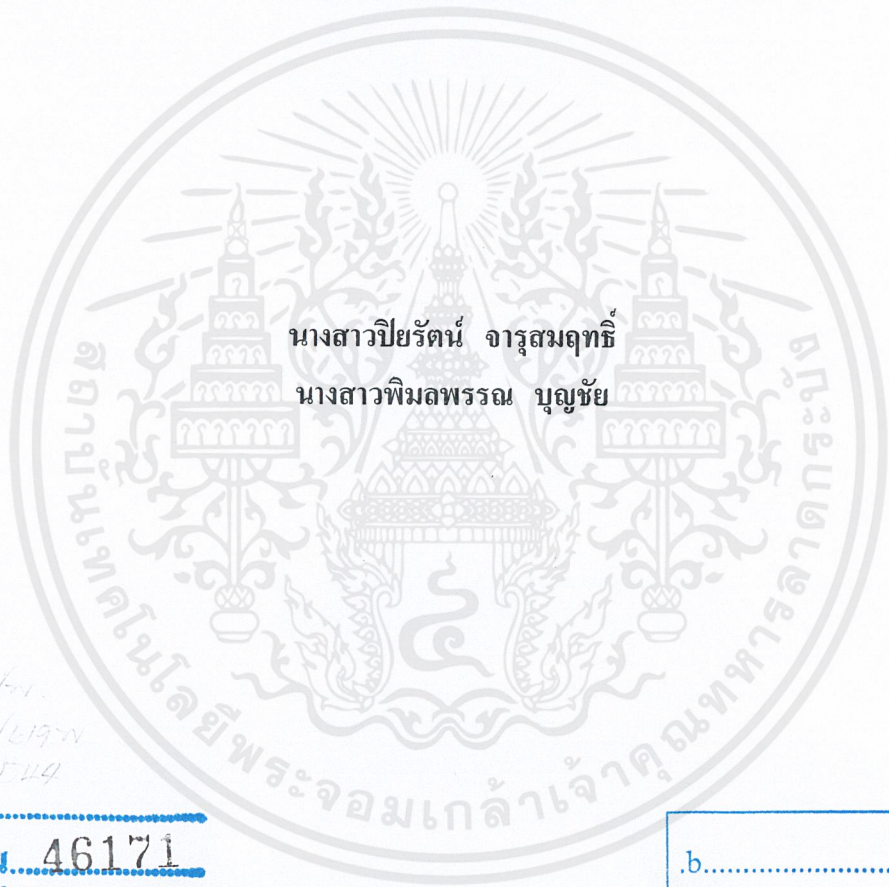


พจนานุกรมอังกฤษ-อังกฤษแบบลอย

Floating Thesaurus



นางสาวปิยรัตน์ จารุสมฤทธิ  
นางสาวพิมพ์พรรณ บุญชัย

๑-๑  
๑/๒๕๖๗  
๕๕๖๙

เลขหม.....  
เลขทะเบียน 46171  
วัน, เดือน, ปี 20 ส.ค. 2546

.b.....  
.i.....

ปริญญาโท เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พจนานุกรมอังกฤษ-อังกฤษแบบลอย

Floating Thesaurus



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2544

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง พจนานุกรมอังกฤษ-อังกฤษแบบลอย

Floating Thesaurus

ผู้จัดทำ

1. นางสาวปิยรัตน์ จารุสมฤทธิ รหัสประจำตัว 41014273
2. นางสาวพิมพ์พรรณ บุญชัย รหัสประจำตัว 41014309



อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## พจนานุกรมอังกฤษ-อังกฤษแบบลอย

ปิยรัตน์ จารุสมฤทธิ

พิมพ์พรรณ บุญชัย

เกียรติคุณ เจียรนัยชนะกิจ อาจารย์ที่ปรึกษา

ปีการศึกษา 2544

### บทคัดย่อ

ปฏิญานิพนธ์นี้จะนำเสนอแนวคิดในการพัฒนาโปรแกรมในเชิงวัตถุ (Object-Oriented Programming) เพื่อติดต่อกับฐานข้อมูล และการประยุกต์ใช้อินเทอร์เน็ต

โปรแกรมพจนานุกรมอังกฤษ-อังกฤษแบบลอยที่ได้พัฒนาขึ้นนี้เป็นโปรแกรมซึ่งทำงานบนโปรแกรมใดๆ เซอร์ภายใต้ระบบปฏิบัติการวินโดวส์ ที่ให้ความสะดวกแก่ผู้ใช้ในการหาความหมายของคำศัพท์ โดยการนำเมาส์ดับเบิลคลิกคำศัพท์ที่ปรากฏ ณ ตำแหน่งใดๆ บนหน้าจอคอมพิวเตอร์ ความหมายของคำศัพท์นั้นจะถูกนำมาแสดงในกรอบคำแปลในตำแหน่งที่เมาส์ชี้อยู่ ผู้ใช้สามารถหาความหมายของคำศัพท์โดยการพิมพ์ในส่วนติดต่อกับผู้ใช้ได้อีกด้วย ถ้าโปรแกรมไม่พบความหมายของคำศัพท์นั้นในฐานข้อมูลของโปรแกรม โปรแกรมสามารถหาความหมายของคำศัพท์นั้นจากอินเทอร์เน็ตได้โดยอัตโนมัติ และผู้ใช้อาจยังสามารถเพิ่ม แก้ไข และลบคำศัพท์ในฐานข้อมูลของโปรแกรมได้อีกด้วย

การพัฒนาโปรแกรมนี้กระทำภายใต้โพรโทคอล TCP/IP โดยใช้ “Microsoft Studio Visual C++ 6.0 Enterprise” เป็นเครื่องมือในการพัฒนา และใช้ MFC (Microsoft Foundation Class) Library เป็นไลบรารีหลักในการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Floating Thesaurus

Piyarat Jarusomridhi  
Pimonpun Boonchai  
Kietikul Jearanaitanakij Advisor  
Year 2001

### Abstract

This thesis represents an idea of Object-Oriented programming development for database accessing and Internet client application.

Floating Thesaurus application works on any programs that resided in Windows system, giving users convenience to search the synonyms of vocabulary by pointing a mouse over and double click on vocabulary that appears at any positions on the computer screen. Its synonyms are shown as Windows tooltip at the mouse pointer. Users can also search the synonyms of vocabularies by typing in the user-interface. If its synonyms are not found in program's database, the program can automatically search for its synonyms on Internet. Users can also add, edit and delete the vocabularies from program's database themselves.

This program was written based on TCP/IP protocol. We used "Microsoft Studio Visual C++ 6.0 Enterprise" as a developing tool and mostly implemented by MFC (Microsoft Foundation Class) Library.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้คงจะไม่อาจสำเร็จได้ หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลายๆ ฝ่ายด้วยกันต้องขอขอบพระคุณ อาจารย์เกียรติคุณ เจียรนัยธนะกิจ อาจารย์ที่ปรึกษาปริญญาานิพนธ์เป็นอย่างมาก ที่ได้คอยให้ความเอาใจใส่ แนะนำสิ่งต่างๆ และคอยช่วยเหลือเสมอมา ซึ่งเป็นส่วนสำคัญที่สุดที่ทำให้ปริญญาานิพนธ์นี้สำเร็จลงได้

บุคคลสำคัญที่ทำให้ข้าพเจ้ามีวันนี้ได้คือ บิดา มารดา อันเป็นที่เคารพรักยิ่งตลอดมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก ที่ได้เลี้ยงดูและอบรมข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ ได้ให้กำลังใจในการทำงาน และเอาใจใส่ดูแลในทุกๆ ด้าน ไม่สามารถหาที่เปรียบได้ จึงขอกราบขอบพระคุณมา ณ ที่นี้ด้วย และสุดท้ายนี้ ต้องขอขอบคุณเพื่อนๆ ทุกคนที่ได้คอยให้คำปรึกษา และความช่วยเหลือต่างๆ ในการทำงาน ทำให้ปริญญาานิพนธ์นี้สำเร็จได้ด้วยดี

ปิยรัตน์ จารุสมฤทธิ์  
พิมลพรรณ บุญชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VIII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ	3
บทที่ 2 OOP (Object Oriented Programming)	4
2.1 แนวคิดของ OOP	4
2.2 คลาสและออบเจกต์ในภาษา C++	5
2.2.1 การสืบทอดของคลาส (Inheritance)	6
2.2.2 ลำดับชั้น (Hierarchy)	7
2.2.3 การเข้าถึงข้อมูลภายในคลาส	8
2.2.3.1 การกำหนดระดับแบบ private	8
2.2.3.2 การกำหนดระดับแบบ public	8
2.2.3.3 การกำหนดระดับแบบ protected	9
2.2.4 คอนสตรัคเตอร์และดีสตรัคเตอร์	10
2.2.5 การโอเวอร์โหลดคอนสตรัคเตอร์	11
2.2.6 ฟังก์ชันเวอร์ซอล	12
บทที่ 3 การเข้าถึงข้อมูลในฐานข้อมูล	13
3.1 เทคโนโลยีสำหรับการเข้าถึงข้อมูลในฐานข้อมูล	13
3.1.1 DAO (Data Access Object)	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.1	Microsoft Jet	14
3.1.1.2	ODBC Direct	15
3.1.2	RDO (Remote Data Object)	15
3.1.3	เทคโนโลยีการเข้าถึงข้อมูลในแนวความคิด UDA	15
3.1.3.1	ODBC (Open Database Connectivity)	16
3.1.3.2	OLE DB (Object-Linking and Embedding Database)	17
3.1.3.3	ADO (ActiveX Data Object)	17
3.2	ข้อสรุปของการติดต่อกับฐานข้อมูลด้วยการใช้เทคโนโลยีการอินเทอร์เน็ตต่างๆ	18
บทที่ 4	MFC (Microsoft Foundation Class)	19
4.1	ประวัติของ MFC (Microsoft Foundation Class)	19
4.2	สถาปัตยกรรมของ MFC	20
4.2.1	คลาส CObject	20
4.2.2	คลาส CCmdTarget	21
4.2.3	คลาส CWnd	22
4.2.4	คลาส CDialog	23
4.2.5	คลาสคอนโทรล	23
4.2.6	คลาสอะเรย์ และคลาสลิสต์	23
4.2.7	คลาสที่ใช้ในการวาดภาพ	24
4.2.8	คลาสที่ใช้สำหรับอินเทอร์เน็ต	24
4.2.9	คลาสที่ไม่อยู่ในตระกูล CObject	24
4.3	ประโยชน์จากการใช้คลาส MFC	24
4.4	การใช้งาน MFC ในโปรแกรม	25
4.5	MFC ที่ใช้ในโปรแกรม	25
4.5.1	คลาสต่างๆ ของ DAO	25
4.5.1.1	คลาส CDaoWorkspace	25
4.5.1.2	คลาส CDaoDatabase	26
4.5.1.3	คลาส CDaoRecordset	27
4.5.2	Internet Client Application	28
4.5.2.1	WinInet Class	28
4.5.2.2	ขั้นตอนการอิมพลีเมนต์ HTTP Client Application	30
4.5.3	MFC WinInet Classes	31
4.5.3.1	คลาส CInternetSession	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3.2	คลาส CInternetConnection	31
4.5.3.3	คลาส CInternetFile	32
4.5.3.4	คลาส CHttpConnection	32
4.5.3.5	คลาส CHttpFile	33
4.5.3.6	คลาส CInternetException	33
4.6	วิธีการเขียนโปรแกรมโดยใช้ MFC Library	33
บทที่ 5	การสร้างไดนามิคลิงค์ไลบรารี (Dynamic Link Library)	36
5.1	ทฤษฎีพื้นฐานเกี่ยวกับไดนามิคลิงค์ไลบรารี	36
5.2	การสร้างและชนิดของ DLL	36
5.2.1	DLL แบบ Regular DLL	37
5.2.2	DLL แบบ MFC Extension DLL	38
บทที่ 6	แนวคิดและการออกแบบโปรแกรมเชิงวัตถุ	39
6.1	Use Case Diagram	39
6.2	Sequence Diagram and Collaboration Diagram	40
6.3	Floating Thesaurus Class Diagram	55
6.4	Floating Thesaurus Activity Diagram	56
6.4.1	Dictionary Class	56
6.4.2	My Vocabulary Class	60
6.4.3	Display Class	61
6.4.4	Game Class	63
บทที่ 7	การทำงานของโปรแกรมพจนานุกรมแบบลอย	64
7.1	การทำงานของโปรแกรม	64
7.1.1	การสร้างหน้าต่างหลักของโปรแกรม	64
7.1.2	การติดต่อกับฐานข้อมูล	68
7.1.3	การค้นหาคำศัพท์ เพิ่ม แก้ไข และลบคำศัพท์	69
7.1.4	การทำทวนคำศัพท์และเกม	72
7.1.5	การติดต่อกับระบบปฏิบัติการเพื่อทำงานร่วมกับโปรแกรมอื่นๆได้	75
7.2	คุณสมบัติของโปรแกรม	76
7.3	ข้อจำกัดของโปรแกรม	76
7.4	การใช้งานโปรแกรมพจนานุกรม	76
7.5	จุดเด่นและจุดบกพร่องในการทำงานของระบบ	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5.1 จุดเด่นของระบบ	79
7.5.2 จุดบกพร่องของระบบ	80
บทที่ 8 สรุปและวิจารณ์	81
8.1 ปัญหาและอุปสรรค	81
8.2 แนวทางการวิจัยและพัฒนาต่อ	81
8.3 เปรียบเทียบระบบกับผลิตภัณฑ์ที่มีอยู่ในตลาด	82
บรรณานุกรม	83



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

รูปที่ 2-1	แสดงการจัดลำดับชั้นของคลาส	7
รูปที่ 3-1	แสดงการใช้เทคโนโลยีการอินเทอร์เน็ตเฟส	13
รูปที่ 3-2	แสดงการอินเทอร์เน็ตเฟสระหว่างแอปพลิเคชันกับฐานข้อมูล	13
รูปที่ 3-3	แสดงสถาปัตยกรรมของ DAO	14
รูปที่ 3-4	แสดงสถาปัตยกรรมของ UDA	15
รูปที่ 3-5	แสดงสถาปัตยกรรมของ ODBC	16
รูปที่ 3-6	แสดงสถาปัตยกรรมของ OLE DB	17
รูปที่ 3-7	แสดงสถาปัตยกรรมของ ADO	17
รูปที่ 4-1	แสดงลำดับชั้นคลาสของสถาปัตยกรรม MFC	20
รูปที่ 4-2	แสดงการสืบทอดของคลาส CCmdTarget	21
รูปที่ 4-3	แสดงการสืบทอดของคลาส CWnd	22
รูปที่ 4-4	แสดงการสืบทอดของคลาส CDialog	23
รูปที่ 4-5	แสดงการจัดการเกี่ยวกับการโปรแกรมของ WinInet	29
รูปที่ 4-6	แสดงขั้นตอนการร้องขอบริการจาก HTTP Server	30
รูปที่ 6-1	แสดง Use Case Diagram	39
รูปที่ 6-2	แสดง Sequence Diagram ของ Search by Floating-Online	40
รูปที่ 6-3	แสดง Collaboration Diagram ของ Search by Floating-Online	40
รูปที่ 6-4	แสดง Sequence Diagram ของ Search by Floating-no repeat	41
รูปที่ 6-5	แสดง Collaboration Diagram ของ Search by Floating-no repeat	41
รูปที่ 6-6	แสดง Sequence Diagram ของ Search by Floating-play game	42
รูปที่ 6-7	แสดง Sequence Diagram ของ Search by Floating-play game	42
รูปที่ 6-8	แสดง Sequence Diagram ของ Search by Floating-no game	43
รูปที่ 6-9	แสดง Collaboration Diagram ของ Search by Floating-no game	43
รูปที่ 6-10	แสดง Sequence Diagram ของ Search by Kay-in-online	44
รูปที่ 6-11	แสดง Collaboration Diagram ของ Search by Key-in-online	44
รูปที่ 6-12	แสดง Sequence Diagram ของ Search by Key-in-no repeat	45
รูปที่ 6-13	แสดง Collaboration Diagram ของ Search by Key-in-no repeat	45
รูปที่ 6-14	แสดง Sequence Diagram ของ Search by Key-in-play game	46
รูปที่ 6-15	แสดง Collaboration Diagram ของ Search by Key-in-play game	46
รูปที่ 6-16	แสดง Sequence Diagram ของ Search by Key-in-no game	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6-17 แสดง Collaboration Diagram ของ Search by Key-in-no game	47
รูปที่ 6-18 แสดง Sequence Diagram ของ Search On-line	48
รูปที่ 6-19 แสดง Collaboration Diagram ของ Search On-line	48
รูปที่ 6-20 แสดง Sequence Diagram ของ Add - success Method	49
รูปที่ 6-21 แสดง Collaboration Diagram ของ Add – success Method	49
รูปที่ 6-22 แสดง Sequence Diagram ของ Add - error Method	50
รูปที่ 6-23 แสดง Collaboration Diagram ของ Add - error Method	50
รูปที่ 6-24 แสดง Sequence Diagram ของ Update - success Method	51
รูปที่ 6-25 แสดง Collaboration Diagram ของ Update - success Method	51
รูปที่ 6-26 แสดง Sequence Diagram ของ Update - error Method	52
รูปที่ 6-27 แสดง Collaboration Diagram ของ Update - error Method	52
รูปที่ 6-28 แสดง Sequence Diagram ของ Delete - success Method	53
รูปที่ 6-29 แสดง Collaboration Diagram ของ Delete – success Method	53
รูปที่ 6-30 แสดง Sequence Diagram ของ Delete - error Method	54
รูปที่ 6-31 แสดง Collaboration Diagram ของ Delete - error Method	54
รูปที่ 6-32 แสดง Class Diagram ของ Floating Thesaurus	55
รูปที่ 6-33 แสดง Activity Diagram ของ Dictionary Class	56
รูปที่ 6-34 แสดง Activity Diagram ของ Search by Floating Method	57
รูปที่ 6-35 แสดง Activity Diagram ของ Search by Key-in Method	57
รูปที่ 6-36 แสดง Activity Diagram ของ Search On-line Method	58
รูปที่ 6-37 แสดง Activity Diagram ของ Add Method	58
รูปที่ 6-38 แสดง Activity Diagram ของ Update Method	59
รูปที่ 6-39 แสดง Activity Diagram ของ Delete Method	59
รูปที่ 6-40 แสดง Activity Diagram ของ My Vocabulary Class	60
รูปที่ 6-41 แสดง Activity Diagram ของ Get Vocab + Meaning	60
รูปที่ 6-42 แสดง Activity Diagram ของ Set Vocab + Meaning	61
รูปที่ 6-43 แสดง Activity Diagram ของ Display Class	61
รูปที่ 6-44 แสดง Activity Diagram ของ Interface Display	62
รูปที่ 6-45 แสดง Activity Diagram ของ Floating Display Method	62
รูปที่ 6-46 แสดง Activity Diagram ของ Game Method	63
รูปที่ 7-1 แสดงหน้าต่างค้นหาคำศัพท์ของโปรแกรมพจนานุกรม	77
รูปที่ 7-2 แสดงการค้นหาคำศัพท์แบบลอย	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7-3	แสดงหน้าต่างการเพิ่มและแก้ไขคำศัพท์ตามหน้าที่ของคำ	78
รูปที่ 7-4	แสดงหน้าต่างคำศัพท์ที่ต้องการแก้ไข	78
รูปที่ 7-5	แสดงหน้าต่างขอคำยืนยันจากผู้ใช้ในการลบคำศัพท์	79
รูปที่ 7-6	แสดงหน้าต่างเกมทบทวนคำศัพท์	79



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่ 4-1	แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CDaoWorkspace	25
ตารางที่ 4-2	แสดงฟังก์ชันดาต้าเมมเบอร์ และหน้าที่ต่างๆ ของคลาส CDaoDatabase	26
ตารางที่ 4-3	แสดงฟังก์ชันดาต้าเมมเบอร์ และหน้าที่ต่างๆ ของคลาส CDaoRecordset	27
ตารางที่ 4-4	แสดงขั้นตอนการอิมพลิเมนต์ HTTP Client Application	30
ตารางที่ 4-5	แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CInternetSession	31
ตารางที่ 4-6	แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CInternetConnection	32
ตารางที่ 4-7	แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CInternetFile	32
ตารางที่ 4-8	แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CHttpConnection	32
ตารางที่ 4-9	แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CHttpFile	33
ตารางที่ 4-10	แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CInternetException	33
ตารางที่ 8-1	แสดงการเปรียบเทียบระบบกับผลิตภัณฑ์อื่น	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

พจนานุกรมเป็นหนังสือรวบรวมคำที่มีใช้อยู่ในภาษา โดยมากจัดเรียงตามตัวอักษร ให้ความรู้เรื่องอักษรวิธี บอกลีขาน และนิยามความหมาย ตลอดจนบอกประวัติของคำเท่าที่จำเป็น นอกจากนี้หนังสือพจนานุกรมยังเป็นเครื่องแสดงให้เห็นว่าชาติผู้เป็นเจ้าของพจนานุกรมนั้น มีวัฒนธรรมอยู่ในระดับไหน สูงต่ำเพียงไร เพราะฉะนั้นหนังสือพจนานุกรมจึงนับได้ว่าเป็นหนังสืออ้างอิงที่สำคัญและจำเป็นมาก

ในปัจจุบันนี้ ภาษาอังกฤษได้เข้ามามีบทบาทต่อชีวิตประจำวันของคนเรามากขึ้น เนื่องจากการศึกษารุดหน้าไปไกลมาก และภาษาอังกฤษย่อมเป็นปัจจัยสำคัญยิ่ง ซึ่งการศึกษาภาษาอังกฤษเพื่อให้ได้ผลสมบูรณ์นั้นย่อมต้องอาศัยพจนานุกรมที่ทันสมัยเป็นคู่มือในการศึกษา ทำให้มีความจำเป็นต้องการใช้พจนานุกรมมากขึ้นด้วย

นอกจากนี้เทคโนโลยีทางด้านคอมพิวเตอร์ได้พัฒนาไปมาก จึงได้มีการสร้างโปรแกรมพจนานุกรมขึ้นมาให้กับผู้ใช้คอมพิวเตอร์เพื่อช่วยอำนวยความสะดวก โปรแกรมพจนานุกรมที่มีอยู่ทั่วไปในปัจจุบันนี้มีหลายลักษณะด้วยกัน แต่โดยส่วนใหญ่แล้วนั้นจะต้องพิมพ์คำศัพท์เพื่อทำการค้นหาคำศัพท์ในฐานข้อมูล แล้วจึงค่อยแสดงความหมายของคำศัพท์ออกมาที่ส่วนติดต่อกับผู้ใช้ อีกทั้งเทคโนโลยีทางด้านอินเทอร์เน็ตก็ได้พัฒนาไปมากด้วยเช่นกัน ซึ่งอินเทอร์เน็ตนี้เป็นที่นิยมใช้กันอย่างแพร่หลายมากในปัจจุบัน และยังเป็นแหล่งรวบรวมความรู้จากทั่วทุกมุมโลกที่สามารถค้นหาข้อมูลต่างๆ ได้อย่างมากมาย มีความน่าเชื่อถือ และยังมีประสิทธิภาพอีกด้วย โดยเฉพาะการค้นหาคำศัพท์ทางอินเทอร์เน็ตนั้น จะมีเว็บไซต์อยู่หลายเว็บไซต์ที่คอยให้บริการการค้นหาคำศัพท์ออนไลน์อยู่มากมาย

อย่างไรก็ตามเป็นที่น่าสังเกตว่า ในโปรแกรมพจนานุกรมที่มีอยู่นั้น ยังไม่สามารถอำนวยความสะดวกให้กับผู้ใช้ได้มากเท่าที่ควร มันยังมีข้อจำกัดอยู่ หากผู้ใช้ต้องการค้นหาคำศัพท์ที่ต้องการซึ่งแสดงอยู่บนหน้าจอคอมพิวเตอร์ จะต้องเปิดโปรแกรมพจนานุกรม หรือต้องเปิดหน้าเว็บไซต์ที่ให้บริการค้นหาคำศัพท์ออนไลน์ขึ้นมา ก่อนจึงจะสามารถหาความหมายของคำศัพท์นั้นได้

เพื่ออำนวยความสะดวกให้กับผู้ใช้คอมพิวเตอร์มากขึ้น โครงการวิจัยนี้เราจึงได้พัฒนาโปรแกรมพจนานุกรมในรูปแบบใหม่ขึ้นมา ซึ่งมีความรวดเร็วในการค้นหาคำศัพท์ และง่ายต่อการใช้ นอกจากนี้ยังได้นำประโยชน์ของการค้นหาคำศัพท์ทางอินเทอร์เน็ตผ่านเว็บไซต์แบบออนไลน์ มาประยุกต์และพัฒนาโปรแกรมค้นหาคำศัพท์ ที่จะช่วยให้ผู้ใช้สามารถค้นหาคำศัพท์ได้ง่ายมากยิ่งขึ้น

## 1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อพัฒนาโปรแกรมพจนานุกรม ที่สามารถช่วยอำนวยความสะดวกในการค้นหาคำศัพท์แก่ผู้ใช้คอมพิวเตอร์มากยิ่งขึ้น
2. เพื่อศึกษาถึงทฤษฎี หลักการ และขั้นตอนการพัฒนาโปรแกรมเชิงวัตถุ (Object Oriented Programming) พร้อมทั้งนำความรู้ที่ได้รับนี้ไปประยุกต์ใช้ในการพัฒนาโปรแกรม
3. เพื่อศึกษาการเขียนโปรแกรมในการติดต่อกับฐานข้อมูล และศึกษาการเขียนโปรแกรมประยุกต์ ให้สามารถใช้ติดต่อกับอินเทอร์เน็ตได้

## 1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะเป็นการออกแบบและพัฒนาโปรแกรมในเชิงวัตถุ ซึ่งโปรแกรมที่ได้พัฒนาขึ้นมานั้นเป็นโปรแกรมพจนานุกรมอังกฤษ-อังกฤษแบบลอย ที่ช่วยอำนวยความสะดวกแก่ผู้ใช้ในการหาความหมายของคำศัพท์ ได้มีการศึกษาถึงวิธีการติดต่อกับฐานข้อมูลด้วยการใช้เทคโนโลยีการอินเทอร์เน็ตเฟสต่างๆ รูปแบบของโปรแกรมนั้นจะแสดงคำเหมือนของคำศัพท์ นิยามความหมาย และหน้าที่ของคำในกรอบคำแปล เมื่อนำเมาส์ไปดับเบิลคลิกตรงคำศัพท์ที่ต้องการค้นหาบนหน้าจอคอมพิวเตอร์ อีกทั้งยังสามารถหาความหมายของคำศัพท์ได้โดยการพิมพ์ในส่วนติดต่อกับผู้ใช้ได้อีกด้วย

นอกจากนี้หากว่าคำศัพท์ที่ผู้ใช้ต้องการค้นหานั้น ไม่ปรากฏอยู่ในฐานข้อมูลของโปรแกรม จะสามารถหาความหมายของคำศัพท์นั้นจากอินเทอร์เน็ตได้ โดยทำการขอไปยังเว็บไซต์ที่ให้บริการพจนานุกรมแบบออนไลน์โดยอัตโนมัติ (ในที่นี้จะเลือกใช้ [www.babylon.com](http://www.babylon.com)) แล้วนำมาเก็บไว้ในฐานข้อมูลของโปรแกรม และผู้ใช้ยังสามารถเพิ่ม แก้ไข และ ลบคำศัพท์ในฐานข้อมูลของโปรแกรมได้

อีกทั้งตัวโปรแกรมยังสามารถตรวจสอบการค้นหาคำศัพท์ของผู้ใช้ได้ โดยถ้าหากว่าผู้ใช้ได้มีการค้นหาคำศัพท์คำนั้นมากกว่าหนึ่งครั้งภายในช่วงระยะเวลาที่กำหนด จะมีส่วนช่วยเหลือของโปรแกรมซึ่งเป็นเกมที่จะช่วยทบทวน คำศัพท์ให้แก่ผู้ใช้

ในการพัฒนาระบบนั้นจะใช้ Microsoft Visual Studio C++ 6.0 Enterprise เป็นเครื่องมือในการพัฒนา และใช้ MFC (Microsoft Foundation Class) Library เป็นไลบรารีหลักในการเขียนโปรแกรม

## 1.4 วิธีการดำเนินงาน

งานวิจัยในโครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีหลักพื้นฐานที่เกี่ยวข้องกับงานวิจัย คือการศึกษาแนวคิดและวิธีการออกแบบโปรแกรมเชิงวัตถุ (Object Oriented Programming)

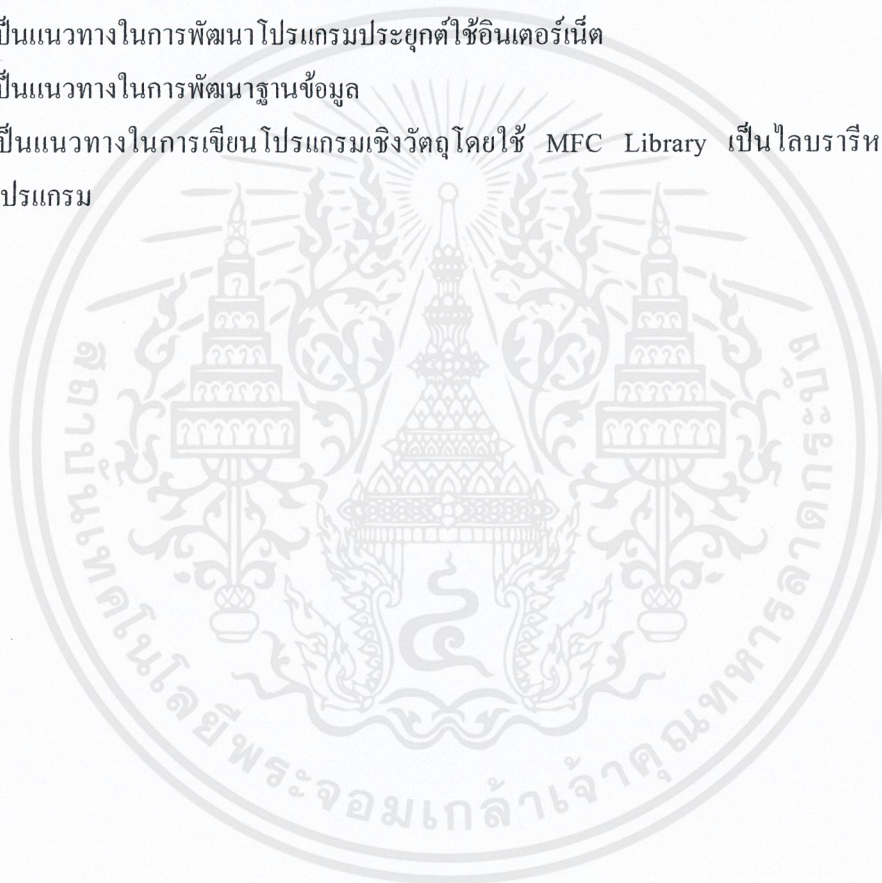
ศึกษาทฤษฎีในการพัฒนาโปรแกรม ซึ่งจะประกอบด้วยเรื่องหลักๆ อยู่ 3 เรื่องด้วยกัน คือเทคโนโลยีการเข้าถึงข้อมูลในฐานข้อมูลด้วยอินเทอร์เน็ตเฟสต่างๆ สถาปัตยกรรมของ MFC (Microsoft Foundation Class) และการใช้งาน MFC ในโปรแกรม และการสร้างไดนามิกลิงค์ไลบรารี (Dynamic Link Library - DLL)

จากนั้นนำเอาความรู้ที่ได้ศึกษามาทั้งหมดมาพัฒนาโปรแกรมพจนานุกรม โดยเริ่มจากการออกแบบโครงสร้างของโปรแกรมพจนานุกรม ซึ่งจะเป็นการแสดงถึงองค์ประกอบโดยรวมของโปรแกรม และขั้นตอนการทำงานหลักๆ ของโปรแกรม จากนั้นจึงเริ่มการพัฒนาโปรแกรมพจนานุกรมตามที่ได้ออกแบบเอาไว้

เมื่อได้พัฒนาโปรแกรมเสร็จสมบูรณ์แล้ว ก็ทำการทดสอบโปรแกรมพจนานุกรมรวมทั้งหมด และแก้ไขปรับปรุงโปรแกรมพจนานุกรมให้ดีขึ้น

### 1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

1. เป็นแนวทางในการพัฒนาโปรแกรมติดต่อกับระบบปฏิบัติการวินโดวส์โดยตรง
2. เป็นแนวทางในการพัฒนาโปรแกรมประยุกต์ใช้อินเทอร์เน็ต
3. เป็นแนวทางในการพัฒนาฐานข้อมูล
4. เป็นแนวทางในการเขียนโปรแกรมเชิงวัตถุโดยใช้ MFC Library เป็นไลบรารีหลักในการเขียนโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### OOP (Object Oriented Programming)

#### 2.1 แนวคิดของ OOP

โอโอพี (OOP) หรือออบเจกต์โอเรียนเท็ด โพรแกรมมิ่ง (Object Oriented Programming) เป็นแนวคิดในการเขียนโปรแกรมแบบหนึ่ง ที่ผู้รู้หลายๆ ท่านได้สรุป หรือได้ให้คำนิยามไว้ว่า เป็นการเขียนโปรแกรมเชิงวัตถุ บางท่านก็สรุปว่าเป็นการเขียนโปรแกรมแบบอ้างอิงวัตถุบ้าง ซึ่งก็เป็นการให้ความหมายที่ตรงมากที่สุด

ถ้าเราไม่มองในแง่มุมมองของการเขียนโปรแกรมเพียงอย่างเดียว ให้เรามองไปในภาพรวม มองไปในสิ่งรอบๆ ตัวเรา เราสามารถบอกได้ว่า แนวคิดของ OOP ก็คือ “*ธรรมชาติของวัตถุ*” หมายความว่า OOP จะมองสิ่งแต่ละสิ่งถือเป็น “*วัตถุชิ้นหนึ่ง*” มันจะมีสีแดงหรือสีเขียว จะยาวหรือสั้น มันก็คือวัตถุชิ้นหนึ่งเหมือนกัน ซึ่งวัตถุแต่ละสิ่งนั้น ย่อมมีคุณสมบัติที่ต่างกัน แต่อาจจะมีส่วนที่เหมือนกันบ้าง และเราก็สามารถกำหนดประเภทหรือคลาสให้กับวัตถุเหล่านั้นได้ เช่น วัตถุสีแดงก็มารวมอยู่ในกลุ่มเดียวกัน หรือวัตถุที่มีขนาดยาวก็มารวมอยู่ในกลุ่มเดียวกัน เป็นต้น

นอกจากนี้แล้ว เมื่อ OOP มองทุกสิ่งถือเป็นวัตถุชิ้นหนึ่งแล้ว มันยังคิดต่อไปอีกว่า “*วัตถุแต่ละอย่างนั้น ต่างก็จะมีลักษณะและวิธีการใช้งานเป็นของตัวเอง*” หมายความว่า วัตถุแต่ละชนิด หรือแต่ละชิ้น ต่างก็มีรูปร่างลักษณะ และการใช้งาน (การกระทำ) ที่แตกต่างกันออกไป เราจะเรียกคุณลักษณะของวัตถุว่า แอตทริบิวต์ (Attribute) และเราจะเรียกวิธีการใช้งานของวัตถุว่า เมธอด (Method) ยกตัวอย่างเช่น

*“ดินสอเป็นวัตถุที่มีลักษณะยาวเรียว, ภายในเป็นไส้ถ่านใช้สำหรับเขียน การใช้งานดินสอ ทำได้โดยใช้มือจับเขียนลงบนวัสดุรองรับ”*

จากประโยคในข้างต้น เราสามารถจับใจความได้ว่า คุณลักษณะของวัตถุ (Attribute) ก็คือ “*ยาวเรียว, ภายในเป็นไส้ถ่าน*” ส่วนการใช้งาน (Method) ก็คือ “*ใช้มือจับและเขียนลงบนวัสดุรองรับ*”

จากการยกตัวอย่างข้างต้น คราวนี้เราสามารถสรุปได้แล้วว่า ถ้าเกิดวัตถุใดมีลักษณะยาวเรียว, มีไส้เป็นถ่าน เมื่อจะใช้งานมัน เราจะต้องใช้มือจับ และเขียนลงบนวัสดุรองรับ เราก็สามารถบอกได้เลยว่า สิ่งนั้นคือ “*ดินสอ*” นั่นเอง

เราจะเห็นได้ว่า แนวคิดของ OOP นั้นจะมีลักษณะที่คล้ายกับธรรมชาติของสิ่งของสิ่งหนึ่ง ซึ่งเราสามารถแบ่งแยกสิ่งต่างๆ ออกเป็นประเภทๆ ได้ ถ้าเราได้นำเอาแนวคิด OOP มาใช้ในการเขียนโปรแกรมและการจัดการข้อมูล เราจะพบว่าโปรแกรมหรือฟังก์ชัน จะมีความเป็นอิสระแก่กันอย่างเห็นได้ชัด อธิบายง่ายๆ ก็คือ โปรแกรมหรือฟังก์ชันแต่ละตัวถึงแม้จะมาจากที่เดียวกัน แต่มันสามารถทำงานในคนละหน้าที่ เก็บข้อมูลคนละค่าได้ โดยจะไม่มายุ่งเกี่ยวกับเลย

## 2.2 กลาสและออบเจกต์ในภาษา C++

การเขียนโปรแกรมแบบ OOP มีลักษณะเช่นเดียวกับการเขียนโปรแกรมแบบโครงสร้าง โดยใช้หลักการที่คล้ายกันแต่ OOP นั้น จะมีประสิทธิภาพที่สูงกว่า เพราะเป็นการนำเอาคุณลักษณะ (Attribute) และวิธีการ (Method) เช่น ข้อมูล, ตัวแปร หรือฟังก์ชันของวัตถุนั้นมารวมไว้ในกลุ่มๆ เดียวกัน ที่เราเรียกว่า คลาส (Class)

การที่คลาสห่อหุ้มข้อมูล และฟังก์ชันการทำงานต่างๆ ของวัตถุเอาไว้ ซึ่งภายในคลาสสามารถป้องกันหรือซ่อนไว้โดยคอมไพเลอร์ เรียกว่า “การปกป้องข้อมูล (Encapsulation)” โดยผู้ที่เขียนโปรแกรมสามารถกำหนดได้ว่าจะป้องกันส่วนไหนของโปรแกรมที่พัฒนาขึ้น ดังนั้นผู้ที่เข้ามาใช้โปรแกรมจะไม่สามารถเข้าใช้ที่ส่วนนั้นๆ ได้ ซึ่งจะมีความสะดวกในการใช้งาน และเป็นการเพิ่มความปลอดภัยให้กับข้อมูลด้วย

เราจะนำเอาแนวคิดของ OOP มาเขียนโปรแกรม โดยจะสมมติคลาสขึ้นมา 1 คลาส โดยให้เป็นคลาสของดินสอ ใช้ชื่อคลาสว่า CPencil

```
Class CPencil {
    int length;
    void write(char *what);
};
```

