

บทที่ 1

บทนำ



ความเป็นมาและความสำคัญของปัญหา

ในการใช้คอมพิวเตอร์เพื่อประมวลผลงานต่าง ๆ นั้น จำเป็นจะต้องมีการใช้ข้อมูล (Data) เป็นจำนวนมากเพื่อประมวลผลด้วยเสมอ ข้อมูลเหล่านั้นก็ต้องมีที่จัดเก็บที่มีประสิทธิภาพ ที่สามารถให้โปรแกรมการทำงานต่าง ๆ เข้าถึงและเรียกค้นได้อย่างรวดเร็ว ซึ่งการเก็บข้อมูลเพื่อให้มีประสิทธิภาพดีนั้น จะต้องเก็บในลักษณะที่มีรูปแบบและโครงสร้างที่แน่นอน ตัวอย่างเช่น โครงสร้างข้อมูลแบบแถวลำดับ (ARRAY) โครงสร้างข้อมูลแบบกองซ้อน (STACK) โครงสร้างข้อมูลแบบต้นไม้ (TREE) เป็นต้น ซึ่งโครงสร้างแต่ละแบบ มีลักษณะเฉพาะตัวแตกต่างกันไป โครงสร้างข้อมูลแบบแถวลำดับ จะเป็นในลักษณะของตารางที่เป็นช่อง ๆ แต่ละช่องสามารถเก็บข้อมูลได้ 1 หน่วย อาจเป็นตัวเลขหนึ่งตัวหรือตัวอักษรหลาย ๆ ตัว ขนาดของแต่ละช่องต้องเท่ากันหมด โครงสร้างแบบกองซ้อนมีคุณสมบัติที่ว่าเมื่อทำการเพิ่มข้อมูลหรือลบข้อมูล จะกระทำที่ปลายข้างเดียวกัน ปลายนั้นเรียกว่า TOP OF STACK โครงสร้างแบบต้นไม้มีลักษณะคล้ายกับกิ่งก้านของต้นไม้ที่งอกจากข้างบนลงมาข้างล่าง

จากรูปแบบของโครงสร้างข้อมูลบางชนิดที่กล่าวมา เมื่อนำไปใช้เก็บข้อมูลจริง ๆ นั้น นักเขียนโปรแกรมจะต้องใช้หน่วยความจำของคอมพิวเตอร์ในการเก็บข้อมูล แต่ใช้หลักการของโครงสร้างข้อมูลที่เหมาะสมกับโปรแกรมประยุกต์นั้น ๆ ซึ่งภาษาที่ใช้ในการเขียนโปรแกรมนั้น ก็มีโครงสร้างข้อมูลบางอย่างให้กับนักเขียนโปรแกรมอยู่แล้ว เช่น โครงสร้างข้อมูลแบบแถวลำดับ แต่ยังมีโครงสร้างข้อมูลอื่น ๆ อีกมากมายที่ภาษาโปรแกรมไม่มีให้ นักเขียนโปรแกรมต้องใช้ภาษาโปรแกรมนั้นสร้างขึ้นมาเอง เช่น โครงสร้างข้อมูลแบบต้นไม้ โครงสร้างแบบกองซ้อน เป็นต้น จึงทำให้การพัฒนาโปรแกรมต้องใช้เวลาส่วนหนึ่งในการสร้างโครงสร้างของข้อมูลที่ต้องการขึ้นมาอีกด้วย

ขณะนี้ทางบริษัทไมโครซอฟต์ได้พัฒนาภาษาโปรแกรมใหม่ขึ้นมาภาษาหนึ่ง เรียกว่า ภาษาวิชวลเบสิก (VISUAL BASIC) มีพื้นฐานที่ใช้กับระบบปฏิบัติการดอส (DOS) และใช้กับวินโดวส์ของไมโครซอฟต์เอง โดยเฉพาะรุ่นที่ใช้กับวินโดวส์นั้น ทำให้การพัฒนาโปรแกรมสำหรับวินโดวส์ทำได้ง่ายและสะดวกยิ่งขึ้น โดยผู้เขียนโปรแกรมสำหรับวินโดวส์ไม่จำเป็นต้องศึกษาถึงหลักการทํางานของโปรแกรมต่าง ๆ ภายใต้วินโดวส์ การบริหารหน่วยความจำ การใช้ทรัพยากรต่าง ๆ ร่วมกันภายใต้วินโดวส์

หรือศึกษาถึง API (APPLICATION PROGRAM INTERFACE) ใด ๆ ของวินโดวส์เหมือนเมื่อครั้ง วินโดวส์เพิ่งเริ่มเป็นที่นิยมแพร่หลายแต่ก่อนเลย

