

การพัฒนาแอปพลิเคชันโดยใช้ภาษาจาวา

Application Development by Java



โดย

นางสาวสุชาดา คิวานนท์ 37014499

นางสาวสุปรียา ตะล้อมลิน 37014517

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เลขหมู่.....
เลขทะเบียน.....

30510

วัน, เดือน, ปี 17 ก.ค. 2541

เอกสารนี้เป็นเอกสารของหอสมุดกลาง สำนักหอสมุดกลาง มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่สามารถนำเอกสารไปเผยแพร่ได้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2540

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง การพัฒนาแอปพลิเคชันโดยใช้ภาษาจาวา

ผู้จัดทำ

1. นางสาวสุชาดา คิวานนท์
2. นางสาวสุปรียา ตะล่อมสิน

.....
/ ม ม /
(อาจารย์ธรรมา หงษ์สุวรรณ)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาแอปพลิเคชัน โดยใช้ภาษาจาวา

นางสาวสุชาดา ทิวานนท์
นางสาวสุปรียา ตะล่อมสิน

อ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2540

บทคัดย่อ

โครงการนี้เป็นการศึกษาเกี่ยวกับภาษาจาวาโดยทั่วไป เนื่องจากได้ตระหนักว่าภาษาจาวาเป็นภาษาที่มีประโยชน์ และเป็นภาษาที่จะใช้กันอย่างกว้างขวางในอนาคตอันใกล้นี้ ด้วยคุณสมบัติหลายๆประการดังจะได้กล่าวไว้ในปริญญานิพนธ์นี้ และนำเอาความรู้ที่ได้มาทำการพัฒนาเป็นโปรแกรม เพื่อนำไปใช้ประโยชน์ต่อไป

โดยในที่นี้ได้นำเอาความรู้ที่ได้มาพัฒนา จัดทำเป็นโฮมเพจของสำนักทะเบียนและประมวลผลกลางของคณะวิศวกรรมศาสตร์ ซึ่งมุ่งเน้นที่จะให้บริการด้านข้อมูลเบื้องต้น เช่น ผลการเรียนในแต่ละเทอม การลงทะเบียนในเทอมนั้นๆ รายชื่อนักศึกษา เป็นต้น ซึ่งจะอำนวยความสะดวกให้กับอาจารย์ เจ้าหน้าที่ และนักศึกษาโดยทั่วไป

Application Development by Java

Miss Suchada Kiwanon

Miss Supreeya Talomsin

Thana Hongsuwan

1997

ABSTRACT

This project is the study about Java programming language. Because the realization that Java is a very useful language and it has a trend to be widespread in the future with many properties which appear in this thesis, we decided to study about Java and its components which used to develop an application.

The Register office's homepage of engineering faculty is the application in this project which serves general services of register office such as checking grades in each semester, checking the registration at the present semester, showing list of students name, etc. This application will make all the teachers, officials and students easy to interact with register office's database over the internet.

สารบัญ

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
สารบัญ.....	III
สารบัญตาราง.....	IV
สารบัญรูปภาพ.....	V
บทที่ 1 บทนำ.....	1
บทที่ 2 ความรู้ทั่วไปเกี่ยวกับจาวา.....	4
บทที่ 3 การพัฒนาจาวาด้วย Java Bean.....	47
บทที่ 4 JDBC (Java Database Connectivity)	53
บทที่ 5 การออกแบบและการสร้าง.....	72
บทที่ 6 ผลงาน.....	79
บทที่ 7 สรุปและวิเคราะห์.....	85
ภาคผนวก ก. ODBC (Open Database Connectivity)	90
ภาคผนวก ข. IDS Server.....	94
ภาคผนวก ค. ตารางและรายละเอียดในฐานข้อมูล.....	99
กิตติกรรมประกาศ.....	VII
บรรณานุกรม.....	VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 2-1 ชนิดของตัวแปรในจาวา.....	12
ตารางที่ 2-2 keyword ในจาวา.....	13
ตารางที่ 2-3 binary arithmetic operation ของจาวา.....	15
ตารางที่ 2-4 เครื่องหมายที่ใช้บอกค่าของตัวถูกดำเนินการ.....	15
ตารางที่ 2-5 การใช้ prefix และ postfix ในการเพิ่มและลดค่า.....	15
ตารางที่ 2-6 relational operation ในจาวา.....	16
ตารางที่ 2-7 condition operator ในจาวา.....	16
ตารางที่ 2-8 Bitwise operator ในจาวา.....	16
ตารางที่ 2-9 assignment operator เพิ่มเติม.....	17
ตารางที่ 2-10 ประเภทของ statement ที่ใช้ควบคุมการทำงาน.....	17
ตารางที่ 2-11 ระดับการเข้าถึงของแต่ละ specifier.....	25
ตารางที่ 2-12 ข้อแตกต่างระหว่าง Application กับ Applet.....	39
ตารางที่ 4-1 รายชื่อผู้ผลิตซอฟต์แวร์สำหรับเจดีย์ซี.....	70