จากโปรแกรมในข้างต้นจะเห็นได้ว่า การประกาศคลาส CPencil จะมีลักษณะคล้ายกับการประกาศโครงสร้าง (struct) โดยภายในคลาสจะประกอบไปด้วยคุณลักษณะ (Attribute) และวิธีการ (Method) โดยตัวแปร length จะเป็นคุณลักษณะที่ใช้สำหรับเก็บค่าความยาวของดินสอ และฟังก์ชัน write() เป็นวิธีการใช้งานดินสอนั้นเอง จะเห็นได้อย่างชัดเจนถึงความแตกต่างระหว่างการเขียนโปรแกรมแบบโครงสร้าง และแบบ OOP นั่นคือ การเขียนโปรแกรมแบบ OOP จะสามารถเพิ่มฟังก์ชันไปในกลุ่มข้อมูลได้ แต่การเขียนโปรแกรมแบบโครงสร้าง (struct) จะเป็นการรวมเอาข้อมูลที่แตกต่างกัน รวมเข้าไว้ด้วยกัน ภายใต้ชื่อเดียวกัน ซึ่งจะเก็บตัวแปรเพียงอย่างเดียว

เมื่อเราต้องการสร้างดินสอ 1 แท่งเอาไว้ใช้งาน เราจะต้องเขียนโค้ดโปรแกรมดังต่อไปนี้

```
CPencil pencil1;
```

การประกาศค่าในบรรทัดข้างต้น เราเรียกว่า “การประกาศออบเจกต์ (Object)” ซึ่งเป็นการกำหนดให้ pencil1 เป็นตัวแปรคลาส CPencil โดยมี length ใช้สำหรับเก็บค่าความยาวของ pencil1 และมีฟังก์ชัน write() เป็นวิธีการใช้งานของ pencil1 นั้นเอง ถ้ามีดินสอหลายๆ แท่ง เราก็สามารถประกาศออบเจกต์หลายๆ ตัวได้ เช่น pencil1, pencil2, pencil3 ดังนี้

```
CPencil pencil1, pencil2;
```

ออบเจกต์ pencil1 และ pencil2 ต่างก็จะมีตัวแปร length และมีฟังก์ชัน write() เป็นของตัวเอง ที่เห็นได้ชัดก็คือ ออบเจกต์แต่ละตัวจะเป็นอิสระซึ่งกันและกัน นั่นคือ pencil1.length จะไม่มีทางไปยุ่งวุ่นวายกับ pencil2.length ได้เลย เช่นเดียวกันกับคินสองแห่ง pencil1 อาจจะใช้ฟังก์ชัน write() เขียนวงกลม ในขณะที่คินสอง pencil2 ใช้ฟังก์ชัน write() วาดรูปสี่เหลี่ยมก็ได้ เราจะเห็นความเป็นอิสระของออบเจกต์อย่างชัดเจน ในการเขียนโปรแกรมจริงๆ นั้น เราจะได้พบลักษณะของการใช้งานออบเจกต์ที่มาจากคลาสเดียวกัน แต่ใช้งานที่แตกต่างกัน

การใช้งานตัวแปรแบบโครงสร้างในภาษา C++ และการใช้งานคลาสแบบ OOP จะเห็นว่ามีความคล้ายกันมาก โดยเฉพาะการรวมกันของตัวแปรต่างๆ ในโครงสร้าง แต่การใช้งานคลาสจะเห็นได้ชัดว่า ภายในคลาสเราสามารถแทรกฟังก์ชันลงไปได้ ซึ่งจะต่างจากการใช้ตัวแปรแบบโครงสร้างของภาษา C++ ที่ไม่สามารถกำหนดฟังก์ชันลงไปโครงสร้างได้ และยังไม่สามารถกำหนดขอบเขตต่างๆ ลงไปได้อีกด้วย

**คลาส (Class)** คือ การรวมคุณลักษณะและการใช้งานของวัตถุ อย่างน้อยหนึ่งอย่างมาไว้ในกลุ่มเดียวกัน คลาสเป็นพื้นฐานของการเขียนโปรแกรมแบบ OOP ซึ่งคลาสมีลักษณะเหมือนกับ Type ในภาษาแบบคลาสสิก (Classic Language) นอกจากนี้จะเก็บค่าขนาดและข้อมูลเกี่ยวกับโครงสร้างแบบ Type แล้วคลาวยังเก็บการกระทำ (Operation) ของมันเองไว้อีกด้วย คลาวยังมีลักษณะคล้ายกับ Abstract Data Type ในภาษาซี หรือปาสคาลอีกด้วย เพราะมีการเก็บข้อมูลเกี่ยวกับชนิดของข้อมูลที่ใช้ในการปฏิบัติการ (Procedure) ต่างๆ ในโครงสร้างนั้นๆ ด้วย

**ออบเจกต์ (Object)** คือ วัตถุที่เป็นตัวแปรคลาส เป็นรูปแบบของคลาสที่มีตัวตน ที่เราสามารถนำไปใช้งานได้ วัตถุเป็นค่าของคลาวยุติขณะปฏิบัติการ (Run-time value) ถ้าเปรียบคลาวยุติเป็น Type แล้ววัตถุจะเหมือนกับตัวแปรของ Type นั้นๆ คลาวยุติทำหน้าที่รวบรวมและกำหนดคุณสมบัติของวัตถุทั้งหมด

### 2.2.1 การสืบทอดของคลาส (Inheritance)

ถ้าเราสมมติให้ผีเสื้อถือเป็นคลาวยุติหนึ่ง การสืบทอดคลาวยุติ (การสืบทอด) ก็คือผีเสื้อกลายเป็นผีเสื้อ มันจะยังคงความเป็นผีเสื้ออยู่แน่นอน แต่มันจะมีการเพิ่มบางสิ่ง หรือคุณลักษณะบางอย่างเข้าไป และอาจถูกเปลี่ยนชื่อจากเดิมไปเป็นอีกชื่อหนึ่ง (หรืออาจจะไม่มีการเปลี่ยนแปลงใดๆ เลย) เช่นผีเสื้อพันธุ์หนึ่งมีชื่อว่าพันธุ์ AAA เมื่อมันกลายเป็นผีเสื้อพันธุ์ ผีเสื้อมีปีกใหญ่ขึ้น และมีลวดลายมากขึ้น มันก็เลยถูกกำหนดให้เป็นพันธุ์ใหม่มีชื่อว่า พันธุ์ AAABBB เป็นต้น

เช่นเดียวกับคินสอง ถ้าเราบอกว่ามีปากกาอยู่ชนิดหนึ่ง ซึ่งปากกาชนิดนี้มีไส้เป็นถ่าน เพราะฉะนั้นเราจะสามารถบอกได้ว่า ปากกานี้เป็นคินสองประเภทหนึ่งเช่นเดียวกัน แต่เป็นคินสองแบบใหม่ที่สืบทอดมาจากคินสองแบบเก่าอีกทีหนึ่ง ที่เราเรียกกันว่า “คินสองกด” นั่นเองซึ่งความสามารถของคินสองก็เพิ่มมากขึ้น หรืออาจจะตัดส่วนที่ไม่ดีของคินสองแบบเก่าออกไป เช่น คินสองกด เป็นคินสองแบบใหม่ที่เราไม่จำเป็นต้องเหลาบ่อยๆ

ลักษณะของการกลายพันธุ์ หรือการเปลี่ยนรูปแบบจากแบบเก่า มาเป็นแบบใหม่ในแนวคิดของ OOP เราเรียกว่า “การสืบทอดคลาวยุติ หรืออินเฮริเทนซ์ (Inheritance)” หมายความว่า คลาวยุติแต่ละคลาวยุติ (คลาวยุติแม่)

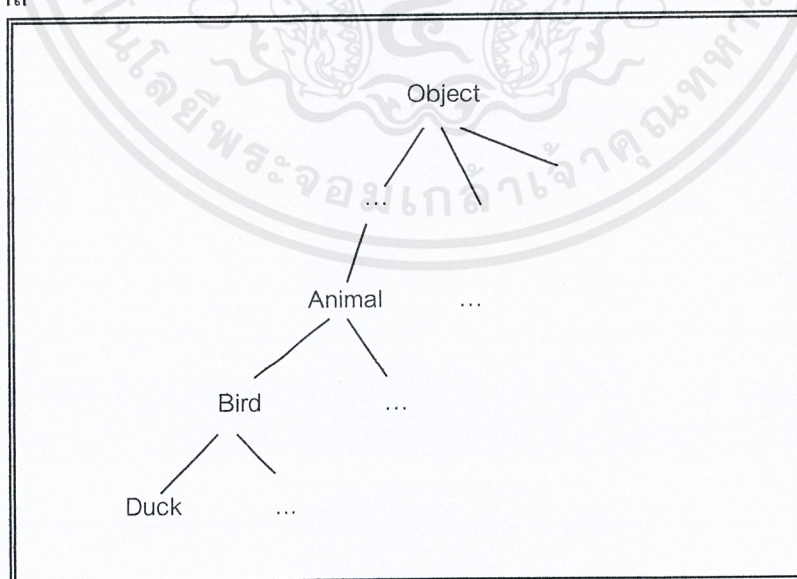
สามารถสืบทอดมาเป็นคลาสใหม่ได้ (คลาสลูก) โดยคลาสใหม่ที่สืบทอดออกมานี้ จะยังมีคุณสมบัติเหมือนกับคลาสเดิมทุกประการ ไม่มีเปลี่ยนแปลง

```
class CPencil {
    int length;
    void write(char *what);
};
class CPushPencil:CPencil {
};
```

จากโปรแกรมนี้เป็นการสร้างคลาสใหม่ที่ชื่อ CPushPencil โดยให้คลาสใหม่นี้สืบทอดคุณสมบัติมาจากคลาสเดิมคือ CPencil ซึ่งการสืบทอดนี้ไม่ว่าเมื่อสืบทอดมาแล้ว คลาส CPencil จะสลายหายไป คลาส CPencil จะยังคงอยู่ และสามารถใช้งานได้เหมือนเดิมทุกประการ สมาชิกที่อยู่ในคลาส CPencil ก็จะปรากฏอยู่ในคลาสใหม่ CPushPencil ด้วยและนอกจากนี้ เรายังสามารถเพิ่มคุณสมบัติใหม่ๆ ลงไปในคลาส CPushPencil ได้ เช่น เราอาจจะเพิ่มตัวแปรบางตัวที่คลาสเดิมไม่มี หรือเพิ่มวิธีการ (Method) ใหม่ๆ เข้าไป

## 2.2.2 ลำดับชั้น (Hierarchy)

คลาสใน OOP มีการจัดรูปแบบเป็นลำดับชั้นคล้ายทรี (Tree) โดยที่ “ซูเปอร์คลาส (Super Class)” ของคลาสนั้นๆ จะเป็นคลาสที่อยู่ลำดับชั้นบนสุดของทรีส่วนคลาสที่อยู่ด้านล่างเรียกว่า “ซับคลาส (Sub Class)” ความหมายของลำดับชั้นคือ คลาสจะมีคุณสมบัติทุกอย่างที่ซูเปอร์คลาสมี การจัดลำดับชั้นช่วยในการเพิ่ม Logic เพิ่มรวบรวมคลาสและยังช่วยให้คลาสที่คล้ายกันใช้คุณสมบัติร่วมกันได้โดยผ่านลำดับชั้นที่ต่ำกว่า ในภาษา C++ คลาสพื้นฐาน (Base class) หมายถึงซูเปอร์คลาสและคลาสที่ได้รับการสืบทอด (Derived class) หมายถึงซับคลาส



รูปที่ 2-1 แสดงการจัดลำดับชั้นของคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.3 การเข้าถึงข้อมูลภายในคลาส

การเข้าถึงข้อมูลหรือสมาชิกภายในคลาส เช่น ตัวแปรที่อยู่ภายในคลาส (data member) หรือฟังก์ชัน (member function) ภายในคลาสนั้น เราก็สามารถทำได้โดยใช้เครื่องหมายจุด (.) คั่นกลางระหว่างออบเจกต์กับตัวแปรหรือฟังก์ชัน แต่เมื่อเราเขียนโปรแกรมจริงๆ เราจะต้องดูประเภทของตัวแปรและฟังก์ชันที่อยู่ภายในคลาสนั้นเสียก่อน เพราะการประกาศสมาชิกภายในคลาส ได้มีการกำหนดระดับเอาไว้ เพื่อใช้สำหรับรักษาคุณค่าหรือข้อมูลภายในคลาส การกำหนดระดับของสมาชิกมีด้วยกัน 3 แบบ ดังนี้

#### 2.2.3.1 การกำหนดระดับแบบ private

การประกาศตัวแปรหรือฟังก์ชันให้เป็นแบบ private นี้ เป็นการป้องกันไม่ให้กระบวนการใดๆ ที่อยู่ภายนอกคลาสเรียกใช้ได้ สิ่งที่จะเรียกใช้ได้อีกคือ กระบวนการที่อยู่ภายในคลาสนั้นเท่านั้น คลาสลูกจะไม่สามารถเข้าถึงสมาชิกประเภทนี้ได้ การเขียนโปรแกรมเราสามารถละคำว่า private ได้ เช่น

<pre>class CPencil {     int length;     void write(char *what); };</pre>	<pre>class CPencil {     private:     int length;     void write(char *what); };</pre>
---	--

ตัวแปร length และฟังก์ชัน write() ได้ถูกประกาศให้เป็นแบบ private เมื่อเราสร้างออบเจกต์ของคลาสนี้ เราจะไม่สามารถเรียกใช้ตัวแปร length และฟังก์ชัน write() ได้เลย เพราะออบเจกต์เป็นกระบวนการที่อยู่ภายนอกคลาสนี้ ดังนี้

<pre>CPencil pencil1; pencil1.length = 10; pencil1.write(10);</pre>	<pre>//ไม่สามารถเรียกใช้ได้ //ไม่สามารถเรียกใช้ได้</pre>
---	--

#### 2.2.3.2 การกำหนดระดับแบบ public

การประกาศตัวแปรหรือฟังก์ชันของคลาสแบบ public นี้ จะตรงข้ามกับแบบ private นั่นคือ กระบวนการที่อยู่ภายในคลาส และภายนอกคลาสนั้นทั้งหมดรวมทั้งออบเจกต์ของคลาสนี้จะสามารถเข้าถึงได้หมด หรือเรียกอีกอย่างหนึ่งว่า การประกาศสมาชิกแบบสาธารณะ เช่น

<pre>Class CPencil {     int length;     public:     void SetLength (int how) { length = how; } };</pre>
--

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราสร้างออบเจ็กต์ pencil ขึ้นมา ถ้าต้องการกำหนดค่าให้กับตัวแปร length ภายในคลาสนี้ เราจะไม่สามารถใช้ออบเจ็กต์กำหนดค่าให้กับมันได้ เพราะตัวแปร length ยังเป็นแบบ private ออบเจ็กต์และกระบวนการภายนอกคลาสไม่สามารถเข้าถึงตัวแปรนี้ได้ แต่เมื่อเรามาดูที่ฟังก์ชัน CPencil::SetLength() ซึ่งจะเห็นได้ว่าฟังก์ชันนี้สามารถเรียกใช้ตัวแปร length ได้ นั่นก็เป็นเพราะว่าฟังก์ชัน SetLength() เป็นกระบวนการที่กระทำอยู่ภายในคลาสนั่นเอง

ดังนั้น เราจึงสามารถกำหนดค่าให้กับตัวแปร length ภายในคลาส CPencil ซึ่งเป็นแบบ private ได้ โดยการเรียกผ่านฟังก์ชัน SetLength() อีกทีหนึ่ง

```
p.SetLength(10);
```

เมื่อฟังก์ชัน SetLength() รับค่าพารามิเตอร์ 10 เข้าไป กระบวนการภายในคลาสนี้จะเริ่มขึ้น ฟังก์ชัน SetLength() ก็จะกำหนดค่าพารามิเตอร์ที่อยู่ในตัวแปร how ให้กับตัวแปร length ที่เป็นแบบ private ได้

### 2.2.3.3 การกำหนดระดับแบบ protected

การประกาศสมาชิกคลาสแบบ protected นี้ จะมีลักษณะเช่นเดียวกับแบบ private แต่จะต่างกันตรงที่ตัวแปรแบบ protected จะเพิ่มสิทธิ์ให้กับสมาชิกภายในคลาสลูกสามารถ เข้าถึงตัวแปรแบบ protected นี้ได้ด้วย นั่นคือคลาสลูกที่สืบทอดมาจากคลาสแม่ กระบวนการที่เกิดขึ้นภายในคลาสลูกก็จะสามารถเข้าถึงข้อมูลแบบ protected ในคลาสมแม่ได้ (ซึ่งจะต่างกับแบบ private) เช่น

```
Class CPencil {
    protected:
        int length;
};
class CPushPencil:public CPencil {
    public:
        void UseIt() { length = 10; }
};
```

จากตัวอย่าง เป็นการประกาศคลาส CPencil และประกาศตัวแปร length เป็นแบบ protected เมื่อทำการสืบทอดคลาสลูกมาเป็นคลาส CPushPencil ตัวแปร length ซึ่งเป็นแบบ protected ก็สามารถถูกเรียกใช้ได้ โดยกระบวนการภายในคลาสลูกนี้ โดยใช้ฟังก์ชัน UseIt() นั่นเอง ถ้าหากเราเปลี่ยนจาก protected ให้เป็นแบบ private ตัวแปร length ก็จะไม่สามารถเรียกใช้ได้จากการภายในคลาสลูกเลย

เมื่อเราประกาศออบเจ็กต์ของคลาส CPushPencil ออบเจ็กต์จะไม่สามารถเข้าถึงตัวแปร length ได้โดยตรง แต่สามารถเรียกใช้ฟังก์ชัน UseIt() เพื่อกำหนดค่าให้กับตัวแปรได้ เพราะฟังก์ชัน UseIt() เป็นกระบวนการภายในคลาส

```
CPushPencil pencil1;
Pencil1.UseIt();
```

## 2.2.4 คอนสตรัคเตอร์และดีสตรัคเตอร์ (Constructor and Destructor)

การเขียนโปรแกรมแบบ OOP เราสามารถใช้คุณสมบัติของฟังก์ชันคอนสตรัคเตอร์ (Constructor) และดีสตรัคเตอร์ (Destructor) เพื่อเพิ่มความยืดหยุ่นให้กับโปรแกรมได้ ในการสร้างคลาสแต่ละครั้ง เราไม่จำเป็นต้องสร้างคอนสตรัคเตอร์และดีสตรัคเตอร์ทุกครั้งไป การสร้างคอนสตรัคเตอร์และดีสตรัคเตอร์นั้นขึ้นอยู่กับความจำเป็นในแต่ละสถานการณ์ และการดำเนินงานของคลาส (ไม่ว่าคลาสทุกคลาสจะต้องมี)

ในคลาสๆ หนึ่ง สามารถมีคอนสตรัคเตอร์หรือมีดีสตรัคเตอร์เพียงตัวใดตัวหนึ่งก็ได้ หรือจะมีทั้งสองตัวก็ได้ หรือไม่มีทั้งสองตัวก็ได้ ไม่มีข้อกำหนดตายตัวเอาไว้ หน้าที่ของฟังก์ชันคอนสตรัคเตอร์และดีสตรัคเตอร์ก็คือ

ฟังก์ชันคอนสตรัคเตอร์ เป็นฟังก์ชันที่จะถูกเรียกให้ทำงาน เมื่อมีการสร้างออบเจกต์ของคลาสเกิดขึ้น ฟังก์ชันดีสตรัคเตอร์ เป็นฟังก์ชันที่จะถูกเรียกให้ทำงาน เมื่อสิ้นสุดการทำงานของออบเจกต์ในคลาสนั้นๆ เช่น จบโปรแกรม หรือมีการลบออบเจกต์ออกจากหน่วยความจำ เป็นต้น

ลักษณะของคอนสตรัคเตอร์ จะเป็นฟังก์ชันหนึ่งที่อยู่ภายในคลาส โดยมีชื่อฟังก์ชันเหมือนกับชื่อคลาส และดีสตรัคเตอร์ก็เป็นฟังก์ชันหนึ่ง ที่ใช้ชื่อเดียวกันกับชื่อคลาส แต่จะมีเครื่องหมาย ~ นำหน้าฟังก์ชัน

```
class CPencil {
    int length;
public:
    CPencil() { length = 10; }
    ~CPencil() { length = 0; }
};
```

จากตัวอย่างในข้างต้น เป็นการประกาศคลาส CPencil ประกอบด้วยสมาชิก length และคอนสตรัคเตอร์กับดีสตรัคเตอร์ เมื่อใดที่ออบเจกต์ของคลาส CPencil นี้ถูกสร้างขึ้น ฟังก์ชันคอนสตรัคเตอร์ CPencil() ก็จะถูกเรียกทันที บรรทัด length=10; ก็จะถูกกระทำ และเมื่อสิ้นสุดการทำงานของออบเจกต์ ฟังก์ชันดีสตรัคเตอร์ ~CPencil() ก็จะถูกเรียกและบรรทัด length=0; ที่อยู่ภายในฟังก์ชันนี้ก็จะถูกกระทำ

นอกจากนี้คอนสตรัคเตอร์และดีสตรัคเตอร์ ยังสามารถนำไปใช้ในการจองหน่วยความจำ และคืนหน่วยความจำได้อีกด้วย ซึ่งเป็นที่นิยมให้กันมาก เพราะผู้ใช้ไม่จำเป็นต้องเขียนโค้ดโปรแกรม เพื่อคืนหน่วยความจำเมื่อถึงเวลาที่กำหนด เช่น

```
class CPencil {
    CObject *i;
public:
    CPencil() { i = new CObject(); }
    ~CPencil() { delete i; }
};
```

## 2.2.5 การโอเวอร์โหลดคอนสตรัคเตอร์ (Constructor Overloading)

โอเวอร์โหลดในภาษาที่เราู้จักกัน หมายถึง การที่มีสิ่งต่างๆ หรือส่วนประกอบอื่นๆ มากเกินกว่าจะรับไว้ได้ แต่สำหรับโอเวอร์โหลดใน OOP นั่นก็คือ การที่มีฟังก์ชันคอนสตรัคเตอร์หลายๆ ตัวใน 1 คลาส ซึ่งปกติแล้วอาจคิดว่าในคลาสหนึ่งๆ จะสามารถมีฟังก์ชันคอนสตรัคเตอร์ได้เพียงตัวเดียว แต่จริงๆ แล้วสามารถสร้างฟังก์ชันคอนสตรัคเตอร์ขึ้นมาได้หลายตัวโดยที่ใช้ชื่อเดียวกัน แต่จะต่างกันที่ตรงที่การกำหนดพารามิเตอร์และค่าที่รีเทิร์น (Return) ให้กับฟังก์ชัน เพื่อความยืดหยุ่นในการใช้คลาส ลักษณะนี้เรียกว่า “การโอเวอร์โหลดฟังก์ชันคอนสตรัคเตอร์ (Constructor Overloading Function)”

ตัวอย่างการโอเวอร์โหลดฟังก์ชัน แสดงได้ดังนี้

```
Class CPencil {
    Public:
        CPencil();
        CPencil(int Long);
        CPencil(BOOL visible);
        CPencil(CString name);
};
```

จากตัวอย่างโปรแกรม จะเห็นได้ว่า คลาส CPencil มีคอนสตรัคเตอร์อยู่ด้วยกัน 4 ตัวตามลำดับ จะสังเกตเห็นได้อย่างชัดเจน และสามารถสรุปได้ว่า ฟังก์ชันโอเวอร์โหลดของคอนสตรัคเตอร์นั้น จะมีชื่อเหมือนกับคลาส แต่จะต่างกันตรงที่การส่งค่าพารามิเตอร์ไปยังฟังก์ชันคอนสตรัคเตอร์นั่นเอง เมื่อประกาศออบเจกต์ของคลาส จะมีทางเลือกในการสร้างออบเจกต์มากขึ้น เพราะสามารถผ่านค่าไปให้กับคอนสตรัคเตอร์ของคลาสได้ เช่น

```
CPencil p;
```

จากบรรทัดข้างต้น เป็นการประกาศออบเจกต์ของคลาส CPencil ฟังก์ชันคอนสตรัคเตอร์ตัวแรก จะถูกเรียกให้ทำงานโดยอัตโนมัติ เพราะเราไม่ได้ผ่านค่าใดๆ ไปให้ นอกจากนี้เรายังสามารถประกอบออบเจกต์ในแบบต่างๆ โดยการผ่านค่าพารามิเตอร์ไปให้กับคอนสตรัคเตอร์ ดังนี้

```
CPencil mypen(10);
CPencil yourpen(TRUE);
CPencil Dadpen("You");
```

ถ้าคลาสพบว่ามีกรโอเวอร์โหลดฟังก์ชันรองรับเอาไว้ ฟังก์ชันคอนสตรัคเตอร์ตัวนั้นก็จะถูกเรียกใช้งานอัตโนมัติ

## 2.2.6 ฟังก์ชันเวอร์ชวล (Virtual Function)

ฟังก์ชันเวอร์ชวล เป็นฟังก์ชันหนึ่งที่คลาสแม่อนุญาตให้คลาสลูกประกาศได้โดยใช้ชื่อ และรูปแบบฟังก์ชันที่เหมือนกัน เช่น โปรแกรมต่อไปนี้ แสดงการประกาศคลาส CPencil และฟังก์ชันเวอร์ชวล Write()

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
class CPencil {
public:
    virtual void Write() { printf("pencil write\n"); }
};
class CPushPencil: public CPencil {
public:
    void Write() { printf("PushPencil write\n"); }
};
```

จะเห็นว่าในคลาสลูก CPushPencil ก็ได้มีการประกาศฟังก์ชัน Write() โดยใช้รูปแบบของฟังก์ชันที่เหมือนกันกับในคลาสแม่ทุกอย่าง ในเวลาใช้งานคลาส เมื่อเราสร้างออบเจกต์ของคลาส CPushPencil และเรียกใช้ฟังก์ชัน Write() เราจะต้องเขียนโค้ดโปรแกรมดังบรรทัดต่อไปนี้

