจากประเภทของระบบที่ทำการบำรุงรักษา โดยแนวคิดของโคโดโมจำแนกประเภทของระบบออกเป็น 3 ประเภท ซึ่งพิจารณาจากลักษณะเฉพาะของระบบ

10. ระบบรายงานผลการเรียนของนักศึกษามีจำนวนคำสั่งโปรแกรมขนาด 60,000 บรรทัด โดยทีมงานมีความเชี่ยวชาญและมีประสบการณ์ค่อนข้างสูงในการพัฒนาระบบดังกล่าว เนื่องจากมีรายงานปัญหาการใช้งานจากนักศึกษาทำให้ต้องมีการปรับปรุงแก้ไขระบบนี้โดยเร็วที่สุด จงประเมินค่าบำรุงรักษาซอฟต์แวร์ดังกล่าวต่อไปนี้

$$\text{MA} = \text{EAF} * F * \text{DEV}$$

EAF คือการหาผลคูณค่าน้ำหนักรวมของทั้ง 15 ปัจจัย ตามตารางที่ 7-1 จากโจทย์ที่ให้เกี่ยวกับประสบการณ์ของบุคลากรและข้อจำกัดด้านเวลาสามารถประเมินและกำหนดค่าน้ำหนักให้ปัจจัยด้านคุณลักษณะของบุคลากร (ปัจจัยที่ 8-12) ให้อยู่ในระดับ “สูง” และปัจจัยด้านเวลา (ปัจจัยที่ 15) อยู่ในระดับ “น้อยมาก” ส่วนปัจจัยอื่น ๆ อยู่ในระดับ “ปานกลาง” ซึ่งจะทำให้เราได้ค่าน้ำหนักปัจจัยทั้ง 15 ปัจจัย ดังนี้

$$\begin{aligned} \text{EAF} &= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.15 * 1.00 * 0.71 * 0.82 * 0.70 * 0.90 * 0.95 * \\ &1.00 * 1.00 * 1.23 = 0.49 \end{aligned}$$

สมมติว่า ระบบนี้มีการร้องขอเพื่อแก้ไขปรับปรุงโดยเฉลี่ย 1 ครั้งต่อปี จะได้ว่า $F=1$

$$\text{KLOC} = 60$$

$$a = 3.0$$

$$b = 1.12$$

$$\text{จาก } Dev = a(KLOC)^b$$

$$Dev = 3.0 * (60)^{1.12} = 294.2 \text{ เดือน}$$

ดังนั้นจากสมการ จะได้ว่า

$$MA = 0.49 * 1 * 294.2 = 144.15 \text{ เดือน (man-month)}$$

จากผลที่ได้จะเห็นได้ว่า จำนวนเดือนมีค่าสูงมาก ดังนั้น จึงต้องพิจารณาว่าในทีมมีจำนวนสมาชิกมากกว่าหนึ่งคน เช่นมีจำนวน 24 คน จะเหลือจำนวน $MA = 144/24 = 6$ เดือน

ถ้าค่าจ้างทีมพัฒนาต่อคนต่อเดือนคือ 17,000 บาท จะได้ว่า ค่าบำรุงรักษาซอฟต์แวร์ต่อปี เป็นดังนี้

$$17000 * 24 * 6 = 2,448,000 \text{ บาท}$$

บทที่ 8

1. จงบอกข้อดีและข้อจำกัดของแบบจำลองแก้ไขเร่งด่วน (Quick-fix model)

ข้อดี คือ สามารถแก้ปัญหาได้อย่างรวดเร็ว และสามารถดำเนินการได้โดยเพียงคนเดียว
ข้อเสีย คือ ปัญหาที่ได้รับการแก้ไขอาจจะเกิดขึ้นซ้ำได้อีก เนื่องจากไม่ได้รับการวิเคราะห์อย่างดีพอ

2. จงบอกแนวทางในการทำเอกสารที่ช่วยให้การบำรุงรักษาซอฟต์แวร์สามารถกระทำได้อย่างมีประสิทธิภาพ

