บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

ในการพัฒนาระบบงานและการประมวลผลงานต่าง ๆ ด้วยคอมพิวเตอร์ จำเป็นจะต้องใช้ข้อมูล (Data) ในรูปแบบต่าง ๆ เพื่อการดำเนินการอยู่เสมอ ข้อมูลต่าง ๆ ที่ใช้จึงจำเป็นต้องมีการจัดการอย่างมีประสิทธิภาพ เพื่อให้โปรแกรมที่ต้องใช้ข้อมูลเหล่านั้น สามารถจัดเก็บและเข้าถึงได้อย่างสะดวก และมีการแบ่งปันการใช้ข้อมูลระหว่างโปรแกรมอื่นได้ตามรูปแบบที่กำหนด ซึ่งนับเป็นปัจจัยหนึ่งที่ช่วยให้การประมวลผลทางคอมพิวเตอร์มีประสิทธิภาพที่ดีขึ้น การให้ได้มาซึ่งการจัดเก็บข้อมูลที่มีประสิทธิภาพจึงจำเป็นต้องมีการออกแบบโครงสร้างข้อมูลที่เหมาะสมกับวัตถุประสงค์การใช้ข้อมูลนั้น โครงสร้างข้อมูลมีหลายรูปแบบ ซึ่งแต่ละแบบมีคุณลักษณะเฉพาะของมัน ตัวอย่างเช่น โครงสร้างข้อมูลแบบแถวลำดับ (Array) จะมีลักษณะของตารางเป็นช่อง ๆ โครงสร้างข้อมูลแบบกองซ้อน (Stack) มีคุณสมบัติที่ว่าเมื่อทำการเพิ่มข้อมูลเข้าไป หรืออ่านข้อมูลออกมา จะกระทำที่ปลายข้างเดียว โครงสร้างข้อมูลแบบต้นไม้ (Tree) จะมีการแตกกิ่งก้านของแต่ละหน่วยข้อมูลออกมาในลักษณะคล้ายกับการงอกของต้นไม้ที่งอกจากบนลงล่าง

จากรูปแบบของโครงสร้างข้อมูลบางชนิดที่กล่าวมา เมื่อโปรแกรมคอมพิวเตอร์นำไปใช้ขณะกำลังประมวลผลอยู่นั้น จะต้องใช้หน่วยความจำของคอมพิวเตอร์ในการจัดเก็บข้อมูลตามรูปแบบของโครงสร้างข้อมูลที่ใช้ โครงสร้างข้อมูลบางรูปแบบก็มีพร้อมที่จะให้เรียกใช้ได้อยู่แล้วในภาษาคอมพิวเตอร์บางภาษา เช่น โครงสร้างข้อมูลแบบแถวลำดับมีในภาษาซี ภาษาปาสคาล ภาษาเบสิค เป็นต้น แต่ยังมีโครงสร้างข้อมูลอีกบางรูปแบบที่ไม่สามารถเรียกใช้ได้ทันที ผู้พัฒนาโปรแกรมต้องสร้างโครงสร้างข้อมูลขึ้นมาใช้เองเช่น โครงสร้างข้อมูลแบบกองซ้อน โครงสร้างข้อมูลแบบแถวคอย เป็นต้น จึงทำให้การพัฒนาโปรแกรมต้องเสียเวลาส่วนหนึ่งไปในการสร้างโครงสร้างข้อมูลที่ต้องการใช้ขึ้นมาก่อน

ในการพัฒนาโปรแกรมที่มีขนาดใหญ่นั้น จะมีความยุ่งยากในการแก้ไขจุดบกพร่อง (Debug) ของโปรแกรมมากขึ้นเป็นเท่าตัว หนทางหนึ่งในการช่วยลดความยุ่งยากให้กับผู้พัฒนาโปรแกรมคือ การที่มีช่องทางที่จะช่วยให้การตรวจสอบและทดสอบโปรแกรมเป็นไปโดยง่าย (Kruse,

Leung, and Tondo, 1991) จากที่กล่าวมาข้างต้นว่า โครงสร้างข้อมูลมักจะเป็นส่วนหนึ่งในการพัฒนาโปรแกรม จึงควรใช้โครงสร้างข้อมูลที่มีส่วนช่วยให้การตรวจแก้ไขจุดบกพร่องของโปรแกรมทำได้ง่ายขึ้น