สารบัญรูปภาพ

รูปที่ 2-1	การทำงานของจาวา.....	6
รูปที่ 2-2	ส่วนประกอบของซอฟต์แวร์ออปเจ็ค.....	6
รูปที่ 2-3	แบบจำลองรถจักรยานในรูปของซอฟต์แวร์ออปเจ็ค.....	7
รูปที่ 2-4	การส่งข่าวสารระหว่างออปเจ็ค.....	8
รูปที่ 2-5	การส่งพารามิเตอร์ไปพร้อมกับ ข่าวสาร.....	9
รูปที่ 2-6	ส่วนประกอบของ class.....	10
รูปที่ 2-7	ส่วนประกอบของ class ของรถจักรยาน.....	10
รูปที่ 2-8	ลำดับชั้นของ class.....	11
รูปที่ 2-9	superclass และ subclass.....	29
รูปที่ 2-10	class heirarchy.....	29
รูปที่ 2-11	ตัวอย่างของ exception heirarchy.....	31
รูปที่ 2-12	class heirarchy ของ Throwable class.....	34
รูปที่ 2-13	สถานะของ Thread ในจาวา.....	36
รูปที่ 2-14	Thread group class.....	38
รูปที่ 2-15	heirarchy ของ Applet class.....	40
รูปที่ 3-1	Button Bean.....	47
รูปที่ 3-2	Sliders Bean.....	48
รูปที่ 3-3	Calendar Bean.....	48
รูปที่ 3-4	Spreadsheets Bean.....	48
รูปที่ 3-5	เครื่องคิดเลขขณะสร้างในเครื่องมือที่ใช้สร้าง.....	49
รูปที่ 3-6	เครื่องคิดเลขขณะกำลังทำงาน.....	49
รูปที่ 3-7	Mojo ตัวอย่าง Builder Tool.....	50
รูปที่ 3-8	หน้าต่างของเครื่องรับบริการ.....	51
รูปที่ 3-9	ฟอร์ม.....	51
รูปที่ 3-10	property sheet.....	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-1	โมเดลชั้นการทำงานของเจดีบีซี.....	53
รูปที่ 4-2	รูปแบบที่ใช้ในเจดีบีซี.....	54
รูปที่ 4-3	การติดต่อไปยังฐานข้อมูลโดยใช้ CGI.....	62
รูปที่ 4-4	การติดต่อไปยังฐานข้อมูลโดยผ่านเจดีบีซี.....	63
รูปที่ 4-5	Two-Tier model.....	63
รูปที่ 4-6	Three-Tier model.....	64
รูปที่ 4-7	JDBC –ODBC Bridge.....	64
รูปที่ 4-8	Native (Partly-Java) Drivers.....	67
รูปที่ 4-9	Net-Protocol (All-Java) Drivers.....	68
รูปที่ 4-10	Native-Protocol (All-Java) Drivers.....	69
รูปที่ 5-1	ตัวอย่างการออกแบบหน้าจอ.....	74
รูปที่ 6-1	หน้าจอแรกของโฮมเพจสำนักทะเบียน.....	79
รูปที่ 6-2	หน้าจอแสดงรายการที่ให้บริการ.....	80
รูปที่ 6-3	dialog แสดงหน้าจอการจัดเก็บไฟล์.....	81
รูปที่ 6-4	dialog แสดงรายชื่อนักศึกษาตามรายวิชา.....	81
รูปที่ 6-5	แสดงหน้าจอการตรวจสอบการลงทะเบียน.....	82
รูปที่ 6-6	แสดงหน้าจอการเข้าใช้บริการตรวจสอบผลการเรียน.....	82
รูปที่ 6-7	แสดงหน้าจอผลการเรียน.....	83
รูปที่ 6-8	แสดงหน้าจอที่ใช้ในการฝากข่าว.....	84
รูปที่ 7-1	dialog แสดงการ Grant java security ใน Netscape Communicator.....	88
รูปที่ ก-1	องค์ประกอบของ ODBC.....	92
รูปที่ ข-1	การติดตั้ง IDS Server บน Personal Computer.....	95
รูปที่ ข-2	การติดตั้ง IDS Server แบบแยกองค์ประกอบบน 3 เครื่อง.....	95
รูปที่ ข-3	การติดตั้ง IDS Server แบบแยกองค์ประกอบบน 4 เครื่อง.....	96
รูปที่ ข-4	แบบแผนของ IDS Server.....	98