ควรปรับปรุงเอกสารต่าง ๆ เมื่อซอฟต์แวร์ได้รับการแก้ไข หรือเปลี่ยนแปลง

3. จงยกตัวอย่างเครื่องมือช่วยบำรุงรักษาซอฟต์แวร์แบบอัตโนมัติ อย่างน้อย 2 รายการ

- 1) Debugger
- 2) Refactoring tool

4. ในการรื้อปรับซอฟต์แวร์ ควรพิจารณาถึงเรื่องใดบ้าง

- ค่าใช้จ่ายรายปีสำหรับการบำรุงรักษาซอฟต์แวร์
- ค่าใช้จ่ายรายปีสำหรับการใช้งานซอฟต์แวร์
- รายรับที่ได้จากการใช้งานซอฟต์แวร์
- ประเมินการค่าใช้จ่ายรายปีสำหรับการบำรุงรักษาซอฟต์แวร์ ภายหลังจากดำเนินการรื้อปรับระบบ
- ประเมินการค่าใช้จ่ายรายปีสำหรับการใช้งานซอฟต์แวร์ ภายหลังการดำเนินการรื้อปรับระบบ
- ประเมินการรายรับที่ได้ ภายหลังดำเนินการรื้อปรับระบบจากการใช้ซอฟต์แวร์
- ประเมินการค่าใช้จ่ายในการดำเนินการรื้อปรับซอฟต์แวร์
- ประเมินการระยะเวลาที่ใช้ในการรื้อปรับซอฟต์แวร์
- ประเมินการความเสี่ยงของผลสำเร็จในการรื้อปรับซอฟต์แวร์
- ประเมินการระยะเวลาการใช้ระบบซอฟต์แวร์

5. จงอธิบายความแตกต่างระหว่างแบบจำลองกระบวนการบำรุงรักษาซอฟต์แวร์ ISO/IEC 14764 และแบบจำลอง IEEE

แบบจำลอง ISO/IEC 14764 จะประกอบด้วยกิจกรรมเพียงไม่กี่กิจกรรม เช่น การวิเคราะห์ปัญหาและการปรับปรุงซอฟต์แวร์ การบำรุงรักษาซอฟต์แวร์ และการประเมินซอฟต์แวร์ที่ถูกปรับปรุงแก้ไข แต่ในส่วน ของ

IEEE จะมีกิจกรรมที่มีลักษณะคล้ายกับกิจกรรมในการพัฒนาซอฟต์แวร์ ได้แก่ การวิเคราะห์ การออกแบบ การแก้ไข การทดสอบ การยอมรับการทดสอบ การส่งมอบ

6. เหตุใดจึงต้องดำเนินการวิเคราะห์ความคุ้มค่าของการรี้อปรับซอฟต์แวร์

ความคุ้มค่าของการรี้อปรับซอฟต์แวร์เป็นตัวชี้วัดที่สามารถนำมาใช้ในการพิจารณาความคุ้มค่าของการรี้อปรับระบบซอฟต์แวร์

7. ข้อมูลต่อไปนี้เป็นโมดูลในระบบการจ่ายเงินเดือนขององค์กรหนึ่ง จงลำดับความเร่งด่วนของการรี้อปรับระบบ

โมดูล	ความคุ้มค่าของการรี้อปรับระบบ
ระบบการเชื่อมต่อกับธนาคาร	150,000
ระบบคำนวณเงินเดือนและสวัสดิการ	200,000
ระบบจัดทำรายงานสำหรับผู้บริหาร	100,000

การรี้อปรับระบบการจ่ายเงินเดือนขององค์กรหนึ่ง สามารถลำดับความเร่งด่วนของการรี้อปรับระบบได้ดังนี้ ระบบคำนวณเงินเดือนและสวัสดิการ ระบบการเชื่อมต่อกับธนาคาร ระบบจัดทำรายงานสำหรับผู้บริหาร

8. อะไรคือเทคนิคที่ใช้ในการบำรุงรักษาซอฟต์แวร์

การบำรุงรักษาซอฟต์แวร์ให้สามารถจัดการโครงสร้างกระบวนการ และจัดการความเปลี่ยนแปลงได้อย่างสะดวก สามารถดำเนินการได้โดย

- การจัดการโครงแบบของซอฟต์แวร์
- การใช้งานเครื่องมือช่วยบำรุงรักษาซอฟต์แวร์แบบอัตโนมัติ