สำหรับแนวทางการพัฒนาโปรแกรมในปัจจุบัน ได้เริ่มใช้แนวคิดทางด้านการพัฒนาโปรแกรมเชิงวัตถุ (Object Oriented Programming) (Wiener and Pinson, 1992) และการพัฒนาโปรแกรมเชิงทัศน์ (Visual Programming) โดยวิชวลเบสิค (Visaul Basic) ของบริษัทไมโครซอฟต์ นับเป็นผลิตภัณฑ์สำหรับการเขียนโปรแกรมที่มีลักษณะการใช้งานใกล้เคียงกับแนวคิดทั้งสองดังกล่าวข้างต้น และมีวัตถุประสงค์เพื่อให้การพัฒนาโปรแกรมสำหรับวินโดว์ทำได้ง่ายและสะดวก (กิตติ เลิศพิริยสุวัฒน์, 2536) โดยผู้เขียนโปรแกรมสำหรับวินโดว์ไม่จำเป็นต้องศึกษาถึงหลักการทำงานของโปรแกรมต่าง ๆ ภายใต้วินโดว์ ตลอดจนการบริหารหน่วยความจำ การใช้ทรัพยากรต่างๆ ร่วมกันภายใต้วินโดว์ ซึ่งแตกต่างจากการเขียนโปรแกรมรูปแบบเดิมที่ต้องออกแบบหน้าจอเพื่อกำหนดตำแหน่งการแสดงผลและการรับข้อมูลเข้า จากนั้นจึงเขียนโปรแกรมสั่งงานให้คอมพิวเตอร์ทำงานตามลำดับขั้นตอน แต่การเขียนโปรแกรมด้วยวิชวลเบสิคจะใช้หลักการของวัตถุหรือตัวควบคุม (Control) และการมองเห็น โดยการนำวัตถุต่าง ๆ เช่น ข้อความ (Label), ช่องรับข้อความ (Text box), หรือปุ่มควบคุม (Command button) ที่ต้องการจะใช้งานมาประกอบกันบนฟอร์ม (Form) และกำหนดคุณสมบัติต่าง ๆ ของวัตถุแต่ละตัวบนฟอร์ม แล้วจึงเขียนโปรแกรมเชื่อมวัตถุเหล่านั้น ให้ทำงานตามเหตุการณ์ต่างๆ ที่เกิดขึ้นกับวัตถุเหล่านั้น โดยในปัจจุบันได้มีวัตถุสำเร็จรูปสำหรับการทำงานในด้านต่าง ๆ อยู่เป็นจำนวนมาก (Cilwa and Duntemann, 1994)

ดังนั้นจึงควรที่จะพัฒนาตัวควบคุมสำหรับเป็นโครงสร้างข้อมูลขึ้นมา ที่พร้อมให้นำไปประยุกต์ใช้งานได้ทันที และสามารถแสดงให้เห็นการเปลี่ยนแปลงของข้อมูลภายในโครงสร้างข้อมูลเนื่องจากการทำงานของโปรแกรมได้ เพื่อเป็นส่วนหนึ่งที่ช่วยให้ผู้พัฒนาโปรแกรม นำไปใช้ในการติดตามการทำงานของโปรแกรม ทำให้การตรวจแก้ไขจุดบกพร่องของโปรแกรมทำได้สะดวกขึ้น นอกจากนี้ยังจะเป็นประโยชน์ทางอ้อมสำหรับการเรียนการสอน โดยตัวควบคุมสำหรับโครงสร้างข้อมูลนี้จะช่วยสร้างเสริมความเข้าใจและแสดงภาพจำลองของโครงสร้างข้อมูลในระหว่างการเรียนการสอนได้ อย่างไรก็ตามโครงสร้างข้อมูลมีหลายรูปแบบหลายชนิด ซึ่งนายบุญศิริ บุญยกได้พัฒนาตัวควบคุมสำหรับโครงสร้างข้อมูลแบบต้นไม้ไว้แล้ว และวิทยานิพนธ์นี้จะพัฒนาตัวควบคุมสำหรับโครงสร้างข้อมูลแบบฮีปชนิดทวิภาค (Binary Heap) โดยจะนำมาใช้ประโยชน์ในงานที่เกี่ยวข้องกับการจัดบุริมภาพ (Priority) ของงานต่าง ๆ ส่วนตัวควบคุมสำหรับโครงสร้างข้อมูลรูปแบบอื่น ๆ จะสามารถใช้ตัวควบคุมนี้เป็นต้นแบบได้