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันเทคโนโลยีการติดต่อสื่อสารอินเทอร์เน็ต ที่ใช้ระบบเชื่อมโยงของเครือข่าย ที่เรียกว่า เวิลด์ไวด์เว็บ (World-Wide Web :WWW) ได้มีการใช้กันอย่างแพร่หลาย มีการใช้ระบบเครือข่ายในการ ค้นหาข้อมูล แลกเปลี่ยน หรือมีการแสดงบทความ ข่าวสาร สารต่าง ๆ ผ่านทางโฮมเพจ โดยการแสดง ข้อความหรือข่าวสารต่างๆ ในเครือข่ายเวิลด์ไวด์เว็บจะทำโดยภาษาโปรแกรมที่ใช้ในการเขียนโฮมเพจคือ เอชทีเอ็มแอล (HyperText Markup Language : HTML) ซึ่งมีความสามารถในการแสดงตัวอักษรในรูปแบบต่างๆ และ รูปภาพเท่านั้นซึ่งดูไม่น่าสนใจ แต่เมื่อเทคโนโลยีมีความเจริญก้าวหน้าขึ้น ก็ได้มีผู้ คิดค้นให้โฮมเพจสามารถแสดงข้อมูลแบบมัลติมีเดียได้ โดยวิธีการหนึ่งที่ได้ค้นพบก็คือการใช้ภาษาจาวา (Java Programming Language) ที่ผลิตโดยบริษัท ซันไมโครซิสเต็มส์ ซึ่งจะช่วยเพิ่มสีสันและทำให้ โฮมเพจมีชีวิตชีวาขึ้น เช่น มีภาพเคลื่อนไหว เสียง

นอกจากนี้แล้วภาษาที่ใช้ในเขียนโปรแกรมเพื่อทำงานบนระบบเครือข่ายได้ด้วย ซึ่งนี่เอง เป็นคุณสมบัติสำคัญที่ทำให้จาวาในปัจจุบันนี้มีบทบาทอย่างมากให้ระบบเครือข่าย และทำให้มีการใช้ และการพัฒนาจาวาอยู่ตลอดเวลา เช่น การพัฒนาจาวาให้สามารถติดต่อกับระบบฐานข้อมูล ได้อย่างมีประสิทธิภาพ หรือ ที่เรียกว่า Java Database Connectivity (JDBC), การพัฒนาจาวาสำหรับระบบมัลติมีเดีย, JavaOS, JAVA Server เป็นต้น

จากการที่เวิลด์ไวด์เว็บได้รับความนิยมอย่างสูง ประกอบกับการที่มีผู้ใช้ทั้งในระดับบุคคลและ องค์กรต่างๆ ได้เห็นถึงความสำคัญของระบบฐานข้อมูล ซึ่งเมื่อรวมทั้งสองอย่างนี้เข้าด้วยกันผลที่ได้ก็คือ การขยายขอบเขตของข้อมูล การให้บริการ และการขยายฐานลูกค้าขององค์กรต่างๆออกไปทั่วโลก ซึ่ง เว็บบางเว็บเหล่านี้จะให้คุณค่ามากกว่าเว็บเพจแบบสแตติกทั่วไป เนื่องจากเว็บเหล่านี้ส่งผลให้เกิดการโต้ตอบสองทิศทางระหว่างเจ้าของฐานข้อมูลกับผู้ใช้ ตัวอย่างเช่น บริษัทสามารถโฆษณาบนโฮมเพจเกี่ยวกับผลิตภัณฑ์ล่าสุด (จากฐานข้อมูลผลิตภัณฑ์) ของคนได้ทันที หรือการเผยแพร่โฆษณาชนิดที่มีอุปชัณ สำหรับผู้ใช้ โดยสามารถป้อนแอดเดรสของคนเข้าไป เพื่อขอรับข้อมูลข่าวสารเพิ่มเติมเกี่ยวกับผลิตภัณฑ์นั้นๆ หรือ เป็นโฮมเพจที่ใช้ประกาศผลสอบเอ็นทรานซ์ จากทั้งสามตัวอย่างที่กล่าวมาเป็นตัวอย่าง การสร้างโฮมเพจที่มุ่งเน้นการให้ข้อมูลข่าวสารเป็นสำคัญ