บทที่ 9

1. จงอธิบายสาเหตุที่ต้องมีการบริหารการเปลี่ยนแปลงซอฟต์แวร์

สาเหตุที่ต้องมีการบริหารการเปลี่ยนแปลงซอฟต์แวร์ เพื่อที่จะป้องกันความสับสนและการสูญเสียข้อมูลที่จัดเก็บเกี่ยวกับสิ่งที่เปลี่ยนแปลงในซอฟต์แวร์

2. จงระบุความเสียหายที่อาจเกิดจากการบริหารการเปลี่ยนแปลงที่ไม่ดี

ความเสียหายที่อาจเกิดจากการบริหารการเปลี่ยนแปลงที่ไม่ดีนั้น รวมถึงการแก้ไขซอฟต์แวร์เวอร์ชันที่ผิด การจัดส่งเวอร์ชันที่ไม่ถูกต้องให้ลูกค้า ตลอดจนอาจไม่สามารถระบุได้ว่าจัดเก็บซอร์สโค้ดสำหรับเวอร์ชันของซอฟต์แวร์หรือคอมพิวเตอร์ที่ต้องการไว้ที่ไหน

3. จงอธิบายกิจกรรมหลักของการบริหารการเปลี่ยนแปลง

กิจกรรมหลักของการบริหารการเปลี่ยนแปลง ประกอบด้วย กิจกรรมการควบคุมเวอร์ชัน ซึ่งเป็นกิจกรรมที่เกี่ยวข้องกับการติดตามการเปลี่ยนแปลงของเวอร์ชันต่าง ๆ ขององค์ประกอบของซอฟต์แวร์ กิจกรรมการควบคุมการเปลี่ยนแปลง ซึ่งเป็นกิจกรรมที่เกี่ยวข้องกับการติดตามการร้องขอให้แก้ไขหรือเปลี่ยนแปลงซอฟต์แวร์ การประมาณการค่าใช้จ่ายและผลกระทบจากการเปลี่ยนแปลงที่จะเกิดขึ้น และการตัดสินใจว่าจะทำการเปลี่ยนแปลงหรือไม่ และจะดำเนินการเปลี่ยนแปลงในช่วงเวลาไหนกิจกรรมการสร้างระบบ (System building) ซึ่งเป็นกระบวนการนำองค์ประกอบต่าง ๆ มาประกอบเข้าด้วยกันเพื่อสร้างระบบที่พร้อมใช้งาน และกิจกรรมการส่งมอบหรือการจัดการวางจำหน่าย (Release management) ซึ่งเป็นกิจกรรมที่เกี่ยวข้องกับการเตรียมซอฟต์แวร์เพื่อที่จะส่งมอบหรือจัดจำหน่าย

4. กระบวนการควบคุมการเปลี่ยนแปลงเริ่มต้นและสิ้นสุดเมื่อใด

กระบวนการควบคุมการเปลี่ยนแปลง เริ่มต้นเมื่อลูกค้ารวมถึงผู้เกี่ยวข้องและผู้มีส่วนได้ส่วนเสียที่ไม่ใช่ นักพัฒนาระบบ กรอกแบบฟอร์มขอให้มีการเปลี่ยนแปลงของระบบและส่งข้อมูลที่กรอกเรียบร้อยแล้วมาให้ กระบวนการควบคุมนี้จะสิ้นสุดเมื่อคำขอให้เปลี่ยนแปลงได้รับการปิด ซึ่งการปิดนั้นมี 2 แบบ คือ การปิดโดยไม่ดำเนินการใด ๆ และการปิดที่เกิดจากการที่ได้ดำเนินการเปลี่ยนแปลงเรียบร้อยแล้ว

5. จงระบุตัวอย่างการร้องขอที่ไม่ต้องมีการดำเนินการแก้ไข

ตัวอย่างการร้องขอที่ไม่ต้องมีการดำเนินการใด ๆ เช่น การร้องขอให้มีการแก้ไขข้อผิดพลาดที่ลูกค้าค้นพบ ซึ่งเมื่อตรวจสอบแล้วอาจพบว่าข้อผิดพลาดนี้ได้มีการดำเนินการไปแล้ว รายงานข้อผิดพลาดที่ไม่ใช่ข้อผิดพลาดแต่เป็นความเข้าใจผิดของลูกค้าเกี่ยวกับความคาดหวังว่าระบบจะตอบสนองอย่างไรต่อเหตุการณ์บางอย่างที่จะเกิดขึ้น และการร้องขอให้มีการพัฒนาฟังก์ชันเพิ่มเติม ซึ่งฟังก์ชันที่ร้องขอได้มีการพัฒนาไว้ในระบบอยู่แล้วแต่ลูกค้าไม่ทราบ เป็นต้น

