

# บทที่ 1

## บทนำ



### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจัยสำคัญอย่างหนึ่งที่น่าสนใจในวงการอุตสาหกรรมพัฒนาซอฟต์แวร์ คือ เทคนิคการนำกลับมาใช้ใหม่ (Reuse techniques) เพื่อช่วยลดค่าใช้จ่ายในการพัฒนาซอฟต์แวร์ เพิ่มคุณภาพ และช่วยให้การปรับปรุงบำรุงรักษาซอฟต์แวร์ทำได้ง่าย Viljamaa, A. [1] ได้กล่าวถึงการนำกลับมาใช้ใหม่ (Reuse) ที่นำมาใช้ในการพัฒนาซอฟต์แวร์มีอยู่ 3 ระดับ คือ

#### 1. การนำกลับมาใช้ภายใน (Internal reuse)

ในการออกแบบซอฟต์แวร์ที่เป็นแบบนามธรรม (Software's abstraction) ภายในโปรแกรมเดียวกันสามารถนำส่วนที่ออกแบบแล้วนั้นมาใช้ได้กับส่วนอื่นๆ ได้อีก

#### 2. การนำชุดคำสั่งกลับมาใช้ (Code reuse)

เป็นการรวบรวม ฟังก์ชัน และ คลาสพื้นฐานเก็บไว้เป็นห้องสมุด (Libraries) หรือ ชุดเครื่องมือ (Toolkit) สามารถนำไปประยุกต์ใช้กับโปรแกรมอื่นๆ ได้

#### 3. การนำสถาปัตยกรรมการออกแบบโปรแกรมกลับมาใช้ (Architectures reuse)

เป็นการใช้เทคนิคการนำกลับมาใช้ในระดับที่สูงกว่าสองวิธีแรก โดยสามารถนำสถาปัตยกรรมการออกแบบโปรแกรม กลับมาใช้ได้อีก

การออกแบบและพัฒนาโปรแกรมแบบเชิงวัตถุ(Object-oriented design and programming) เป็นการพัฒนาโปรแกรมที่สนับสนุนวิธีการนำกลับมาใช้ใหม่ โดยการเรียกใช้ คลาสไลบรารี (Class libraries) ซึ่งจะทำให้การเอื้ออำนวยเฉพาะสิ่งที่เป็นคลาสพื้นฐานเป็นการนำชุดคำสั่งกลับมาใช้ แต่ไม่ได้ให้บริการการนำกลับมาใช้ในระดับสถาปัตยกรรม นั่นจึงทำให้การพัฒนาโปรแกรมแบบเชิงวัตถุได้เปลี่ยนแปลงจากการใช้คลาสไลบรารีมาเป็นโครงสร้างสำหรับพัฒนาโปรแกรมประยุกต์ (Application framework)

รูปแบบการออกแบบ (Design pattern) เป็นแนวคิดวิธีการที่ใช้ในการแก้ปัญหาในการออกแบบพัฒนาโปรแกรม โดยได้เก็บรวบรวมจากรูปแบบวิธีการออกแบบที่พบบ่อยๆ ซึ่งพิสูจน์แล้วว่าสามารถนำมาใช้แก้ปัญหาได้จริง โดยทำการแยกปัญหาออกเป็นส่วนย่อยๆ ในการแก้ปัญหานั้นๆ อาจเลือกรูปแบบหนึ่งวิธี แต่ภายในปัญหานั้นๆ อาจมีปัญหาคู่อื่นซ่อนอยู่ซึ่งสามารถใช้รูปแบบวิธีอื่นๆ แก้ไขและในการสร้างโครงสร้างสำหรับพัฒนาโปรแกรมประยุกต์ จะประกอบไปด้วยรูปแบบการออกแบบหลายๆ วิธี

วิทยานิพนธ์เรื่องนี้ ได้สนับสนุนแนวคิดการนำสถาปัตยกรรมการออกแบบโปรแกรมกลับมาใช้ใหม่ โดยการนำรูปแบบการออกแบบมาพัฒนาเป็นส่วนหนึ่งของโครงร่างสำหรับสร้างโปรแกรมประยุกต์ ผู้ใช้สามารถนำโครงร่างที่ได้มาใช้สร้างโปรแกรมประยุกต์โดยการเขียนคำสั่งเพิ่มเติมในส่วนที่แสดงคุณลักษณะเฉพาะของโปรแกรมที่ต้องการเท่านั้น โดยภาษาโปรแกรมที่นำมาใช้ในการพัฒนา คือ ภาษาจาวา