เนื่องจากได้เห็นถึงคำสำคัญและคุณสมบัติที่โดดเด่นของภาษาจาวา จึงได้ทำการศึกษาค้นคว้า เกี่ยวกับภาษานี้โดยทั่วไป และจากแนวความคิดเกี่ยวกับเว็บเว็บบางเว็บเหล่านี้ ทำให้ตัดสินใจที่จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาแอปพลิเคชันในรูปแบบของโฮมเพจของสำนักทะเบียนและประมวลผลกลาง ซึ่งจะให้บริการเกี่ยวกับข่าวสารข้อมูลต่างๆ ไปของทางสำนักทะเบียน เช่น การตรวจสอบการลงทะเบียน, การตรวจสอบรายชื่อนักศึกษา, การตรวจสอบผลการเรียนในแต่ละภาคเรียน เป็นต้น ซึ่งมุ่งเน้นที่จะอำนวยความสะดวกให้กับนักศึกษาทั่วไป

1.2 ขอบเขตการศึกษา

การศึกษาในโครงการนี้มุ่งเน้นที่จะศึกษาถึงคุณสมบัติ หลักการทำงาน การเขียนโปรแกรม และ แนวทางการพัฒนาภาษาจาวา รวมถึงโปรแกรมที่ช่วยในการพัฒนาภาษา จาวา เช่น จาวา Development Kit (JDK) ของบริษัท ซันไมโครซิสเต็มส์ และ Visual Cafe Pro 1.0 ของบริษัทซีเมนเท็ค เพื่อให้สามารถนำมาใช้ในการพัฒนาเป็นแอปพลิเคชัน เพื่อจะทำไปใช้ประโยชน์ต่อไป

สำหรับการพัฒนาโฮมเพจของสำนักทะเบียนและประมวลผลกลาง จำเป็นที่จะต้องศึกษาเพิ่มเติมในรายละเอียดอื่นๆอีก ได้แก่

1. Java Database Connectivity (JDBC) ซึ่งเป็น API ที่ช่วยในการติดต่อสื่อสารกับฐานข้อมูล
2. ศึกษาการออกแบบและเขียนโฮมเพจ (ภาษา HTML) รวมถึงการใช้โปรแกรมที่ช่วยในการพัฒนาโฮมเพจ เช่น Assistant Pro 3.0 และ Hot Dog PRO 4.0 เป็นต้น
3. IDSServer ของบริษัท IDSSoftware เป็นซอฟต์แวร์ที่เป็น middle tier ซึ่งจะช่วยส่งเสริมการติดต่อระหว่าง ไคลน์เอ็นท์ กับ ฐานข้อมูล
4. ระบบ Security ของ Browser ต่างๆ

1.3 ประโยชน์

1. ได้รับความรู้และประสบการณ์ในการเขียนโปรแกรมด้วยภาษา จาวา
2. ทราบถึงข้อแตกต่างในการติดต่อระหว่างไคลน์เอ็นท์ กับ ฐานข้อมูลระหว่างการใช้ CGI กับ JDBC
3. ได้ทราบถึงแนวทาง และ ขั้นตอนในการพัฒนาเว็บดาต้าเบส เพื่อนำไปใช้ในอนาคต
4. เป็นการใช้ประโยชน์จากทรัพยากรที่มีอยู่ให้คุ้มค่าที่สุดที่สุด
5. ทำให้ทราบถึงปัญหาที่เกิดขึ้นในการปฏิบัติงานจริง และแนวทางให้การแก้ไขปัญหา