6. กลุ่มบุคคลที่เกี่ยวข้องกับการจัดการการเปลี่ยนแปลงมีอะไรบ้าง จงอธิบาย

กลุ่มบุคคลที่เกี่ยวข้องกับการจัดการการเปลี่ยนแปลงมี 3 กลุ่ม ดังนี้

- 1) กลุ่มผู้สนับสนุน เป็นกลุ่มบุคคลที่ต้องการให้มีการเปลี่ยนแปลง เป็นผู้เริ่มต้นสร้างคำขอให้มีระบบใหม่ ซึ่งโดยทั่วไปคือผู้บริหารอาวุโสของหน่วยงานในองค์กรที่ต้องการให้มีการปรับปรุงและใช้งานระบบใหม่
- 2) กลุ่มผู้กระทำการเปลี่ยนแปลง เป็นกลุ่มบุคคลที่ต้องใช้ความพยายามเป็นอย่างมากเพื่อกระทำการเปลี่ยนแปลงให้บรรลุความสำเร็จและเกิดประโยชน์
- 3) กลุ่มผู้ที่นำระบบที่เปลี่ยนแปลงแล้วไปใช้ เป็นกลุ่มเป้าหมายของการจัดการเปลี่ยนแปลง บุคคลกลุ่มนี้เป็นผู้ที่ทำให้ระบบต้องได้รับการออกแบบใหม่ และเป็นผู้ที่จะเลือกใช้หรือไม่ใช้ระบบที่ได้รับการเปลี่ยนแปลง

7. ในแบบฟอร์มคำร้องขอให้มีการเปลี่ยนแปลง ต้องแสดงข้อมูลใดบ้าง

ในแบบฟอร์มคำร้องขอให้มีการเปลี่ยนแปลง ต้องระบุว่าต้องการให้เปลี่ยนแปลงอะไร ระบุสถานการณ์ปัจจุบัน ระบุเหตุผลที่ต้องการให้เปลี่ยนแปลง ระบุความคาดหวังระบบที่แก้ไขแล้วจะเป็นอย่างไร ระบุวันที่ต้องการให้แล้วเสร็จ ลงนามโดยผู้ยื่นคำร้อง และมีผู้มีอำนาจหรือหัวหน้างานลงนามรับรองเมื่อคำร้องขอได้ผ่านขั้นตอนการวิเคราะห์ผลกระทบแล้ว ฝ่ายที่ดูแลการปรับแก้ไขซอฟต์แวร์จะเติมข้อมูลผลการวิเคราะห์ผลกระทบ ระยะเวลาที่ใช้ในการปรับแก้ไขซอฟต์แวร์ และค่าใช้จ่ายโดยประมาณ ระบุสถานะของใบคำร้อง จัดลำดับความสำคัญเร่งด่วน ระบุวันที่คาดว่าจะดำเนินการแล้วเสร็จ และลงนามรับรองโดยผู้ประมาณการ หัวหน้าโครงการ และผู้ประเมินคุณภาพ

8. การอนุมัติให้มีการเปลี่ยนแปลง พิจารณาจากความเหมาะสมในประเด็นใดบ้าง

การอนุมัติให้มีการเปลี่ยนแปลง จะพิจารณาจากความเหมาะสมในประเด็นดังนี้

- 1) อะไรจะเกิดขึ้นถ้าไม่ดำเนินการเปลี่ยนแปลงตามคำร้องขอ โดยพิจารณาตามระดับของความรุนแรงที่จะเกิดขึ้น
- 2) ประโยชน์ที่ได้รับจากการเปลี่ยนแปลง ต้องพิจารณาว่าการเปลี่ยนแปลงที่เกิดขึ้นก่อให้เกิดประโยชน์กับผู้ใช้โดยรวม หรือมีประโยชน์เฉพาะผู้ร้องขอให้เปลี่ยนแปลงเท่านั้น
- 3) จำนวนผู้ใช้ที่ได้รับผลกระทบจากการเปลี่ยนแปลง ถ้ามีผู้ใช้ที่ได้รับผลกระทบจำนวนน้อย สามารถกำหนดความสำคัญของการเปลี่ยนแปลงให้อยู่ในระดับต่ำได้
- 4) ค่าใช้จ่ายที่ใช้ในการเปลี่ยนแปลง ถ้าต้องปรับแก้ไขหลายส่วนหรือต้องใช้เวลาในการแก้ไขเปลี่ยนแปลงนาน จะทำให้ค่าใช้จ่ายเพิ่มขึ้น