```
CPushPencil p1;
p1.Write();
```

ผลที่ได้จากการรันโปรแกรมก็คือ ข้อความ "PushPencil write" จะแสดงบนจอภาพ นั่นเป็นเพราะว่า p1 เป็นออบเจกต์ของคลาส CPushPencil ฟังก์ชัน Write() ที่อยู่ภายในคลาส CPushPencil จะถูกเรียกให้ทำงาน ซึ่งจำเป็นมากที่จะต้องกำหนดฟังก์ชันที่เป็นแบบเวอร์ชวล ให้มีรูปแบบที่เหมือนกันกับคลาสแม่ เพราะถ้าหากออบเจกต์หาฟังก์ชัน Write() ในคลาสลูกไม่พบก็จะขึ้นไปหาฟังก์ชัน Write() ที่คลาสแม่โดยอัตโนมัติ

ในการประกาศฟังก์ชันเวอร์ชวลนี้ จะช่วยให้สามารถสร้างฟังก์ชันที่มีชื่อและรูปแบบซ้ำกันกับฟังก์ชันในคลาสแม่ได้โดยใส่คำสั่งว่า virtual นำหน้าฟังก์ชันที่ต้องการในคลาสแม่ เรียกการประกาศในลักษณะนี้ว่า การโอเวอร์ไรด์ฟังก์ชัน (Function Overriding) หรือการเขียนฟังก์ชันซ้ำนั่นเอง

ในการเขียนฟังก์ชันเวอร์ชวลนี้ จะทำให้สามารถเพิ่มความสามารถของฟังก์ชันภายในคลาสลูกได้โดยไม่เกี่ยวกับคลาสแม่

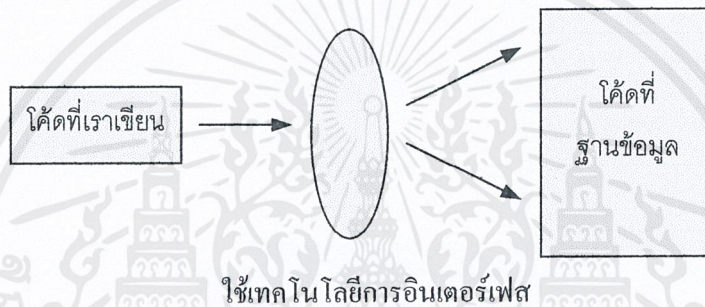
## บทที่ 3

### การเข้าถึงข้อมูลในฐานข้อมูล

#### 3.1 เทคโนโลยีสำหรับการเข้าถึงข้อมูลในฐานข้อมูล

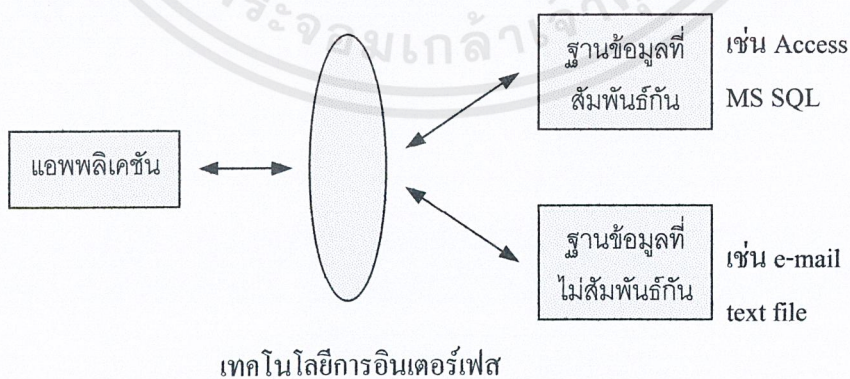
เนื่องจากฐานข้อมูล คือซอฟต์แวร์ชิ้นหนึ่งที่มีความซับซ้อนมาก (เทคโนโลยีการอินเทอร์เน็ตเฟสกับฐานข้อมูลมีความซับซ้อนน้อยกว่าฐานข้อมูล) มีผลให้การเขียนโปรแกรมเพื่อติดต่อกับฐานข้อมูลด้วยอินเทอร์เน็ตเฟสแบบธรรมดาเป็นงานเขียนค่อนข้างซับซ้อนมาก แต่ถ้าใช้เทคโนโลยีการอินเทอร์เน็ตเฟสกับฐานข้อมูลแล้ว จะสามารถเขียนโปรแกรมด้วยโค้ดที่มีจำนวนลดลง (โค้ดนี้อยู่ที่ฐานข้อมูล) ไปมากทีเดียว

การอินเทอร์เน็ตเฟสฐานข้อมูลที่ดีนั้น เหมือนกับเลนส์ที่ขยายโค้ดให้มีขนาดใหญ่ขึ้นดังรูป



รูปที่ 3-1 แสดงการใช้เทคโนโลยีการอินเทอร์เน็ตเฟส

การเขียนโปรแกรมเพื่อสื่อสารกับฐานข้อมูลด้วยวิธีการธรรมดา นอกจากจะมีความซับซ้อนแล้ว ยังมีผลให้แอปพลิเคชันที่ได้ไม่ยืดหยุ่นต่อการใช้งานและมีข้อจำกัด รวมทั้งถ้าเปลี่ยนไปใช้กับฐานข้อมูลอื่นๆ จะต้องแก้ไขโค้ด และสูญเสียเวลาไปมาก



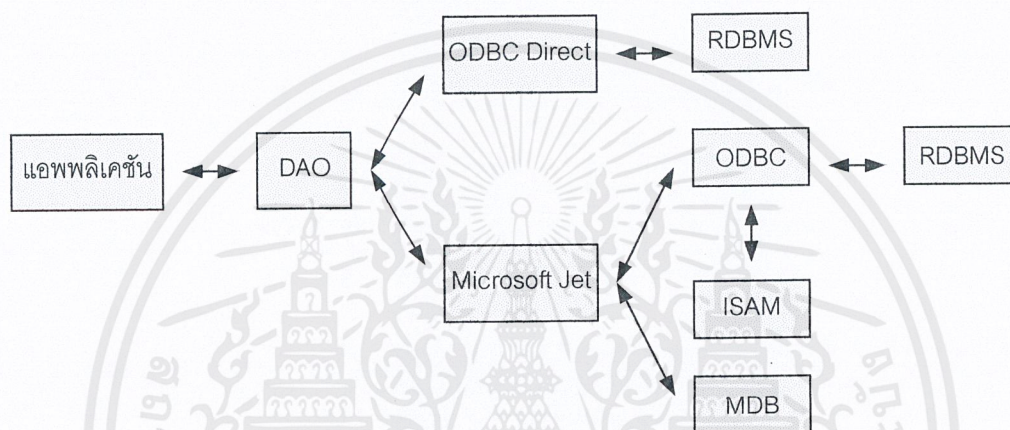
รูปที่ 3-2 แสดงการอินเทอร์เน็ตเฟสระหว่างแอปพลิเคชันกับฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีการอินเทอร์เน็ตเฟสได้ให้รูปแบบการอินเทอร์เน็ตเฟสแบบเดียวสำหรับการสื่อสารกับฐานข้อมูลที่แตกต่างกัน เทคโนโลยีการอินเทอร์เน็ตเฟสสมัยใหม่ทำให้สามารถเขียนโค้ดได้ง่าย และสามารถทำกระบวนการซับซ้อนกับฐานข้อมูลชนิดใดๆ ก็ได้

เทคโนโลยีการอินเทอร์เน็ตเฟสกับฐานข้อมูลมีด้วยกันหลายชนิด เช่น ODBC (Open Database Connectivity), RDO (Remote Data Objects), DAO (Data Access Object), OLE DB (Object-Linking and Embedding Database) และ ADO (ActiveX Data Object)

### 3.1.1 DAO (Data Access Object)



รูปที่ 3-3 แสดงสถาปัตยกรรมของ DAO

ในปี ค.ศ. 1995 บริษัทไมโครซอฟท์ได้แนะนำ DAO สำหรับ Microsoft Jet Database Engine ซึ่งเป็นเซตของการอินเทอร์เน็ตเฟสแบบ COM Automation โดย DAO ทำให้สามารถใช้ภาษาโปรแกรมคอมพิวเตอร์เข้าถึงและจัดการข้อมูลในฐานข้อมูลได้

DAO เข้าถึงข้อมูลด้วย Object Model แทนที่จะเป็นการเรียกกลุ่มฟังก์ชันใน DLL (Dynamic Link Library) ด้วยการใช้โมเดลออบเจกต์ทำให้การเรียกดู หรือการกระทำข้อมูลในฐานข้อมูลได้ง่าย นอกจากนี้แล้วออบเจกต์โมเดล DAO ได้รวมคลาสต่างๆ และออบเจกต์นี้ยังห่อหุ้มด้วยโครงสร้างต่างๆ ของฐานข้อมูล Access เช่น table, query, index และอื่นๆ อีก นั่นหมายความว่า DAO อนุญาตให้สามารถแก้ไขโครงสร้างของฐานข้อมูล Access ได้โดยตรง โดยปราศจากการใช้คำสั่ง SQL statement ได้

DAO สามารถติดต่อกับฐานข้อมูลได้สองแบบคือ ด้วย Microsoft Jet หรือ ODBC Direct

#### 3.1.1.1 Microsoft Jet

สามารถเข้าถึงข้อมูลในฐานข้อมูล Access ได้โดยตรง และ Microsoft Jet นี้ต่อกับ ODBC เพื่อติดต่อกับฐานข้อมูล ISAM (Indexed-Sequential Access Model) เช่น FoxPro, Paradox หรือ Lotus 1-2-3 และฐานข้อมูล RDBMS อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณลักษณะที่เด่นของ DAO เมื่อใช้ Microsoft Jet นั้น Microsoft Jet มีฟังก์ชันมากมาย ซึ่งมีคุณลักษณะเด่นดังต่อไปนี้

1. การจัดการกับข้อมูล (Data Manipulation) สามารถที่จะ Add, Update และ Delete ได้
2. Join กับตารางในฐานข้อมูลอื่นได้ (Heterogeneous Join) เราสามารถทำการ Join กับตารางในฐานข้อมูลอื่นๆ เช่น สามารถ Join ตารางใน Access กับตารางใน MS SQL Server ได้
3. การทำซ้ำ (Replication) Microsoft Jet อนุญาตให้สามารถทำสำเนาฐานข้อมูลได้ ดังนั้นผู้ใช้ที่อยู่ต่างๆ กัน สามารถทำงานกับฐานข้อมูลทีละปีมา ณ เวลาเดียวกันและใช้ร่วมกัน หรือทำซิงค์โครไนซ์กัน เช่น เมื่อคนที่ 1 update ข้อมูล คนอื่นๆ ที่อยู่อีกที่หนึ่งก็จะเห็นข้อมูลถูก update ด้วย

### 3.1.1.2 ODBC Direct

สามารถเข้าถึงข้อมูลในฐานข้อมูล Access โดยติดต่อกับฐานข้อมูล RDBMS อื่นๆ ซึ่งจะไม่ผ่าน Microsoft Jet

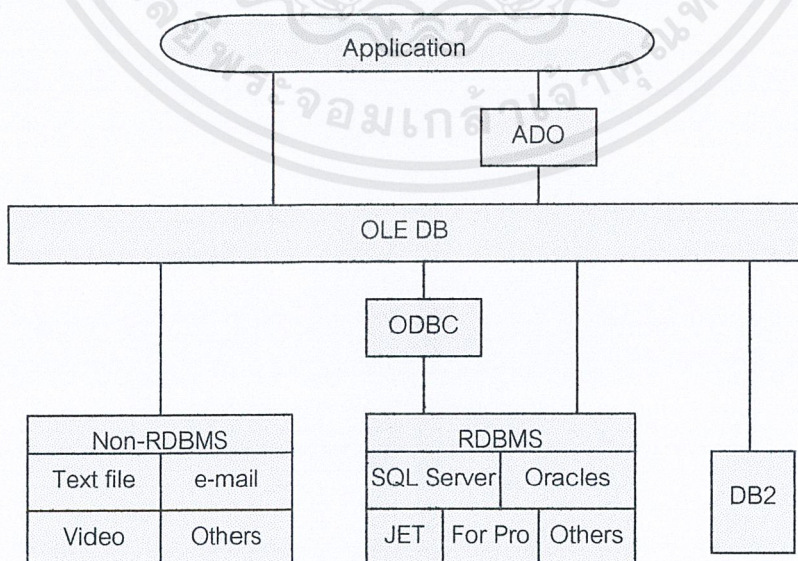
### 3.1.2 RDO (Remote Data Object)

RDO ใช้โมเดลออบเจกต์ในการเข้าถึงข้อมูล ODBC โมเดลการโปรแกรมมิ่ง RDO คล้ายกับโมเดลของ DAO ยกเว้นว่ามีจุดประสงค์สำหรับการเข้าถึงฐานข้อมูล SQL มากกว่าที่จะเป็นข้อมูล ISAM ดังนั้น RDO จึงเน้นไปทางด้าน Stored Procedure มากกว่า

RDO ถูกพัฒนาขึ้นเพื่อ โปรแกรมภาษา Visual Basic มีความสามารถเข้าถึงข้อมูล ODBC โดยไม่ต้องเขียนโค้ด ODBC API ด้วยตนเอง ดังนั้น RDO จึงถูกจัดให้เป็นการอินเทอร์เฟซในระดับสูง

นอกจากนี้แล้ว RDO ถูกออกแบบให้เอาข้อดีของฐานข้อมูล SQL Server คือการใช้ query ที่ซับซ้อน

### 3.1.3 เทคโนโลยีการเข้าถึงข้อมูลในแนวความคิด Universal Data Access (UDA)



รูปที่ 3-4 แสดงสถาปัตยกรรมของ UDA

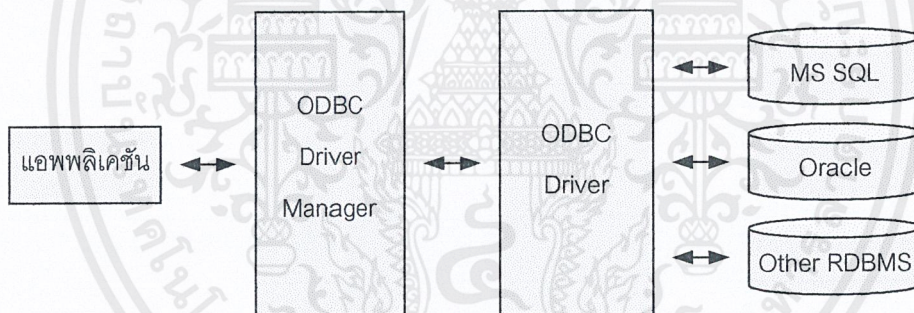
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UDA เป็นสถาปัตยกรรมใหม่ของ Microsoft ที่ทำให้การเข้าถึงข้อมูลเป็นไปได้โดยมีประสิทธิภาพ ทั้งแบบฐานข้อมูลที่สัมพันธ์กัน และที่ไม่สัมพันธ์กัน (Relational and Nonrelational) และได้หลาย Platform ด้วยการใช้ UDA ทำให้เราสร้างหรือเขียน โปรแกรมได้ง่าย รวมทั้งสามารถใช้กับเครื่องมือ และภาษาโปรแกรมมิ่งของทุกวันนี้ได้ทั้งหมด

UDA เป็นระบบเปิดและไม่จำเป็นต้องใช้ผลิตภัณฑ์ของผู้ผลิตฐานข้อมูลรายเดียว ในความจริงแล้ว UDA สามารถใช้กับ Platform ฐานข้อมูลที่มีอยู่ทุกวันนี้ได้ทั้งหมด UDA ประกอบด้วยเทคโนโลยีหลักๆ 3 เทคโนโลยีด้วยกัน

จากรูปแสดงสถาปัตยกรรมของ UDA สมมุติว่าข้อมูลอยู่ใน RDBMS ซึ่งมี Driver ODBC และไม่มี OLE DB Provider แอปพลิเคชันของเราสามารถใช้ ODBC ติดต่อกับข้อมูลในฐานข้อมูล หรือแอปพลิเคชัน เราสามารถใช้ OLE DB Consumer ติดต่อกับ OLE DB Provider ไปยัง Driver ODBC จึงจะสามารถติดต่อกับข้อมูลในฐานข้อมูลได้ ถ้าข้อมูลอยู่ในดาต้าซอส (Data Source) ที่มี OLE DB Provider แอปพลิเคชันของเราใช้ OLE DB Consumer เพื่อติดต่อโดยตรงกับ OLE DB Provider โดยไม่ผ่าน Driver ODBC (ดูรูป สถาปัตยกรรมของ OLE DB)

3.1.3.1 ODBC (Open Database Connectivity)

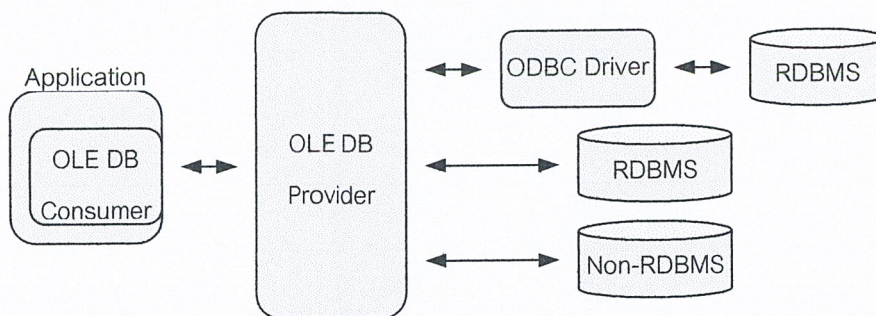


รูปที่ 3-5 แสดงสถาปัตยกรรมของ ODBC

ODBC สร้างขึ้นเมื่อช่วงปลายทศวรรษ 1980 และช่วงต้น 1990 เพื่อให้การเขียนโปรแกรมติดต่อกับฐานข้อมูลแบบสัมพันธ์ ใช้การอินเตอร์เฟสมีรูปแบบเดียวกัน ODBC ได้ให้ API สำหรับแอปพลิเคชันที่จะติดต่อกับฐานข้อมูลต่างๆ โดยผ่าน Driver ODBC

เมื่อเปรียบเทียบกับการอินเตอร์เฟสอื่นๆ แล้ว ODBC API ถูกจัดเป็นกลุ่มการอินเตอร์เฟสระดับล่าง (Low Level Interface) ดังนั้น ODBC API ทำให้แอปพลิเคชันสามารถกระทำกับฐานข้อมูลในระดับล่าง ซึ่ง ODBC นั้นได้รับความยอมรับให้เป็นมาตรฐานในการอินเตอร์เฟสกับฐานข้อมูลแบบสัมพันธ์

### 3.1.3.2 OLE DB (Object-Linking and Embedding Database)



รูปที่ 3-6 แสดงสถาปัตยกรรมของ OLE DB

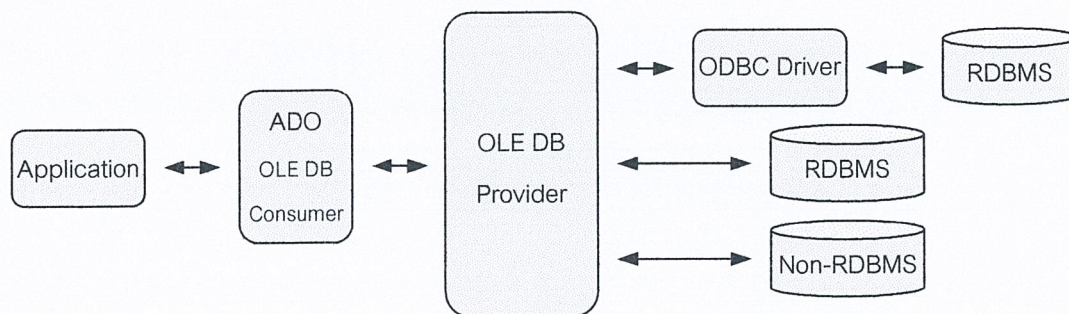
OLE DB เป็นวิธีการเข้าถึงข้อมูลชนิดใหม่จาก Microsoft และเป็นเซตของ COM (Component Object Model) ซึ่งประกอบด้วยชุดของการอินเทอร์เฟสที่เป็นมาตรฐาน OLE DB ถูกออกแบบมาเพื่อเข้าถึงข้อมูลเดต้าซอสได้หลากหลายชนิด โดยไม่คำนึงถึงฐานข้อมูลว่าจะสัมพันธ์กันหรือไม่ ความยืดหยุ่นนี้ทำให้ผู้พัฒนาใช้ลักษณะเด่นของ OLE DB ได้อย่างมีประสิทธิภาพ OLE DB จะแบ่งเป็น Consumer และ Provider

จากรูปแสดงสถาปัตยกรรมของ OLE DB ด้าน Consumer นี้ แสดงถึงผู้ใช้ที่ต้องการสร้างแอปพลิเคชัน ส่วน Provider จะเป็นการจัดวางข้อมูลให้อยู่ในรูปแบบตารางและส่งไปยังผู้ใช้ ทำให้ผู้ใช้สามารถดึงข้อมูลจากฐานข้อมูลเพื่อกระทำกับข้อมูลเหล่านั้น เช่น การ update ข้อมูล, การลบข้อมูล เป็นต้น

- **OLE DB Consumer:** สามารถเข้าถึงข้อมูลจาก Provider ใดๆ ก็ได้ เนื่องจาก Provider เป็นออบเจ็กต์ COM ดังนั้น Consumer สามารถเข้าถึงข้อมูลโดยใช้ภาษาใดๆ ก็ได้ เช่น ภาษา C++, ภาษา Java เป็นต้น

- **OLE DB Provider:** จะจัดวางข้อมูลต่างๆ ให้อยู่ในรูปแบบตาราง ทำให้ Consumer สามารถอ่านข้อมูลรูปแบบนั้นเป็นแบบเรคคอร์ดเซต หรือกระทำกับข้อมูลนั้น ฉะนั้น Provider สามารถเป็นได้ทั้งแบบง่ายๆ หรือแบบซับซ้อน โดยขึ้นอยู่กับข้อมูลนั้นๆ

### 3.1.3.3 ADO (ActiveX Data Object)



รูปที่ 3-7 แสดงสถาปัตยกรรมของ ADO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาใดๆ ของเอกสารนี้ โดยส่งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADO เป็นการอินเทอร์เฟซข้อมูลในระดับสูง (High-Level Interface) ADO มีประสิทธิภาพสูง และมีความเชื่อถือได้ในการเข้าถึงข้อมูล ADO ถูกออกแบบมาให้ง่ายต่อการอินเทอร์เฟซระดับแอปพลิเคชันกับ OLE DB ที่มีประสิทธิภาพสูง ไม่ว่าจะเป็นฐานข้อมูลแบบสัมพันธ์กัน หรือแบบไม่สัมพันธ์กัน เช่น E-mail, File Systems, Text, Graphic และอื่นๆ

ADO ใช้การอินเทอร์เฟซแบบ Dual (Dual Interface) (ซึ่งเป็นผลจากการรวมคุณลักษณะเด่นของ Remote Data Object และ DAO ไว้ด้วยกัน) อินเทอร์เฟซนี้ทำให้ ADO สามารถใช้ภาษาสคริปต์ได้ และ ADO ยังมีคุณลักษณะเด่นในการทำ Query และการจัดการข้อมูล เช่น การใช้คำสั่ง SQL Statement นอกจากนี้แล้ว ADO ยังได้ให้การ Query แบบ Asynchronous ที่อนุญาตให้เราควบคุมการ Query ที่จะถูกส่งกลับมายังแอปพลิเคชันก่อนที่ Query จะทำสำเร็จ การทำอย่างนี้อนุญาตให้เราให้สิทธิแก่ผู้ใช้สามารถยกเลิก หรือกระทำต่อการ Query ที่ใช้เวลานาน เป็นต้น

### 3.2 ข้อสรุปของการติดต่อกับฐานข้อมูลด้วยเทคโนโลยีการอินเทอร์เฟซต่างๆ

**OLE DB** มีประสิทธิภาพดีที่สุดในเมื่อเทียบกับการอินเทอร์เฟซแบบอื่นๆ และใช้เวลาในการเข้าถึงข้อมูลน้อยกว่า ADO สำหรับการเขียนโปรแกรมแล้ว ภาษา Visual C++ จะเหมาะสมที่สุด แต่อย่างไรก็ตาม การเขียนโค้ดเพื่อติดต่อกับฐานข้อมูล ทั้งแบบสัมพันธ์กัน และแบบที่ไม่สัมพันธ์กันจะมีความซับซ้อนพอสมควร

ADO มีประสิทธิภาพดีมากในการเขียนโค้ดเพื่อติดต่อกับฐานข้อมูล ทั้งแบบสัมพันธ์กันและแบบที่ไม่สัมพันธ์กัน ในการเขียนโค้ดสามารถใช้โปรแกรมภาษา Visual Basic หรือภาษาสคริปต์ได้ เช่น VBScript และ JavaScript แต่ไม่เหมาะสมกับภาษา Visual C++ นอกจากนี้ ADO ยังมีฟังก์ชันการใช้งานมากเมื่อเปรียบเทียบกับฟังก์ชันที่อินเทอร์เฟซอื่นมี

ทั้ง OLE DB และ ADO มีประสิทธิภาพที่ดีกว่าการใช้เทคโนโลยีแบบอื่นๆ ด้วยเหตุนี้ทางผู้ผลิตเทคโนโลยีทั้งสองนี้ จึงได้วางไว้เป็นมาตรฐานในการเข้าถึงข้อมูลได้ทุกชนิด และข้าม Platform ได้

**DAO** มีประสิทธิภาพดีสำหรับการใช้กับฐานข้อมูล Access การเขียนโปรแกรมง่าย และใช้ได้กับหลายๆ ภาษา

**RDO** ใช้ในการติดต่อกับฐานข้อมูลที่ตั้งอยู่ Local และตั้งอยู่ระยะไกล เหมาะสำหรับฐานข้อมูล SQL Server โดยที่ RDO นี้ถูกออกแบบมาสำหรับการใช้ภาษา Visual Basic โดยเฉพาะ

**ODBC** มีประสิทธิภาพในการติดต่อกับฐานข้อมูลที่สัมพันธ์กันได้ดี ODBC เป็นการอินเทอร์เฟซในระดับล่าง จึงทำให้ควบคุมข้อมูลกับฐานข้อมูลในระดับล่างได้ดี เมื่อเปรียบเทียบกับ ADO ที่เป็นการอินเทอร์เฟซในระดับสูง

## บทที่ 4

### MFC (Microsoft Foundation Class)

#### 4.1 ประวัติของ MFC (Microsoft Foundation Class)

ในช่วงปลายทศวรรษ 1980 ภาษา BASIC แอสแซมบลี และ PASCAL เป็นภาษาที่นิยมใช้กันทั่วไป สำหรับการพัฒนาซอฟต์แวร์บนดอส (DOS: Disk Operation System) ช่วงนี้ภาษา C ก็ได้เกิดขึ้นโดย Dennis Ritchie ที่ Bell Lab และเริ่มต้นใช้ในระบบปฏิบัติการยูนิกซ์ และมีประสิทธิภาพสูงมาก ทางไมโครซอฟท์ได้เห็นความสำคัญนี้จึงได้เลือกใช้กับภาษา C (บวกกับภาษาแอสแซมบลี) มาพัฒนาระบบปฏิบัติการ Windows

เมื่อปี 1987 ไมโครซอฟท์ได้มีการแนะนำโปรแกรมเมอร์ และนักพัฒนาซอฟต์แวร์ที่ใช้ระบบปฏิบัติการบนดอส ให้มารู้วิธีการเขียนโปรแกรมบนระบบปฏิบัติการ Windows ด้วยการให้ Windows API (Application Programming Interface) สำหรับเครื่องพีซี ซึ่งมีการเพิ่มฟังก์ชันใหม่จำนวนมาก

Windows API รุ่นแรก (เป็นแบบ 16 บิต) ได้ให้มีการใช้สภาพแวดล้อม สภาพแวดล้อมจะเป็น Windows SDK (Software Development Kit) รวมทั้งได้มีคอมไพเลอร์ C มาด้วย นักพัฒนาโปรแกรมได้มีการเริ่มต้นทำแอปพลิเคชันแบบ GUI (Graphic User Interface) ซึ่งเป็นการนำเอาข้อดีของ API มาใช้

โดยทั่วไปแล้วภาษาคอมพิวเตอร์ประกอบด้วย 2 แนวความคิดคือ ข้อมูล (Data) และอัลกอริทึม (Algorithm) ข้อมูลประกอบด้วยข่าวสารต่างๆ ที่โปรแกรมใช้และทำกระบวนการ หรือ โพรเซส อัลกอริทึมเป็นวิธีการ (Method) ที่โปรแกรมใช้ภาษา C เป็นภาษา Procedure ดังนั้นมันจึงเน้นหนักไปทางด้านอัลกอริทึมมากกว่า ส่วนทางด้านข้อมูลนั้น ภาษา C ทำได้ไม่ดี เมื่อนำมาใช้ในการพัฒนาโปรแกรมขนาดใหญ่ ด้วยข้อจำกัดนี้จึงได้มีการพัฒนาภาษา C ใหม่โดย Bjarne Stroustrup ที่ Bell Lab ภาษาใหม่นี้คือ C++ ซึ่งเป็นภาษาเชิงวัตถุ (Object Oriented Language) โดยการนำเอาภาษา C บวกกับการโปรแกรมเชิงวัตถุ (OOP: Object Oriented Programming)

คุณสมบัติของ OOP มีการเน้นไปทางด้านข้อมูลมากกว่าที่จะเน้นทางด้านภาษาแบบ Procedure ดังนั้นภาษาแบบ OOP จะไม่ค่อยมีการเขียน Flow Chart แสดงการทำงานกันนัก ยกเว้นส่วน Algorithm ที่ยังใช้กันอยู่ เมื่อภาษา C++ ได้รับความนิยมมากขึ้น คุณสมบัติของ OOP ได้รวมกับการอินเตอร์เฟสกับ Windows API ท้ายสุดทางไมโครซอฟท์ได้พัฒนาการอินเตอร์เฟสนี้กลายเป็นผลิตภัณฑ์ AFX (Application Framework) และการพัฒนานี้นำไปสู่ผลิตภัณฑ์ MFC ซึ่งปรากฏอยู่ในทุกวันนี้

ใน MFC จะประกอบด้วยไลบรารี MFC ซึ่งเป็นการรวมคลาสของ C++ และ AFX จำนวนมากมาย ที่ถูกออกแบบมาเพื่อสร้างแอปพลิเคชันบนแพลตฟอร์มวินโดวส์ การรวมคลาสต่างๆ นี้ เพื่อขยายขอบเขตภาษา C++ ออกไป โดยการรวมส่วนโครงสร้างพื้นฐานต่างๆ ไว้เป็นหลัก ซึ่งส่วนโครงสร้างพื้นฐานนี้จะถูกนำมาใช้สร้างแอปพลิเคชัน MFC จะแบ่งคลาสออกเป็น 2 กลุ่มใหญ่ คือ

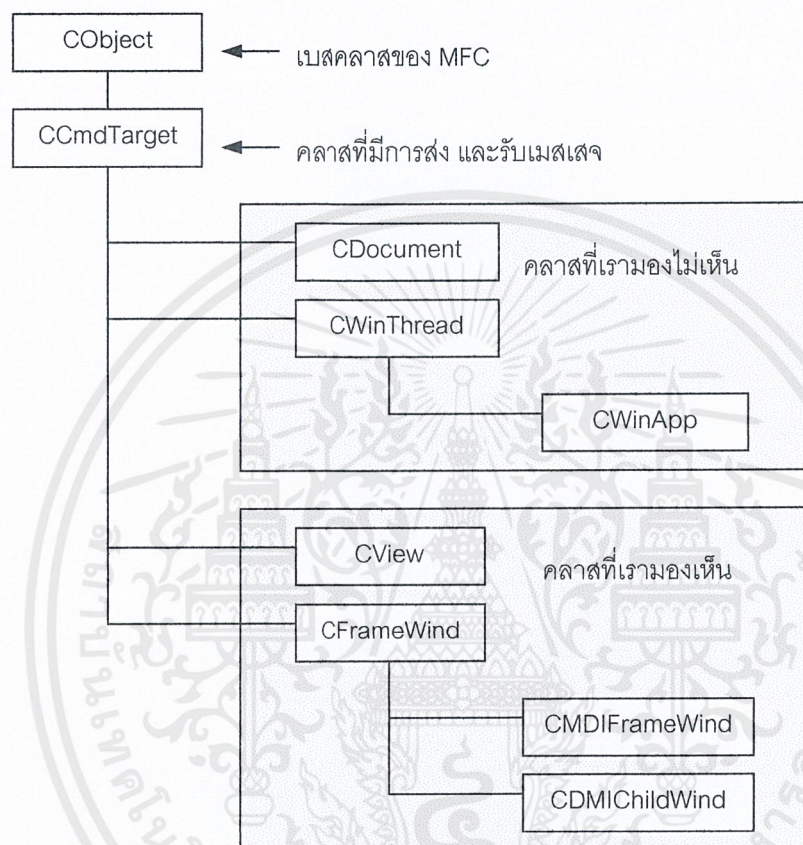
1. คลาสที่มีต้นกำเนิดมาจากคลาสแม่ คือ CObject
2. คลาสที่เป็นคลาสโสด คือ คลาสที่ถูกสร้างขึ้นมาโดยไม่มีการสืบทอดมาจากคลาสใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน MFC จะมีคลาสชื่อ CObject เป็นคลาสแม่ โดยคลาสที่ใช้ในการเขียนโปรแกรมในระบบ เช่น การสร้างวินโดวส์ การสร้างคอนโทรล จะสืบทอดมาจากคลาส CObject นี้ทั้งนั้น

## 4.2 สถาปัตยกรรมของ MFC

คลาสและหน้าที่ของแต่ละคลาส อธิบายได้ดังนี้



รูปที่ 4-1 แสดงลำดับชั้นคลาสของสถาปัตยกรรม MFC

### 4.2.1 คลาส CObject

คลาส CObject เป็นคลาสแม่ของทุกๆ คลาสหรืออาจจะเรียกว่า เบสคลาส (Base Class) ซึ่งมีคลาสต่างๆ ที่สืบทอดจากคลาสนี้มากกว่า 200 คลาส และมีอีก 30 กว่าคลาสที่ไม่ได้สืบทอดมาจากคลาส CObject

คลาส MFC เกือบทั้งหมด (รวมทั้งที่เราสร้างขึ้นใหม่ด้วย) จะต้องสนับสนุนการจัดการกับ Runtime type information และจัดการกับการ Persistence ออบเจกต์ หรือที่พูดกันว่าการทำซีเรียลไลเซชัน (Serialization) และกระทำเอาต์พุตที่วิเคราะห์ไว้สำหรับออบเจกต์ที่สืบทอดมา นอกจากนี้คลาสที่สืบทอดจาก CObject สามารถบรรจุอยู่ในคลาสคอนเทรนเนอร์ MFC ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

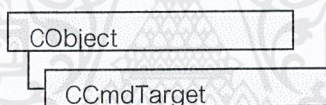
คลาสทั้งหลายที่สืบทอดมาจากคลาส CObject จะต้องมีคุณลักษณะอย่างน้อยอย่างหนึ่งในสามอย่างนี้

1. การสนับสนุนการวิเคราะห์ทั่วไป: คลาสที่สืบทอดมาจะต้องมีคุณลักษณะนี้เป็นส่วนใหญ่ ซึ่งถูกสร้างอยู่ภายในคลาส CObject ที่อนุญาตให้เรากระทำการตรวจสอบออบเจกต์ที่สืบทอดจาก CObject ใช้ได้หรือไม่ และการใส่ข้อมูลการตรวจสอบจากอินพุทที่อ่านได้ และเอาต์พุทที่แสดงในหน้าต่างเอาต์พุทของ Debugger ได้หรือไม่
2. การสนับสนุน Runtime type information (RTTI): เมื่อคลาส MFC สนับสนุน RTTI มันสามารถดึงชื่อคลาสของออบเจกต์ และข้อมูลอื่นๆ ที่เกี่ยวกับออบเจกต์ ณ เวลาทำงาน (Run Time)

3. การสนับสนุน Serialization: การ Serialization เป็นกระบวนการของการไหล (Stream) ข้อมูลที่คงอยู่เรื่อยๆ ของออบเจกต์ไปยังฮาร์ดดิสก์ (การบันทึก) หรือจากฮาร์ดดิสก์ออกไป (การอ่าน)

นอกจากนี้แล้วคลาส CObject ได้ให้ประโยชน์อื่นๆ กับคลาสที่สืบทอดมาจากคลาส CObject เช่น โอเวอร์โหลดโอเปอเรเตอร์ new และ delete เพื่อป้องกันเมมโมรี่รั่วไหล ถ้าเราสร้างออบเจกต์ด้วยการสืบทอดจาก CObject และใช้โอเปอเรเตอร์ new แล้วลืมใช้โอเปอเรเตอร์ delete เมื่อปิดโปรแกรม MFC จะมีข้อความเตือนในหน้าต่างเอาต์พุทของ Debug

#### 4.2.2 คลาส CCmdTarget



รูปที่ 4-2 แสดงการสืบทอดของคลาส CCmdTarget

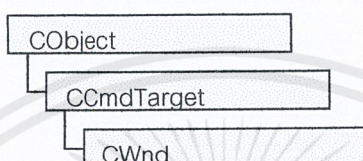
คลาส CCmdTarget เป็นคลาสที่สืบทอดมาจาก CObject ใช้สำหรับจัดการกับกระบวนการแมสเสจแมป มีหน้าที่รับผิดชอบเส้นทางของระบบ และอีเวนตวินโดวส์เพื่อไปยังออบเจกต์ต่างๆ ที่สามารถตอบสนองต่ออีเวนตเหล่านี้ คลาสที่รับระบบและอีเวนตวินโดวส์จะประกอบด้วยคลาสดังนี้

1. คลาส CWinApp ใช้สำหรับจัดการแอปพลิเคชัน และยังแสดงเซรด์ (Thread) หลักของโปรแกรม เมื่อมันทำงานด้วย ผลลัพธ์ที่ได้ก็คือมีเพียงออบเจกต์เดียวในแอปพลิเคชัน MFC ใดๆ กระบวนการเริ่มต้นการรัน โปรแกรม, Instance ของโปรแกรม
2. คลาส CWnd เป็นคลาสแม่ของคลาสวินโดวส์ทั้งหมด รับผิดชอบการสร้างวินโดวส์ ตลอดจนการควบคุมการทำงานของคอนโทรล สามารถรับและจัดการระบบ และอีเวนตวินโดวส์ วินโดวส์ส่วนใหญ่แล้วจะมีการแสดงกราฟิก และตอบสนองกับอินพุทผู้ใช้
3. คลาส CFrameWnd ใช้สำหรับสร้างวินโดวส์แบบเฟรม (สืบทอดมาจากคลาส CWnd อีกที) เป็นวินโดวส์ที่ประกอบด้วย Title bar เมนู system border บอกร์ minimize และ maximize ทูลบาร์ สเตตัสบาร์ และวินโดวส์วิวที่กำลังทำงานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คลาส CView และ CDocument ซึ่งคลาส CView ใช้ในการจัดการกับ View ในโปรแกรม โดยได้ให้ฟังก์ชันพื้นฐานสำหรับคลาสวิวที่ผู้ใช้กำหนด วิวที่ติดกับด็อกคิวเมนต์ และกระทำเหมือนกับสื่อกลางระหว่างด็อกคิวเมนต์และผู้ใช้ วิววาดภาพของด็อกคิวเมนต์บนหน้าจอ หรือเครื่องพิมพ์ และตีความหมายจากอินพุตผู้ใช้ให้ปรากฏบนด็อกคิวเมนต์ สำหรับคลาส CDocument ใช้ในการจัดการกับเอกสารได้ให้ฟังก์ชันพื้นฐานสำหรับคลาสด็อกคิวเมนต์ที่ผู้ใช้กำหนด ด็อกคิวเมนต์แสดงหน่วยของข้อมูลที่โดยปกติผู้ใช้เปิดด้วยคำสั่งไฟล์ Open และบันทึกด้วยคำสั่งไฟล์ Save

#### 4.2.3 คลาส CWnd



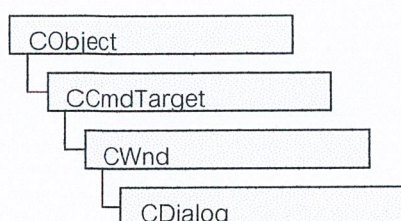
รูปที่ 4-3 แสดงการสืบทอดของคลาส CWnd

คลาส CWnd เป็นคลาสหนึ่งที่เราจะต้องให้ความสนใจกับมันให้มาก เพราะมีความสำคัญมากในการเขียนโปรแกรม คลาสนี้จะเป็นคลาสที่ควบคุมการทำงานของวินโดวส์ และคอนโทรลทั้งหมดในโปรแกรม เป็นคลาสแม่ให้กับคลาสต่อไปนี้

1. คลาสวินโดวส์ CFrameWnd ใช้ในการควบคุมการทำงาน และการแสดงผลของวินโดวส์
2. คลาสวินโดวส์ CSplitterWnd ใช้ในการแสดงผลวินโดวส์แบบแบ่งเฟรม (Splitter Window)
3. คลาสคอนโทรลบาร์ CControlBar ใช้จัดการกับทูลบาร์, ไดอะล็อกบาร์, สเตตัสบาร์ ซึ่งเป็นส่วนหนึ่งของวินโดวส์ในโปรแกรม
4. คลาส CPropertySheet ใช้ในการแสดงวินโดวส์แบบ Property (วินโดวส์ที่มีแท็บซ้อนๆ กัน)
5. คลาส CDialog ใช้ในการควบคุมการแสดงผลของไดอะล็อก ซึ่งคลาส CDialog นี้ยังสืบทอดไปเป็นคลาสย่อยๆ อีกหลายคลาส เช่น CCommonDialog เป็นต้น
6. คลาส CView ใช้สำหรับแสดงผล View ในโปรแกรม เช่น View ของเอกสาร หรือ View ในการวาดภาพของโปรแกรม
7. คลาสคอนโทรล ซึ่งมีอยู่มากมายหลายตัว เช่น CEdit, CComboBox เป็นต้น

จากรายชื่อคลาสในข้างต้น ล้วนเป็นคลาสลูกที่สืบทอดมาจาก CWnd ทั้งสิ้น ซึ่งจะสังเกตเห็นได้อย่างหนึ่งว่า หน้าที่ของแต่ละคลาสล้วนเกี่ยวข้องกับวินโดวส์ และคอนโทรลทั้งสิ้น

#### 4.2.4 คลาส CDialog



รูปที่ 4-4 แสดงการสืบทอดของคลาส CDialog

คลาส CDialog เป็นคลาสที่สืบทอดมาจาก CWnd ซึ่งเป็นคลาสที่จะรับผิดชอบในการแสดงผล และควบคุมไดอะล็อก ไม่ว่าจะเป็นแบบ Modal หรือ Modelless, การกดปุ่ม OK กับ Cancel และการทำงานของฟังก์ชัน OnInitDialog

จากคลาส CDialog นี้ ยังได้มีการสืบทอดไปเป็นคลาสลูกอีก นั่นคือ CCommonDialog ซึ่งเป็นคลาสของคอมมอนไดอะล็อก (Common Dialog) ใช้ในการสร้างไดอะล็อกเปิดไฟล์, บันทึกไฟล์ หรือไดอะล็อกแมสเสจบ็อกซ์

หากเราดูรายละเอียดของคลาส CDialog นี้ เราจะพบกับเมมเบอร์ฟังก์ชันของคลาส CDialog นี้ ที่น่าสนใจมีอยู่ 3 ฟังก์ชันด้วยกัน คือ

- OnOK()
- OnCancel()
- OnInitDialog()

ทั้ง 3 ฟังก์ชันในข้างต้นนี้ เป็นฟังก์ชันที่มีการประกาศไว้แบบ Overridables คือ เป็นฟังก์ชันที่เราสามารถเขียนด้วยฟังก์ชันนี้ลงในคลาสของเราได้ และเราก็สามารถเรียกฟังก์ชันจากคลาสแม่ได้โดยตรงอีกด้วย

#### 4.2.5 คลาสคอนโทรล

คลาสคอนโทรล เป็นคลาสที่สืบทอดมาจากคลาส CWnd ใช้ในการจัดการกับคอนโทรลต่างๆ ซึ่งมีอยู่ด้วยกัน 24 คอนโทรล เช่น CAnimationCtrl, CButton, CComboBox, CDateTimeCtrl, CEdit, CHeaderCtrl, CHotkeyCtrl เป็นต้น

#### 4.2.6 คลาสอะเรย์และคลาสลิสต์

คลาสอะเรย์ (Array Classes) และคลาสลิสต์ (List Classes) เป็นกลุ่มของคลาสที่ใช้ในการจัดการกับข้อมูลแบบอะเรย์ ซึ่งสืบทอดมาจากคลาส CObject เราสามารถสร้างออบเจกต์ของคลาสอะเรย์ เพื่อจัดการกับข้อมูลที่มีหลายๆ ตัวได้ แทนที่จะใช้คลาสอะเรย์ของ MFC จะมีความแน่นอนกว่ามาก

คลาสอะเรย์ของ MFC มีดังนี้ CArray, CByteArray, CDWordArray, CObArray, CPtrArray, CStringArray, CUIntArray, CWordArray นอกจากนี้ MFC ยังได้เตรียมคลาสที่ใช้ในการจัดการลิงคัลลิสต์เอาไว้ให้กับเราด้วย มีอยู่ด้วยกัน 4 คลาส คือ CList, CPtrList, CObList, CStringList

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสลิสต์และคลาสอะเรย์ เราจะใช้ในการจัดการข้อมูลแบบ โครงสร้างเสียเป็นส่วนมาก

#### 4.2.7 คลาสที่ใช้ในการวาดภาพ

การวาดภาพในวินโดวส์ เราจะใช้สิ่งที่เรียกว่า ดีไวซ์คอนเท็กซ์ (Device Context) ซึ่งเราสามารถใช้งานวาดภาพภายในไดอะล็อกได้ โดยเขียนโค้ดที่ใช้ในการวาดภาพไว้ในฟังก์ชัน OnPaint หรือเราสามารถวาดภาพลงใน View ซึ่งเป็นวินโดวส์แบบเฟรมได้

คลาสที่ใช้ในการรองรับการวาดภาพก็คือ คลาส CDC ซึ่งเป็นคลาสที่ทำหน้าที่เหมือนกับพื้นที่ในการวาดภาพ นอกจากนี้ยังมีคลาสอีกกลุ่มหนึ่งที่จะต้องใช้ร่วมกัน คือ คลาสอุปกรณ์ที่ใช้ในการวาดภาพ เช่น คลาสพู่กัน, คลาสแปรง คลาสนี้ก็คือคลาส CGdiObject เป็นคลาสแม่ของคลาสอุปกรณ์ในการวาดภาพต่างๆ ดังนี้ CBitmap, CBrush, CFont, CPalette, CPen, CRgn

#### 4.2.8 คลาสที่ใช้สำหรับอินเทอร์เน็ต

MFC ได้พัฒนาคลาสขึ้นมากลุ่มหนึ่ง ที่ใช้ในการจัดการกับบริการในอินเทอร์เน็ต เราเรียกคลาสเหล่านี้ว่า MFC WinInet Class ประกอบด้วยคลาสที่ใช้สำหรับจัดการบริการในอินเทอร์เน็ต 3 บริการคือ WWW, Ftp และ Gopher ดังนี้ CInternetSession, CInternetConnection, CFileFind, CGopherLocator

นอกจากคลาส WinInet ยังมีคลาสที่ชื่อ CAsyncSocket และ CSocket ซึ่งเป็นคลาส Socket สำหรับการติดต่อสื่อสารผ่าน Socket ซึ่งเป็นพื้นฐานแห่งการสื่อสารในระบบวินโดวส์ทั้งหมด

#### 4.2.9 คลาสที่ไม่อยู่ในตระกูล CObject

คลาสเหล่านี้เป็นคลาสโคด ส่วนมากจะไม่มีคลาสแม่ เป็นคลาสที่สร้างขึ้นมาจัดการกับข้อมูล หรือใช้งานในบางหน้าที่เท่านั้น

คลาสที่ไม่อยู่ในตระกูล CObject แบ่งออกเป็น 9 หมวดใหญ่ๆ ที่เราจะต้องใช้บ่อยๆ ก็คือ คลาส Simple Value Types ซึ่งเป็นคลาสที่ใช้สำหรับกำหนดชนิดของข้อมูล

### 4.3 ประโยชน์จากการใช้คลาส MFC

คลาส MFC ได้ให้ข้อดีหลายๆ อย่างในการพัฒนาแอปพลิเคชันบนวินโดวส์ ถึงแม้ว่าคลาส MFC ไม่ได้ห่อหุ้มฟังก์ชัน Win32API ทั้งหมด แต่มันก็ได้รวบรวมส่วนที่เป็นโครงสร้างเฟรมเวิร์คหลักให้มีผล ให้การพัฒนาแอปพลิเคชันต่างๆ บนวินโดวส์ง่ายขึ้นไปมาก เช่น การสร้าง ActiveX Control และ Component การสร้างแอปพลิเคชันอินเทอร์เน็ต เป็นต้น เราสามารถสรุปข้อดีได้ดังต่อไปนี้

1. ความสามารถเข้ากันได้ดีระหว่างคลาส MFC ในเวอร์ชันที่แล้วๆ กับคลาส C++ ใหม่
2. เฟรมเวิร์ค MFC ใช้เทคนิคของ OOP มีผลให้โค้ดสามารถนำกลับมาใช้ใหม่ได้
3. การสร้างแอปพลิเคชันเฟรมเวิร์คหลักให้ ซึ่งมีผลให้เราลดการเขียนโค้ดลงไปมาก เมื่อเทียบกับการเขียนโดยใช้ SDK
4. เมื่อใช้คลาส MFC แล้ว ตัวคอมไพเลอร์ยังคงสนับสนุนไลบรารีมาตรฐาน ANSI C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนข้อเสียของการใช้ MFC เมื่อเทียบกับการใช้ SDK คือ ใช้โค้ดจำนวนมากกว่า เนื่องจากโค้ด MFC เมื่อคอมไพล์ จะมีการแปลงโค้ด MFC อีกครั้งหนึ่ง และมีผลให้การทำงานของโปรแกรมช้ากว่าเล็กน้อย อย่างไรก็ตามมันยังเร็วกว่าภาษาระดับสูง เช่น Pascal และ Visual Basic เป็นต้น

#### 4.4 การใช้งาน MFC ในโปรแกรม

ในโปรแกรมหนึ่งๆ บนวินโดวส์ ถ้าเราใช้ MFC เป็นไลบรารีหลักในการเขียนโปรแกรมดังเช่นที่เราเขียนมานั้น หลักการเขียนโปรแกรมโดยใช้ MFC ไม่ว่าจะเขียนโปรแกรมแบบใด เช่น Dialog-Based หรือแบบ SDI จะมีกฎที่คล้ายๆ กันดังนี้

1. จะต้องมียุคคลาสหนึ่งทีสืบทอดมาจากคลาส CWinApp เพราะคลาส CWinApp จะใช้ในขั้นตอนการรันโปรแกรม ถ้าไม่มีคลาสนี้ โปรแกรมจะไม่สามารถทำงานได้
2. จะมีคลาสหนึ่งเป็นคลาสของวินโดวส์ ซึ่งจะทำงานร่วมกับคลาส CWinApp เช่น คลาส CDialog, CFrameWnd เป็นต้น ซึ่งคลาสเหล่านี้ทำหน้าที่เป็นวินโดวส์หลักของโปรแกรมซึ่งทุกโปรแกรมจะต้องมี
3. ภายในโปรแกรมจะไม่มีฟังก์ชัน main() การเริ่มต้นของโปรแกรมอยู่ที่ฟังก์ชัน InitInstance() ของคลาส CWinApp จากนั้นการทำงานทั้งหมดจะอยู่ที่กระบวนการเมสเสจแมป หรือเมสเสจลูปของวินโดวส์คลาสต่อไป

#### 4.5 MFC ที่ใช้ในโปรแกรม

คลาสของ MFC ที่ใช้ในการพัฒนาโปรแกรมพจนานุกรมมีดังนี้

##### 4.5.1 คลาสต่างๆ ของ DAO

คลาสต่างๆ ของ DAO ที่จะกล่าวถึงมีดังนี้

##### 4.5.1.1 คลาส CDaoWorkspace

คลาสนี้ถูกใช้เพื่อเข้าถึงฟังก์ชันในค่าเบสเอ็นจิน ด้วยคุณสมบัติที่เด่นในการทำทรานแซคชัน ทำให้คลาสนี้นำมาใช้ในการทำทรานแซคชันเป็นส่วนใหญ่ นอกจากนี้แล้วการทำทรานแซคชัน สามารถเป็นอิสระจากทรานแซคชันอื่นๆ ที่เกิดขึ้นพร้อมกันได้

ฟังก์ชันต่างๆ ของคลาส CDaoWorkspace ที่สำคัญ สามารถสรุปได้ดังตารางต่อไปนี้

ชื่อฟังก์ชัน	หน้าที่
BeginTrans	เริ่มต้นทำทรานแซคชัน
CDaoWorkspace	สร้างออบเจกต์ Workspace หลังจากใช้คำสั่ง Create หรือ Open
Close	ปิด Workspace และออบเจกต์ทั้งหมดที่เปิดอยู่ การทรานแซคชันที่ค้างอยู่จะถูกโรลแบ็ค (RollBack)

ตารางที่ 4-1 แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CDaoWorkspace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟังก์ชัน	หน้าที่
CommitTrans	ทำการทรานแซคชันปัจจุบัน และบันทึกการเปลี่ยนแปลง
Create	สร้างออบเจกต์ DAOWorkspace ใหม่
Open	เปิดออบเจกต์ Workspace ที่ร่วมกับ Workspace ดิฟเฟอเรนเชียลของ DAO ภายหลังจากออบเจกต์ CDaoWorkapce ได้ถูกสร้างขึ้น
RollBack	ทำให้การทรานแซคชันปัจจุบันสิ้นสุดลง และไม่มีการบันทึกการเปลี่ยนแปลงข้อมูล

ตารางที่ 4-1 (ต่อ) แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CDaoWorkspace

RollBack หมายถึง การแก้ไขข้อมูลให้กลับมาเหมือนเดิม เมื่อมีการเปลี่ยนแปลงที่เกิดขึ้นระหว่างการทรานแซคชันล้มเหลว โดยที่ RollBack จะส่งค่าเรคคอร์ดกลับมาในสถานะก่อนที่การทรานแซคชันล้มเหลว

#### 4.5.1.2 คลาส CDaoDatabase

เป็นคลาสหนึ่งของ MFC ที่ถูกใช้สำหรับการเข้าถึงฐานข้อมูล Access เป็นส่วนใหญ่ แต่คลาส CDaoDatabase ไม่ได้รวมฟังก์ชันทรานแซคชันที่เหมือนกับคลาส CDaoWorkspace ไว้ คลาส CDaoDatabase มีฟังก์ชันต่างๆ ดังนี้

ชื่อฟังก์ชัน	หน้าที่
CanTransact	ตรวจสอบว่าฐานข้อมูลสามารถทำการทรานแซคชันได้หรือไม่ ถ้าไม่ได้จะให้ค่าเป็นศูนย์ หรือ FALSE
CanUpdate	ตรวจสอบว่าสามารถอัปเดตข้อมูลได้หรือไม่ ถ้าไม่ได้จะให้ค่าเป็นศูนย์
CDaoDatabase	เป็นคอนสตรัคเตอร์ (Constructor)
Close	ปิดการติดต่อกับฐานข้อมูล
Create	สร้างออบเจกต์ฐานข้อมูล และเปิดการติดต่อกับฐานข้อมูลใหม่ด้วยออบเจกต์ CDaoDatabase
Execute	รัน Query SQL Statement
IsOpen	ตรวจสอบว่าออบเจกต์ CDaoDatabase ติดต่อกับฐานข้อมูลหรือไม่ ถ้าไม่ได้จะให้ค่าเป็นศูนย์
Open	เปิดการติดต่อกับฐานข้อมูล
m_pWorkspace	เป็นพอยน์เตอร์ชี้ไปยังออบเจกต์ CDaoWorkspace ซึ่งประกอบด้วยออบเจกต์ฐานข้อมูล และกำหนดการทรานแซคชัน
m_pDAODatabase	เป็นพอยน์เตอร์ชี้ไปยังออบเจกต์ DAODatabase

ตารางที่ 4-2 แสดงฟังก์ชันดาต้าแมมเบอร์ และหน้าที่ต่างๆ ของคลาส CDaoDatabase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5.1.3 คลาส CDaoRecordset

คลาสนี้แสดงเซตของเรคคอร์ดจากฐานข้อมูล ซึ่งเรียกกันว่า “เรคคอร์ดเซต (Recordset)” MFC AppWizard จะสร้างคลาสเรคคอร์ดนี้ให้กับแอปพลิเคชันโดยอัตโนมัติ คลาส CDaoRecordset มีฟังก์ชันและค่าตัวแปรต่างๆ ในการติดต่อกับฐานข้อมูลดังนี้

ชื่อฟังก์ชัน/ค่าตัวแปร	หน้าที่
AddNew	เตรียมพร้อมสำหรับการเพิ่มเรคคอร์ดใหม่ และยังไม่เพิ่มเรคคอร์ดใหม่จนกว่าจะเรียกฟังก์ชัน Update
CanAppend	ส่งค่ากลับไม่เป็นศูนย์ถ้าประสบความสำเร็จในการเพิ่มเรคคอร์ดจากการใช้ฟังก์ชัน AddNew มิฉะนั้นจะส่งค่ากลับเป็นศูนย์
CanUpdate	ส่งค่ากลับไม่เป็นศูนย์ถ้าเรคคอร์ดเซตสามารถอัปเดตได้ มิฉะนั้นจะส่งค่ากลับเป็นศูนย์
CanBookmark	ตรวจสอบว่าเรคคอร์ดสนับสนุนการทำ Bookmark ได้หรือไม่ ถ้าได้จะไม่เป็นศูนย์
CancelUpdate	ยกเลิกการอัปเดตที่ค้างอยู่ อันเนื่องมาจากการใช้ฟังก์ชัน Edit และ AddNew
CanTransact	ถ้าค่าตัวแปร (Data Source) สนับสนุนการทำทรานแซกชันได้ จะส่งค่ากลับไม่เป็นศูนย์
CDaoRecordset	สร้างออบเจกต์ CDaoRecordset
Close	ปิดเรคคอร์ดเซต
Delete	ลบเรคคอร์ดปัจจุบันจากเรคคอร์ดเซต
Edit	เรียกฟังก์ชันนี้เพื่อเปลี่ยนแปลงเรคคอร์ดปัจจุบัน
GetEditMode	ส่งค่ากลับเป็น TRUE ถ้าตรวจพบเรคคอร์ดปัจจุบันอยู่ในโหมดการแก้ไข
GetDefaultDBName	บรรจุสตริงที่เป็น Path และชื่อฐานข้อมูลสำหรับเรคคอร์ดเซตนี้
GetDefaultSQL	ใช้เพื่อเรียกสตริงที่เป็น SQL Statement
IsBoF	ส่งค่ากลับไม่เป็นศูนย์ ถ้าไม่มีเรคคอร์ดปัจจุบัน หรือเลื่อนเรคคอร์ดถอยหลังก่อนที่จะเลื่อนไปยังเรคคอร์ดแรกสุด
IsEoF	ส่งค่ากลับไม่เป็นศูนย์ ถ้าไม่มีเรคคอร์ดปัจจุบัน ปัจจุบัน หรือเลื่อนเรคคอร์ดไปท้ายสุดแล้วยังเลื่อนไปอีก
IsOpen	จะส่งค่ากลับไม่เป็นศูนย์ ถ้าเรียกฟังก์ชัน Open หรือ Requery ก่อนหน้านี้ หรือเรคคอร์ดเซตเปิดอยู่ก่อนหน้านี้

ตาราง 4-3 แสดงฟังก์ชันค่าตัวแปร และหน้าที่ต่างๆ ของคลาส CDaoRecordset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟังก์ชัน/ค่าตัวแปร	หน้าที่
Move	เลื่อนเรคคอร์ดโดยการผ่านพารามิเตอร์ ถ้าพารามิเตอร์เป็นศูนย์ จะไม่เลื่อน ถ้าเป็นบวกเลื่อนไปข้างหน้า และถ้าเป็นลบจะเลื่อนถอยหลัง
MoveFirst	เลื่อนเรคคอร์ดไปแรกของเรคคอร์ดเซต
MoveLast	เลื่อนเรคคอร์ดไปท้ายสุดของเรคคอร์ดเซต
MoveNext	เลื่อนเรคคอร์ดไปข้างหน้าหนึ่งเรคคอร์ดของเรคคอร์ดเซต
MovePrev	เลื่อนเรคคอร์ดไปข้างหลังหนึ่งเรคคอร์ดของเรคคอร์ดเซต
Open	สร้างเรคคอร์ดใหม่จากตาราง ไลนาเซต หรือ Snapshot
Requery	รันคิวรีของเรคคอร์ดเซตอีกครั้ง เพื่อทำการรีเฟรชเรคคอร์ด
SetFieldNull	กำหนดค่าให้เป็น Null (ไม่มีค่า) ในคอลัมน์ที่ต้องการของเรคคอร์ดเซตปัจจุบัน
m_nParams	เก็บจำนวนค่าตัวแปรพารามิเตอร์ในคลาสเรคคอร์ดเซต (จำนวนพารามิเตอร์ถูกผ่านค่าด้วยการคิวรีของเรคคอร์ดเซต)
m_pDatabase	เก็บค่าพอยน์เตอร์ชี้ไปยังออบเจกต์ CDAODatabase
m_strFilter	เก็บสตริงที่ถูกใช้เพื่อสร้าง WHERE ซึ่งเป็น SQL Statement
m_strSort	บรรจุสตริงที่ถูกใช้เพื่อสร้าง ORDER BY ซึ่งเป็น SQL Statement

ตาราง 4-3 (ต่อ) แสดงฟังก์ชันค่าตัวแปร และหน้าที่ต่างๆ ของคลาส CDAORecordset

#### 4.5.2 Internet Client Application

ในการติดต่อกับ web site ที่ให้บริการค้นหาคำศัพท์นั้น สามารถติดต่อได้โดยการส่ง URL ไปยัง web site นั้น และเก็บข้อมูลที่ส่งกลับมาในรูปแบบ html ไว้ใน temporary file เสียก่อน แล้วค่อยนำไปตัด tag html และส่วนอื่นๆ ให้เหลือเพียงคำแปลของคำศัพท์นั้น แล้วนำไปเก็บในฐานข้อมูลและนำไปแสดงผลให้ผู้ใช้อ่านต่อไป

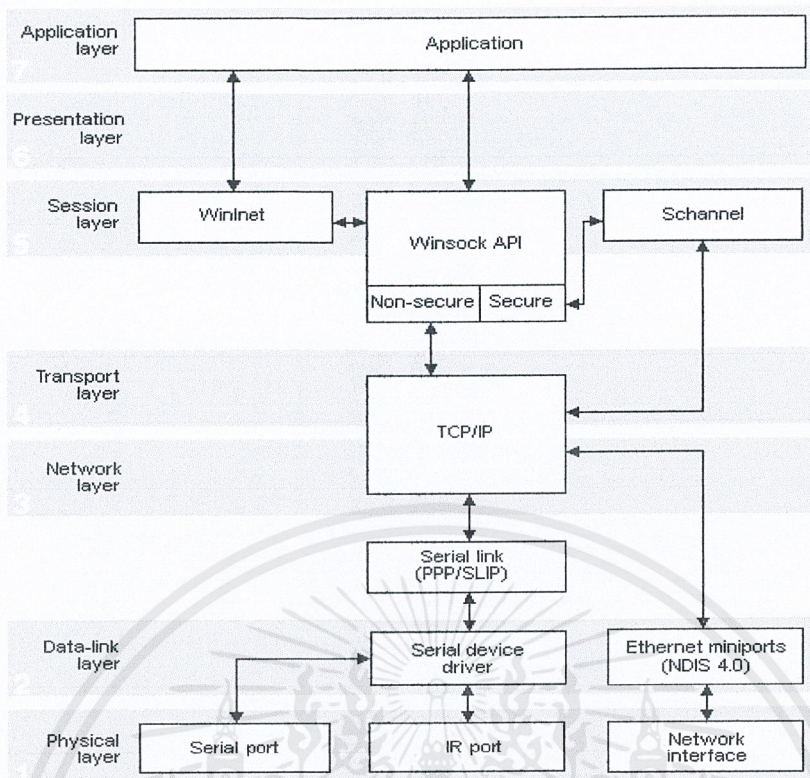
วิธีการติดต่อกับ web site นั้น implement โดยการใช้คลาส WinInet ของ MFC Library ซึ่งสนับสนุนการติดต่อของ Internet Protocol ทั้ง HTTP, FTP และ Gopher โดยที่ไม่จำเป็นต้องเข้าใจถึงการทำงานของ Winsock หรือ TCP/IP ในระดับล่าง โดยจะอธิบายรายละเอียดของคลาส WinInet ต่อไป

##### 4.5.2.1 WinInet Class

WinInet หรือ Win32 Internet Extensions คือ API (Application Programming Interface) ที่ให้แอปพลิเคชันสามารถติดต่อกับอินเทอร์เน็ตได้ โดยการใช้อินเทอร์เน็ตโพรโตคอลทั่วไป เช่น Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP) และ Gopher

WinInet จัดการเกี่ยวกับการโปรแกรม Windows Sockets (Winsock), TCP/IP และอินเทอร์เน็ตโพรโตคอลในชั้น Session Layer ตาม International Organization for Standardization Open Systems Interconnection (ISO/OSI) Model ดังรูปที่ 4-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 แสดงการจัดการเกี่ยวกับการโปรแกรมของ WinInet

การเขียนโปรแกรมด้วยคลาส WinInet สามารถเขียนแอปพลิเคชันฝั่งไคลเอนต์ที่ติดต่อกับอินเทอร์เน็ตได้ในระดับสูงโดยไม่ต้องคำนึงถึง Winsock, TCP/IP หรือรายละเอียดเฉพาะของอินเทอร์เน็ต โพรโตคอล และขนาดของโค้ดก็เล็กกว่าการเขียนโปรแกรมด้วยวิธีอื่นๆอีกด้วย การเขียนโปรแกรมด้วย WinInet มีอยู่ 2 ทางเลือก คือ การเรียกฟังก์ชันของ WinInet โดยตรง และการเรียกใช้ตาม MFC Library

ในที่นี้การเรียกใช้คลาส WinInet ด้วย MFC สามารถทำได้ง่ายกว่าและยังมีข้อดีดังต่อไปนี้

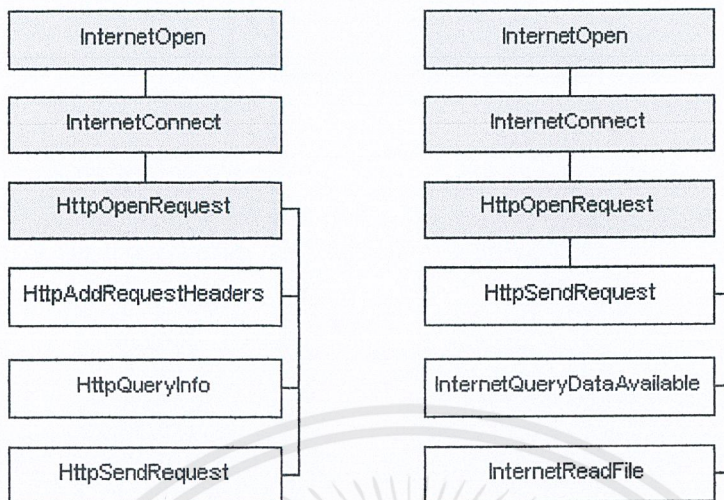
1. Buffered I/O
2. มีการจัดการกับชนิดของข้อมูลให้ (Type-safe handles)
3. กำหนดพารามิเตอร์ (Default parameters) ให้กับหลายๆ ฟังก์ชัน
4. มีการจัดการกับข้อผิดพลาด (Exception handling) ทัวไปทางอินเทอร์เน็ต
5. มีการปิดการจัดการและการติดต่อให้โดยอัตโนมัติ

ซึ่ง MFC WinInet นั้นมีประโยชน์หลายอย่างด้วยกัน ประโยชน์ของ MFC WinInet มีดังนี้

1. สามารถอ่านข้อมูลจากเซิร์ฟเวอร์ไม่ว่าจะเป็น HTTP, FTP หรือ Gopher ได้ง่ายเหมือนอ่านข้อมูลจากฮาร์ดดิสก์
2. สามารถเขียนโปรแกรมในระดับสูงโดยไม่ต้องติดต่อกับ Winsock และ TCP/IP แต่ก็ยังสามารถเขียนโปรแกรมในระดับ socket โดยใช้ Winsock และ TCP/IP ได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.5.2.2 ขั้นตอนการอิมพลิเมนต์ HTTP Client Application



รูปที่ 4-6 แสดงขั้นตอนการร้องขอบริการจาก HTTP Server

รูปแบบพื้นฐานของการร้องขอจากไคลเอนต์ และการตอบรับจากเซิร์ฟเวอร์ในการติดต่อสื่อสารแบบ HTTP ประกอบด้วยขั้นตอนดังต่อไปนี้

1. ไคลเอนต์สร้างการติดต่อแบบ TCP/IP
2. ไคลเอนต์ส่งการร้องขอไปยังเซิร์ฟเวอร์
3. เซิร์ฟเวอร์ส่งการตอบรับมายังไคลเอนต์
4. ไคลเอนต์ปิดการติดต่อแบบ TCP/IP

ขั้นตอน	วิธีการ implement	ผลลัพธ์
เริ่มต้น HTTP เซสชัน	สร้างออบเจกต์ของ CInternetSession	กำหนดค่าเริ่มต้นให้กับ WinInet และติดต่อไปยังเซิร์ฟเวอร์
ติดต่อไปยังเซิร์ฟเวอร์ของ HTTP	ใช้ CInternetSession:: GetHttpConnection	เซิร์ฟเวอร์คืนออบเจกต์ของ CHttpConnection กลับมา
เปิดการร้องขอ HTTP	ใช้ CHttpConnection:: OpenRequest	เซิร์ฟเวอร์คืน CHttpFile กลับมา
ส่งการร้องขอ HTTP	ใช้ CHttpFile:: AddRequestHeaders และ CHttpFile::SendRequest	หาไฟล์ที่ต้องการจากเซิร์ฟเวอร์ และคืนค่า FALSE ถ้าไม่พบไฟล์ที่ต้องการ

ตารางที่ 4-4 แสดงขั้นตอนการอิมพลิเมนต์ HTTP Client Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน	วิธีการ implement	ผลลัพธ์
อ่านจากไฟล์	ใช้ CHttpFile	ทำการอ่านข้อมูลเป็นไบต์ตามที่ได้กำหนดขนาดบัพเฟอร์ไว้
จัดการกับข้อผิดพลาด	ใช้คลาส CInternetException	จัดการกับข้อผิดพลาดทั่วไปของอินเทอร์เน็ต
ปิด HTTP เซสชัน	ทำลายออบเจกต์ของ CInternetSession	ปิดการติดต่อและปิดการจัดการกับไฟล์โดยอัตโนมัติ

ตารางที่ 4-4 (ต่อ) แสดงขั้นตอนการอิมพลิเมนต์ HTTP Client Application

### 4.5.3 MFC WinInet Classes

MFC WinInet Classes เฉพาะที่ใช้โปรแกรมมีดังนี้

#### 4.5.3.1 คลาส CInternetSession

คลาสนี้ถูกใช้เพื่อสร้างและกำหนดค่าอินเทอร์เน็ตเซสชัน (Internet session) หนึ่งเซสชันหรือหลายๆเซสชันและอธิบายการติดต่อกับพรอกซีเซิร์ฟเวอร์ (Proxy server) ถ้าจำเป็น ฟังก์ชันต่างๆ ของคลาส CInternetSession ที่สำคัญ สามารถสรุปได้ดังตารางต่อไปนี้

ชื่อฟังก์ชัน	หน้าที่
CInternetSession	สร้างออบเจกต์ CInternetSession
OpenURL	วิเคราะห์และเปิด URL
GetHttpConnection	เปิด HTTP เซิร์ฟเวอร์เพื่อโปรแกรมที่กำลังพยายามเปิดการติดต่อ
GetFtpConnection	เปิด FTP เซสชันกับเซิร์ฟเวอร์แล้วทำการ log on ผู้ใช้
GetGopherConnection	เปิด Gopher เซิร์ฟเวอร์เพื่อโปรแกรมที่กำลังพยายามเปิดการติดต่อ
EnableStatusCallback	สร้าง Status callback routine ฟังก์ชัน EnableStatusCallback นี้มีสำหรับการทำงานแบบ Asynchronous
Close	ปิดการติดต่อกับอินเทอร์เน็ตเมื่ออินเทอร์เน็ตเซสชันจบลงแล้ว
OnStatusCallback	แก้ไขสถานะการทำงานใหม่เมื่อสถานะของ callback ถูกกระตุ้น

ตารางที่ 4-5 แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CInternetSession

#### 4.5.3.2 คลาส CInternetConnection

คลาสนี้ถูกใช้เพื่อจัดการกับการติดต่อไปยังอินเทอร์เน็ตเซิร์ฟเวอร์

ฟังก์ชันต่างๆ ของคลาส CInternetConnection ที่สำคัญ สามารถสรุปได้ดังตารางต่อไปนี้

ชื่อฟังก์ชัน	หน้าที่
CInternetConnection	สร้างออบเจกต์ CInternetConnection
GetContext	รับคอนเท็กซ์ ID สำหรับออบเจกต์ที่ทำการติดต่อ
GetSession	รับพอยน์เตอร์ที่ชี้ไปยังออบเจกต์ CInterSession ที่สัมพันธ์กับการติดต่อ
GetServerName	รับชื่อของเซิร์ฟเวอร์ที่สัมพันธ์กับการติดต่อ

ตารางที่ 4-6 แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CInternetConnection

#### 4.5.3.3 คลาส CInternetFile

คลาสนี้และคลาสที่สืบทอดจากคลาสนี้ ถูกใช้เพื่อเข้าถึงไฟล์ที่อยู่ในอีกระบบหนึ่งที่ตั้งติดต่ออยู่ด้วย โดยอินเทอร์เน็ตโปรโตคอล

ฟังก์ชันต่างๆ ของคลาส CInternetFile ที่สำคัญ สามารถสรุปได้ดังตารางต่อไปนี้

ชื่อฟังก์ชัน	หน้าที่
CInternetFile	สร้างออบเจกต์ CInternetFile
SetWriteBufferSize	กำหนดขนาดของบัฟเฟอร์ที่ข้อมูลจะถูกเขียน
SetReadBufferSize	กำหนดขนาดของบัฟเฟอร์ที่ข้อมูลจะถูกอ่าน
Read	อ่านข้อมูลเป็นไบต์ตามขนาดที่กำหนด
Write	เขียนข้อมูลเป็นไบต์ตามขนาดที่กำหนด
Flush	ถ่ายข้อมูลในบัฟเฟอร์ที่ข้อมูลจะถูกเขียนและทำให้แน่ใจว่าข้อมูลในหน่วยความจำได้ถูกเขียนลงไปยังเครื่องปลายทาง
Close	ปิดออบเจกต์ CInternetFile และคืน resources

ตารางที่ 4-7 แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CInternetFile

#### 4.5.3.4 คลาส CHttpConnection

คลาสนี้ถูกใช้เพื่อจัดการกับการติดต่อไปยัง HTTP เซิร์ฟเวอร์

ฟังก์ชันต่างๆ ของคลาส CHttpConnection ที่สำคัญ สามารถสรุปได้ดังตารางต่อไปนี้

ชื่อฟังก์ชัน	หน้าที่
CHttpConnection	สร้างออบเจกต์ CHttpConnection
OpenRequest	เปิดการร้องขอแบบ HTTP

ตารางที่ 4-8 แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CHttpConnection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.5.3.5 คลาส CHttpFile

คลาสนี้ถูกใช้เพื่อให้ความสามารถในการค้นหา และอ่านไฟล์ที่อยู่บน HTTP เซิร์ฟเวอร์ ฟังก์ชันต่างๆ ของคลาส CHttpFile ที่สำคัญ สามารถสรุปได้ดังตารางต่อไปนี้

ชื่อฟังก์ชัน	หน้าที่
CHttpFile	สร้างออบเจกต์ CHttpFile
AddRequestHeaders	เพิ่มส่วนหัวให้กับการร้องขอที่ส่งไปยัง HTTP เซิร์ฟเวอร์
SendRequest	ส่งการร้องขอไปยัง HTTP เซิร์ฟเวอร์

ตารางที่ 4-9 แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CHttpFile

### 4.5.3.6 คลาส CInternetException

คลาสนี้ถูกใช้เพื่อแสดงสถานะข้อยกเว้น (Exception) ที่เกี่ยวข้องกับการปฏิบัติการบนอินเทอร์เน็ต ฟังก์ชันต่างๆ ของคลาส CInternetException ที่สำคัญ สามารถสรุปได้ดังตารางต่อไปนี้

ชื่อฟังก์ชัน	หน้าที่
CInternetException	สร้างออบเจกต์ CInternetException
m_dwError	ความผิดพลาดที่ทำให้เกิดข้อยกเว้น
m_dwContext	ค่าคอนเท็กซ์ที่สัมพันธ์กับกระบวนการที่ทำให้เกิดข้อยกเว้น

ตารางที่ 4-10 แสดงฟังก์ชันและหน้าที่ต่างๆ ของคลาส CInternetException

## 4.6 วิธีการเขียนโปรแกรมโดยใช้ MFC Library

การเขียนโปรแกรมเพื่อติดต่อกับฐานข้อมูล