การเขียนโปรแกรมรูปแบบเดิมนั้น เราจะต้องออกแบบหน้าจอ กำหนดตำแหน่งการแสดงผล ตำแหน่งการรับข้อมูลเข้า รูปแบบของข้อมูลออก กำหนดขั้นตอนการทำงานและอื่น ๆ จากนั้นจึงทำการเขียนโปรแกรม โปรแกรมที่ได้จะส่งงานคอมพิวเตอร์เป็นลำดับไป แต่การเขียนโปรแกรมด้วยวิซวลเบสิก จะใช้หลักของภาพและการมองเห็น โดยเริ่มจากออกแบบหน้าต่างหรือในวิซวลเบสิก เรียกว่า ฟอรัม (Form) ในฟอรัมจะประกอบด้วยวัตถุหรือคอนโทรล (Control) ต่าง ๆ ที่จะทำงานร่วมกัน เช่น ข้อความ (Label) , ช่องรับข้อความ (Text Box) , Scrollbar, ปุ่มควบคุม (Command Button) เมื่อกำหนดสิ่งเหล่านี้ครบตามความต้องการแล้ว จึงค่อยกำหนดว่าองค์ประกอบแต่ละอย่างจะทำงานร่วมกันอย่างไร โดยเขียนโปรแกรมเชื่อมเข้าไปกับวัตถุเหล่านั้นให้ทำงานตามเหตุการณ์ (Event) ที่เกิดขึ้น เช่น เมื่อมีการคลิก (Click) หรือดับเบิลคลิก (Double Click) เมาส์ที่ตัววัตถุแล้วจะเกิดอะไรขึ้น นอกจากนี้วัตถุแต่ละตัวจะมีลักษณะหรือคุณสมบัติ (Property) ของตัวเอง เช่น ช่องรับข้อความ จะมี ชื่อ, ชนิดอักษร, ความกว้าง, ความสูง และสี เป็นต้น โดยเราสามารถอ้างอิงหรือเปลี่ยนค่าของคุณสมบัติ เหล่านี้ได้ขณะที่โปรแกรมทำงานอยู่หรือด้วยหน้าต่างคุณสมบัติ (Properties Window) เป็นต้นว่า หากไม่มีการป้อนข้อมูลจะให้เห็นด้วยสีหนึ่งหรือไม่ต้องแสดงบนจอภาพเลย

วัตถุที่เป็นโครงสร้างข้อมูลแบบต้นไม้ที่จะสร้างนี้ ก็เช่นเดียวกับวัตถุอื่น ๆ ในวิซวลเบสิกที่จะมี คุณสมบัติเฉพาะตัวทางด้านโครงสร้างข้อมูลของตัวเอง เช่น มีลักษณะคล้ายต้นไม้ที่ออกจากข้างบนลงมา ข้างล่าง มีลักษณะของบัพ (Node) ที่สามารถเปลี่ยนแปลงได้ สามารถเก็บข้อมูลที่เป็นอักขระและตัวเลข ได้ เป็นต้น ซึ่งจะช่วยในการพัฒนาโปรแกรมประยุกต์โดยใช้ภาษาวิซวลเบสิกสำหรับวินโดวส์ของบริษัท ไมโครซอฟต์ และต้องใช้โครงสร้างข้อมูลแบบต้นไม้ไบนารี (Binary Tree) , ต้นไม้ไบนารีค้นหาข้อมูล (Binary Search Tree) และต้นไม้ที่มีความสูงสมดุล (AVL Tree) เพื่อให้ผู้เขียนโปรแกรมไม่ต้องสิ้นเปลืองเวลาในการสร้างโครงสร้างข้อมูลเหล่านี้ขึ้นมาด้วยวิซวลเบสิกเอง และยังช่วยในการศึกษาถึงการทำ งานของโครงสร้างข้อมูลแบบต้นไม้ เพราะตัววัตถุสามารถแสดงการเปลี่ยนแปลงภายในต้นไม้ให้เห็นอย่าง ชัดเจนได้

วัตถุประสงค์

1. สร้างวัตถุ (Object) ที่มีคุณสมบัติเป็นโครงสร้างข้อมูลแบบต้นไม้ไบนารี , แบบต้นไม้ไบนารีค้นหาข้อมูล และแบบต้นไม้ที่มีความสูงสมดุล เพื่อใช้กับวิซวลเบสิก สำหรับวินโดวส์ รุ่น 2.0 ขึ้นไปของบริษัท ไมโครซอฟต์