บทที่ 10

1. โครงแบบของซอฟต์แวร์หมายถึงอะไร จงอธิบาย

ข้อกำหนดเฉพาะของฟังก์ชันงานและคุณลักษณะทางกายภาพของซอฟต์แวร์ ฮาร์ดแวร์ เฟิร์มแวร์ รวมถึงอุปกรณ์อื่น ๆ ของระบบในแต่ละรุ่น โดยข้อกำหนดเหล่านี้จะถูกระบบไว้ในเอกสารทางด้านเทคนิคตามวัตถุประสงค์ของระบบ

2. กำหนดให้คลาสชื่อ Authenticator ของระบบซอฟต์แวร์หนึ่ง มีการปรับปรุงพัฒนาอย่างต่อเนื่องถึงปัจจุบัน มี 4 เวอร์ชัน ได้แก่ เวอร์ชัน 1.0, 2.0, 3.0 และ 4.0 หากการอ้างถึงคลาสเขียนเป็น Authenticator แสดงว่า เวอร์ชันที่ใช้งานเป็นเวอร์ชันอะไร

4.0

3. จงอธิบายการแก้ปัญหาการจัดการเนื้อที่จัดเก็บของระบบควบคุมเวอร์ชัน เมื่อมีเวอร์ชัน ต่าง ๆ ในระบบ หลายเวอร์ชัน

โดยทั่วไปแล้วจะจัดเก็บรายการที่เปลี่ยนแปลง

4. เครื่องมือสำหรับการจัดการเวอร์ชัน แบ่งได้เป็นกี่ประเภท และแต่ละประเภทมีความแตกต่างกันอย่างไร

- 1) เครื่องมือสำหรับการจัดการเวอร์ชันแบบโลคอล (Local) จากอดีตนักพัฒนาซอฟต์แวร์นั้นต้องทำการสำเนาโค้ดไว้ในแฟ้มข้อมูลสำรองก่อนจะทำการแก้ไขไฟล์
- 2) เครื่องมือสำหรับการจัดการเวอร์ชันแบบรวมศูนย์ (Centralized) เนื่องจากการจัดการเวอร์ชันแบบ Local นั้นเหมาะกับการทำงานคนเดียว ไม่เหมาะกับการทำงานที่เป็นทีมโดยเฉพาะในการใช้ไฟล์ร่วมกัน
- 3) เครื่องมือสำหรับการจัดการเวอร์ชันแบบกระจายศูนย์ (Distributed) จากข้อจำกัดของการสำรองข้อมูลในเซิร์ฟเวอร์กลางกรณีที่ไม่สามารถเชื่อมต่อกับเซิร์ฟเวอร์ได้

5. จงอธิบายประโยชน์ของการแตกกิ่ง (Branch) สำหรับการจัดการเวอร์ชัน

การแตกเส้นการพัฒนาแยกออกจากเส้นหลัก ซึ่งเส้นการพัฒนานี้จะถูกรเรียกว่า กิ่ง (Branch) โดยพัฒนาไปกับเส้นการพัฒนาหลัก และเมื่อพัฒนาไปได้ในระดับหนึ่งแล้วอาจมีการผสาน (Merge) เข้ากับลำต้น ซึ่งก็คือการนำส่วนที่ถูกพัฒนาแยกไว้ในกิ่ง ใส่กลับเข้าไปในลำต้น