## 1.2 วัตถุประสงค์

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาโครงร่างสำหรับสร้างโปรแกรมเชิงวัตถุด้วยรูปแบบการออกแบบ โดยรูปแบบการออกแบบที่นำมาพัฒนาเป็นส่วนหนึ่งของโครงร่างจะช่วยสร้างโครงชุดคำสั่งตามรูปแบบการออกแบบที่ผู้ใช้กำหนด ซึ่งโครงร่างที่พัฒนาจะสร้างโครงชุดคำสั่งเป็นภาษาจาวาให้อัตโนมัติ เพื่อให้ผู้ใช้เติมชุดคำสั่งให้สมบูรณ์ ซึ่งจะเป็นการลดจำนวนชุดคำสั่งที่ต้องเขียน

## 1.3 ขอบเขตของงานวิจัย

วิทยานิพนธ์นี้มีขอบเขตของการวิจัย ดังต่อไปนี้

1. ออกแบบและพัฒนาโครงร่างสำหรับสร้างโปรแกรมเชิงวัตถุด้วยรูปแบบการออกแบบ
2. นำรูปแบบการออกแบบ จำนวน 23 วิธี ของ Gamma, Helm, Johnson, Vlissides (1995) มาเป็นแม่แบบในการสร้างโครงชุดคำสั่งเพื่อใช้ในการสร้างโปรแกรมเชิงวัตถุ
3. โครงร่างสำหรับสร้างโปรแกรมเชิงวัตถุด้วยรูปแบบการออกแบบจะสร้างโครงชุดคำสั่งเป็นภาษาจาวามาตรฐานในปัจจุบัน คือ Java Development Kit 1.2 และ สามารถนำ JavaBeans มาใช้ร่วมในการสร้างโปรแกรมเชิงวัตถุ
4. ทดสอบโดยการสร้างโปรแกรมเชิงวัตถุจำนวน 3 โปรแกรม

## 1.4 ขั้นตอนและวิธีการดำเนินงาน

วิทยานิพนธ์นี้มีขั้นตอนและวิธีการดำเนินงานวิจัยดังต่อไปนี้

1. ศึกษาแนวคิดและทฤษฎี การพัฒนาซอฟต์แวร์เชิงวัตถุ โปรแกรมภาษาจาวา รูปแบบการออกแบบและโครงร่างสำหรับพัฒนาโปรแกรมเชิงวัตถุที่มีใช้ในปัจจุบันเพื่อเป็นตัวอย่าง
2. วิเคราะห์และออกแบบโครงร่างสำหรับสร้างโปรแกรมเชิงวัตถุด้วยรูปแบบการออกแบบ
3. พัฒนา และทดสอบโครงร่างสำหรับสร้างโปรแกรมเชิงวัตถุด้วยรูปแบบการออกแบบ
4. ประเมินผลโดยการสร้างโปรแกรมเชิงวัตถุจำนวน 3 โปรแกรม จากโครงร่างสำหรับสร้างโปรแกรมเชิงวัตถุด้วยรูปแบบการออกแบบ
5. สรุปผล และจัดทำวิทยานิพนธ์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

วิทยานิพนธ์นี้มีประโยชน์ ดังต่อไปนี้

1. โครงร่างสำหรับสร้างโปรแกรมเชิงวัตถุด้วยรูปแบบการออกแบบ สามารถนำมาช่วยในการพัฒนาโปรแกรมประยุกต์ ซึ่งช่วยลดจำนวนชุดคำสั่งที่จะต้องเขียน และทดสอบ
2. สนับสนุนแนวคิดการนำกลับมาใช้ใหม่ มาใช้ในขั้นตอนการพัฒนาโปรแกรม
3. การศึกษารูปแบบการออกแบบและนำมาประยุกต์ใช้ในการออกแบบโปรแกรม ทำให้การสื่อสารกันระหว่างผู้ออกแบบและผู้พัฒนาโปรแกรม มีความเข้าใจในการแก้ไขปัญหาเป็นแนวทางเดียวกัน และสามารถทำให้สื่อสารเข้าใจกันได้ง่ายขึ้น