2. สร้างระบบความช่วยเหลือ (Help) ของวัตถุดังกล่าวทั้ง 3 แบบ

ขอบเขตของงาน

1. สร้างวัตถุ (หรือคอนโทรล) ขึ้นใหม่ 3 วัตถุ ได้แก่
 - 1.1 สร้างวัตถุต้นไม้ไบนารี โดยข้อมูลในแต่ละบัพอาจเป็นอักขระหรือตัวเลข
 - 1.2 สร้างวัตถุต้นไม้ไบนารีค้นหาข้อมูล
 - 1.3 สร้างวัตถุต้นไม้ที่มีความสูงสมดุล
2. วัตถุต้นไม้ ดังกล่าวข้างต้น จะมีลักษณะดังต่อไปนี้
 - 2.1 มีคุณสมบัติ (Property) ที่สำคัญ ได้แก่
 - 2.1.1 ParentNode Property หมายถึง บัพพ่อของแต่ละบัพ
 - 2.1.2 LeftNode Property หมายถึง ลูกทางซ้ายของแต่ละบัพ
 - 2.1.3 RightNode Property หมายถึง ลูกทางขวาของแต่ละบัพ
 - 2.1.4 Animation Property กำหนดว่าจะแสดงการเปลี่ยนแปลงภายในบัพ (เมื่อมีการเพิ่มหรือลบบัพ) ให้เห็นอย่างชัดเจนหรือไม่
 - 2.1.5 TreeLevel Property เก็บค่าของระดับความสูงของวัตถุในขณะนั้น
 - 2.2 มีเหตุการณ์ (Event) ที่สำคัญ ได้แก่
 - 2.2.1 TreeChange Event เกิดขึ้นเมื่อจำนวนบัพในวัตถุมีการเปลี่ยนแปลง
 - 2.2.2 NodeIncrease Event เกิดขึ้นเมื่อจำนวนบัพเพิ่มขึ้น
 - 2.2.3 NodeDecrease Event เกิดขึ้นเมื่อจำนวนบัพลดลง
 - 2.3 มีวิธี (Method) ที่สำคัญ ได้แก่
 - 2.3.1 AddItem Method ทำการเพิ่มบัพให้วัตถุ
 - 2.3.2 RemoveItem Method ลบบัพออกจากวัตถุ
 - 2.3.3 Clear Method ลบบัพทุกบัพภายในวัตถุ
 - 2.4 วัตถุจะตอบสนองในเหตุการณ์ต่อไปนี้
 - 2.4.1 เมื่อดับเบิลคลิกเมาส์ซ้าย ที่บัพใด ๆ จะเป็นการเพิ่มบัพใหม่ให้กับวัตถุ โดยจะปรากฏช่องรับข้อความ เพื่อรับค่าของบัพที่ต้องการเพิ่ม
 - 2.4.2 เมื่อคลิกเมาส์ขวาที่บัพใด ๆ จะเป็นการลบบัพนั้น โดยจะปรากฏช่องรับข้อความ เพื่อถามความแน่ใจของผู้ใช้ว่าต้องการลบบัพนี้แน่นอนหรือไม่

ขั้นตอนและวิธีดำเนินการ

1. ศึกษาถึงทฤษฎีของโครงสร้างข้อมูลที่เกี่ยวข้อง ศึกษาการเขียนโปรแกรมบนวินโดวส์ รุ่น 3.1 ของไมโครซอฟต์, ชุดพัฒนาโปรแกรมของไมโครซอฟต์ (Microsoft Windows Software Development Kit) รุ่น 3.1 , การเขียนโปรแกรมด้วยวิซวลเบสิกสำหรับวินโดวส์ และการใช้งานชุดพัฒนาโปรแกรมของ Microsoft C/C++ รุ่น 7.0
2. วิเคราะห์และออกแบบวัตถุต้นไม้อะบบความช่วยเหลือของวัตถุต้นไม้อื่น
3. พัฒนาและทดสอบการทำงานและความถูกต้องของวัตถุ
4. สรุปผลการวิจัยและข้อเสนอแนะ

ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้เขียนโปรแกรมที่ต้องการใช้โครงสร้างข้อมูลแบบต้นไม้ทั้ง 3 แบบ ไม่ต้องสูญเสียเวลาสร้างโครงสร้างข้อมูลดังกล่าวขึ้นมา เปรียบเสมือนเครื่องมือ (TOOL) อันหนึ่งในการพัฒนาโปรแกรมด้วยวิซวลเบสิกสำหรับวินโดวส์
2. ช่วยให้นักเขียนโปรแกรมสามารถมองเห็นภาพของต้นไม้ได้ในขณะพัฒนาโปรแกรม เช่น เมื่อทำการเพิ่มหรือลบบัพภายในต้นไม้ ซึ่งจะช่วยลดความยากลำบากในการแก้ไขข้อผิดพลาดของโปรแกรม (ERROR) ได้
3. สามารถใช้ศึกษาและอธิบายคุณสมบัติและลักษณะของโครงสร้างข้อมูลทั้ง 3 แบบได้
4. เป็นแนวทางให้กับผู้ที่ต้องการสร้างวัตถุอื่น ๆ ให้กับวิซวลเบสิกสำหรับวินโดวส์ได้ต่อไป