6. จงอธิบายความแตกต่างระหว่าง Working directory และ Remote repository

Working directory คือ ไดเรกทอรีที่อยู่บนเครื่องของนักพัฒนาที่จัดการไฟล์

Remote repository คือ ไดเรกทอรีที่อยู่บนเครื่องแม่ข่าย

7. ระบุคำสั่งใน Git สำหรับการสร้าง Repository ใหม่

git init

8. จงอธิบายความแตกต่างระหว่างคำสั่ง git status และ git log

git status – ตรวจสอบสถานะไฟล์

git log - ดูประวัติของไฟล์

9. จงอธิบายความแตกต่างระหว่างคำสั่ง git commit และ git push

git commit - ยืนยันว่าต้องการเก็บไฟล์ที่อยู่ใน Staging area ลงไปไว้ใน Git repository

git push - การเก็บไฟล์ไว้ใน remote repository

10. จงอธิบายสถานะทั้งหมดของไฟล์ในการใช้ Git

- Untrack - เป็นข้อมูลที่ถูกสร้างขึ้นใหม่ ยังไม่ได้ส่งเข้าไปในระบบไฟล์ของ Git หรือไฟล์ที่ไม่ต้องการให้ Git นำไปเก็บสำรองไว้
- Modified - ได้ทำการแก้ไขไฟล์แล้ว แต่ยังไม่เริ่มกระบวนการจัดเก็บลง Repository
- Staged - ได้ทำเครื่องหมายไฟล์ ที่ถูกแก้ไขเพื่อจะบันทึกในเวอร์ชันหน้า
- Committed - ข้อมูลถูกบันทึกอย่างปลอดภัยใน Repository บน Local Host ซึ่งจะแสดงการเปลี่ยนแปลงสถานะของไฟล์ในส่วนต่าง ๆ

บทที่ 11

1. จงอธิบายความแตกต่างระหว่างวิศวกรรมย้อนกลับและวิศวกรรมก้าวหน้า

วิศวกรรมก้าวหน้า หมายถึงกระบวนการพัฒนาซอฟต์แวร์แบบดั้งเดิมจากความต้องการจนถึงการดำเนินการในรายละเอียด ในขณะที่วิศวกรรมย้อนกลับจะตรงข้ามกับวิศวกรรมก้าวหน้าคือ จะทำจากสิ่งที่พัฒนาไว้เสร็จแล้ว

2. จงระบุประโยชน์ของวิศวกรรมย้อนกลับในการพัฒนาซอฟต์แวร์

- 1) การปรับเอกสารประกอบโปรแกรม
- 2) การค้นหาสถาปัตยกรรม
- 3) การลบร่องรอยโค้ดที่ไม่ดี (Code smell)

3. จงอธิบายความแตกต่างระหว่างกระบวนการวิเคราะห์วากยสัมพันธ์และวิเคราะห์ความหมาย

กระบวนการวิเคราะห์เชิงวากยสัมพันธ์ (Syntax analysis) ทำหน้าที่นำรายการโทเคน (Stream of tokens) ที่เป็นผลผลิตของขั้นตอนการวิเคราะห์คำศัพท์ไปตรวจสอบว่ารายการโทเคนถูกต้องตามหลักไวยากรณ์ของภาษาโปรแกรมหรือไม่

4. จงอธิบายความแตกต่างระหว่าง Fan-in และ Fan-out

Fan-in คือ จำนวนของฟังก์ชันอื่นที่เรียกฟังก์ชันที่กำหนดในโมดูลหนึ่ง

Fan-out คือ จำนวนของฟังก์ชันอื่นที่ถูกเรียกจากฟังก์ชันที่กำหนดในโมดูลหนึ่ง

5. จงยกตัวอย่างเครื่องมือที่เป็นดีคอมไพเลอร์ของภาษาจาวา

JD-Eclipse (<http://java-decompiler.github.io/>)

6. จงยกตัวอย่างประโยชน์ของเบรกพอยต์ในการพัฒนาซอฟต์แวร์

ทำให้ผู้ใช้งานเลือกฟังก์ชันหรือบรรทัดของโค้ดที่ใดก็ได้ในโปรแกรม และระบุให้ดีบั๊กเกอร์หยุดการทำงานของโปรแกรมไว้ชั่วคราวเมื่อถึงบรรทัดดังกล่าว

7. จงอธิบายวัตถุประสงค์ของกระบวนการวิเคราะห์คำศัพท์

คือ ช่วยให้อีกิจกรรมต่าง ๆ ในกระบวนการวิเคราะห์วากยสัมพันธ์ (Syntax analysis) และวิเคราะห์ความหมาย (Semantic analysis) ทำงานได้ง่ายขึ้น