```

CDictSet::CDictSet(CDaoDatabase* pdb) : CDaoRecordset(pdb) {
    //{{AFX_FIELD_INIT(CDictSet)
    m_code = _T("");
    m_eword = _T("");
    m_tant = _T("");
    m_tenglish = _T("");
    m_tsyn = _T("");
    m_tword = _T("");
    //}}AFX_FIELD_INIT
    m_nDefaultType = dbOpenDynaset;
}
    
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโค้ดตัวอย่างนี้ จะอยู่ในคอนสตรัคเตอร์ของคลาสเรคคอร์ดเซต ซึ่งเป็นการเม็ปดาต้าเมมเบอร์ กับเรคคอร์ดของข้อมูลในฐานข้อมูล

```
void CDictSet::DoFieldExchange(CDaoFieldExchange* pFX) {
   //{{AFX_FIELD_MAP(CDictSet)
    pFX->SetFieldType(CDaoFieldExchange::outputColumn);
    DFX_Text(pFX, _T("[code]"), m_code);
    DFX_Text(pFX, _T("[eword]"), m_eword);
    DFX_Text(pFX, _T("[tant]"), m_tant);
    DFX_Text(pFX, _T("[tenglish]"), m_tenglish);
    DFX_Text(pFX, _T("[tsyn]"), m_tsyn);
    DFX_Text(pFX, _T("[tword]"), m_tword);
   //}}AFX_FIELD_MAP
}
```

จากฟังก์ชันข้างบนนี้ คือฟังก์ชัน DoFieldExchange เมื่อมีการเรียกฟังก์ชันนี้ ฟังก์ชันทั้ง DFX (DAO Record Field Exchange) และ RFX (Record Field Exchange) จะทำหน้าที่แลกเปลี่ยนข้อมูลระหว่างฟิลด์ดาต้าเมมเบอร์ในเรคคอร์ดเซต กับคอลัมน์ในฐานข้อมูล ซึ่งฟังก์ชัน DFX นี้ใช้กับคลาสเรคคอร์ดเซต DAO และฟังก์ชัน RFX นี้ใช้กับคลาสเรคคอร์ดเซต ODBC

```
void CDictView::DoDataExchange(CDataExchange* pDX) {
    CDaoRecordView::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CDictView)
    DDX_Control(pDX, IDC_CVOCAB, m_clist);
    DDX_CBString(pDX, IDC_CVOCAB, m_cvocab);
    DDX_Text(pDX, IDC_THAI_MEANING, m_ctmeaning);
    DDX_Text(pDX, IDC_ENG_MEANING, m_cemeaning);    //}}
    AFX_DATA_MAP
}
```

จากตัวอย่างข้างต้นนี้ ได้จากการทำเม็ปดาต้าเมมเบอร์ซึ่งอยู่ในไฟล์อิมพลีเม้นเตชัน (Implementation) ชื่อ DictView ฟังก์ชันนี้ทำหน้าที่รับข้อมูลจากดาต้าเมมเบอร์มาแสดงผลออกทางหน้าจอ และยังทำหน้าที่รับข้อมูลจากผู้ใช้ที่พิมพ์เข้ามา ไปเก็บไว้ในดาต้าเมมเบอร์ด้วย

ฟังก์ชัน Dialog Data Exchange (DDX) อยู่ในคลาสสวีทที่สืบทอดมาจากคลาส CDaoRecordView ภายในแอปพลิเคชันที่สร้างจาก MFC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน DDX นี้ให้เราสามารถโอนย้ายข้อมูลระหว่างคอนโทรลไดอะล็อก หรือฟอร์มวิวกับดาต้าเมมเบอร์ในคลาสเดียวกัน โดย Foreign Object สามารถแลกเปลี่ยนข้อมูลระหว่างคอนโทรลกับดาต้าเมมเบอร์ของออบเจกต์ที่แยกออกมา (ในกรณีนี้คือระหว่างคอนโทรลของเรคคอร์ดวิวกับดาต้าเมมเบอร์ของเรคคอร์ดเซต)

เมื่อเราระบุคลาส CDaoRecordset หรือ CRecordSet ที่สืบทอดมาพร้อมกับ CDaoRecordView หรือ CRecordView เราสามารถตั้งชื่อคลาสเรคคอร์ดเซตที่มีอยู่แล้วหรือสร้างใหม่ได้ คลาสวิซาร์ดได้เพิ่มตัวแปรเมมเบอร์ไปยังคลาสเรคคอร์ดวิว ซึ่งถูกตั้งชื่อไว้โดยดีฟอลต์ “m\_pSet” ชนิดดาต้าของตัวแปรนี้เป็นคลาสเรคคอร์ดเซต เราสามารถแสดงความสัมพันธ์ระหว่างเรคคอร์ดบนหน้าจอ ออบเจกต์เรคคอร์ดวิวและกับออบเจกต์เรคคอร์ดเซต

```
IMPLEMENT_DYNCREATE(CDictView, CDaoRecordView)
BEGIN_MESSAGE_MAP(CDictView, CDaoRecordView)
   //{{AFX_MSG_MAP(CDictView)
    ON_BN_CLICKED(IDC_Search, OnSearch)
    ON_COMMAND(ID_ADDVOCAB, OnAddvocab)
    ON_COMMAND(ID_UPDATEVOCAB, OnUpdatevocab)
    ON_COMMAND(ID_DELETEVOCAB, OnDeletevocab)
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

จากตัวอย่างนี้ คือส่วนของโปรแกรมที่ได้จากการแมปเมสเสจ (Message) ซึ่งการแมปเมสเสจนี้เป็นหัวใจของโปรแกรมวินโดวส์ ซึ่งทำการวนลูปตลอดเวลาใน WinMain() จนกว่าเราจะปิดโปรแกรม ในส่วนนี้จะเป็นการแมปเมสเสจเฉพาะ ON\_BN\_CLICKED และ ON\_COMMAND



และตารางจะถูกใช้เพื่อวางและเรียกแต่ละฟังก์ชันเหล่านี้ ฟังก์ชันใดๆ ก็ตามที่ไม่ได้ส่งออกไปก็จะไม่เพิ่มเข้าไปในตารางนี้ และฟังก์ชันเหล่านั้นจะมองไม่เห็น หรือถูกเรียกโดยแอปพลิเคชันที่อยู่ด้านนอกหรือ DLL

แอปพลิเคชันสามารถเรียกฟังก์ชันใน DLL ได้ 2 วิธี วิธีที่เกี่ยวข้องส่วนมากของการเรียกฟังก์ชันเหล่านี้คือ มองเข้าไปยังตำแหน่งของฟังก์ชันที่ต้องการใน DLL แล้วเอาค่าพอยน์เตอร์ชี้ไปยังฟังก์ชัน ต่อมาพอยน์เตอร์จะถูกใช้เพื่อเรียกฟังก์ชัน อีกวิธีหนึ่งเป็นวิธีที่ง่ายกว่าคือ ลิงค์แอปพลิเคชันด้วยไฟล์ LIB ที่ถูกสร้างพร้อมกับ DLL ไฟล์ LIB นี้ถูกใช้โดยตัวลิงค์ราวกับว่าเป็นไฟล์ไลบรารีมาตรฐาน

อย่างไรก็ตามไฟล์ LIB นี้ ประกอบด้วยสตับ (Stub) สำหรับแต่ละฟังก์ชันที่ถูกส่งออกไปใน DLL สตับคือ ฟังก์ชันเทียม (pseudo) ที่มีชื่อและพารามิเตอร์เหมือนกัน เหมือนกับว่าเป็นฟังก์ชันจริงๆ ภายในสตับฟังก์ชันคือ โค้ดจำนวนหนึ่งทีเรียลฟังก์ชันจริงใน DLL ด้วยการผ่านค่าพารามิเตอร์ทั้งหมดซึ่งจะถูกผ่านมายังสตับ การทำอย่างนี้ออนุญาตให้เรากระทำกับฟังก์ชันใน DLL เหมือนกับว่าฟังก์ชันเหล่านี้เป็นส่วนหนึ่งของโค้ดแอปพลิเคชัน และไม่ใช้ไฟล์ที่ถูกแยกออกมา

มีอยู่ 2 วิธีของการสร้าง DLL ง่ายๆ โดยการใช้ Microsoft Visual C++ คือ Regular DLL และ MFC Extension DLL

### 5.2.1 DLL แบบ Regular DLL

การสร้าง DLL ชนิดนี้จะส่งฟังก์ชันมาตรฐานออก ไม่สามารถส่งคลาสออกมาได้ ดังนั้นการสร้าง DLL นี้เป็นสิ่งที่ง่ายมากรวมทั้งการวางแผนการสร้างไม่ต้อมาก การส่งฟังก์ชันออกมามีรูปแบบดังนี้

```
extern "C" __declspec (dllexport) <function type> WINAPI <function declaration>
```

คำว่า extern "C" บ่งบอกว่าเป็นฟังก์ชันมาตรฐาน C

\_\_declspec (dllexport) เป็นคีย์เวิร์ด ซึ่งต้องอยู่ด้านซ้ายของการประกาศฟังก์ชัน เมื่อใช้คีย์เวิร์ดนี้แล้วเรา ไม่จำเป็นต้องใช้ไฟล์ที่มีนามสกุล .def สำหรับการส่งออกอีกต่อไป

ถ้าเราใช้คลาส MFC ในการสร้าง Regular DLL เราจำเป็นต้องเรียกมาโคร AFX\_MANAGE\_STATE แล้วนำมาใส่ในบรรทัดแรกของฟังก์ชันที่ต้องการส่งออก การทำแบบนี้เป็นสิ่งจำเป็นเพื่อให้ฟังก์ชันที่ถูกส่งออก เป็นเทรดเซฟ (tradesafe) ซึ่งอนุญาตให้ฟังก์ชันของเราสามารถถูกเรียกใช้ได้มากกว่า 2 โปรแกรม (2 เทรด)

มาโครนี้มีแค่หนึ่งพารามิเตอร์ ซึ่งก็คือ พอยน์เตอร์ชี้ไปโครงสร้าง AFX\_MODULE\_STATE เราเรียกฟังก์ชัน AfxGetStaticModuleState เพื่อดึงข้อมูลจากโครงสร้าง AFX\_MODULE\_STATE รูปแบบฟังก์ชันที่ส่งออกต่างๆ ไปดังนี้

```
extern "C" __declspec (dllexport) void WINAPI yourfunction()
{
    AFX_MANAGE_STATE(AfxGetStaticModuleState());
    // your function
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีอย่างหนึ่งของการใช้ Regular DLL คือ เมื่อสร้างไฟล์ DLL ชนิดนี้แล้ว เราสามารถใช้ได้กับภาษาเขียนโปรแกรมได้หลายภาษา เช่น Visual Basic

ข้อเสียของการใช้ Regular DLL ก็คือ มันไม่สามารถส่งคลาส, เมมเบอร์ฟังก์ชัน และโอเวอร์โหลดฟังก์ชันออกได้ เพราะว่าแต่ละคอมไพเลอร์ C++ มีเอกลักษณ์ของตัวเอง

อย่างไรก็ตามเราสามารถ useClass C++ และคลาสไลบรารี MFC ได้ภายใน DLL ชนิด Regular นี้ เมื่อเราสร้าง Regular DLL ด้วย MFC AppWizard เราสามารถเลือกลิงค์แบบ Static หรือลิงค์แบบ Dynamic ที่ลิงค์กับไลบรารี MFC ถ้าเราเลือกการลิงค์แบบ Static DLL ของเรามันจะรวมกับโค้ดไลบรารี MFC ที่จำเป็น และดังนั้นจึงสามารถนำไปใช้ที่ไหนก็ได้โดยไม่ต้องมีไลบรารี MFC อีก แต่ข้อเสียอย่างหนึ่งก็คือไฟล์ DLL ที่ได้จะมีขนาดเล็กกว่ามาก แต่เวลานำไปใช้กับเครื่องที่ไม่มีไลบรารี MFC ไม่สามารถใช้งานได้ ดังนั้นจำเป็นต้องมีไลบรารี MFC

### 5.2.2 DLL แบบ MFC Extension DLL

การสร้าง DLL ชนิดนี้ง่ายกว่าแบบ Regular DLL อีก เราแค่เพียงเพิ่มมาโคร AFX\_EXT\_CLASS เพื่อส่งคลาสออกจาก DLL การประกาศมีรูปแบบดังนี้

```
Class AFX_EXT_CLASS CyourClass
{
    ...
};
```

มาโครนี้จะส่งคลาสออก เราจำเป็นต้องรวมมาโครนี้ในไฟล์เฮดเดอร์ ซึ่งจะถูกใช้โดยแอปพลิเคชันที่ใช้ DLL ข้อเสียของการใช้มาโคร AFX\_EXT\_CLASS ก็คือ DLL ที่สร้างขึ้นมาสามารถใช้ได้กับคอมไพเลอร์ C++ ที่สนับสนุนคลาส MFC เช่น คอมไพเลอร์ของ Borland

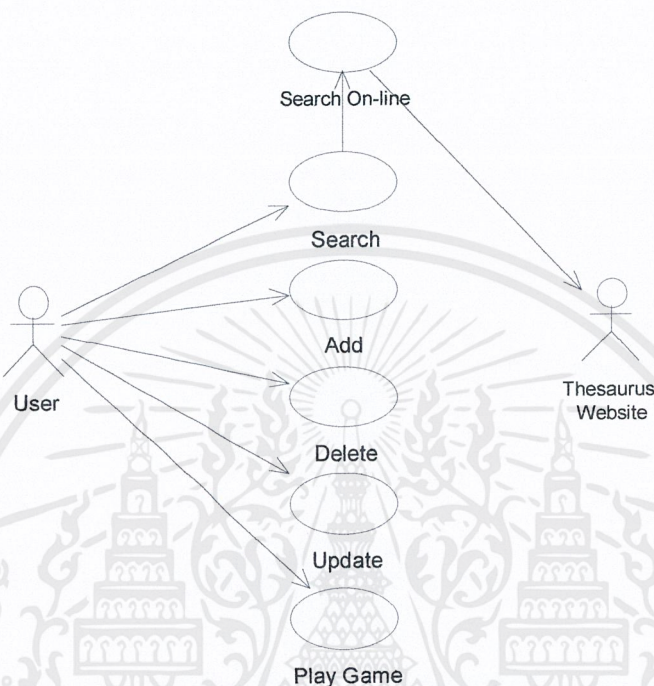
แต่บางครั้งถ้าเราต้องการส่งเฉพาะเมมเบอร์ฟังก์ชัน หรือค้ำเมมเบอร์ เช่น เราต้องการส่งฟังก์ชันของคลาสใดอะต็อก เราจำเป็นต้องส่ง Constructor ของคลาสใดอะต็อกแล้วตามด้วยฟังก์ชันนั้น มีวิธีการดังนี้

```
Class CtestDialog : public CDialog
{
public:
    AFX_EXT_CLASS CtestDialog();
    AFX_EXT_CLASS int
    ...
};
```

## บทที่ 6

### แนวคิดและการออกแบบโปรแกรมเชิงวัตถุ

#### 6.1 Use Case Diagram



รูปที่ 6-1 แสดง Use Case Diagram

Use Case Diagram แสดงความสัมพันธ์ระหว่างการทำงานภายในโปรแกรมและสิ่งแวดล้อมภายนอก โดยมีรายละเอียดดังนี้