1.4 การดำเนินงาน

1. ทำการศึกษาเกี่ยวกับคุณสมบัติ หลักการทำงาน และการเขียนโปรแกรมภาษาจาวา
2. ทำการศึกษาและเปรียบเทียบโปรแกรมที่ช่วยในการพัฒนาโปรแกรมภาษาจาวา (Development Tool)
3. ศึกษาและทดลองใช้ Development Tool ที่ได้เลือกไว้ ในที่นี้ได้เลือก Visual Cafe Pro 1.0
4. กำหนดหัวข้อของแอปพลิเคชัน ซึ่งในที่นี้ก็คือ โฮมเพจของสำนักทะเบียนและประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลาง คณะวิศวกรรมศาสตร์

5. ทำการสอบถาม และ กำหนดของเขตการทำงาน ของ โสมเพจ
6. ทำการศึกษาถึงระบบฐานข้อมูลของทางสำนักทะเบียน การจัดระเบียบตาราง และการขออนุญาตคัดลอกข้อมูลที่จะเป็น
7. ศึกษารูปแบบ และกำหนดการติดต่อกับฐานข้อมูลรวมถึงเอพีไอ(API) หรือ โปรแกรมที่จำเป็น (JDBC-ODBC API, mSQL, IDSServer)
8. ศึกษาและออกแบบ โสมเพจ โดยใช้ Development Tool ที่เลือกไว้ (Assistant Pro 3.0) รวมทั้งทำการออกแบบหน้าจอ (User Interface) ให้สามารถเข้าใจได้ง่ายที่สุด
9. ทำการเขียนโปรแกรมในส่วนของ Input User Interface, การติดต่อกับฐานข้อมูล และ Output User Interface
10. ทดลองใช้เพื่อหาข้อบกพร่อง และทำการแก้ไข
11. รวบรวมข้อมูลและเอกสารทั้งหมดเพื่อเรียบเรียงเป็นปฏิญญาฉบับนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ความรู้ทั่วไปเกี่ยวกับจาวา

ในหัวนี้จะกล่าวถึงภาษาจาวา โดยทั่วไป ทั้งคุณสมบัติ การทำงาน แนวความคิด องค์ประกอบที่จำเป็นเพื่อการเขียนโปรแกรม รวมถึงการพัฒนาภาษาจาวาในรูปแบบต่าง

2.1 คุณสมบัติของ จาวา

จาวาเป็นภาษาที่ใช้ในการเขียนโปรแกรมแบบหนึ่ง ซึ่งถูกพัฒนาโดยบริษัท ซันไมโครซิสเต็มส์ จาวาถูกพัฒนาเพื่อใช้ในการพัฒนาโปรแกรมสำเร็จรูปทางสภาพแวดล้อมที่ต่างกัน และในสภาพแวดล้อมที่มีการกระจายของโปรแกรมไปในเครือข่ายโดยมีลักษณะหรือพื้นฐานมาจากภาษา C++ คุณสมบัติของภาษาจาวาคือ