วัตถุประสงค์

1. เพื่อพัฒนาชั้นของตัวควบคุม (Control Class) สำหรับโครงสร้างข้อมูลแบบฮีปชนิดทวิภาค ซึ่งประกอบด้วยคุณสมบัติ (Property) เหตุการณ์ (Event) วิธี (Method) และฟังก์ชันต่าง ๆ ที่เกี่ยวข้องกัน

2. เพื่อพัฒนาระบบความช่วยเหลือ (Help) สำหรับชั้นของตัวควบคุมในข้อ 1 ซึ่งระบบความช่วยเหลือนี้จะอธิบายถึงความหมายและรายละเอียดของคุณสมบัติ เหตุการณ์ วิธี และฟังก์ชันตลอดจนวิธีการใช้งานของตัวควบคุมนี้

3. เพื่อพัฒนาชุดสาธิต (Demo Set) สำหรับการใช้ชั้นของตัวควบคุมนี้กับวิชวลเบสิค ซึ่งแสดงให้เห็นถึงตัวอย่างการใช้งานตัวควบคุมนี้สำหรับวิชวลเบสิคในรายละเอียดด้านต่างๆ ของตัวควบคุม

ขอบเขตการวิจัย

1. พัฒนาชั้นของตัวควบคุมสำหรับโครงสร้างข้อมูลแบบฮีปที่แต่ละบัพประกอบด้วยส่วนที่เป็นตัวแทนของบัพและส่วนที่เป็นคีย์ (Key) ของบัพ ให้สามารถทำงานบนวิชวลเบสิคภายใต้สภาพปฏิบัติการวินโดว์ โดยมีรายละเอียดที่สำคัญดังนี้

คุณสมบัติของตัวควบคุมที่เพิ่มเติมเข้ามาสำหรับตัวควบคุมนี้โดยเฉพาะ ได้แก่

- Animation กำหนดว่าจะให้แสดงการเปลี่ยนแปลงของโครงสร้างข้อมูลทีละขั้นตอนหรือไม่
- HeapOrder กำหนดว่า การจัดข้อมูลในฮีปจะให้ข้อมูลมากที่สุดหรือน้อยที่สุดอยู่ข้างบน
- InputNode เป็นการทดลองเพิ่มบัพข้อมูลในระหว่างการออกแบบโปรแกรม (Design Time)
- DisplayTree กำหนดว่า การแสดงโครงสร้างข้อมูลจะแสดงเป็นภาพจำลองแบบต้นไม้หรือแบบแถวลำดับ
- BulidHeap กำหนดว่าจะสร้างโครงสร้างข้อมูลให้เป็นแบบฮีปหรือไม่

เหตุการณ์ของตัวควบคุมที่สร้างขึ้นมาใหม่ ได้แก่

- HpNodeAdd เมื่อมีการเพิ่มบัพใหม่เข้าไปในโครงสร้างข้อมูล
- HpNodeDel เมื่อมีการลบบัพออกจากโครงสร้างข้อมูล
- HpNodeChg เมื่อมีการเปลี่ยนแปลงบัพข้อมูลในโครงสร้างข้อมูล

เหตุการณ์ที่สามารถทำงานได้โดยผู้ใช้ไม่ต้องเขียนโปรแกรมประกอบ ได้แก่

- เมื่อผู้ใช้คลิกเมาส์ข้างขวาที่ตัวควบคุมติดกัน 2 ครั้ง แสดงว่าเป็นการเลือกบัพที่ต้องการขึ้นมาแก้ไขข้อมูล ระบบก็จะจัดการในเรื่องการติดต่อกับผู้ใช้(User Interface) เพื่อการแก้ไขหรือลบข้อมูลในบัพ
- เมื่อผู้ใช้คลิกเมาส์ข้างซ้าย 2 ครั้งติดต่อกันที่ตัวควบคุม แสดงว่าต้องการเพิ่มข้อมูลบัพใหม่เข้าไปก็จะแสดงบัพว่างขึ้นมา เพื่อรอการใส่ข้อมูลใหม่เข้าไป