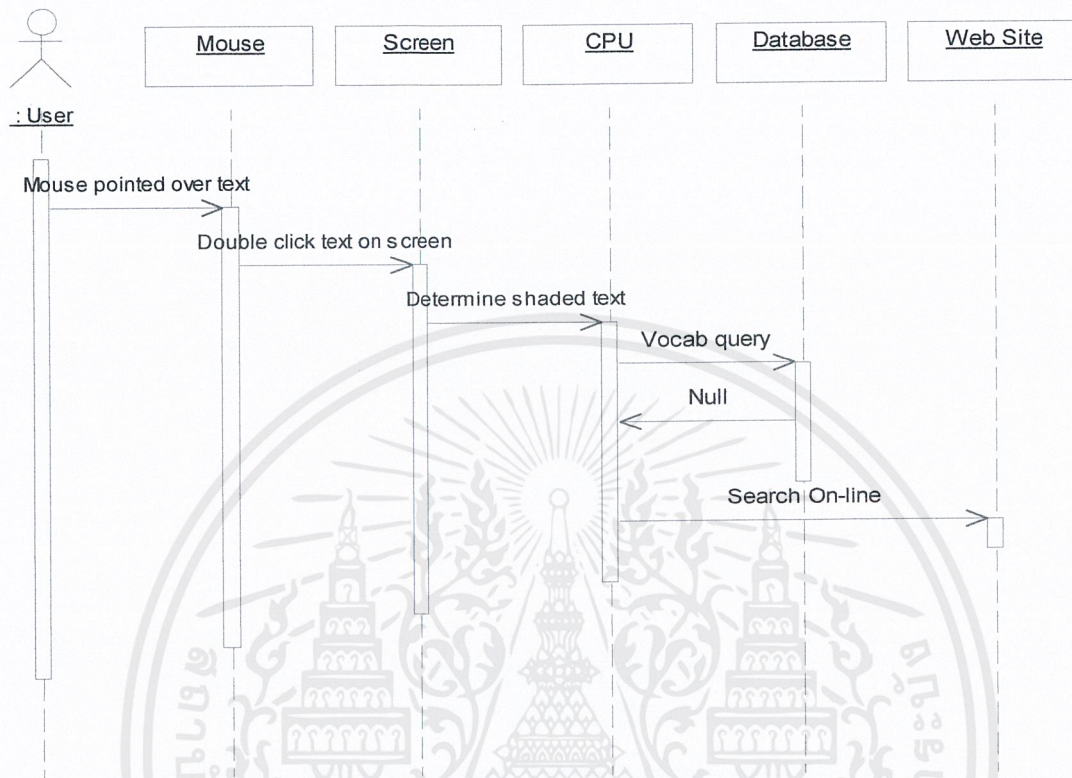
- **Search** ผู้ใช้โปรแกรมสามารถค้นหาคำศัพท์ได้ 2 วิธี คือ การค้นหาคำศัพท์ด้วยเมาส์ และการค้นหาคำศัพท์ด้วยส่วนติดต่อกับผู้ใช้
- **Search On-line** ผู้ใช้สามารถเลือกการค้นหาคำศัพท์ทางอินเทอร์เน็ตในกรณีที่โปรแกรมไม่พบคำศัพท์ที่ผู้ใช้ทำการค้นหา โดยโปรแกรมจะทำการติดต่อกับเว็บไซต์ที่ให้บริการค้นหาคำศัพท์ออนไลน์ แล้วนำคำศัพท์ที่ค้นหาได้นี้ เพิ่มลงในฐานข้อมูลเพื่อประโยชน์ในการค้นหาคำศัพท์ครั้งต่อไป
- **Add** ผู้ใช้สามารถเพิ่มคำศัพท์ใหม่ลงในฐานข้อมูลผ่านส่วนติดต่อกับผู้ใช้ได้ด้วยตนเอง
- **Update** ผู้ใช้สามารถแก้ไขคำศัพท์ที่มีอยู่แล้วในฐานข้อมูลผ่านส่วนติดต่อกับผู้ใช้ได้ด้วยตนเอง
- **Delete** ผู้ใช้สามารถลบคำศัพท์ที่มีอยู่ในฐานข้อมูลผ่านส่วนติดต่อกับผู้ใช้ได้ด้วยตนเอง
- **Play Game** หากคำศัพท์ที่ผู้ใช้ต้องการค้นหานั้น เคยถูกค้นหามาก่อนในช่วงเวลาที่ได้กำหนดไว้ จะให้ผู้ใช้เลือกที่จะเล่นเกมส์เพื่อทบทวนคำศัพท์ที่ได้เคยค้นหามาก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 Sequence Diagram and Collaboration Diagram

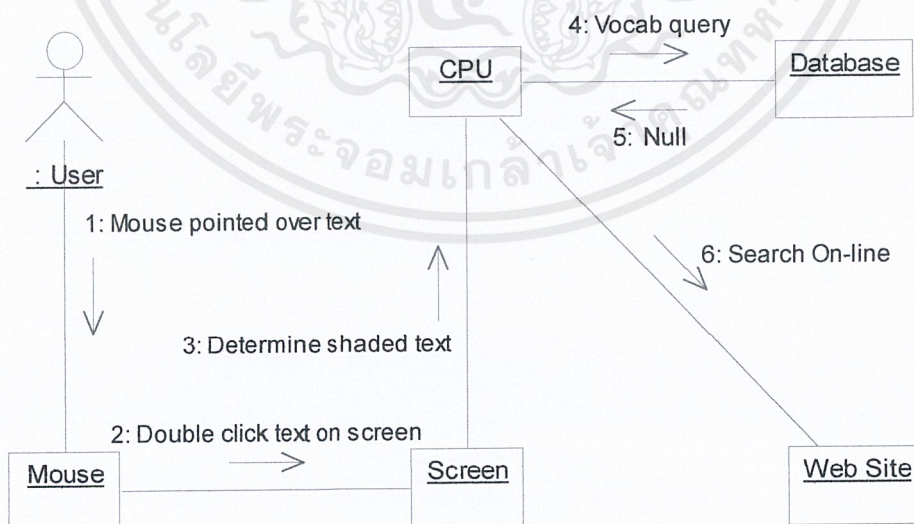
6.2.1 Search by Floating-online

- Sequence Diagram



รูปที่ 6-2 แสดง Sequence Diagram ของ Search by Floating-Online Method

- Collaboration Diagram

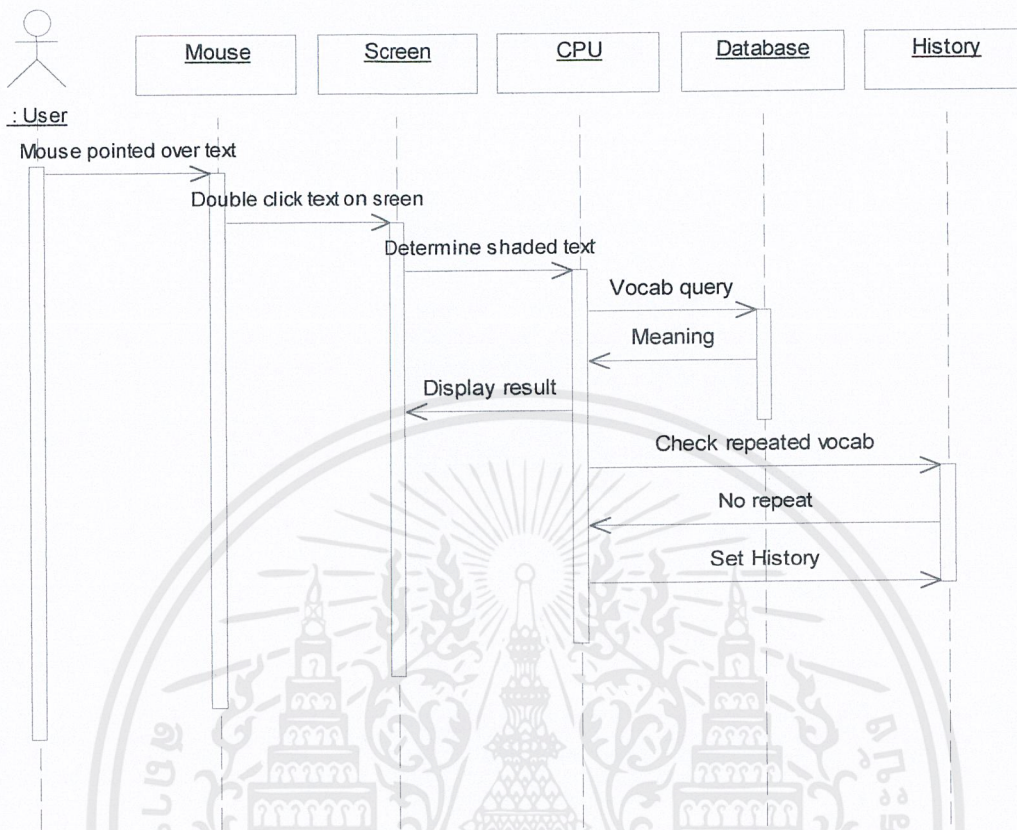


รูปที่ 6-3 แสดง Collaboration Diagram ของ Search by Floating-Online Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

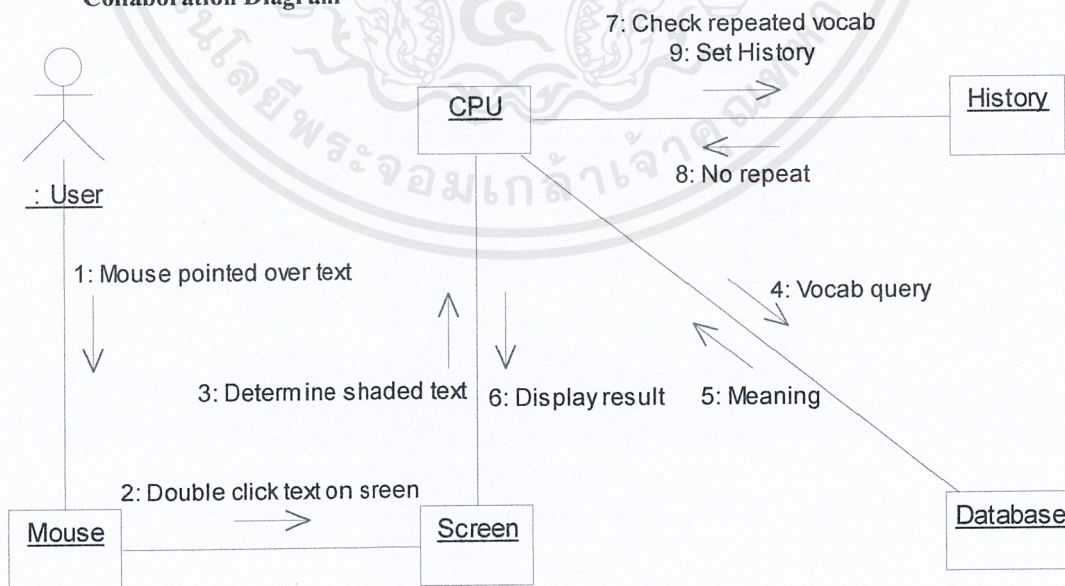
6.2.2 Search by Floating-no repeat

- Sequence Diagram



รูปที่ 6-4 แสดง Sequence Diagram ของ Search by Floating-no repeat Method

- Collaboration Diagram

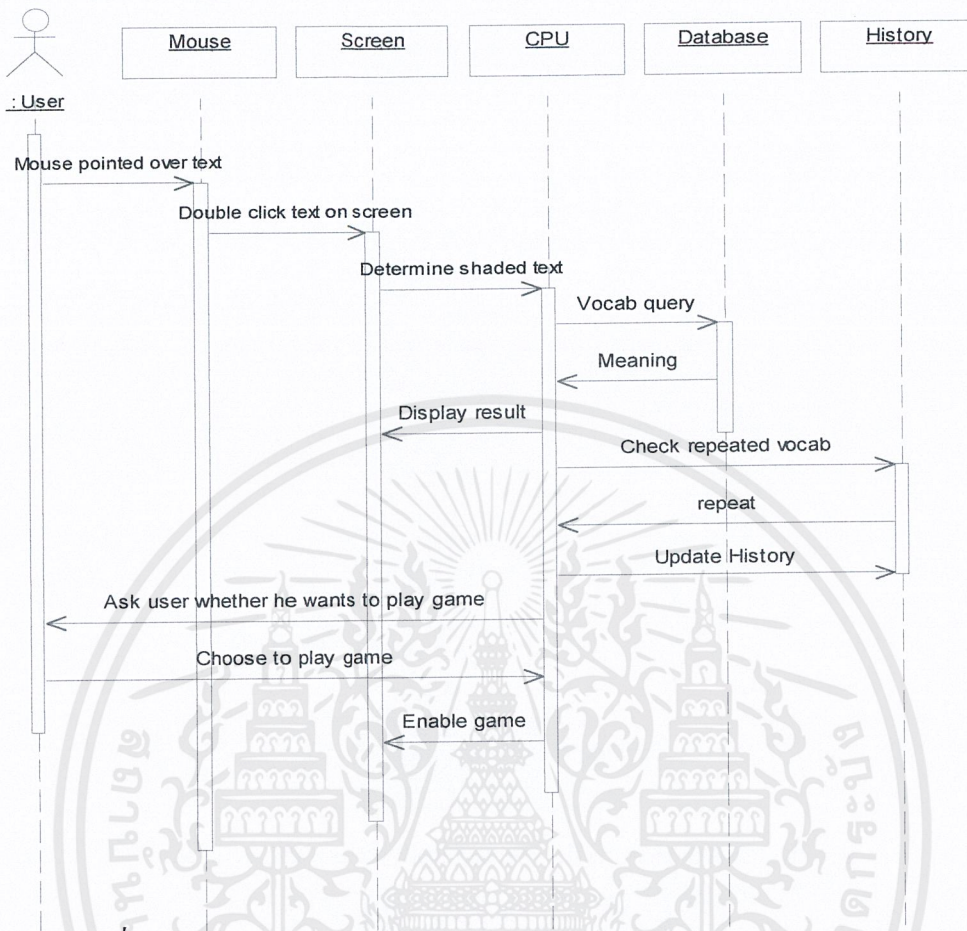


รูปที่ 6-5 แสดง Collaboration Diagram ของ Search by Floating-no repeat Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

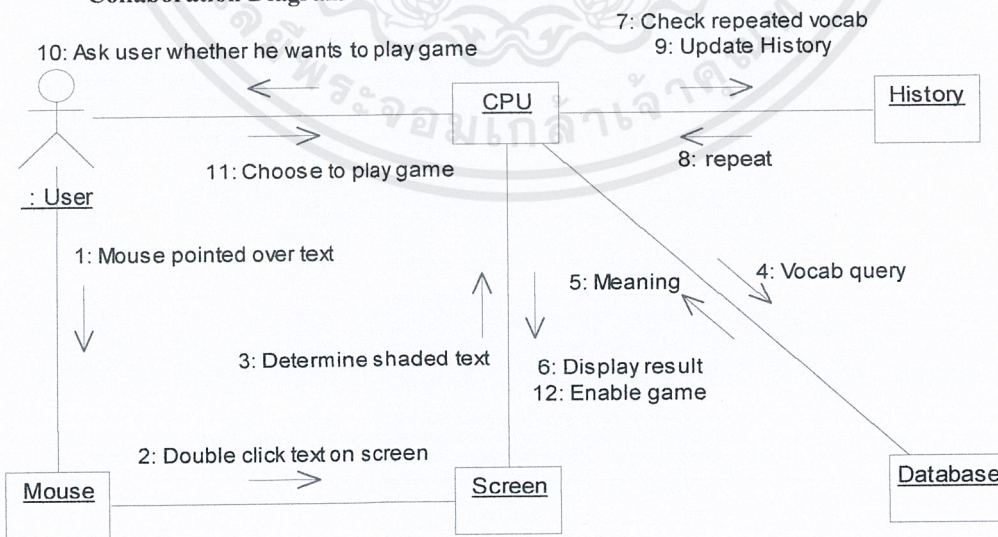
6.2.3 Search by Floating-play game

- Sequence Diagram



รูปที่ 6-6 แสดง Sequence Diagram ของ Search by Floating-play game Method

- Collaboration Diagram

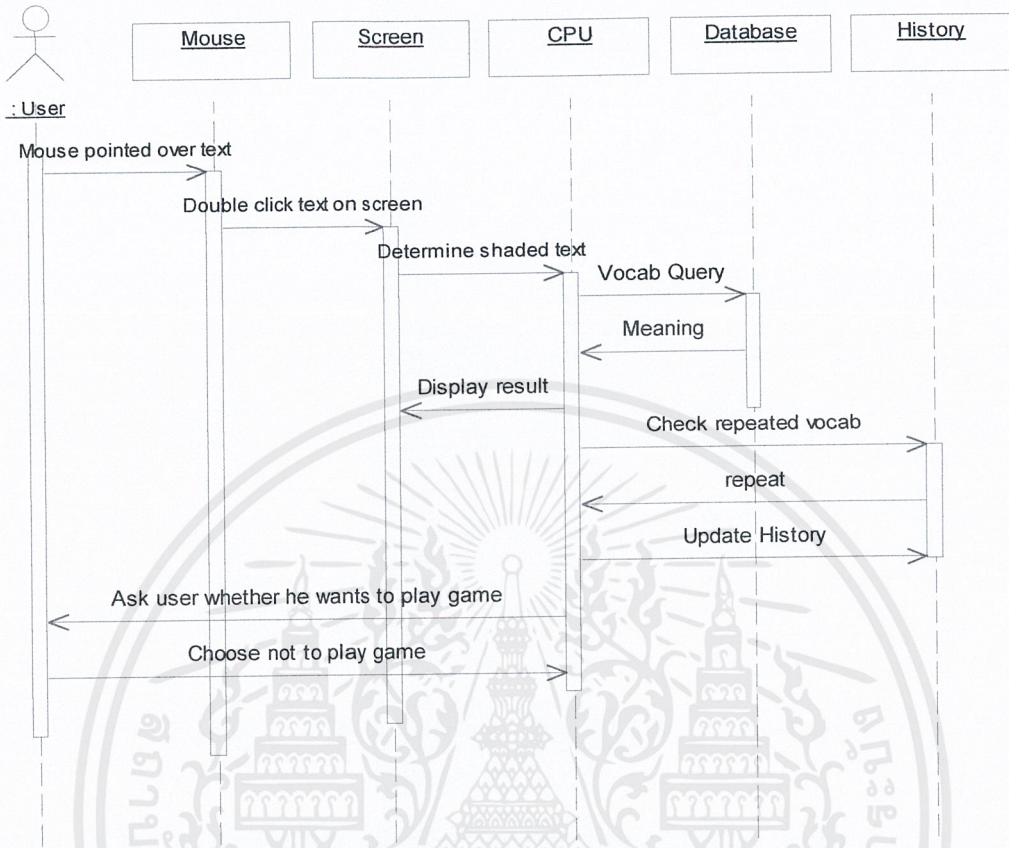


รูปที่ 6-7 แสดง Sequence Diagram ของ Search by Floating-play game Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

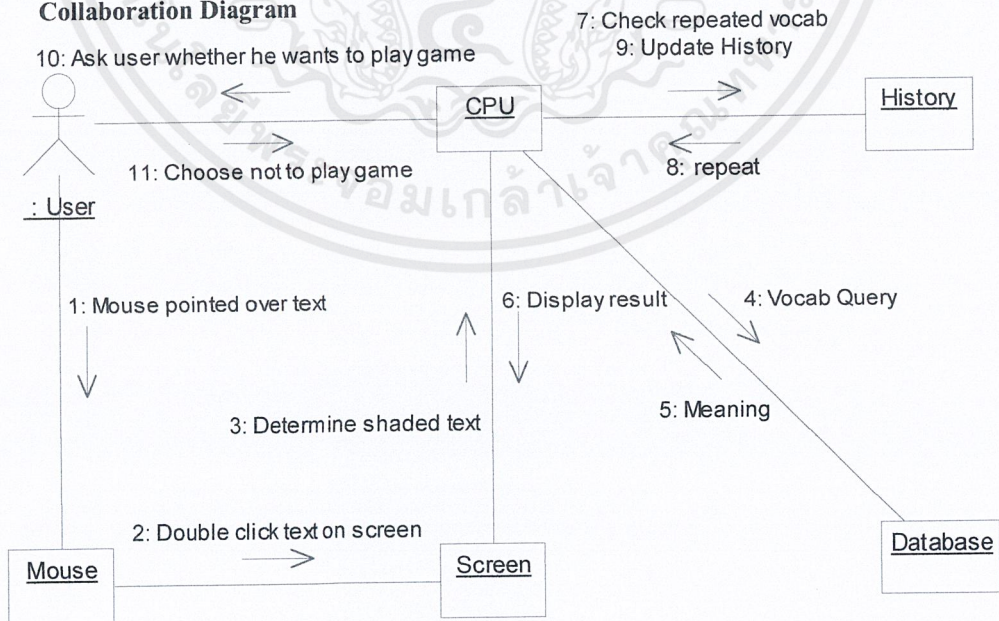
6.2.4 Search by Floating-no game

- Sequence Diagram



รูปที่ 6-8 แสดง Sequence Diagram ของ Search by Floating-no game Method

- Collaboration Diagram

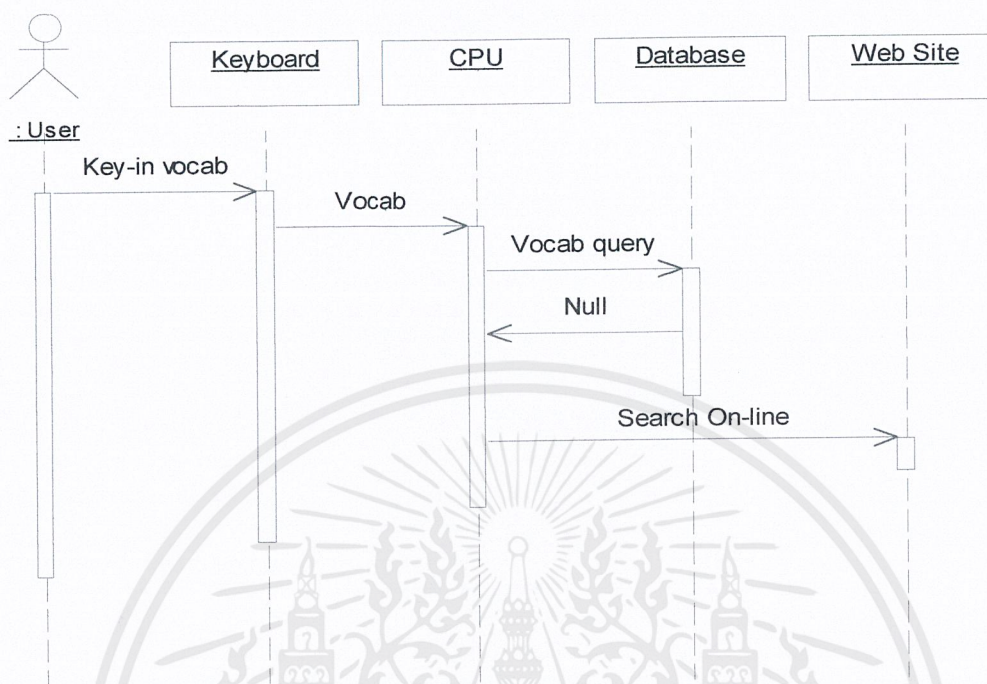


รูปที่ 6-9 แสดง Collaboration Diagram ของ Search by Floating-no game Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

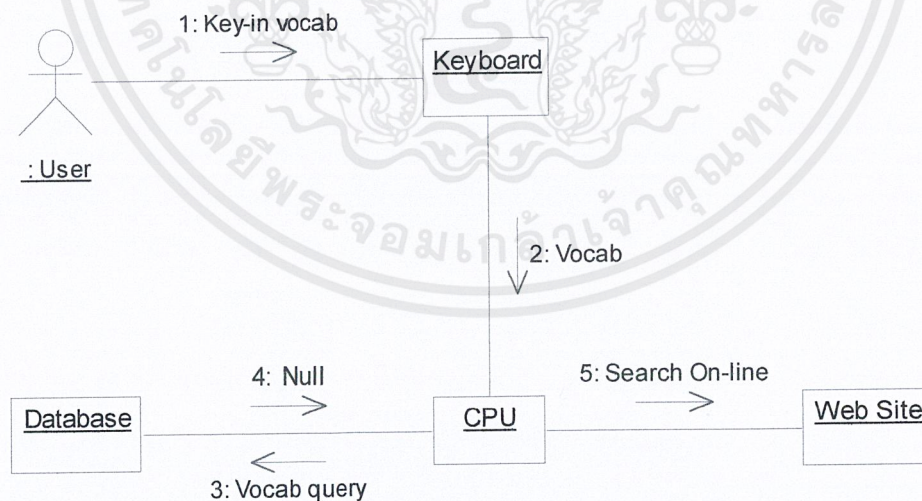
6.2.5 Search by Key-in-online

- Sequence Diagram



รูปที่ 6-10 แสดง Sequence Diagram ของ Search by Kay-in-online Method

- Collaboration Diagram

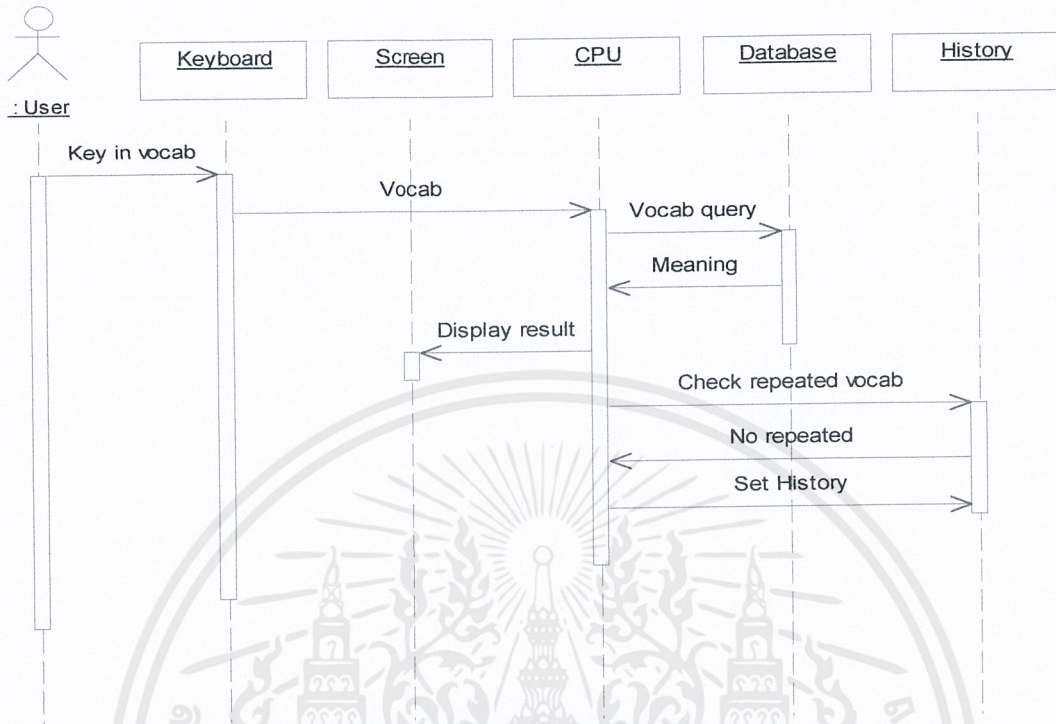


รูปที่ 6-11 แสดง Collaboration Diagram ของ Search by Key-in-online Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

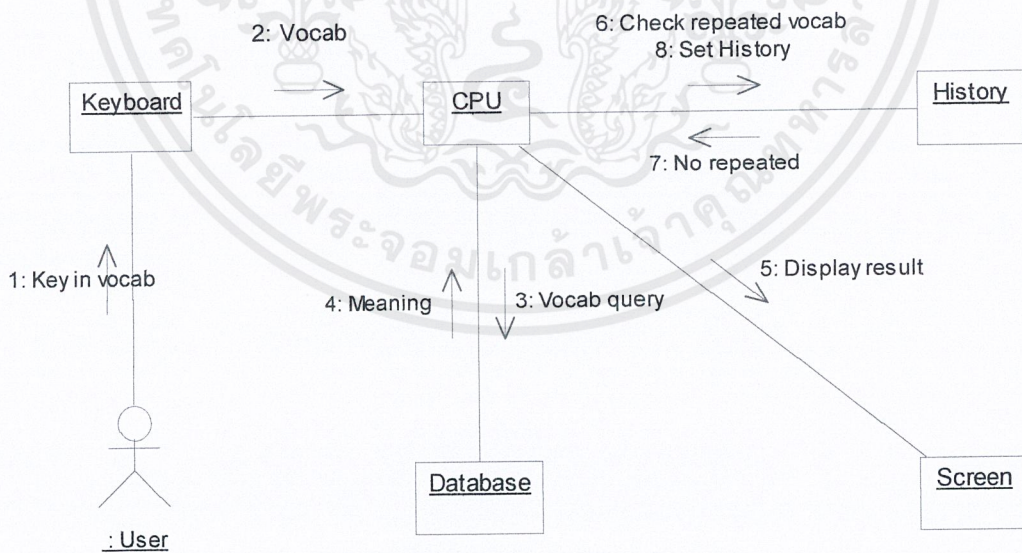
6.2.6 Search by Key-in-no repeat

- Sequence Diagram



รูปที่ 6-12 แสดง Sequence Diagram ของ Search by Key-in-no repeat Method

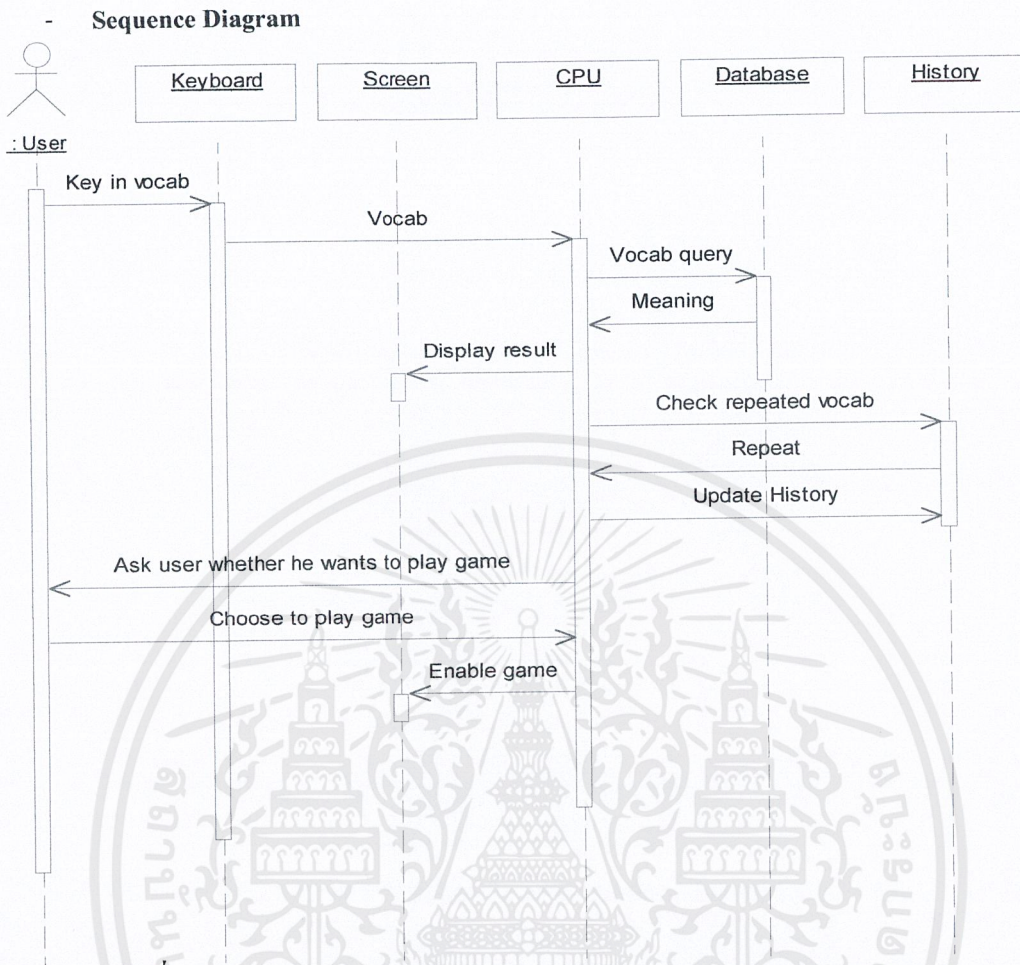
- Collaboration Diagram



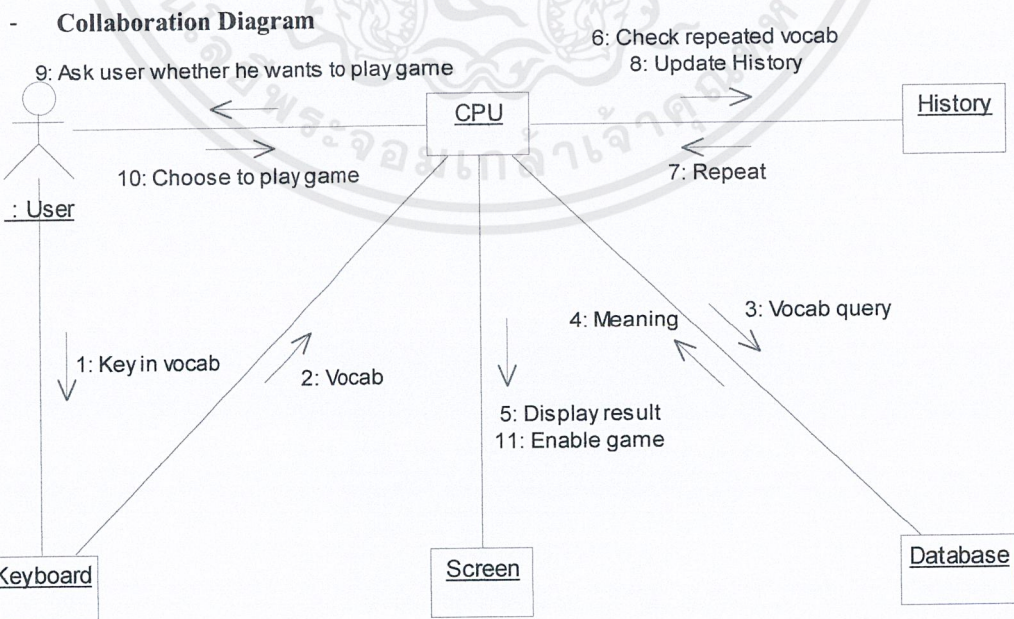
รูปที่ 6-13 แสดง Collaboration Diagram ของ Search by Key-in-no repeat Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.7 Search by Key-in-play game



รูปที่ 6-14 แสดง Sequence Diagram ของ Search by Key-in-play game Method

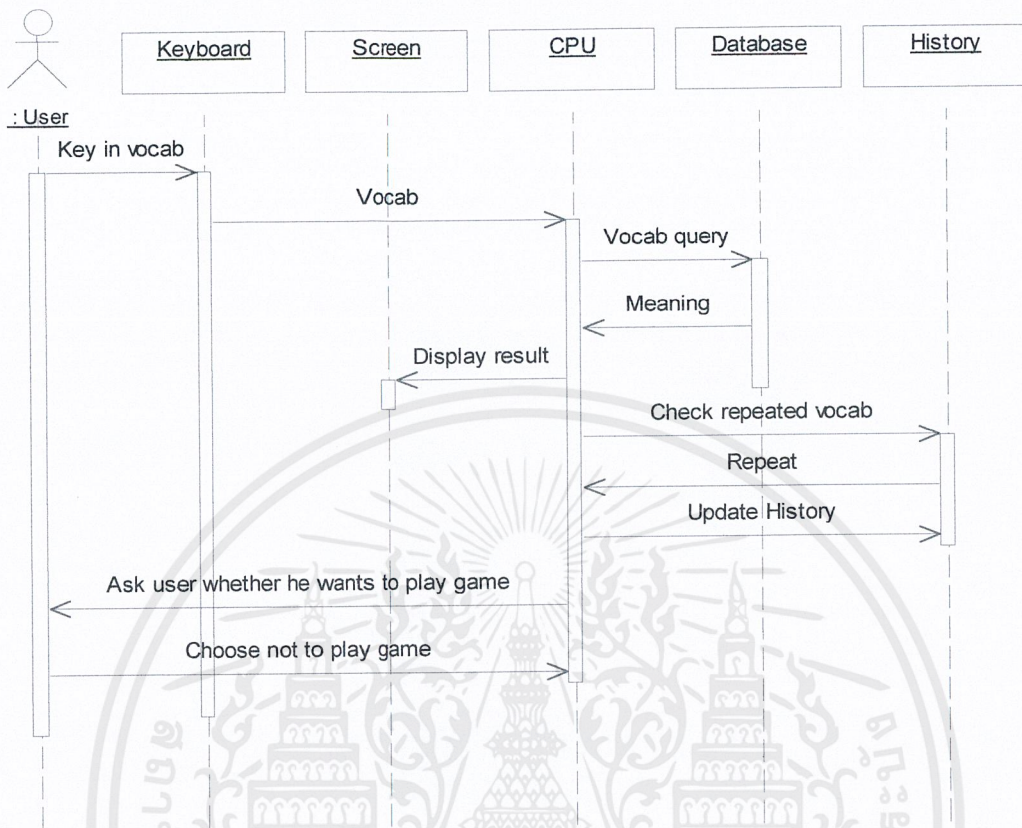


รูปที่ 6-15 แสดง Collaboration Diagram ของ Search by Key-in-play game Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