- 1.ง่าย ลักษณะอย่างแรกของภาษาจาวาก็คือเป็นภาษาที่เข้าใจได้ง่าย เนื่องจากจาวาจะคล้ายกับภาษา C++ แต่ได้กำจัดส่วนที่เป็นปัญหาออก ทำให้จาวาเป็นภาษาที่ง่าย สามารถเรียนรู้ได้ง่าย
- 2.เป็น Object-Oriented ทำให้คลาสของจาวาอยู่ในรูปของออบเจกต์ ทำให้สามารถเคลื่อนย้ายระหว่างคอมพิวเตอร์เครื่องที่ให้บริการ (host computer) และ ตัวแปลภาษาจาวา (java interpreter) ผ่านเครือข่ายได้สะดวก
- 3.มีสภาพแวดล้อมที่การกระจาย โดยมีการให้ไลบรารีของคลาสสำหรับการเข้าถึงอินเทอร์เน็ต คุณ สามารถใช้ออบเจกต์จากที่ไหนก็ได้ในเครือข่ายมาทำโปรแกรมสำเร็จรูปของคุณ หรือสร้างออบเจกต์ซึ่งสามารถถูกเข้าถึงหรือถูกจำลองโดยผู้พัฒนาโปรแกรมคนอื่น ซึ่งจะช่วยให้คุณได้ว่าคุณได้ใช้คลาสที่จะลงมาถึงใหม่ที่สุด
- 4.ตัวแปลภาษา เริ่มแรกคอมไพเลอร์จะอ่าน ตัวโค้ดและแปลงเป็นไบต์โค้ด (อยู่ระหว่างตัวโค้ดกับ ไบต์โค้ด) ซึ่งจะถูกส่งไปดำเนินการต่อที่ตัวแปลภาษาในเครื่องใดๆที่ตัวดำเนินการและระบบ run-time ตั้งอยู่ ตัวแปลภาษาจะอ่านไบต์โค้ดและจะดำเนินการคำสั่งภาษาเครื่องที่เหมาะสม ซึ่งเป็นการช่วยแก้ปัญหาในสภาพแวดล้อมที่มีหลายแพลตฟอร์ม
- 5.มีประสิทธิภาพสูง การที่ตัวแปลภาษาสามารถรันด้วยความเร็วเต็มที่ โดยไม่ต้องทำการตรวจสอบสภาพแวดล้อมในขณะเวลารัน เนื่องจากมี Java Virtual Machine อยู่ และการกำจัดข้อมูลที่ไม่ความหมายโดยอัตโนมัติ (automatic garbage collector) ซึ่งจะรันเสมือนเป็นเทรคอยู่เบื้องหลัง เพื่อทำให้มีหน่วยความจำใช้งานได้เมื่อต้องการ ทำให้ประสิทธิภาพโดยรวมในการทำงานสูงขึ้น
- 6.มีความแข็งแรงและน่าเชื่อถือ การทำ dynamic linking มีผล 2 ประการคือ การคอมไพล์จะทำการตรวจสอบการอ้างอิงไปถึงคลาสอื่นทุกตัวเพื่อความถูกต้องของชนิด และ โปรแกรมจะถูกเชื่อมกับคลาสที่ใหม่ที่สุดที่โปรแกรมใช้อยู่ ไม่จำเป็นต้องทำการคอมไพล์ซ้ำๆเมื่อคลาสมีการเปลี่ยนแปลง และขณะที่การคิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อของคลาสเหล่านั้นเหมือนเดิม ทำให้สามารถใช้คลาสเหล่านั้นได้เหมือนเดิมโดยไม่ต้องเปลี่ยนโค้ดของคุณ นอกจากนี้จาวายังได้ยกเลิกการใช้พ้อยเตอร์ ซึ่งทำให้ง่ายต่อการจัดการหน่วยความจำทำให้มีความน่าเชื่อถือมากยิ่งขึ้น

7.มีความปลอดภัย เพื่อความปลอดภัยในจาวาจะไม่มีวิธีใดที่ใช้ทำให้เกิดอันตรายหรือเชื่อมไปยังโค้ดที่คอมไพล์แล้วเพื่อทำการที่ทำให้เกิดอันตรายได้ จาวาจะทำการทดสอบไฟล์สำหรับการอนุญาตให้เข้าถึงก่อนที่จะอนุญาตให้โปรแกรมปิดไฟล์นั้นๆ ตัวแปลภาษาจะทำการตรวจทานไบต์โค้ดเพื่อตรวจจับการโน้มน้าว เช่น คอมไพล์เลอร์ที่บางคนเขียนให้ใช้พ้อยเตอร์ได้ และจากการทำการเฝ้าหน่วยความจำในขณะรัน ทำให้ป้องกันผู้บุกรุกที่พยายามเข้าถึงโครงสร้างของข้อมูลได้

8.มีสถาปัตยกรรมที่เป็นกลาง เนื่องจากจาวาถูกออกแบบมาเพื่อสนับสนุนโปรแกรมสำเร็จรูป ซึ่งจะถูกกระจายไปยังสภาพแวดล้อมของเครือข่ายที่แตกต่างกัน ในสภาพแวดล้อมเหล่านั้น โปรแกรมจะต้องสามารถดำเนินการบนสถาปัตยกรรมของฮาร์ดแวร์ที่แตกต่างกัน ระบบปฏิบัติการที่ต่างกันได้ เพื่อให้สามารถทำงานในสภาพแวดล้อมที่มีการจัดการที่หลากหลาย คอมไพล์เลอร์ของจาวาจะสร้างไบต์โค้ดเป็นรูปแบบของสถาปัตยกรรมที่เป็นกลาง ออกแบบเพื่อให้สามารถส่งโค้ดได้อย่างมีประสิทธิภาพไปยังฮาร์ดแวร์และซอฟต์แวร์ที่มีรูปแบบหลากหลาย ซึ่งการแปลจาวาจะช่วยแก้ปัญหา การกระจาย binary และ ปัญหาเรื่องรุ่น (version)