วิธีและฟังก์ชันของตัวควบคุมที่จะพัฒนาประกอบด้วย

- Clear เป็นวิธีเพื่อลบข้อมูลทั้งหมดในโครงสร้างข้อมูล
- HpGetCount เป็นฟังก์ชันเพื่อนับจำนวนบัพในโครงสร้างข้อมูล
- HpMerge เป็นฟังก์ชันเพื่อรวมข้อมูลในฮีป 2 ตัวเข้าด้วยกัน
- HpSort เป็นฟังก์ชันเพื่อการจัดเรียงลำดับข้อมูลในโครงสร้างข้อมูล
- HpGetNode เป็นฟังก์ชันเพื่อการอ่านข้อมูลทั้ง 2 ส่วนจากหมายเลขบัพข้อมูลที่ระบุเข้ามา
- HpPopNode เป็นฟังก์ชันเพื่อการอ่านข้อมูลทั้ง 2 ส่วนของบัพรากออกจากโครงสร้างข้อมูล
- HpAddNode เป็นฟังก์ชันเพื่อการเพิ่มข้อมูลทั้ง 2 ส่วนของบัพใหม่เข้าไปในโครงสร้างข้อมูล
- HpChgNode เป็นฟังก์ชันเพื่อการเปลี่ยนแปลงข้อมูลทั้ง 2 ส่วนของบัพในโครงสร้างข้อมูล
- HpDelNode เป็นฟังก์ชันที่ระบุค่าคีย์ของบัพข้อมูลที่จะถูกลบออกจากโครงสร้างขัอมูล

2. พัฒนาระบบความช่วยเหลือ (Help) สำหรับตัวควบคุมในข้อ 1 ซึ่งอธิบายถึงคุณสมบัติ เหตุการณ์ วิธี และฟังก์ชัน ตลอดจนวัตถุประสงค์การใช้งานของตัวควบคุมนี้

3. พัฒนาชุดสาธิตซึ่งเป็นโปรแกรมวิชวลเบสิคแสดงให้เห็นตัวอย่างการใช้งานตัวควบคุมในรายละเอียดด้านต่าง ๆ

ขั้นตอนและวิธีการดำเนินการวิจัย

1. ศึกษาลักษณะโครงสร้างข้อมูลแบบฮีป
2. ศึกษาลักษณะการทำงานของวิชวลเบสิคและชั้นของตัวควบคุมที่เป็นมาตรฐานต่าง ๆ
3. ศึกษาวิธีการพัฒนาชั้นของตัวควบคุม
4. วิเคราะห์ ออกแบบ และพัฒนาชั้นของตัวควบคุมสำหรับโครงสร้างข้อมูลแบบฮีปชนิดทวิภาค
5. พัฒนาระบบความช่วยเหลือส
6. พัฒนาชุดสาธิตสำหรับตัวควบคุมที่ได้พัฒนาขึ้นมา
7. ทดสอบความถูกต้องโดยรวม

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำตัวควบคุมที่พัฒนาขึ้นมานี้ไปใช้เป็นตัวแปรพิเศษแบบหนึ่ง ซึ่งสามารถจัดเก็บข้อมูลในลักษณะโครงสร้างข้อมูลแบบฮีปชนิดทวิภาคได้ และสามารถมีการกระทำต่างๆ (Operation) เหมือนกับโครงสร้างข้อมูลทั่วไปได้ เช่น การเพิ่ม การลบ การจัดเรียงข้อมูลและการเรียกใช้ข้อมูล เป็นต้น

2. สามารถนำตัวควบคุมที่พัฒนาขึ้นมานี้ไปช่วยในการตรวจแก้จุดบกพร่อง (Debug) โปรแกรมได้สะดวกขึ้น เป็นการใช้งานแบบเชิงทัศน์ (Visual) ได้ กล่าวคือสามารถแสดงให้เห็นถึงการเปลี่ยนแปลงที่เกิดขึ้นกับข้อมูลภายในโครงสร้างไปทีละขั้นตอน ตามการทำงานของโปรแกรมผู้ใช้งาน

3. เป็นประโยชน์ทางอ้อมในด้านการเรียน การสอน ในเรื่องโครงสร้างข้อมูลแบบฮีป โดยสามารถนำมาแสดงภาพจำลองของโครงสร้างข้อมูลแบบฮีปชนิดทวิภาคได้ เพื่อให้เห็นถึงภาพการเปลี่ยนแปลงที่เกิดขึ้นกับโครงสร้างข้อมูล เนื่องจากการกระทำต่าง ๆ ต่อโครงสร้างข้อมูล เช่น การเพิ่ม การลบ และการจัดเรียงข้อมูล เป็นต้น

4. เป็นแนวทางสำหรับการพัฒนาชั้นของตัวควบคุม เพื่อประโยชน์การใช้งานด้านอื่นๆ ในโอกาสต่อไป