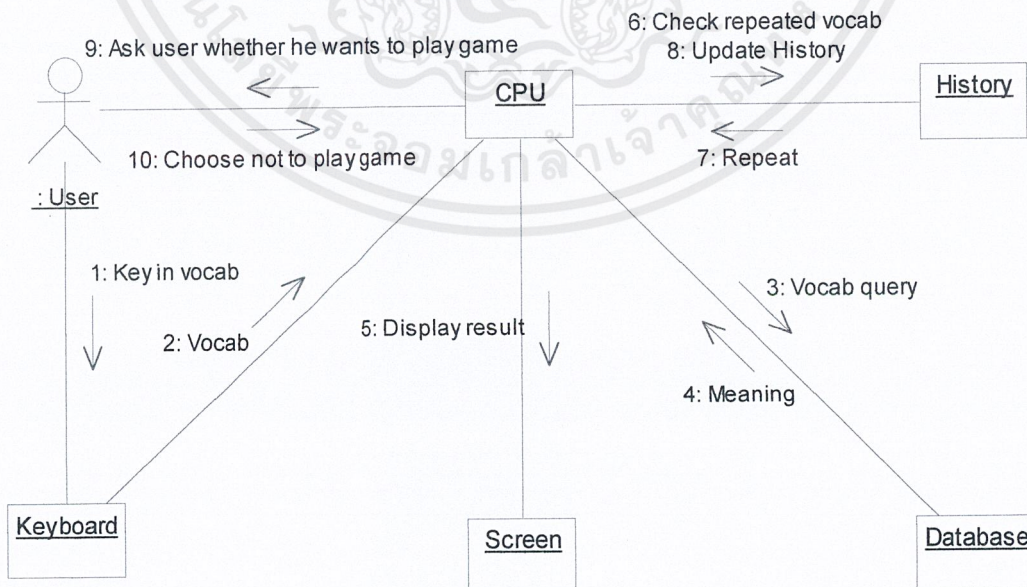
6.2.8 Search by Key-in-no game

- Sequence Diagram



รูปที่ 6-16 แสดง Sequence Diagram ของ Search by Key-in-no game Method

- Collaboration Diagram

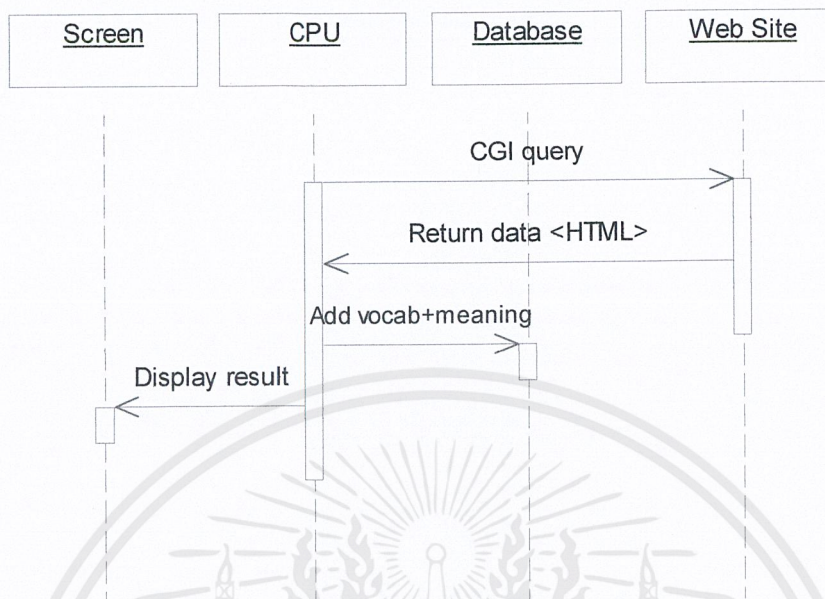


รูปที่ 6-17 แสดง Collaboration Diagram ของ Search by Key-in-no game Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

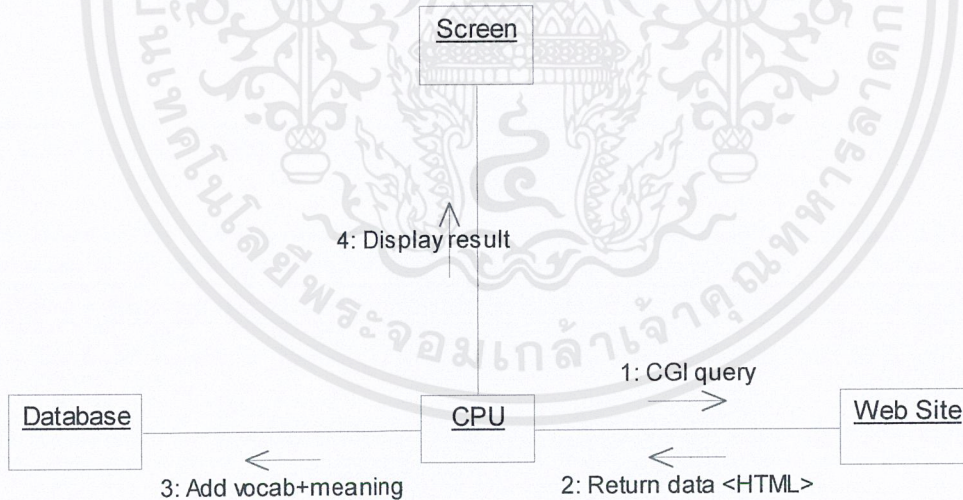
6.2.9 Search On-line

- Sequence Diagram



รูปที่ 6-18 แสดง Sequence Diagram ของ Search On-line Method

- Collaboration Diagram

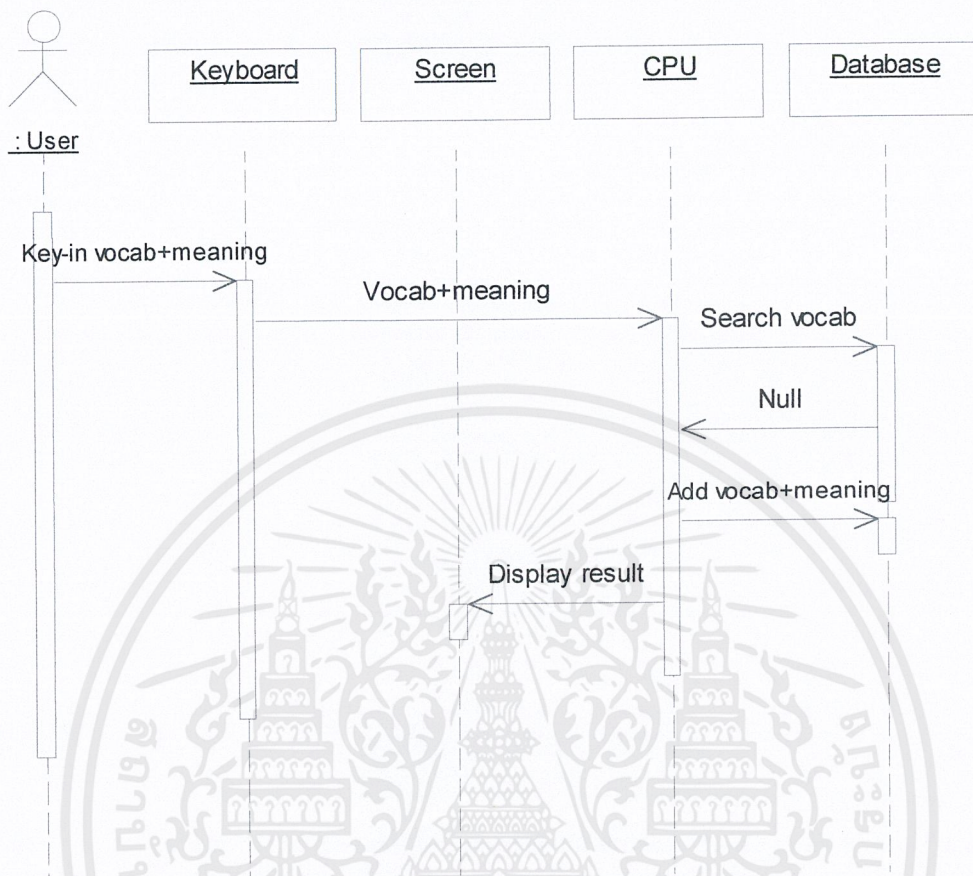


รูปที่ 6-19 แสดง Collaboration Diagram ของ Search On-line Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

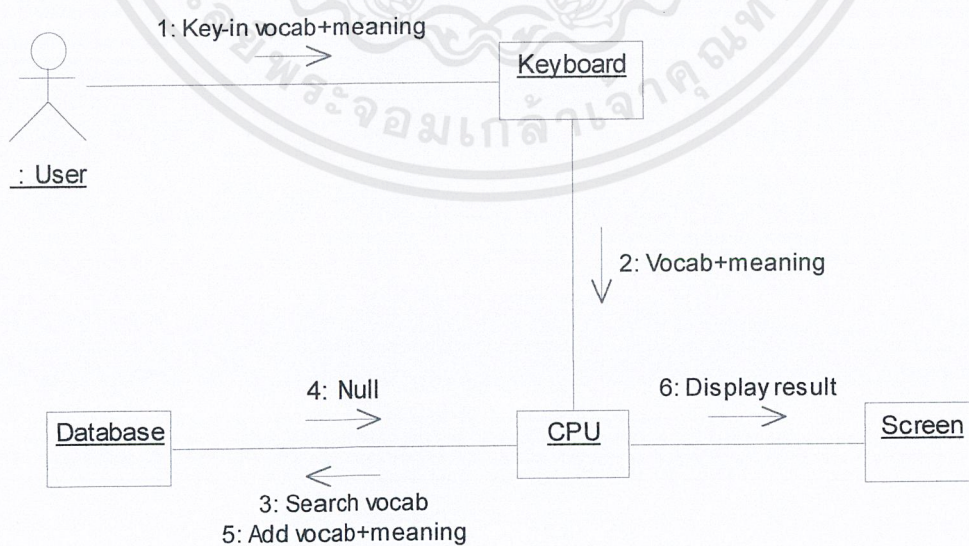
6.2.10 Add-success

- Sequence Diagram



รูปที่ 6-20 แสดง Sequence Diagram ของ Add-success Method

- Collaboration Diagram

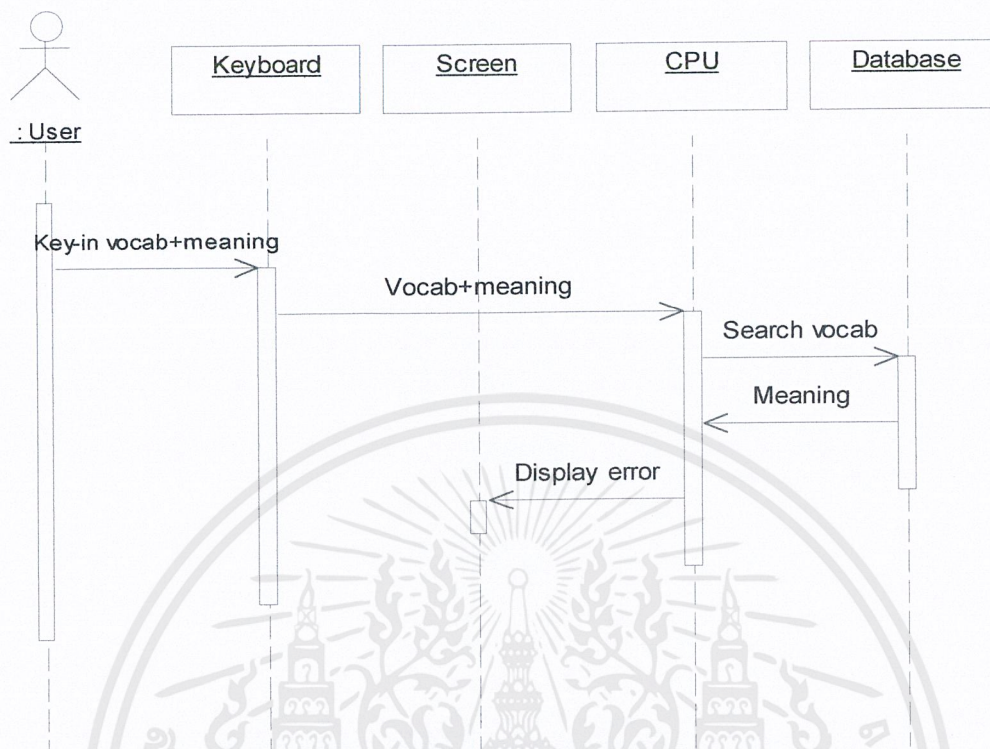


รูปที่ 6-21 แสดง Collaboration Diagram ของ Add-success Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

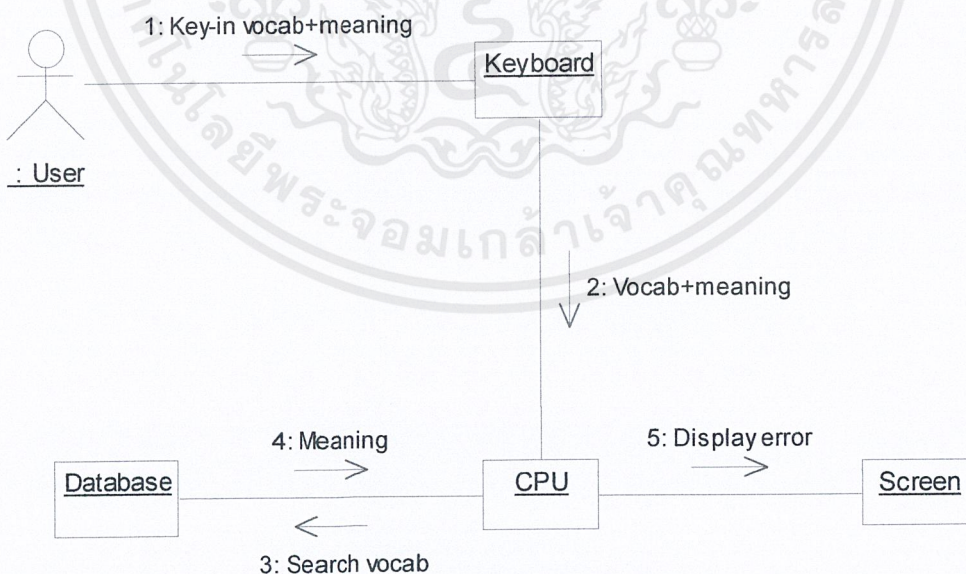
## 6.2.11 Add-error

## - Sequence Diagram



รูปที่ 6-22 แสดง Sequence Diagram ของ Add-error Method

## - Collaboration Diagram

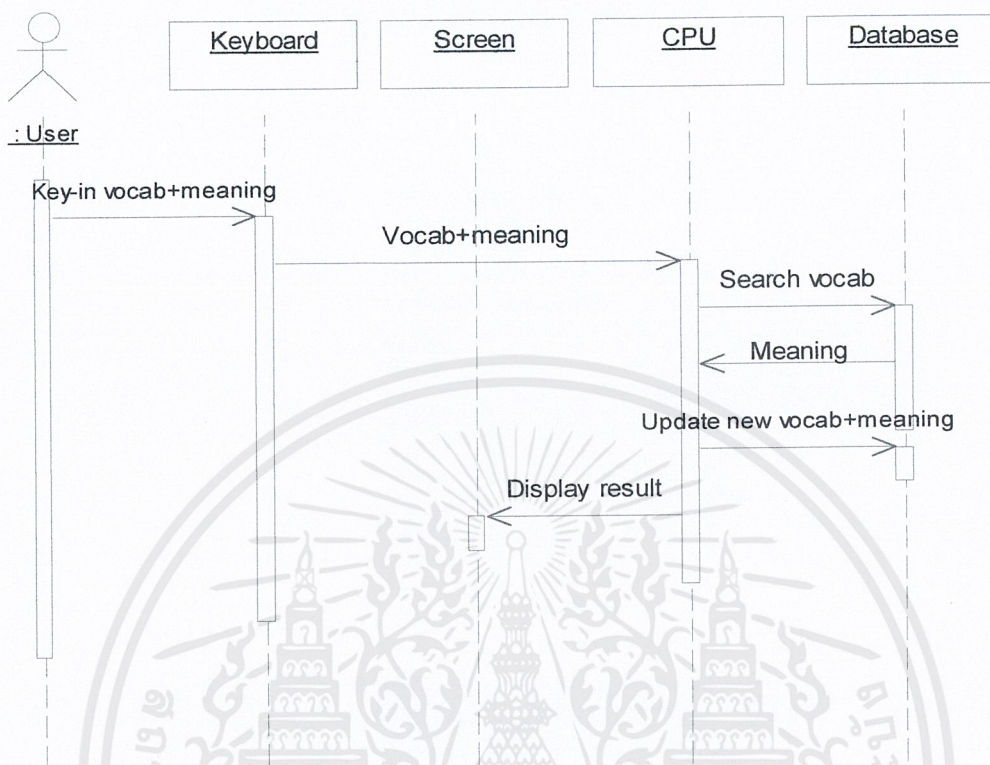


รูปที่ 6-23 แสดง Collaboration Diagram ของ Add-error Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

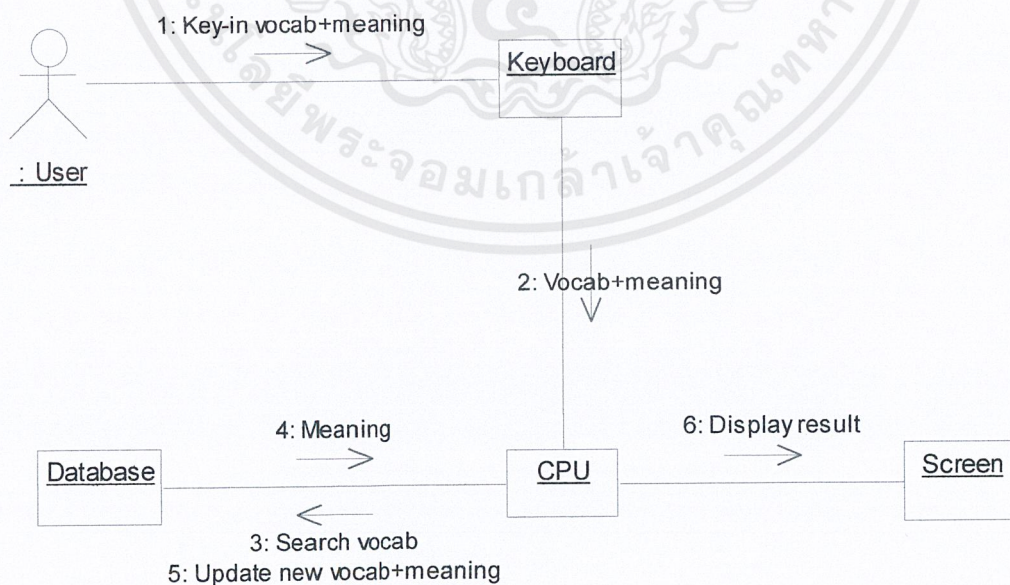
6.2.12 Update-success

- Sequence Diagram



รูปที่ 6-24 แสดง Sequence Diagram ของ Update-success Method

- Collaboration Diagram

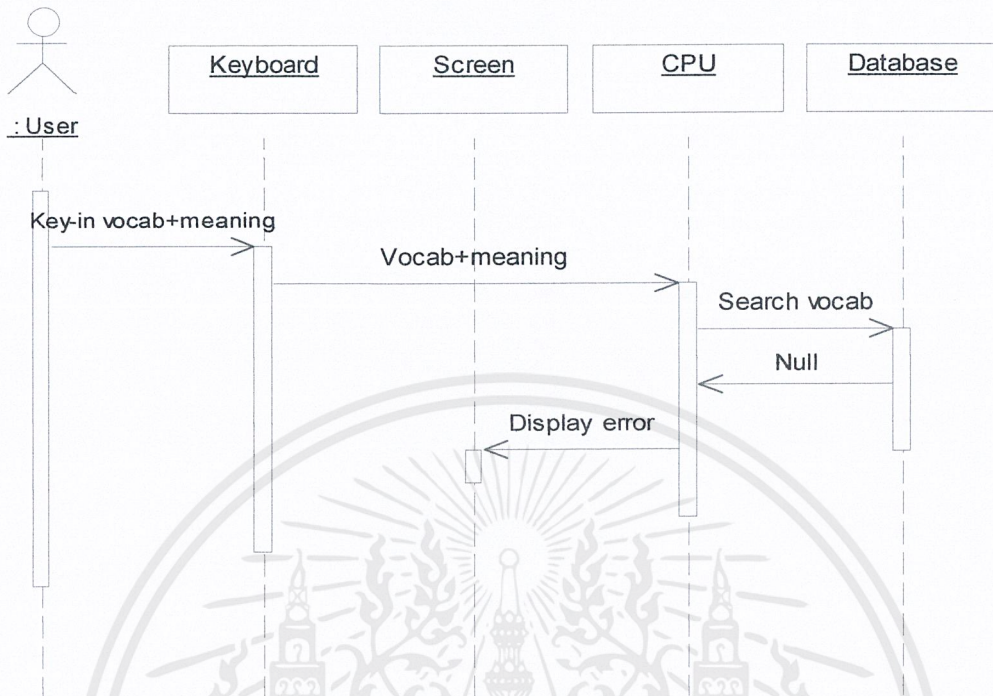


รูปที่ 6-25 แสดง Collaboration Diagram ของ Update-success Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

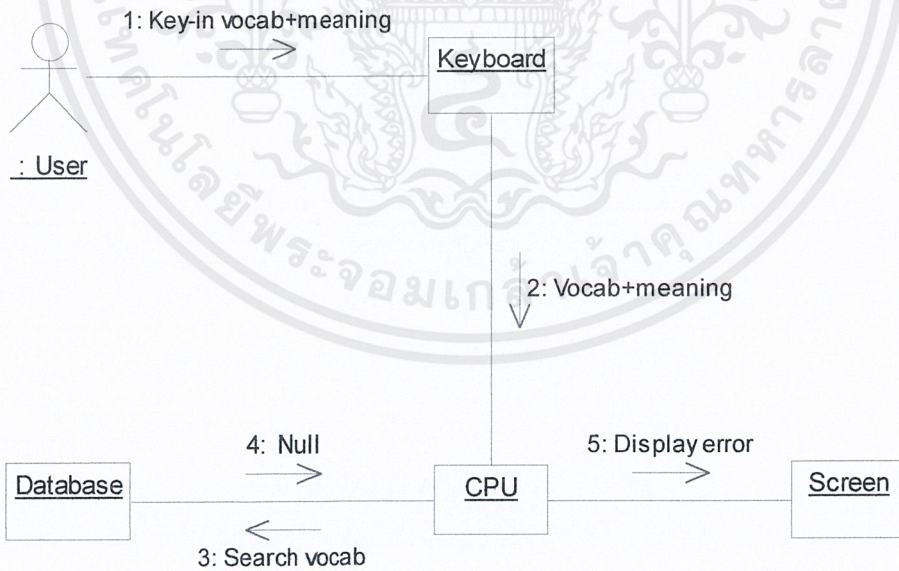
6.2.13 Update-error

- Sequence Diagram



รูปที่ 6-26 แสดง Sequence Diagram ของ Update-error Method

- Collaboration Diagram

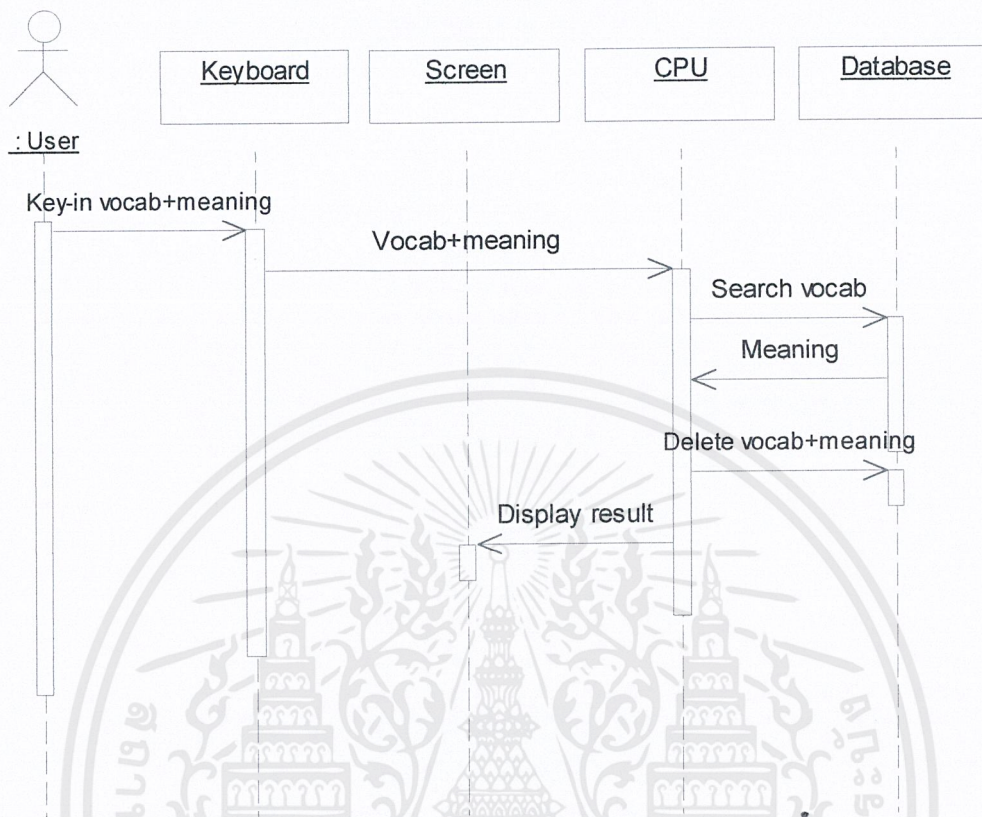


รูปที่ 6-27 แสดง Collaboration Diagram ของ Update-error Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

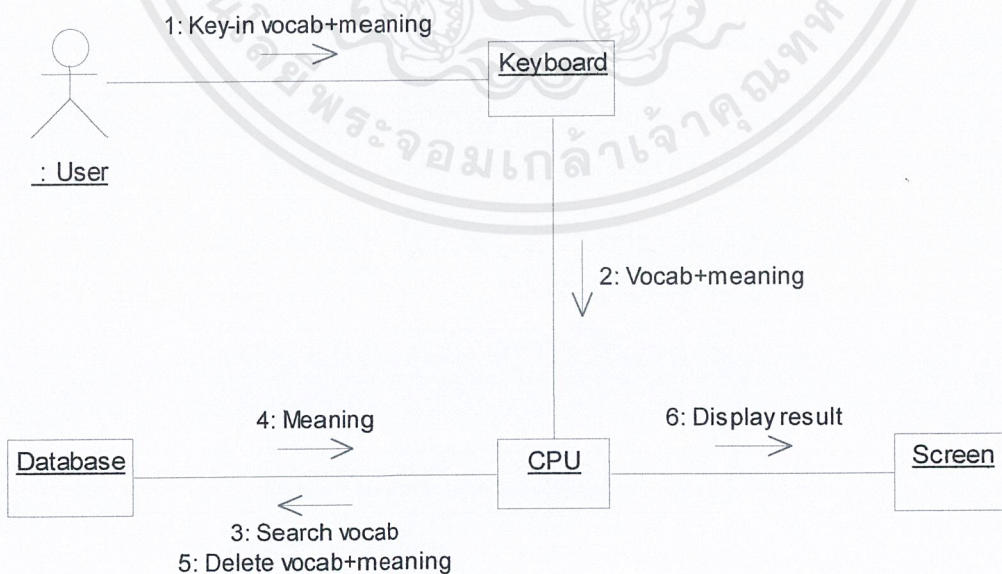
6.2.14 Delete-success

- Sequence Diagram



รูปที่ 6-28 แสดง Sequence Diagram ของ Delete-success Method

- Collaboration Diagram

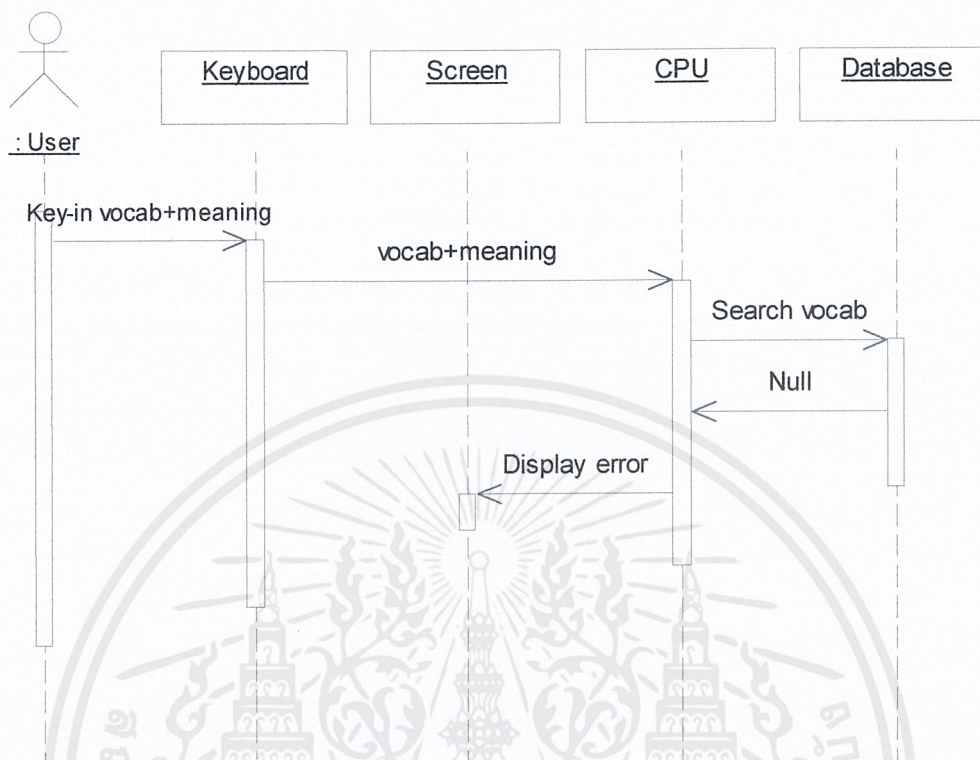


รูปที่ 6-29 แสดง Collaboration Diagram ของ Delete-success Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

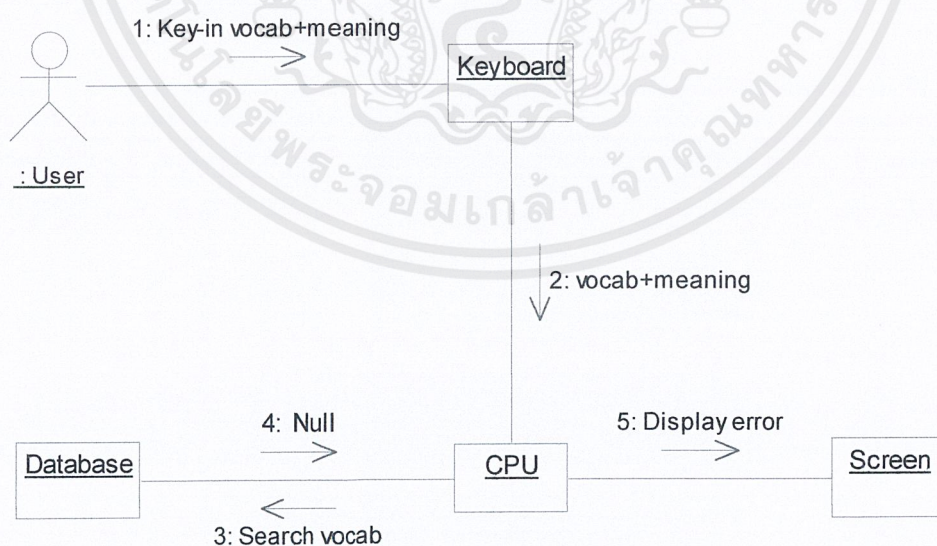
## 6.2.15 Delete-error

## - Sequence Diagram



รูปที่ 6-30 แสดง Sequence Diagram ของ Delete-error Method

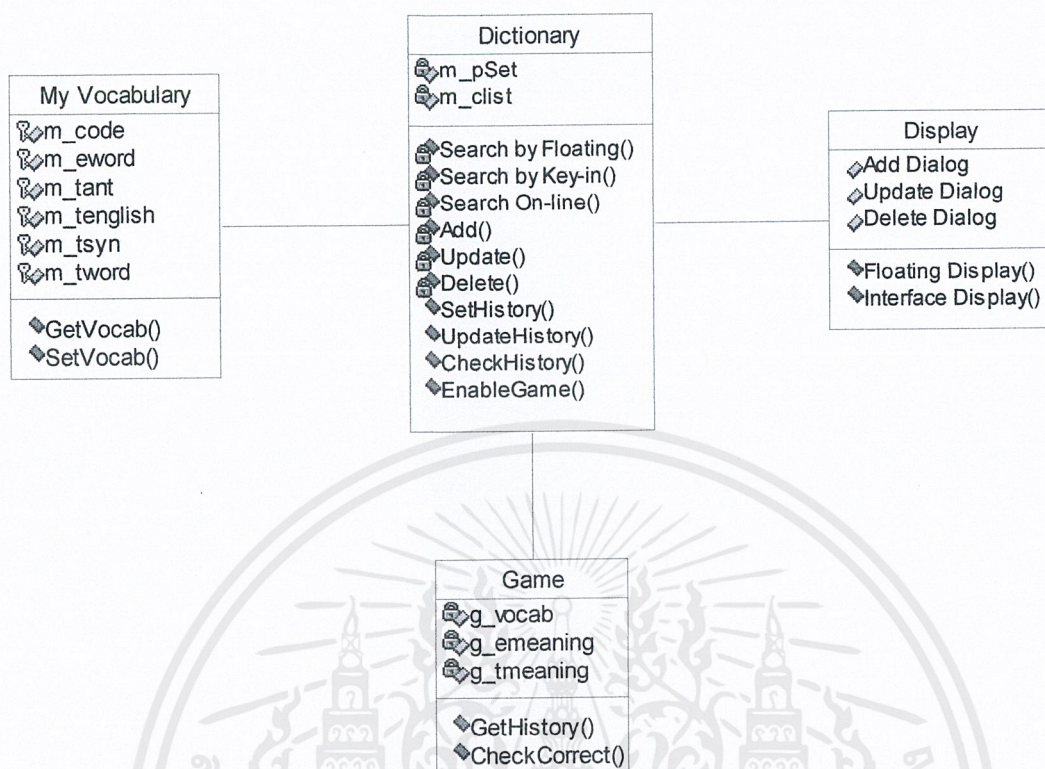
## - Collaboration Diagram



รูปที่ 6-31 แสดง Collaboration Diagram ของ Delete-error Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.3 Floating Thesaurus Class Diagram



รูปที่ 6-32 แสดง Class Diagram ของ Floating Thesaurus

Class Diagram ของ Floating Thesaurus แสดงการออกแบบความสัมพันธ์ระหว่างคลาสแต่ละคลาส

ดังนี้

#### Dictionary Class

คลาสนี้ทำหน้าที่ในการทำฟังก์ชันการทำงานหลัก ประกอบด้วยการ

- ค้นหาคำศัพท์แบบลอย (Search by Floating)
- การค้นหาคำศัพท์ในส่วนติดต่อกับผู้ใช้ (Search by Key-in)
- การค้นหาคำศัพท์ทางอินเทอร์เน็ต (Search On-line)
- การเพิ่มคำศัพท์ลงในฐานข้อมูล (Add)
- การแก้ไขคำศัพท์ในฐานข้อมูล (Update)
- การลบคำศัพท์ในฐานข้อมูล (Delete)

คลาสนี้ทำการขอข้อมูลในฐานข้อมูลจาก My Vocabulary Class และส่งข้อมูลเพื่อนำไปแสดงผลโดย

Display Class ดังนั้น Dictionary Class นี้จึงมีความสัมพันธ์กับคลาสอื่นๆแบบ association

นอกจากนี้ยังทำหน้าที่ในการตรวจสอบประวัติการค้นหาคำศัพท์ของผู้ใช้ ถ้าหากว่าผู้ใช้มีการค้นหาคำศัพท์ซ้ำภายในระยะเวลาที่กำหนด จะไปเรียกคลาส Game เพื่อแสดงเกมเพื่อทบทวนความหมายของคำศัพท์ให้กับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### My vocabulary Class

คลาสนี้ทำหน้าที่เม็ปดาต้าเมมเบอร์ กับเรคคอร์ดของข้อมูลในฐานข้อมูล เพื่อให้ Dictionary Class เรียกใช้ดาต้าเมมเบอร์ในการทำฟังก์ชันต่างๆ ทั้งนี้เพื่อไม่ให้มีการเรียกใช้ข้อมูลในฐานข้อมูลโดยตรง โดยเป็นการซ่อนข้อมูลเอาไว้เพื่อไม่ให้คลาสอื่นๆ สามารถแก้ไขข้อมูลในฐานข้อมูลได้โดยตรง ที่เรียกว่า Information Hiding

### Display Class

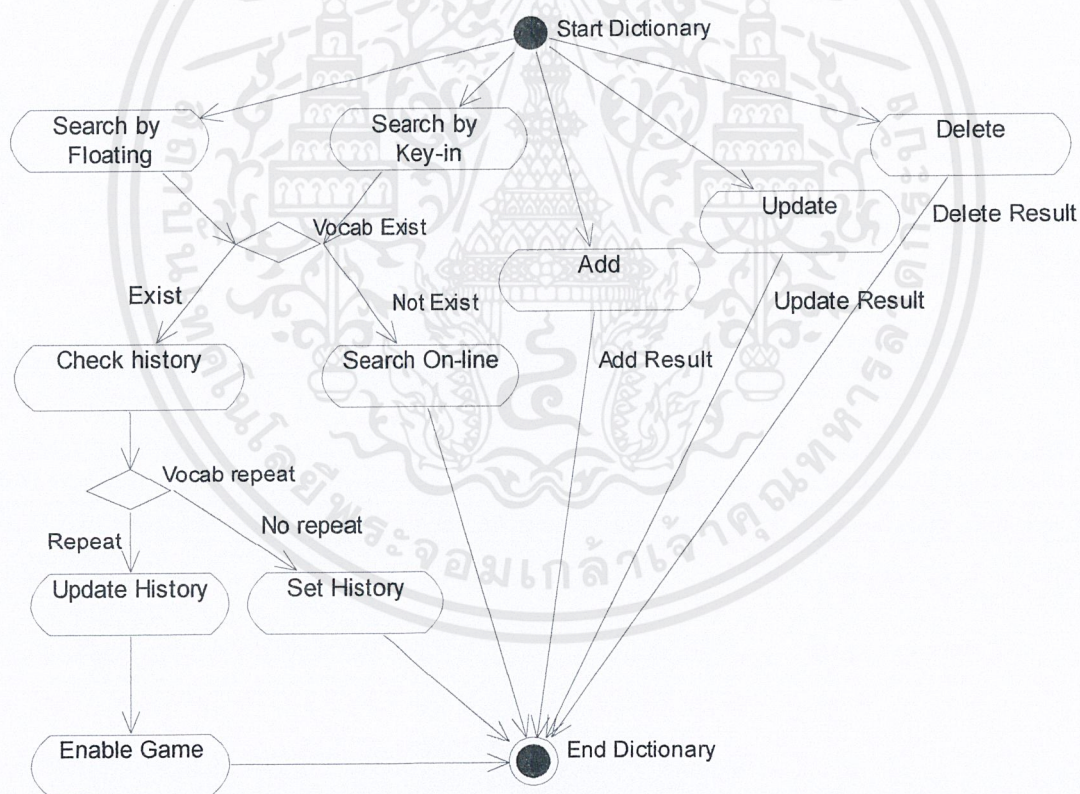
คลาสนี้ทำหน้าที่แสดงผลออกหน้าจอผ่านส่วนติดต่อกับผู้ใช้ และตำแหน่งที่เมาส์ชี้อยู่ในรูปแบบของหน้าต่าง Tool Tip

### Game Class

คลาสนี้จะเป็นการสร้างเกมที่ใช้ทบทวนคำศัพท์แก่ผู้ใช้

## 6.4 Floating Thesaurus Activity Diagram

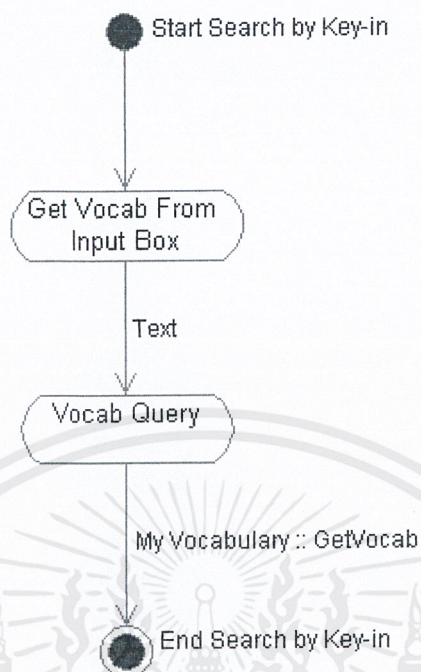
### 6.4.1 Dictionary Class



รูปที่ 6-33 แสดง Activity Diagram ของ Dictionary Class

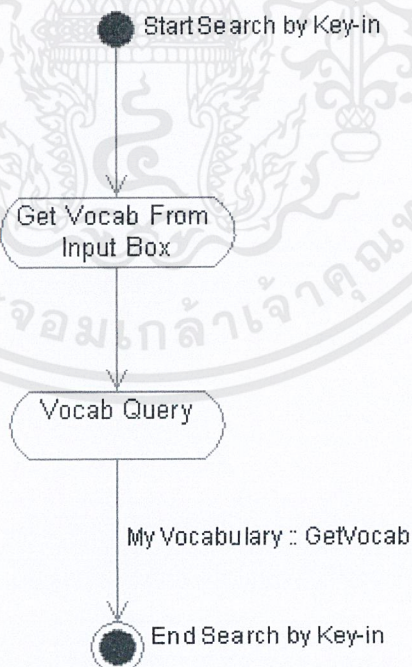
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 6.4.1.1. Search by Floating