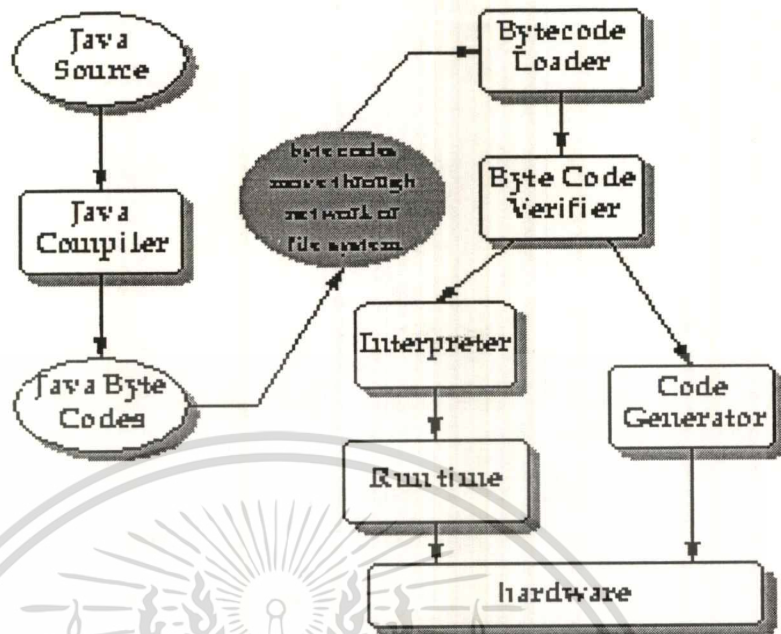
9.สามารถเคลื่อนย้ายได้ การสถาปัตยกรรมที่เป็นกลางก็เป็นส่วนหนึ่งของระบบที่สามารถเคลื่อนย้ายได้ จาวาใช้ความสามารถในการเคลื่อนย้ายได้ในการกำหนดขอบเขตของภาษาพื้นฐาน โดยกำหนดขนาดชนิดของข้อมูลพื้นฐานและพฤติกรรมของตัวดำเนินการทางคณิตศาสตร์ ทำให้โปรแกรมของคุณเหมือนกันในทุกๆรูปแบบ ซึ่งจะชนิดของข้อมูลทุกตัวสามารถเข้ากับสถาปัตยกรรมของฮาร์ดแวร์และซอฟต์แวร์ ที่พาดกันอยู่ได้

2.2 การทำงานของจาวา

เมื่อเราทำการสร้างไฟล์จาวาขึ้นมาได้แล้ว เราต้องทำให้ไฟล์นี้สามารถใช้งานได้ โดยการนำไฟล์จาวาที่สร้างขึ้นมาเป็นไฟล์ทำการคอมไพล์ หลังจากคอมไพล์แล้วจะไปได้ไฟล์ที่เรียกว่า คลาสไฟล์ ซึ่งเป็นไฟล์ที่เก็บไบต์โค้ดไว้ (ไบต์โค้ดจะมีลักษณะใกล้เคียงคำสั่งเครื่องมากที่สุด) โดยกระบวนการที่กล่าวมาทั้งหมดจะกระทำที่เครื่องรับบริการ ไบต์โค้ดที่ได้นี้จะถูกรันบน Java Virtual Machine (JVM) เป็นสิ่งที่สมมติขึ้นซึ่งพฤติกรรมได้ถูกกำหนดมาแล้วโดยบริษัทซันไมโครซิสเต็มส์ ลักษณะของ JVM ได้มีการอธิบายถึงพฤติกรรมที่คาดหวังของเครื่องคอมพิวเตอร์ในการรันไบต์โค้ดที่ให้ไป ไบต์โค้ดที่จะถูกรันบน JVM จะถูกส่งให้ตัวแปลภาษา ซึ่งจะทำการตรวจสอบไบต์โค้ดที่ได้ จากนั้นจะอ่านไบต์โค้ดและดำเนินการคำสั่งเครื่องที่เหมาะสม