รูปที่ 6-34 แสดง Activity Diagram ของ Search by Floating Method

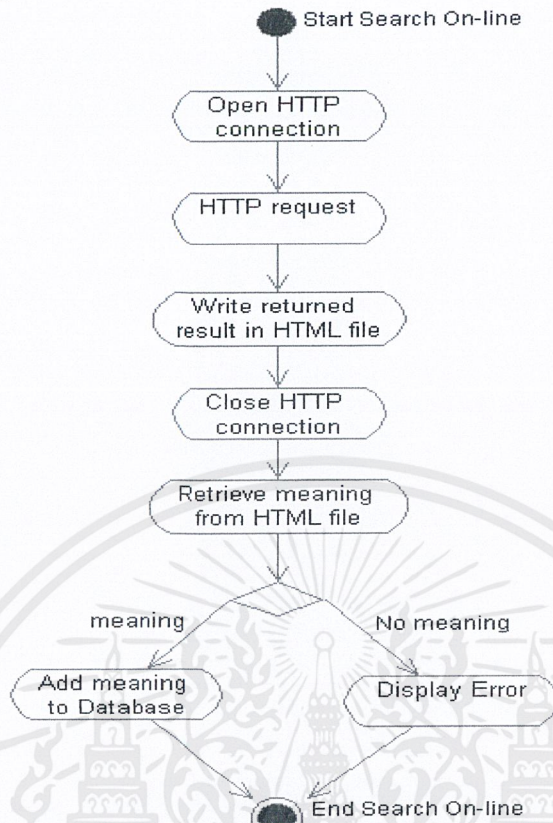
#### 6.4.1.2 Search by Key-in



รูปที่ 6-35 แสดง Activity Diagram ของ Search by Key-in Method

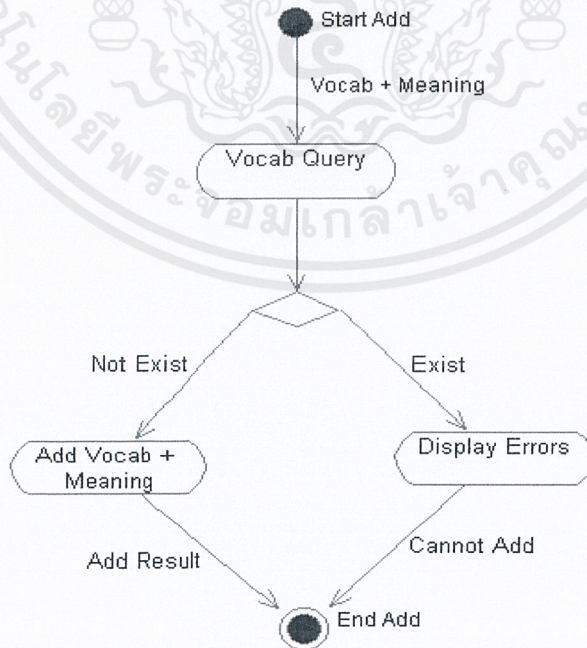
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.1.3 Search On-line



รูปที่ 6-36 แสดง Activity Diagram ของ Search On-line Method

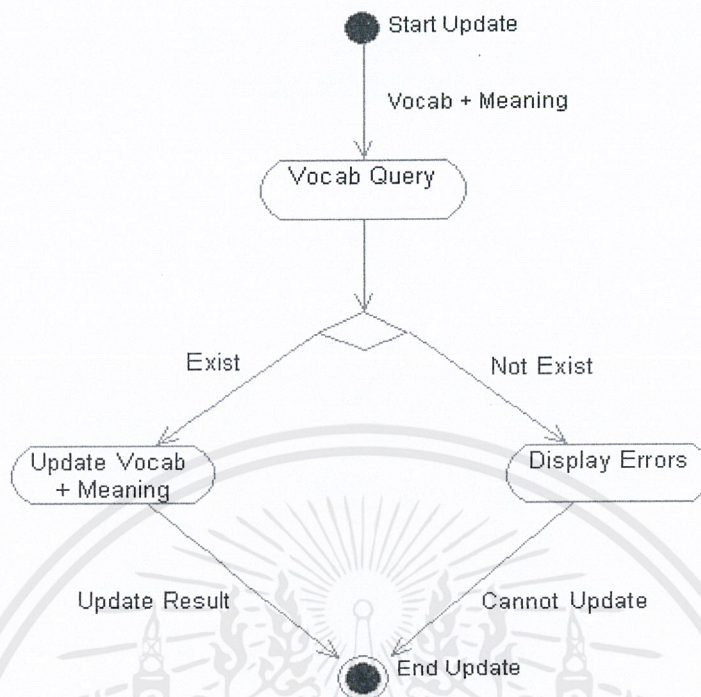
6.4.1.4 Add



รูปที่ 6-37 แสดง Activity Diagram ของ Add Method

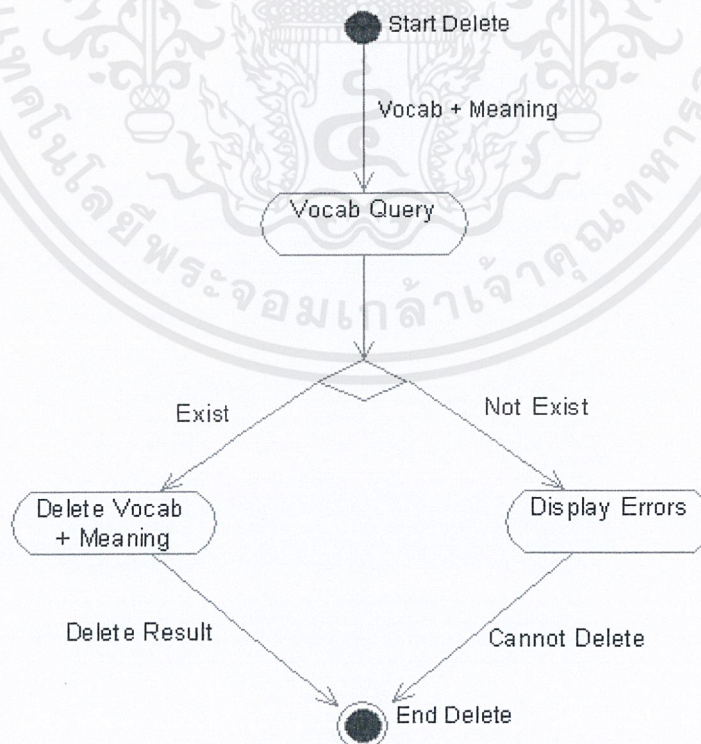
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.4.1.5 Update



รูปที่ 6-38 แสดง Activity Diagram ของ Update Method

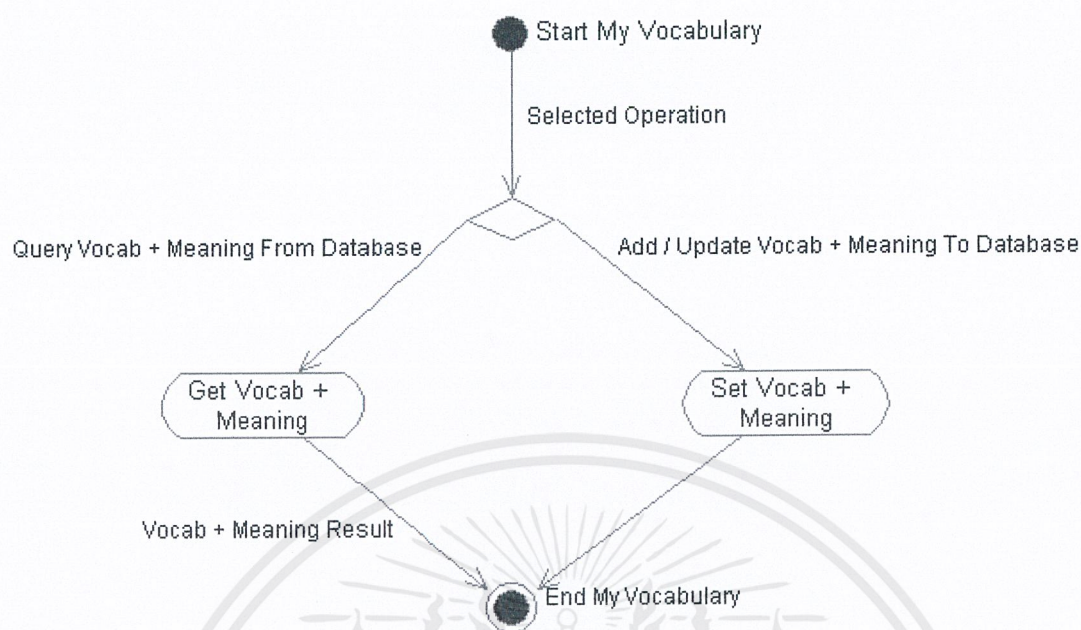
## 6.4.1.6 Delete



รูปที่ 6-39 แสดง Activity Diagram ของ Delete Method

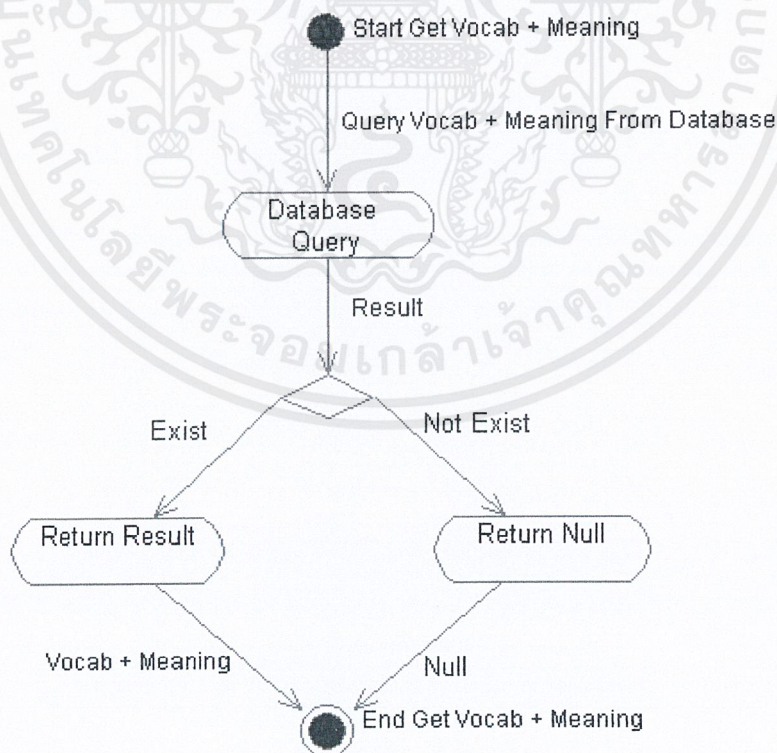
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6.4.2 My Vocabulary Class



รูปที่ 6-40 แสดง Activity Diagram ของ My Vocabulary Class

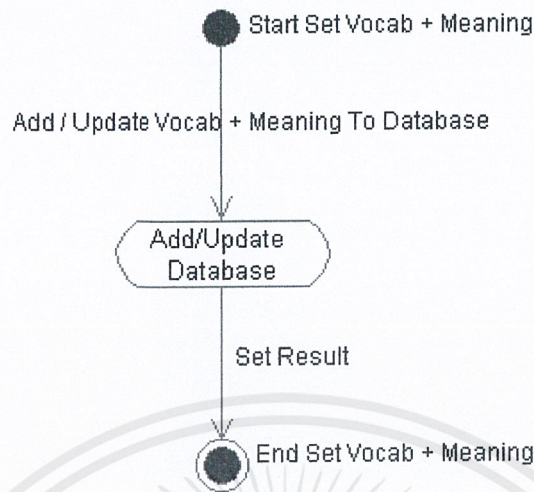
### 6.4.2.1 Get Vocab + Meaning



รูปที่ 6-41 แสดง Activity Diagram ของ Get Vocab + Meaning Method

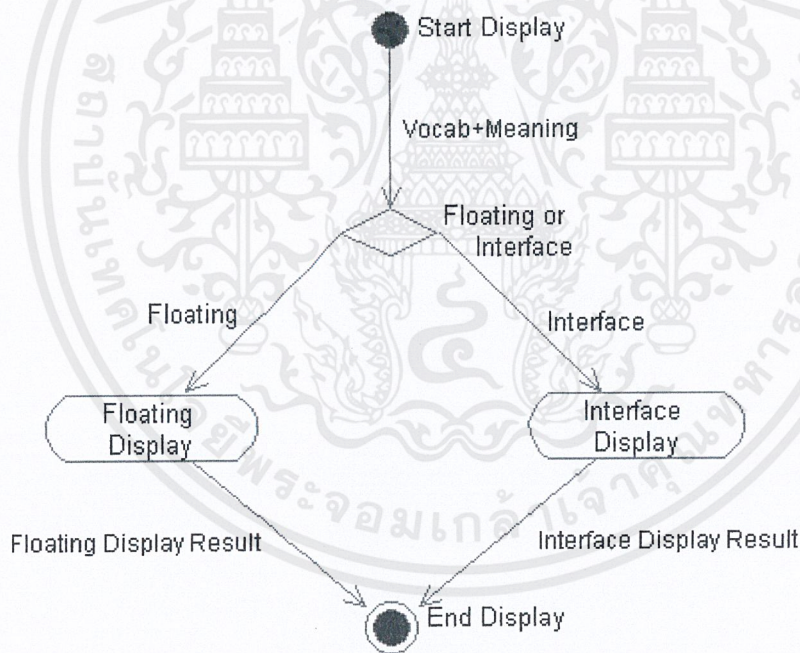
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.4.2.2 Set Vocab + Meaning



รูปที่ 6-42 แสดง Activity Diagram ของ Set Vocab + Meaning Method

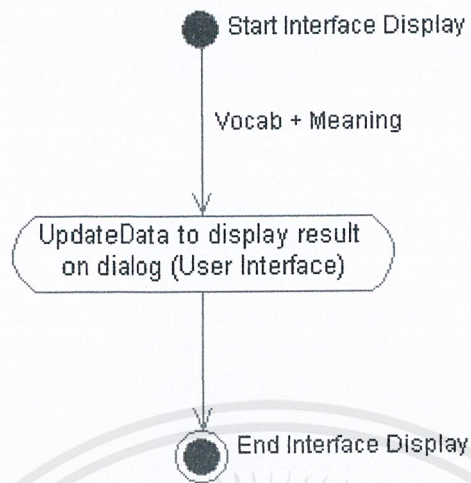
### 6.4.3 Display Class



รูปที่ 6-43 แสดง Activity Diagram ของ Display Class

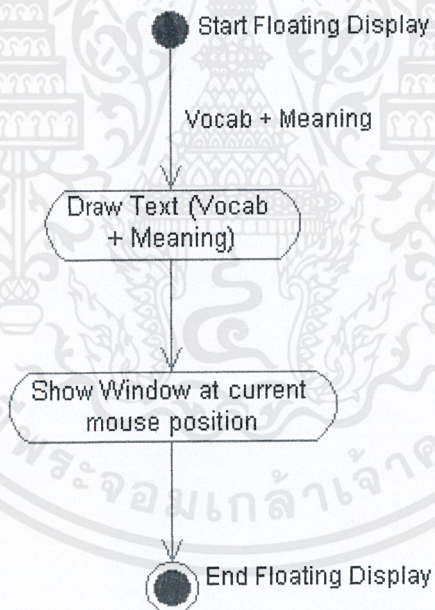
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 6.4.3.1 Interface Display



รูปที่ 6-44 แสดง Activity Diagram ของ Interface Display Method

### 6.4.3.2 Floating Display



รูปที่ 6-45 แสดง Activity Diagram ของ Floating Display Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## บทที่ 7

### การทำงานของโปรแกรมพจนานุกรมแบบลอย

#### 7.1 การทำงานของโปรแกรม

การทำงานของโปรแกรมพจนานุกรมแบบลอยนั้น แบ่งออกได้เป็น

1. การสร้างหน้าต่างหลักของโปรแกรม
2. การติดต่อกับฐานข้อมูล
3. การค้นหาคำศัพท์ เพิ่ม แก้ไข และลบคำศัพท์
4. การทบทวนคำศัพท์ และเกม
5. การติดต่อกับระบบปฏิบัติการเพื่อทำงานร่วมกับโปรแกรมอื่นๆ ได้

#### 7.1.1 การสร้างหน้าต่างหลักของโปรแกรม

การสร้างหน้าต่างหลักของโปรแกรมนั้น ประกอบด้วยคลาสดังนี้

##### 7.1.1.1 CMainFrm.h

คลาส CMainFrm เป็นคลาสที่ใช้สร้างหน้าต่างหลักของโปรแกรม มีฟังก์ชันการทำงานหลักดังนี้

##### - OnCreate

ทำหน้าที่สร้างหน้าต่างหลักของโปรแกรมพจนานุกรม โดยมีขั้นตอนการสร้างหน้าต่างดังนี้

1. สร้าง Frame Window หลักของโปรแกรม
2. สร้างทูลบาร์ (Toolbar)
3. สร้างสเตตัสบาร์ (Statusbar)
4. สร้างไอคอนของโปรแกรม

ผลลัพธ์ที่ได้จากฟังก์ชันนี้ คือค่า 0 และ -1 ถ้าหากว่าค่าที่คืนกลับมาเป็น 0 แสดงว่าสามารถสร้างหน้าต่างได้สำเร็จ แต่ถ้าหากเป็น -1 หน้าต่างนั้นจะถูกทำลาย

##### - OnShowIcon

ทำหน้าที่แสดงไอคอนบนซิสเต็มเทรย์ (SystemTray) โดยจะเรียกใช้จากฟังก์ชัน ShowIcon ของคลาส CSystemTray

##### - OnHideIcon

ทำหน้าที่ซ่อนไอคอนจากซิสเต็มเทรย์ โดยจะเรียกใช้จากฟังก์ชัน HideIcon ของคลาส CSystemTray

##### - OnUpdateShowIcon

จะตรวจสอบว่าที่ซิสเต็มเทรย์แสดงไอคอนของโปรแกรมอยู่หรือไม่ ถ้าหากว่าไม่ได้แสดงไว้จะทำการแสดงไอคอนที่ซิสเต็มเทรย์

- *OnUpdateHideIcon*  
จะตรวจสอบว่าที่ซิสเต็มเทรย์แสดงไอคอนของโปรแกรมอยู่หรือไม่ ถ้าหากว่าแสดงไว้จะทำการซ่อนไอคอนที่ซิสเต็มเทรย์
- *OnClose*  
ทำหน้าที่ลบไอคอนออกจากซิสเต็มเทรย์เมื่อปิดโปรแกรม โดยเรียกใช้จากคิสตรีคเตอร์ของคลาส CMainFrm
- *OnDemoMinimise*  
ทำหน้าที่ย่อหน้าต่างของโปรแกรมที่ปรากฏอยู่จากหน้าจอลงมาเป็นไอคอนบนซิสเต็มเทรย์โดยเรียกใช้จากฟังก์ชัน MinimiseToTray ของคลาส CSystemTray
- *OnDemoMaximise*  
ทำหน้าที่ขยายหน้าต่างของโปรแกรมจากไอคอนบนซิสเต็มเทรย์ให้แสดงอยู่บนหน้าจอ โดยจะเรียกใช้จากฟังก์ชัน MaximizeFromTray ของคลาส CSystemTray
- *OnUpdateMinimise*  
ตรวจสอบว่าหน้าต่างของโปรแกรมถูกย่อไว้หรือไม่ ถ้าหากว่าหน้าต่างของโปรแกรมนั้นถูกย่อไว้ อยู่ และสามารถแสดงหน้าต่างได้ จะทำการแสดงหน้าต่างของโปรแกรมบนหน้าจอ
- *OnUpdateMaximise*  
ตรวจสอบว่าหน้าต่างของโปรแกรมถูกย่อไว้หรือไม่ ถ้าหากว่าหน้าต่างของโปรแกรมนั้นถูกย่อไว้
- *OnHangman*  
ทำหน้าที่ในการเรียกเกมทบทวนคำศัพท์ (Hangman) ขึ้นมาให้ผู้ใช้เล่น

#### 7.1.1.2 CAboutDlg.h

คลาสนี้ทำหน้าที่สร้างหน้าต่าง About Dict ใช้สำหรับแสดงรายละเอียดต่างๆ ของโปรแกรม

#### 7.1.1.3 CSystemTray.h

คลาส CSystemTray นี้มีหน้าที่ทำฟังก์ชันที่เกี่ยวข้องกับซิสเต็มเทรย์ ซึ่งมีฟังก์ชันการทำงานดังนี้

- *Create*  
มีหน้าที่หลักในการสร้างเทรย์ไอคอน (Tray Icon) โดยจะต้องทำการตรวจสอบก่อนว่าระบบปฏิบัติการ (Operating System) สนับสนุนการสร้างเทรย์ไอคอนหรือไม่ ถ้าสนับสนุนแล้วจะมีขั้นตอนการสร้างดังนี้
  1. สร้างหน้าต่างของโปรแกรมในลักษณะป๊อปอัพ (POP-UP Window)
  2. กำหนดโครงสร้างของ NOTIFYICONDATA ซึ่งประกอบด้วยข้อมูลที่ระบบต้องการเพื่อใช้ในกระบวนการสร้างข้อความสถานะ (Message) สำหรับสถานะต่างๆ ของทาสก์บาร์ (Taskbar)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *SetTooltipText* และ *GetTooltipText*

ทำหน้าที่ในการกำหนดข้อความของ ToolTip และคืนค่าข้อความของ ToolTip โดยจะต้องตรวจสอบก่อนว่าระบบปฏิบัติการนั้นสนับสนุนการสร้างเทอร์ย์ไอคอน

- *SetIcon* และ *GetIcon*

เป็นฟังก์ชันที่ใช้ทำหน้าที่กำหนดค่าของเทอร์ย์ไอคอน และคืนค่าของเทอร์ย์ไอคอน ซึ่งในการกำหนดค่าเทอร์ย์ไอคอนนั้นจะต้องตรวจสอบว่าระบบปฏิบัติการนั้นสนับสนุนเทอร์ย์ไอคอน แล้วจึงกำหนดค่าของเทอร์ย์ไอคอน โดยส่งข้อความไปบอก Status area ของทาสก์บาร์ว่ามีการแก้ไขไอคอนใน Status area

- *SetStandardIcon*

สร้างไอคอนโดยจะโหลดไอคอนจากชื่อที่ได้รับเข้ามา ซึ่งจะโหลดไอคอนก็ต่อเมื่อไอคอนนั้นไม่เคยถูกโหลดมาก่อน ส่วนกรณีอื่นนั้นจะคืนค่าไอคอนที่มีอยู่แล้ว จากนั้นเรียกใช้ฟังก์ชัน *SetIcon* เพื่อกำหนดเทอร์ย์ไอคอน

- *ShowIcon* และ *HideIcon*

ทำหน้าที่แสดงไอคอน และลบไอคอนจากซิสเต็มเทอร์ย์โดยจะเรียกใช้ฟังก์ชัน *AddIcon* ในการแสดงและ *RemoveIcon* ในการลบไอคอน ซึ่งจะมีการตรวจสอบก่อนว่าระบบปฏิบัติการนั้นสนับสนุนเทอร์ย์ไอคอนหรือไม่ และเทอร์ย์ไอคอนถูกซ่อนหรือลบออกไปหรือไม่

- *AddIcon*

เป็นฟังก์ชันที่ทำหน้าที่สร้างไอคอนให้แสดงบนซิสเต็มเทอร์ย์ มีขั้นตอนการทำงานดังนี้

1. ตรวจสอบว่ามีเทอร์ย์ไอคอนแสดงอยู่หรือไม่ ถ้ามีอยู่ให้ลบไอคอนนั้นออกไปก่อน ซึ่งจะเรียกใช้ฟังก์ชัน *RemoveIcon*
2. กำหนดค่าแฟล็กอะเรย์ (Flag Array) ใน NOTIFYICONDATA ที่ใช้ระบุว่ามีข้อมูลที่ต้องการ ซึ่งในที่นี้จะเป็นค่าของเมสเสจไอเดนติไฟเออร์ (Message identifier) ของแอปพลิเคชัน ไอคอน และพอยน์เตอร์ที่ชี้ไปยังข้อความ Tooltip
3. ส่งข้อความไปยัง Status area ของทาสก์บาร์ว่าได้มีการเพิ่มไอคอนเข้าไป ผลลัพธ์ที่ได้จะคืนค่าจริง ถ้าหากว่าเพิ่มไอคอนได้สำเร็จ

- *RemoveIcon*

เป็นฟังก์ชันที่ใช้ในการลบไอคอนออกจากซิสเต็มเทอร์ย์ มีขั้นตอนการทำงานดังนี้

1. ตรวจสอบว่า ถ้าหากระบบปฏิบัติการไม่สนับสนุนเทอร์ย์ไอคอนหรือและไม่มีไอคอนแสดงอยู่จะออกจากฟังก์ชัน แต่ถ้าไม่จะทำขั้นต่อไป
2. กำหนดค่าแฟล็กอะเรย์ (Flag Array) ให้เป็น 0
3. ส่งข้อความไปยัง Status area ของทาสก์บาร์ว่าได้มีการลบไอคอนออกจาก Status area ผลลัพธ์ของฟังก์ชัน คืนค่าจริง เมื่อสามารถลบไอคอนได้สมบูรณ์ ส่วนในกรณีอื่น คืนค่าเท็จ

- *SetMenuDefaultItem* และ *GetMenuDefaultItem*  
ทำหน้าที่ในการเปลี่ยนแปลงดีฟอลต์เมนู (Default menu) ของแอปพลิเคชันซึ่งจะทำการกำหนดค่าโดยเรียกใช้ฟังก์ชัน *SetMenuDefaultItem* ของ MFC Class และคืนค่าเมนูของแอปพลิเคชันตามลำดับ
- *SetNotificationWnd* และ *GetNotificationWnd*  
ทำหน้าที่กำหนดให้พอยน์เตอร์ชี้ไปยังหน้าต่างเพื่อที่จะส่งข้อความไป และคืนค่าของหน้าต่างที่ต้องการจะส่งข้อความไป
- *SetTargetWnd* และ *GetTargetWnd*  
ทำหน้าที่เปลี่ยนแปลงหน้าต่างที่ต้องการจะส่งคำสั่งไป และคืนค่าของหน้าต่างที่กำหนดหรือหน้าต่างหลักของโปรแกรม
- *SetCallbackMessage* และ *GetCallbackMessage*  
ทำหน้าที่เปลี่ยนแปลง และคืนค่าของข้อความเพื่อแจ้งไปยังหน้าต่างของแอปพลิเคชัน
- *RemoveTaskbarIcon*  
ลงทะเบียนคลาส Window ของโปรแกรม และกำหนดให้พอยน์เตอร์ชี้ไปยัง Parent window ของโปรแกรม (Child window) ซึ่งถ้าหากว่าหน้าต่างของโปรแกรมแสดงอยู่ Window จะทำการวาดใหม่อีกครั้ง
- *MinimiseToTray*  
ทำหน้าที่ย่อหน้าต่างของโปรแกรม โดยจะเรียกใช้ฟังก์ชัน *RemoveTaskbarIcon* แล้วกำหนดค่าให้ Parent window สามารถมองเห็นได้
- *MaximiseFromTray*  
ทำหน้าที่ขยายหน้าต่างของโปรแกรมจากซิสเต็มเทรย์ มีขั้นตอนการทำงานดังนี้
  1. กำหนดให้ Parent window ของโปรแกรมเป็น NULL
  2. วาดหน้าต่างของโปรแกรมขึ้นมาใหม่
  3. กำหนดให้หน้าต่างของโปรแกรมแสดงอยู่บนหน้าจอ

#### 7.1.1.4 XInfoTip.h

ทำหน้าที่ในการสร้าง ToolTip ซึ่งเป็นส่วนที่ใช้แสดงผลการค้นหาคำศัพท์แบบลอย และใช้แสดงข้อมูลเพิ่มเติม (Tips) ให้กับโปรแกรม โดยมีฟังก์ชันการทำงานดังนี้

- *Create*  
ทำหน้าที่ในการสร้างหน้าต่าง ToolTip ให้กับ control ของโปรแกรม
- *AddTool*  
ทำหน้าที่กำหนดข้อความและไอคอนที่จะนำไปแสดงในหน้าต่าง ToolTip ของแต่ละ control ของโปรแกรม

- *RemoveTool*  
ทำหน้าที่ลบข้อมูลที่จะนำไปแสดงในหน้าต่าง ToolTip ของแต่ละ control ของโปรแกรม
- *Show*  
ทำหน้าที่กำหนดข้อความที่จะนำไปแสดงในหน้าต่าง ToolTip ตามตำแหน่งที่ต้องการ ในที่นี้คือตำแหน่งที่เมาส์ชี้อยู่ และกำหนดเวลาเริ่มแสดงหน้าต่าง ToolTip
- *Hide*  
ทำหน้าที่ซ่อนหน้าต่าง ToolTip เมื่อมีการเลื่อนเมาส์หรือมีการคลิกเมาส์
- *setIcon*  
ทำหน้าที่กำหนดขนาดของไอคอนที่จะวางลงไปบนหน้าต่าง ToolTip
- *SetFont*  
ทำหน้าที่กำหนดรูปแบบของตัวอักษรที่จะนำไปแสดงในหน้าต่าง ToolTip
- *RelayEvent*  
ทำหน้าที่ส่ง mouse message ไปยัง control ที่ต้องการแสดงหน้าต่าง ToolTip ฟังก์ชันนี้ถูกเรียกขึ้นมาใช้งานโดยฟังก์ชัน *PreTranslateMessage* ของคลาสที่สร้าง control ที่ต้องการแสดงหน้าต่าง ToolTip
- *GetWindowRegion*  
ทำหน้าที่คำนวณพื้นที่และกำหนดขนาดของหน้าต่าง ToolTip ที่จะนำไปแสดง
- *OnPaint*  
ทำหน้าที่วาดหน้าต่าง ToolTip ลงไปบนหน้าจอ โดยเรียกฟังก์ชัน *GetWindowRegion* เพื่อให้ทราบขนาดของหน้าต่าง ToolTip แล้วจึงวาดข้อความและไอคอนลงไป
- *OnTimer*  
ทำหน้าที่เริ่มแสดงหรือซ่อนหน้าต่าง ToolTip ตามเวลาที่กำหนดไว้ในฟังก์ชัน *SetTimer*
- *OnDestroy*  
ทำหน้าที่ลบเวลาที่ตั้งไว้เพื่อแสดงหน้าต่าง ToolTip และทำลายหน้าต่างด้วย

## 7.1.2 การติดต่อกับฐานข้อมูล

ในส่วนของการติดต่อกับฐานข้อมูลนั้น ทำงานโดยคลาส *CDictSet* ดังนี้

### 7.1.2.1 *CDictSet.h*

คลาส *CDictSet* เป็นคลาสที่สืบทอดมาจากคลาส *CDaoRecordset* ฟังก์ชันการทำงานประกอบด้วย

- *GetDefaultDBName*  
เป็นฟังก์ชันที่ใช้ในการคืนค่าที่อยู่ และชื่อของไฟล์ของฐานข้อมูลที่ใช้ใน *CDaoRecordset*
- *GetDefaultSQL*  
เป็นฟังก์ชันที่ใช้ระบุ SQL statement สำหรับฐานข้อมูลของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *DoFieldExchange*

ทำหน้าที่ในการแลกเปลี่ยนระหว่างค่าตัวเมมเบอร์ของออบเจกต์ กับคอลัมน์ของเรคคอร์ดจากฐานข้อมูลที่ใช้ในแอปพลิเคชัน

### 7.1.3 การค้นหาคำศัพท์ เพิ่ม แก้ไข และลบคำศัพท์

ในส่วนนี้โปรแกรมจะรับคำศัพท์เข้ามา เพื่อทำการค้นหาคำศัพท์ เพิ่ม แก้ไข และลบคำศัพท์ขึ้นอยู่กับความต้องการของผู้ใช้ ซึ่งการทำงานนั้นจะแบ่งออกเป็นหลายคลาสด้วยกัน ประกอบด้วย

1. คลาส *CDictView* เป็นคลาสที่ใช้แสดงเรคคอร์ดของฐานข้อมูลในคอนโทรล ซึ่งจะติดต่อกับออบเจกต์ของคลาสที่สืบทอดมาจาก *CDaoRecordSet* โดยตรง
2. คลาส *CSearchOnline* เป็นคลาสที่ใช้แสดงไดอะล็อก สำหรับการค้นหาคำศัพท์ผ่านอินเทอร์เน็ต
3. คลาส *CAdd* เป็นคลาสที่ใช้แสดงไดอะล็อก สำหรับการเพิ่มคำศัพท์ลงฐานข้อมูล
4. คลาส *CUpdate* และ *CUpdateBox* เป็นคลาสที่ใช้แสดงไดอะล็อกสำหรับการแก้ไขคำศัพท์
5. คลาส *CDelete* เป็นคลาสที่ใช้แสดงไดอะล็อก สำหรับการลบคำศัพท์ออกจากฐานข้อมูล

#### 7.1.3.1 *CDictView.h*

- *DoDataExchange*

ทำหน้าที่ในการแลกเปลี่ยนข้อมูลกับไดอะล็อกคานดาของโปรแกรม

- *OnGetRecordSet*

เป็นฟังก์ชันที่จะคืนค่าพอยน์เตอร์ของ *m\_pSet* ซึ่งเป็นเซตของข้อมูลของฐานข้อมูล

- *OnPreSearch*

ทำหน้าที่ในการค้นหาคำศัพท์ผ่านส่วนติดต่อกับผู้ใช้ โดยจะเรียกใช้ฟังก์ชันในการค้นหาคำศัพท์มีขั้นตอนการทำงานดังนี้

1. อัปเดตข้อมูลจากไดอะล็อกที่ผู้ใช้ใส่เข้ามาเพื่อต้องการค้นหาความหมายของคำศัพท์
2. เพิ่มคำศัพท์ใหม่ลงใน Combo box
3. ค้นหาคำศัพท์โดยเรียกใช้ฟังก์ชัน *CheckGrammar* ในการทำงานซึ่งถ้าหากว่าคำศัพท์ที่ต้องการค้นหาไม่มีอยู่ในฐานข้อมูลจะเรียกใช้ฟังก์ชัน *OnSearchNet* เพื่อค้นหาคำศัพท์ผ่านอินเทอร์เน็ต
4. แสดงความหมายของคำศัพท์ที่ได้มา ที่ส่วนติดต่อกับผู้ใช้ของโปรแกรม
5. ตรวจสอบประวัติในการหาคำศัพท์ของผู้ใช้ ถ้าหากว่าภายในเวลา 1 เดือนมีการหาคำศัพท์คำนี้มากกว่า 1 ครั้งจะสร้างออบเจกต์ของคลาส *C/gifAnimation* เพื่อติดต่อกับส่วนเกมเพื่อทบทวนคำศัพท์ที่ได้เคยค้นหาไปแล้ว

- *CheckGrammar*

ฟังก์ชันนี้มีหน้าที่หลักในการตรวจสอบคำศัพท์ที่ผู้ใช้ป้อนเข้ามา ซึ่งจะทำการแก้ไขให้ถูกต้องตามหลักไวยากรณ์ในภาษาอังกฤษแล้วจึงทำการค้นหาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *CheckHistory*  
ทำหน้าที่ในการตรวจสอบประวัติการค้นหาคำศัพท์ของผู้ใช้จากไฟล์ ซึ่งจะคืนค่าเป็นจริงเมื่อพบว่าคำศัพท์นั้นเคยถูกค้นหามาก่อน
- *KeepHistory*  
ทำหน้าที่ในการเก็บประวัติการค้นหาคำศัพท์ของผู้ใช้ลงไฟล์ ซึ่งจะลบข้อมูลของคำศัพท์ที่ได้ค้นหาไปแล้วเกิน 1 เดือนออกจากไฟล์ด้วย
- *ReadFile*  
อ่านค่าจากไฟล์ประวัติ เก็บลงอะเรย์ของโปรแกรม และจะคืนค่าจำนวนคำศัพท์ออกมาด้วย
- *OnSearch*  
ตรวจสอบจำนวนผลลัพธ์ที่ได้จากการค้นหาคำศัพท์ในฐานข้อมูล โดยในการค้นหาคำศัพท์จากฐานข้อมูลนั้นจะเรียกจากฟังก์ชัน Search และจะทำการเก็บประวัติการค้นหาคำศัพท์ของผู้ใช้ลงไฟล์
- *Search*  
เป็นฟังก์ชันที่ใช้ในการค้นหาคำศัพท์ในฐานข้อมูลของโปรแกรม ซึ่งจะใช้เคต้าเมมเบอร์ของคลาส CDaoRecordSet เพื่อกำหนดโครงสร้างของ WHERE Clause สำหรับ SQL Statement แล้วจึงเรียกการทำงานของ query เพื่อค้นหาคำศัพท์ นอกจากนี้แล้วฟังก์ชันนี้ยังทำการจัดรูปแบบการแสดงผลของความหมายของคำศัพท์อีกด้วย ผลลัพธ์ของฟังก์ชันจะคืนค่าจำนวนของความหมายที่พบในฐานข้อมูล
- *OnSearchNet*  
เป็นฟังก์ชันที่ใช้สำหรับส่วนติดต่อกับผู้ใช้ โดยจะเรียกฟังก์ชัน SearchOnline ทำงานเพื่อค้นหาคำศัพท์บนอินเทอร์เน็ต
- *SearchOnline*  
ทำหน้าที่ในการค้นหาคำศัพท์บนอินเทอร์เน็ต มีขั้นตอนในการทำงานดังนี้
  1. เชื่อมต่อกับอินเทอร์เน็ตโดยจะเรียกใช้ฟังก์ชัน InitConnection แล้วรับไฟล์ HTML มาเก็บไว้
  2. ตัดคำจากไฟล์ HTML ที่ได้มาเพื่อจะได้คำเหมือนของคำแปล
  3. เก็บคำเหมือนของคำศัพท์ลงฐานข้อมูล เรียกจากฟังก์ชัน NetVocab
  4. แสดงคำเหมือนของคำศัพท์ที่ได้จากการค้นหาผ่านอินเทอร์เน็ต
- *InitConnection*  
ทำหน้าที่ในการเปิด-ปิดการติดต่อเพื่อค้นหาคำศัพท์บนอินเทอร์เน็ต และเก็บผลที่ได้จากการค้นหาในรูปแบบของไฟล์ HTML
- *NetVocab*  
นำคำศัพท์ที่ได้จากการค้นหาบนอินเทอร์เน็ตมาจัดเรียงตามรูปแบบของฐานข้อมูล พร้อมทั้งเก็บความหมายของคำศัพท์ลงในฐานข้อมูลตามหน้าที่ของคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *OnLDbIClk*  
ทำหน้าที่ในการค้นหาคำศัพท์ เหมือนกับฟังก์ชัน OnPreSearch แต่จะรับอินพุตจาก Mouse Message โดยการดับเบิลคลิกที่คำศัพท์เป็นข้อความแล้วทำการค้นหาคำศัพท์จากฐานข้อมูล เรียกใช้ฟังก์ชัน CheckGrammar และ SearchOnline ในการค้นหาคำศัพท์
- *OnFloat*  
ทำหน้าที่เปิดและปิดการทำงานของการทำงานของการค้นหาคำศัพท์แบบลอยของโปรแกรม
- *OnAddVocab*  
สร้างออบเจกต์ของคลาส CAdd เพื่อรับคำศัพท์ที่ผู้ใช้ป้อนเข้ามาเพื่อเพิ่มคำศัพท์ลงในฐานข้อมูล ซึ่งจะเรียก AddVocab ในการเพิ่มคำศัพท์
- *AddVocab*  
ทำหน้าที่ในเพิ่มคำศัพท์ลงในฐานข้อมูลตามหน้าที่ของคำ ผลลัพธ์จะคืนค่าเป็นจริงก็ต่อเมื่อมีการเพิ่มคำศัพท์ลงในฐานข้อมูลได้เรียบร้อยแล้ว
- *OnUpdateVocab*  
สร้างออบเจกต์ของคลาส CUpdate และ CUpdateBox เพื่อรับคำศัพท์ที่ต้องการแก้ไข ต้องตรวจสอบว่าคำศัพท์ที่ผู้ใช้ต้องการแก้ไขนั้นมีอยู่ในฐานข้อมูลหรือไม่ ถ้าหากว่ามีจึงจะทำการแก้ไขคำศัพท์ตามที่ผู้ใช้ป้อนเข้ามา
- *OnDeleteVocab*  
สร้างออบเจกต์ของคลาส CDelete เพื่อรับคำศัพท์ที่ต้องการลบ ซึ่งจะตรวจสอบว่าคำศัพท์ที่ผู้ใช้ต้องการลบนั้นมีอยู่ในฐานข้อมูลหรือไม่ จึงจะทำการลบ
- *DeleteVocab*  
ลบคำศัพท์ที่ผู้ใช้ป้อนเข้ามาออกจากฐานข้อมูล ซึ่งผลลัพธ์ที่ได้จะคืนค่าเป็นจริงเมื่อสามารถลบคำศัพท์ได้สำเร็จ
- *OnMove*  
ทำหน้าที่ในการอัปเดตข้อมูลของเรคคอร์ดลงในฐานข้อมูลของโปรแกรม ซึ่งเป็นการยืนยันการเพิ่มเรคคอร์ดอย่างสมบูรณ์

### 7.1.3.2 CSearchOnline.h

เป็นคลาสที่สืบทอดมาจากคลาส CDialog มีหน้าที่ในการแสดงหน้าต่างไอคอนของโปรแกรม เพื่อแสดงการค้นหาคำศัพท์ผ่านอินเทอร์เน็ต มีฟังก์ชันในการทำงานดังนี้

- *OnOnlineOK*  
ฟังก์ชันนี้จะถูกเรียกมาทำงานเมื่อผู้ใช้งานมีการคลิกที่ปุ่ม OK ของไอคอน
- *OnOnlineCANCEL*  
ฟังก์ชันนี้จะถูกเรียกขึ้นมาทำงานเมื่อผู้ใช้งานมีการคลิกที่ปุ่ม CANCEL ของไอคอน หรือเมื่อผู้ใช้งานกดคีย์ Esc บนคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 7.1.3.3 CAdd.h

มีหน้าที่ในการสร้างหน้าต่างสำหรับการเพิ่มคำศัพท์ลงในฐานข้อมูล ซึ่งสมาชิกของคลาสจะเก็บความหมายของคำศัพท์ตามหน้าที่ของคำ

- *DoDataExchange*

ทำหน้าที่ในการแลกเปลี่ยนข้อมูลกับไดอะล็อกคั่วของโปรแกรม คือแลกเปลี่ยนข้อมูลที่ผู้ใช้ใส่เข้ามาในไดอะล็อก กับคั่วเมมเบอร์ของคลาส

- *OnAddok*

ฟังก์ชันนี้จะถูกเรียกมาทำงานเมื่อผู้ใช้มีการคลิกที่ปุ่ม OK ของไดอะล็อกเพิ่มคำศัพท์ และจะมีการอัปเดตข้อมูลจากไดอะล็อกที่ผู้ใช้ใส่เข้ามาเพื่อเพิ่มคำศัพท์ กับคั่วเมมเบอร์ของคลาสด้วย

### 7.1.3.4 CUpdate.h

มีหน้าที่ในการสร้างหน้าต่าง สำหรับการแก้ไขคำศัพท์ในฐานข้อมูล ซึ่งจะแสดงหน้าต่างเพื่อแก้ไขคำศัพท์ตามหน้าที่ของคำ มีฟังก์ชันในการทำงานคือ

- *DoDataExchange*

ทำหน้าที่ในการแลกเปลี่ยนข้อมูลกับ ไดอะล็อกคั่วของ โปรแกรม คือแลกเปลี่ยนข้อมูลที่ผู้ใช้ใส่เข้ามาในไดอะล็อกเพื่อแก้ไขคำศัพท์ กับคั่วเมมเบอร์ของคลาส

### 7.1.3.5 CUpdate.Box.h

มีหน้าที่ในการสร้างหน้าต่าง สำหรับป้อนคำศัพท์ที่ผู้ใช้ต้องการแก้ไข

- *DoDataExchange*

ทำหน้าที่ในการแลกเปลี่ยนข้อมูลกับ ไดอะล็อกคั่วของ โปรแกรม ซึ่งจะแลกเปลี่ยนคำศัพท์ที่ผู้ใช้ป้อนเข้ามาในไดอะล็อก กับคั่วเมมเบอร์ของคลาส

### 7.1.3.6 CDelete.h

เป็นคลาสที่สืบทอดมาจากคลาส CDialog มีหน้าที่ในการสร้างไดอะล็อกสำหรับป้อนคำศัพท์ที่ผู้ใช้ต้องการลบออกจากฐานข้อมูล

- *DoDataExchange*

ทำหน้าที่ในการแลกเปลี่ยนข้อมูลกับ ไดอะล็อกคั่วของ โปรแกรม ซึ่งจะแลกเปลี่ยนคำศัพท์ที่ผู้ใช้ป้อนเข้ามาในไดอะล็อกเพื่อต้องการจะลบ กับคั่วเมมเบอร์ของคลาส

## 7.1.4 การทบทวนคำศัพท์ และเกม

การทบทวนคำศัพท์และเกมนี้ เป็นส่วนที่ใช้ในหน้าต่างเพื่อถามผู้ใช้งานว่าต้องการเล่นเกมทบทวนคำศัพท์หรือไม่ ถ้าหากว่าต้องการจึงจะสร้างออบเจกต์ของเกมขึ้นมา ประกอบด้วยคลาสต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 7.1.4.1 CPictureEx.h

มีหน้าที่ในการแสดงรูปภาพออกไปยังหน้าต่าง ไดอะล็อกของโปรแกรม มีฟังก์ชันการทำงานดังนี้

##### - Load

ทำหน้าที่ในการโหลดภาพ GIF และเตรียมออบเจกต์สำหรับการวาดภาพ ซึ่งฟังก์ชันในการโหลดจะสามารถทำได้ 3 แบบด้วยกันคือ

1. โหลดรูปภาพจากชื่อไฟล์ที่รับเข้ามา
2. ในการโหลดแบบนี้จะรับ handle จาก global memory block เพื่อโหลดรูปภาพ และ
3. โหลดโดยรับมาเป็นชื่อของ resource และชื่อชนิดของ resource นั้น

ซึ่งผลลัพธ์ที่ได้จะคืนค่าจริงเมื่อสามารถโหลดรูปภาพได้สำเร็จ

##### - Draw

ฟังก์ชันนี้มีหน้าที่คือ วาดภาพ และแสดงแอนิเมชันของภาพ ซึ่งในการวาดภาพจะแบ่งออกเป็น 2 กรณีด้วยกันคือ ภาพแอนิเมชันและภาพที่ไม่ใช่แอนิเมชัน

##### - Stop

ทำหน้าที่ในการหยุดแสดงภาพแอนิเมชัน

##### - UnLoad

ทำหน้าที่ในการหยุดแสดงภาพแอนิเมชัน ซึ่งจะเรียกฟังก์ชัน Stop ทำงานพร้อมทั้งคืน resource ทั้งหมดที่ใช้

##### - SetBkColor และ GetBkColor

ทำหน้าที่ในการกำหนดค่าสีของรูปภาพในส่วนที่เหลือของภาพ ที่ไม่ได้กำหนดเอาไว้ และคืนค่าสีที่ใช้เป็น COLORREF คือค่าสี RGB

##### - IsGIF

ฟังก์ชันนี้คืนค่าเป็นจริงก็ต่อเมื่อ ภาพที่ใช้แสดงนั้นมีชนิดของรูปภาพเป็น GIF

##### - IsAnimatedGIF

คืนค่าเป็นจริงเมื่อภาพที่ใช้เป็นภาพแอนิเมชัน ซึ่งมีชนิดของรูป GIF

##### - IsPlaying

ทำหน้าที่ในการตรวจสอบการแสดงผลของภาพนั้น โดยจะคืนค่าเป็นจริงเมื่อภาพที่แสดงอยู่นั้นเป็นภาพแอนิเมชัน

##### - GetSize

ฟังก์ชันนี้ใช้หาค่าขนาดความสูงและความยาวของภาพ โดยผลลัพธ์ที่ได้เป็น SIZE

##### - GetFrameCount()

ตรวจสอบว่าภาพที่แสดงอยู่นั้นเป็นภาพแอนิเมชันอยู่หรือไม่ ถ้าหากว่าใช่จะนับจำนวนเฟรมของภาพที่ใช้แสดง แล้วคืนค่านั้นเป็นจำนวนเต็ม

##### - GetPaintRect

ทำหน้าที่คืนค่าพื้นที่ที่ใช้ในการวาดรูปอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *SetPaintRect*  
ใช้ในการกำหนดพื้นที่ในการวาดรูป

#### 7.1.4.2 CGifAnimation.h

ทำหน้าที่ในการสร้างไคอะลือกเพื่อแสดงรูปภาพสำหรับถามผู้ใช้งานว่าต้องการเล่นเกมเพื่อทบทวนความหมายของคำศัพท์ที่ลืมไปหรือไม่ โดยมีฟังก์ชันการทำงานดังนี้

- *DoDataExchange*  
ทำหน้าที่ในการแลกเปลี่ยนข้อมูลกับไคอะลือกคตาต้าของโปรแกรม ซึ่งในที่นี้จะแลกเปลี่ยนระหว่างรูปภาพที่ต้องการแสดงในไคอะลือก
- *OnSetCursor*  
กำหนดค่าของเคอร์เซอร์เมื่อเลื่อนเมาส์มาตำแหน่งของภาพที่แสดงอยู่ในกรอบหน้าต่าง
- *OnLButtonDown*  
ทำหน้าที่ในการเรียกเกมทบทวนคำศัพท์ โดยสร้างออบเจกต์ของ CHangmanTutor เมื่อมีการคลิกบนภาพที่ปรากฏในกรอบหน้าต่าง

#### 7.1.4.3 CHangmanTutor.h

คลาสนี้เป็นคลาสที่จะแสดงเกม Hangman เพื่อจะช่วยให้ทบทวนความหมายของคำศัพท์ให้กับผู้ใช้ จะช่วยให้ผู้ใช้จำคำศัพท์ได้มากยิ่งขึ้น

- *DoDataExchange*  
ทำหน้าที่ในการแลกเปลี่ยนข้อมูลกับไคอะลือกคตาต้าของโปรแกรม ซึ่งในที่นี้จะแลกเปลี่ยนระหว่างคำศัพท์ที่จะใช้เพื่อทบทวนความหมายของคำศัพท์ให้กับผู้ใช้
- *OnInitDialog*  
ทำหน้าที่ในการกำหนดค่าเริ่มต้นต่างๆ ของเกม ซึ่งจะเรียกฟังก์ชัน ReadFile ให้อ่านไฟล์และ InitGame ให้เลือกคำศัพท์มาแสดง
- *ReadFile*  
ทำหน้าที่ในการอ่านไฟล์ที่เก็บข้อมูลของคำศัพท์ที่เคยค้นหามาก่อนหน้านี้ภายใน 1 เดือน
- *InitGame*  
เลือกคำศัพท์ที่อยู่ในไฟล์มาเล่นเกมโดยการเดาคู่ ม แสดงค่าต่างๆ ออกทางหน้าต่างของเกม และ กำหนดค่าเริ่มต้นของตัวอักษรที่ถูกใช้ไปแล้วในเกมจากฟังก์ชัน InitUsed
- *OnChar*  
ตรวจสอบปุ่มที่ผู้ใช้กดบนแป้นคีย์บอร์ด ซึ่งถ้าหากเป็นตัวอักษรจะเรียกฟังก์ชัน PlayGame

- *PlayGame*

ทำหน้าที่หลักของการเล่นเกม มีขั้นตอนการทำงานดังนี้

1. ตรวจสอบตัวอักษรที่กดเข้ามาว่าเป็นตัวอักษรที่ไม่เคยใช้และมีอยู่ในคำศัพท์หรือไม่ ถ้าไม่มีให้ลบจำนวนครั้งในการเล่นลง 1 ครั้งจากนั้นแสดงผลทางหน้าต่างของเกม
2. ตรวจสอบว่าเล่นเกมจบหรือยัง ถ้าจบแล้วให้ตรวจสอบต่อว่าเล่นชนะหรือแพ้ ถ้าเล่นแพ้ให้แสดงคำตอบที่ถูกต้อง
3. ถามผู้ใช้งานที่ต้องการเล่นเกมต่อหรือไม่ ถ้าต้องการจะเรียกฟังก์ชัน `InitGame` แต่ถ้าไม่จะออกจากเกม โดยใช้ฟังก์ชัน `OnClose`

- *CheckVocab*

ทำหน้าที่ในการตรวจสอบตัวอักษรที่ผู้ใช้ป้อนเข้ามา คำนวณเป็นจริงเมื่อมีอักษรนั้นอยู่ในคำศัพท์

- *DrawHangman*

ทำหน้าที่ในการแสดงรูปภาพของเกม Hangman โดยจะเรียกใช้จากคลาส `CPictureEx`

- *OnClose*

สร้างออบเจกต์ของคลาส `CHangResult` เพื่อแสดงผลสรุปของการเล่นเกม

### 7.1.5 การติดต่อกับระบบปฏิบัติการเพื่อทำงานร่วมกับโปรแกรมอื่นๆได้

การเขียนโปรแกรมให้สามารถติดต่อกับระบบปฏิบัติการเพื่อทำงานร่วมกับโปรแกรมอื่นๆได้ เมื่อมีอีเวนต์การดับเบิลคลิกในโปรแกรมอื่นๆนั้น จำเป็นต้องเขียนส่วนที่ทำการติดต่อกับระบบปฏิบัติการในรูปของ DLL (Dynamic Link Library) โดยมีฟังก์ชันการทำงานดังนี้

- *DllMain*

เมื่อมีการเรียกใช้ DLL การทำงานของ DLL จะเริ่มต้นที่ `DllMain` นี้ โดยทำหน้าที่ลงทะเบียนวินโดวส์แมสเชจเฉพาะของโปรแกรม Floating Thesaurus กับทุกๆ โพรเซสที่กำลังทำงานอยู่, ทำการแชร์วินโดวส์แฮนเดิล (Windows Handle) และทำการถอน hook procedure ที่ได้ติดตั้งแล้ว ออกจาก hook chain

- *SetMyHook*

ทำหน้าที่ติดตั้ง hook procedure ที่กำหนดขึ้นเองลงใน hook chain เพื่อทำการตรวจสอบแมสเชจอีเวนต์ใดๆที่เกิดขึ้นในระบบ

- *ClearMyHook*

ทำหน้าที่ถอน hook procedure ที่ได้ติดตั้งแล้ว ออกจาก hook chain และยกเลิกการแชร์วินโดวส์แฮนเดิล

- *Msghook*

ทำหน้าที่ตรวจสอบว่าแมสเชจอีเวนต์ที่เกิดขึ้นในระบบคือ การดับเบิลคลิกหรือไม่ แล้วทำการส่งแมสเชจเฉพาะของโปรแกรม Floating Thesaurus ไปที่แมสเชจคิวของระบบ เพื่อให้โปรแกรมดักจับแมสเชจนี้เพื่อนำไปค้นหาคำศัพท์แบบลอยต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.2 คุณสมบัติของโปรแกรม

โปรแกรมพจนานุกรมอังกฤษ-อังกฤษแบบลอยนั้น มีความสามารถในการทำงาน ดังนี้

1. ค้นหาความหมายของคำศัพท์จากฐานข้อมูลมาแสดง ซึ่งจะสามารถรับคำศัพท์มาค้นหาได้ใน 2 ลักษณะด้วยกัน คือ
  - จากส่วนติดต่อกับผู้ใช้
  - จากการดับเบิลคลิกที่คำศัพท์บนหน้าจอคอมพิวเตอร์ที่ต้องการค้นหา
2. หากคำศัพท์ที่ต้องการค้นหานั้น ไม่มีอยู่ในฐานข้อมูล จะติดต่อไปยังเว็บไซต์ [www.babylon.com](http://www.babylon.com) ที่ให้บริการค้นหาคำศัพท์แบบออนไลน์ แล้วนำมาเก็บไว้ในฐานข้อมูลของโปรแกรม
3. สามารถเพิ่ม แก้ไข และลบคำศัพท์จากฐานข้อมูลของโปรแกรมได้
4. มีการเก็บประวัติการค้นหาคำศัพท์ ซึ่งถ้าหากมีการค้นหาคำศัพท์คำนั้นมากกว่า 1 ครั้งภายใน 1 เดือนจะมีเกม Hangman ที่จะช่วยทบทวนคำศัพท์ให้กับผู้ใช้
5. เป็นโปรแกรมที่ฝังตัวกับระบบปฏิบัติการ ซึ่งสามารถทำงานได้กับบราวเซอร์
6. สามารถ Disable โปรแกรมพจนานุกรมอังกฤษ-อังกฤษแบบลอยได้
7. สามารถใช้โปรแกรมนี้ได้กับระบบปฏิบัติการ Windows 9X, ME และ 2000

## 7.3 ข้อจำกัดของโปรแกรม

โปรแกรมพจนานุกรมอังกฤษ-อังกฤษแบบลอยที่สร้างขึ้นนี้ มีข้อจำกัดของการใช้งานอยู่บ้างบางประการ ดังต่อไปนี้

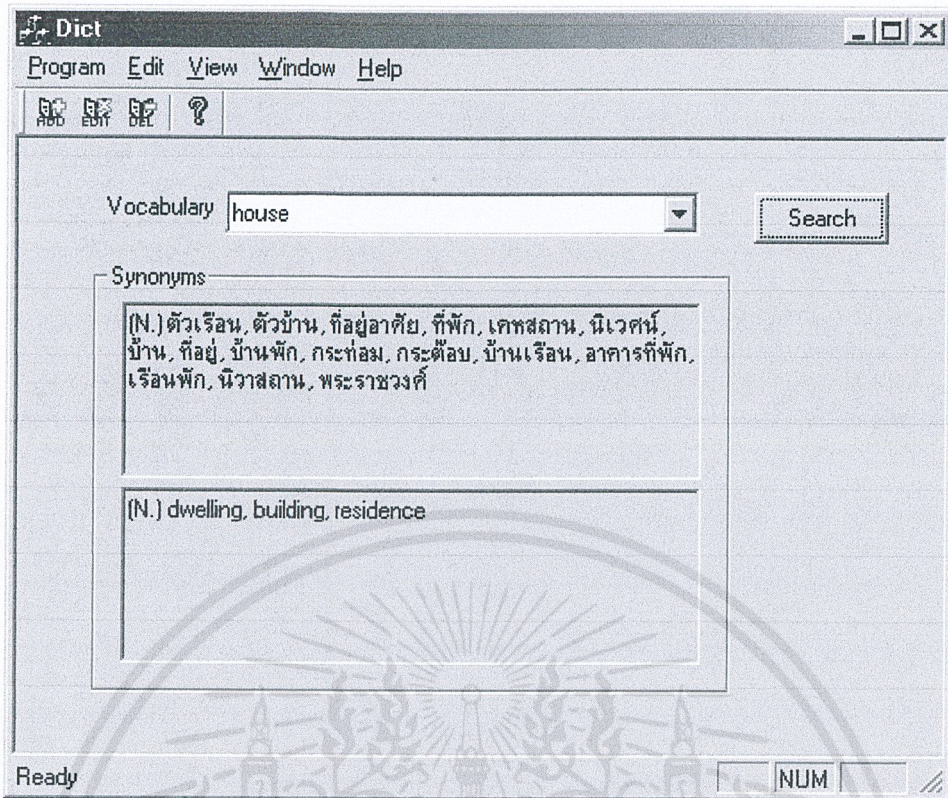
1. ตอนเริ่มต้นการใช้งานจะรู้สึกว่โปรแกรมพจนานุกรมนี้นี้คำศัพท์น้อยไปบ้าง เนื่องจากคำศัพท์บางตัวไม่มีในฐานข้อมูลของโปรแกรม จึงต้องทำการเชื่อมต่อกับอินเทอร์เน็ตเพื่อค้นหาคำศัพท์แล้วนำมาเก็บไว้ในฐานข้อมูลของโปรแกรม
2. ในการติดต่อกับอินเทอร์เน็ตนั้น ถ้าหากว่าเว็บไซต์ [www.babylon.com](http://www.babylon.com) ที่เราเลือกใช้ในการการค้นหาคำศัพท์แบบออนไลน์ ไม่สามารถให้บริการได้ในช่วงเวลาหนึ่ง จะทำให้โปรแกรมไม่สามารถใช้ความสามารถในการค้นหาคำศัพท์แบบออนไลน์ได้
3. ไม่สามารถค้นหาคำศัพท์ด้วยการชี้เมาส์ไปยังคำศัพท์นั้นได้ เพราะไม่มีฟังก์ชันรับคำจากหน้าต่างในภาษาที่เลือกใช้ในการเขียนโปรแกรม ต้องทำการเปลี่ยนจากบิตแมปเป็นตัวอักษรด้วย OCR (Optical Character Recognition) ซึ่งเป็นการยุ่งยาก และอยู่นอกขอบเขตของโครงการ

## 7.4 การใช้งานโปรแกรมพจนานุกรม

การใช้งานโปรแกรมพจนานุกรมอังกฤษ-อังกฤษแบบลอย สามารถเรียกใช้ได้โดยการเรียกแอปพลิเคชัน Dict.exe ซึ่งจะแสดงไอคอนที่ซิสเต็มเทรย์ ซึ่งในการค้นหาคำศัพท์นั้นจะสามารถทำได้ 2 วิธีคือ

1. ส่วนติดต่อกับผู้ใช้ โดยป้อนคำศัพท์ลงในช่อง Vocabulary แล้วคลิกปุ่ม Search จะแสดงความหมายของคำศัพท์ที่หน้าต่างของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-1 แสดงหน้าต่างค้นหาคำศัพท์ของโปรแกรมพจนานุกรม

- แบบลอย เมื่อผู้ใช้ดับเบิลคลิกคำศัพท์ที่ต้องการค้นหาบนหน้าจอคอมพิวเตอร์จะแสดงความหมายของคำศัพท์ตรงตำแหน่งนั้น ซึ่งจะต้องมีการ Enable โปรแกรมก่อนจึงจะสามารถใช้ได้ วิธีการ Enable นั้นก็เพียงแค่คลิกที่เมนู แล้วเลือก Floating Thesaurus

#### Parameters

This message has no parameters.

Ref  message

(N.) สาร, จดหมาย, หนังสือ, ข่าวคราว, ข้อความ  
(N.) letter, word, document

Remarks

When sent to a combo box, the **WM\_COPY** message is handled by its edit control. This message has no effect when sent to a combo box with the CBS\_DROPDOWNLIST style.

#### Requirements

**Windows NT/2000:** Requires Windows NT 3.1 or later.  
**Windows 95/98:** Requires Windows 95 or later.  
**Header:** Declared in Winuser.h; include Windows.h.

#### See Also

[Clipboard Overview](#), [Clipboard Messages](#), [WM\\_CLEAR](#), [WM\\_CUT](#), [WM\\_PASTE](#)

Built on Thursday, November 11, 1999

รูปที่ 7-2 แสดงการค้นหาคำศัพท์แบบลอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

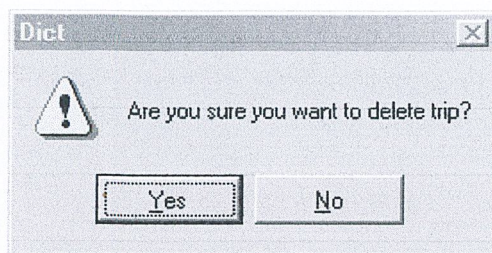
สำหรับการเพิ่ม แก้ไข และลบคำศัพท์จากฐานข้อมูลนั้นจะสามารถทำได้เฉพาะในส่วนติดต่อกับผู้ใช้เท่านั้น โดยการคลิกที่ปุ่ม ADD, EDIT และ DEL ตามลำดับ ในการเพิ่มและแก้ไขคำศัพท์นั้นจะให้ผู้ใช้ใส่คำเหมือนในภาษาอังกฤษ และความหมายของคำในภาษาไทยตามหน้าที่ของคำ ซึ่งจะมีหน้าจอการทำงานดังนี้

รูปที่ 7-3 แสดงหน้าต่างการเพิ่มและแก้ไขคำศัพท์ตามหน้าที่ของคำ

ในการแก้ไขคำศัพท์นั้น ก่อนที่จะมีการเข้าสู่หน้าต่างที่ใช้แก้ไขความหมายของคำศัพท์ตามหน้าที่ของคำนั้น จะให้ใส่คำศัพท์ที่ต้องการค้นหา ก่อน ซึ่งถ้าหากคำศัพท์นั้นไม่ปรากฏอยู่ในฐานข้อมูลของโปรแกรม จะแสดงข้อความเตือนให้ผู้ใช้ทราบ

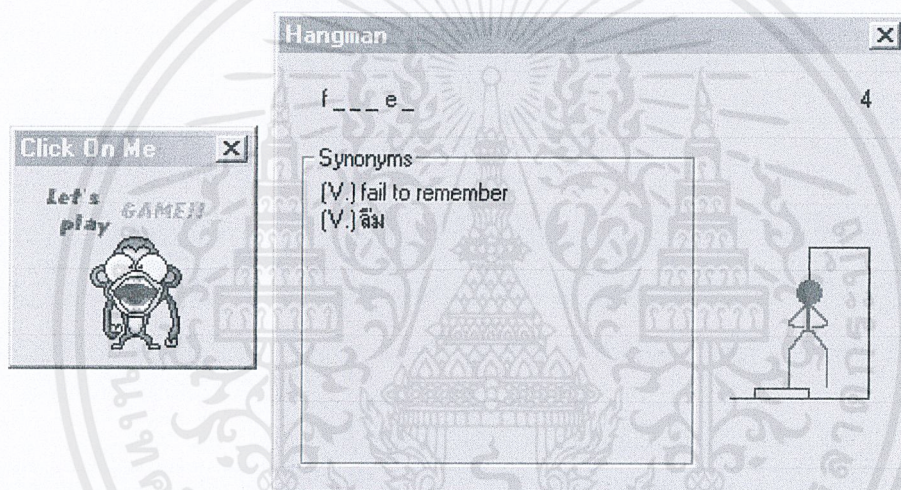
รูปที่ 7-4 แสดงหน้าต่างคำศัพท์ที่ต้องการแก้ไข

สำหรับการลบคำศัพท์ออกจากฐานข้อมูลนั้น จะให้ผู้ใช้ป้อนคำศัพท์ที่ต้องการลบนั้น โดยจะมีหน้าต่างเหมือนกันกับ หน้าต่างที่ใช้ใส่คำศัพท์ที่ต้องการแก้ไขในการแก้ไขคำศัพท์ หลังจากนั้นจะแสดงหน้าต่างขอคำยืนยันจากผู้ใช้ว่าต้องการลบคำศัพท์แน่นอน แล้วจึงลบคำศัพท์นั้นออกจากฐานข้อมูล



รูปที่ 7-5 แสดงหน้าต่างขอคำยืนยันจากผู้ใช้ในการลบคำศัพท์

อีกทั้งในการทำงานนั้น ระบบยังมีส่วนที่ช่วยผู้ใช้งานที่พบความหมายของคำศัพท์โดยที่จะคอยตรวจสอบการค้นหาคำศัพท์ของผู้ใช้ ซึ่งถ้าหากมีการค้นหาคำศัพท์คำนั้นมากกว่า 1 ครั้งภายใน 1 เดือนจะแสดงเกม Hangman ขึ้นมาให้ผู้ใช้งานเล่นเพื่อช่วยทบทวนให้ผู้ใช้งานสามารถจำความหมายของคำศัพท์นั้นได้ดีขึ้น



รูปที่ 7-6 แสดงหน้าต่างเกมทบทวนคำศัพท์

## 7.5 จุดเด่นและจุดบกพร่องในการทำงานของระบบ

### 7.5.1 จุดเด่นของระบบ

โปรแกรมพจนานุกรมที่พัฒนาขึ้นมา นี้ มีลักษณะเด่น ดังต่อไปนี้

1. สามารถค้นหาคำศัพท์โดยดับเบิลคลิกที่หน้าจอคอมพิวเตอร์ได้ทุกหน้าต่าง
2. ทั้งยังสามารถค้นหาได้ด้วยส่วนติดต่อกับผู้ใช้
3. อาจจะเลือก (disable) ไม่ให้โปรแกรมทำการค้นหาคำศัพท์โดยการดับเบิลคลิกก็ได้
4. สามารถเพิ่ม แก้ไข และลบคำศัพท์ได้ตามความต้องการของผู้ใช้
5. มีเกมที่ช่วยทบทวนคำศัพท์ได้ ซึ่งสามารถเลือกที่จะเล่นหรือไม่เล่นก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 7.5.2 จุดบกพร่องของระบบ

โปรแกรมพจนานุกรมนี้ยังมีจุดบกพร่องอยู่บ้าง ได้แก่

1. สามารถให้ผู้ใช้ลบคำศัพท์ได้ ซึ่งหากผู้ใช้ลบคำศัพท์ทิ้งไป ซึ่งถ้าคำศัพท์ถูกลบออกไปมากทำให้ต้องค้นหาคำศัพท์จากอินเทอร์เน็ต ซึ่งจะต้องใช้เวลานานมากกว่า
2. สามารถให้ผู้ใช้แก้ไขคำศัพท์ได้ ซึ่งถ้าหากผู้ใช้แก้ความหมายของคำศัพท์ที่ถูกต้องอยู่แล้ว ให้ผิดพลาดไป ทำให้เมื่อมีการค้นหาคำศัพท์คำนั้นใหม่ก็จะได้ความหมายที่ไม่ถูกต้องได้
3. ไม่สามารถค้นหาคำศัพท์ด้วยการชี้เมาส์ไปยังคำศัพท์นั้นได้ เพราะต้องทำการเปลี่ยนจากบิตแมปเป็นตัวอักษรด้วย OCR (Optical Character Recognition) ซึ่งเป็นการยุ่งยาก และอยู่นอกขอบเขตของโครงการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 8

### สรุปและวิจารณ์

#### 8.1 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคในช่วงการออกแบบและพัฒนาโปรแกรมมีดังนี้

1. การติดต่อกับฐานข้อมูลนั้น จะมีเทคโนโลยีสำหรับการเข้าถึงข้อมูลในฐานข้อมูลอยู่ด้วยกันหลายวิธี ซึ่งแต่ละวิธีต่างก็มีข้อดีและข้อเสียแตกต่างกันไป จึงต้องมีการศึกษารายละเอียดต่างๆ ของแต่ละวิธีให้ชัดเจน เพื่อเลือกวิธีที่ดีที่สุดมาใช้
2. การเลือกใช้รูปแบบของฐานข้อมูลนั้น จะต้องเลือกให้สอดคล้องกับวิธีที่ใช้ในการติดต่อกับข้อมูลในฐานข้อมูล
3. โลยารีของ MFC ประกอบไปด้วยคลาสต่างๆ มากมาย จึงต้องมีการพิจารณาการทำงานของคลาสต่างๆ แล้วเลือกคลาสที่เกี่ยวข้องมาใช้งาน ซึ่งต้องใช้เวลา
4. ความไม่แน่นอนของการออกแบบฐานข้อมูล และลักษณะการนำเสนอของโปรแกรม ทำให้การพัฒนาโปรแกรมเป็นไปได้ช้า และเสียเวลาในการศึกษามาก
5. การค้นหาคำศัพท์เมื่อชี้เมาส์บนคำศัพท์นั้น ไม่มีฟังก์ชันการทำงานในโลยารี MFC จึงต้องเปลี่ยนวิธีซึ่งใช้เวลาในการพิจารณาอยู่นาน เพื่อหาฟังก์ชันการค้นหาคำศัพท์ใหม่ และให้ง่ายต่อการใช้งาน

#### 8.2 แนวทางการวิจัยและพัฒนาต่อ

เนื่องจากโปรแกรมพจนานุกรมอังกฤษ-อังกฤษแบบลอยที่ได้พัฒนาขึ้นมาี้ สามารถค้นหาคำศัพท์เมื่อคลิกเบิ้ลคลิกเมาส์ที่คำศัพท์ ซึ่งไม่ตรงกับที่วางไว้ตอนแรก จึงสามารถนำไปพัฒนาต่อได้โดย

1. ควรพัฒนาด้านการค้นหาคำศัพท์แบบลอย ให้สามารถค้นหาคำศัพท์ได้เมื่อชี้เมาส์บนคำศัพท์นั้นได้ด้วยการทำ OCR (Optical Character Recognition) หรือโดยไม่ใช้ OCR แต่ให้โปรแกรมมีความรวดเร็วในการทำงาน
2. ควรพัฒนาให้มีความสามารถในการหาคำศัพท์ที่แม่นยำมากยิ่งขึ้น โดยการปรับปรุงฐานข้อมูลคำศัพท์ให้เป็นระบบมากกว่านี้
3. ควรพัฒนาฐานข้อมูลของโปรแกรม โดยอาจมีการจัดหมวดหมู่ของคำศัพท์ เช่น อาจแบ่งเป็นหมวดคอมพิวเตอร์ หมวดการแพทย์ เป็นต้น
4. ควรพัฒนาให้ผู้ใช้สามารถค้นหาคำศัพท์แบบลอยด้วยคีย์ลัดอื่นๆ ได้ เช่น คลิกขวา, Ctrl + คลิกขวา เป็นต้น
5. ควรพัฒนาให้มีการแสดงภาพประกอบคำอธิบายของคำศัพท์ อาจเป็นภาพแอนิเมชัน เพื่อให้โปรแกรมพจนานุกรมที่จะพัฒนาขึ้นมาั้น น่าใช้มากยิ่งขึ้น
6. ควรพัฒนาโดยการสร้างเกมต่างๆ ที่จะช่วยทบทวนคำศัพท์เพิ่มเข้าไป ซึ่งเกมเหล่านี้ควรจะเป็นเกมที่จะสามารถช่วยให้ผู้ใช้โปรแกรมจำความหมายของคำศัพท์ได้แม่นยำมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 8.3 เปรียบเทียบระบบกับผลิตภัณฑ์อื่น

ปัจจุบัน โปรแกรมพจนานุกรมที่พบเห็นอยู่ในตลาดนั้น ส่วนใหญ่แล้วจะเป็นโปรแกรมพจนานุกรมที่จะต้องพิมพ์คำศัพท์เพื่อทำการค้นหาคำศัพท์ในฐานข้อมูล แล้วจึงค่อยแสดงความหมายของคำศัพท์ออกมาที่ส่วนติดต่อกับผู้ใช้ จะมีอยู่ไม่กี่ผลิตภัณฑ์เท่านั้นที่อินเทอร์เน็ตเฟสแตกต่างออกไป เท่าที่พบเห็นจะมีอยู่บาง โปรแกรมที่จะแสดงความหมายของคำศัพท์เมื่อนำเมาส์ไปวางไวบนคำศัพท์บนที่ใดๆ บนหน้าจอคอมพิวเตอร์ แต่มันจะทำให้เครื่องช้ามาก นอกจากนี้ยังมีบางโปรแกรมที่จะแสดงคำศัพท์บนหน้าจอ เมื่อกด Ctrl + คลิกเมาส์ขวา

คุณสมบัติ	Floating Dictionary	POPular Dictionary	Sabyedee Dictionary
การค้นหาคำศัพท์	ดับเบิลคลิก	เมาส์ชี้	เมาส์ชี้
พจนานุกรม	อังกฤษ-ไทย อังกฤษ-อังกฤษ	อังกฤษ-ไทย	อังกฤษ-ไทย
ความแม่นยำ	มาก	น้อย	มาก
ค้นหาผ่านอินเทอร์เน็ต	✓	✗	✗
ประวัติการค้นหา	✓	✓	✗
เกมทบทวนคำศัพท์	✓	✓	✗

ตารางที่ 8-1 แสดงการเปรียบเทียบระบบกับผลิตภัณฑ์อื่น

โปรแกรมที่ได้พัฒนาขึ้นมาเป็นโปรแกรมพจนานุกรมอังกฤษ-ไทย และอังกฤษ-อังกฤษที่มีวิธีการค้นหาคำศัพท์และการแสดงผลที่ดีกว่าพจนานุกรมที่มีในท้องตลาด ทั้งยังสามารถค้นหาคำศัพท์ทางอินเทอร์เน็ตได้อีกด้วย และยังมีประวัติการค้นหาคำศัพท์ของผู้ใช้ โดยจะมีเกมที่ช่วยทบทวนความหมายของคำศัพท์ที่ผู้ใช้ได้เคยค้นหาไปแล้ว ซึ่งจะสามารถทำให้ผู้ใช้จำคำศัพท์ได้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

### หนังสืออ้างอิง

1. อาจารย์อภิเนตร อุณาภูล, “*Object-Oriented Analysis and Design*”, พิมพ์ครั้งที่ 1, แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, มิถุนายน 2543
2. นิรุช อำนวยศิลป์, “คู่มือการเขียนโปรแกรม *Microsoft Visual C++ Version 6.0* ฉบับเพื่อการใช้งานจริง”, พิมพ์ครั้งที่ 4, บริษัท ชัคเซส มีเดีย จำกัด, มีนาคม 2544
3. ยุทธนา ลีลาศวัฒนกุล, “คู่มือการเขียนโปรแกรม *Microsoft Visual C++ Version 6.0* ฉบับ *Database Programming*”, บริษัท ชัคเซส มีเดีย จำกัด
4. ยุทธนา ลีลาศวัฒนกุล, “คู่มือการเขียนโปรแกรม และใช้งาน *Visual C++ 6.0* ฉบับโปรแกรมเมอร์”, พิมพ์ครั้งที่ 1, สำนักพิมพ์อินโฟเพรส, กรกฎาคม 2544

### เว็บไซต์อ้างอิง

1. <http://www.babylon.com>
2. <http://www.codebeach.com>
3. <http://www.codeguru.com>
4. <http://www.codeproject.com>
5. <http://www.cprogramming.com>
6. <http://www.google.com>
7. <http://www.microsoft.com>
8. <http://www.planetsourcecode.com>
9. <http://www.thaidev.com>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้