Compile Time

Run Time



รูปที่ 2-1 การทำงานของจาวา

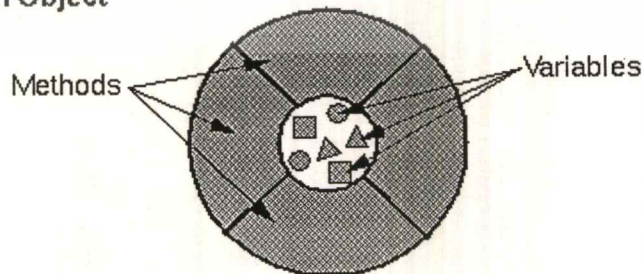
2.3 แนวความคิดของ Object Oriented Programming

แนวความคิดของ Object Oriented Programming (OOP) เป็นการมองในลักษณะของออบเจ็กต์ และ คลาส ซึ่งเป็นแนวความคิดพื้นฐานที่สำคัญของภาษาคอมพิวเตอร์ ในปัจจุบัน เช่น Delphi เป็นต้น

2.3.1 ออบเจ็กต์ คืออะไร

ออบเจ็กต์เป็นหน่วยของซอฟต์แวร์ ที่เก็บ ตัวแปร และ วิธี ที่เกี่ยวข้องไว้ด้วยกัน ออบเจ็กต์ในความเป็นจริงจะมีลักษณะที่เหมือนกันอยู่ 2 ประการ คือ state และ behavior สำหรับซอฟต์แวร์ state จะอยู่ในรูปของ ตัวแปร ส่วน behavior จะอยู่ในรูปวิธี

An Object

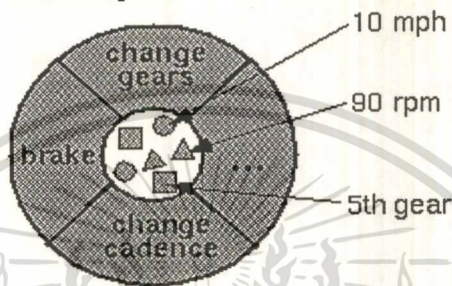


รูปที่ 2-2 ส่วนประกอบของ ซอฟต์แวร์ออบเจ็กต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถแสดง ซอฟต์แวร์ออปเจ็ก ได้ โดยทุกอย่างที่ออปเจ็ก รู้ (state) และ ทำ (behavior) แสดงออกในรูปของตัวแปรและวิธี ใน เช่น ซอฟต์แวร์ ออปเจ็ก ที่เป็นจำลองถึงจักรยานในความเป็นจริง จะมี ตัวแปร ซึ่งจะแสดงถึงความเร็วในขณะนั้น :10 เมตรต่อชั่วโมง ความเร็วรอบ :90 รอบต่อนาที และ เกียร์ในขณะนั้น : เกียร์ 5 นอกจากนี้ใน ซอฟต์แวร์ ออปเจ็ก ที่ใช้แทนจักรยาน ยังมี วิธี ที่ใช้ในการหยุดรถ เปลี่ยนการหมุนของกระโถนจักรยาน เปลี่ยนเกียร์ ตัวแปร และ วิธี เหล่านี้เรียกอีกชื่อหนึ่งว่า instance variable และ instance method

Your Bicycle



รูปที่ 2-3 แบบจำลองรถจักรยานในรูปของ ซอฟต์แวร์ ออปเจ็ก

จากรูปจะเห็นว่าวิธี จะล้อมรอบตัวแปรไว้ ซึ่งจะซ่อนตัวแปร เหล่านี้จากออปเจ็กของโปรแกรมอื่น ซึ่งการทำเช่นนี้เรียกว่า “encapsulation” ในความเป็นจริง encapsulation ใช้ในการซ่อนรายละเอียดที่ไม่สำคัญจากออปเจ็ก อื่น เช่นเมื่อคุณหยุดรถ คุณไม่จำเป็นต้องทราบถึงการทำงานของที่ห้ามล้อ เหมือนกับในโปรแกรม คุณไม่จำเป็นต้องรู้ว่าclassถูกสร้างได้อย่างไร แค่ว่ามีวิธีใดที่อยู่ในนั้นบ้าง ฉะนั้น รายละเอียดภายในสามารถเปลี่ยนแปลงเมื่อใดก็ได้ โดยไม่ส่งผลกระทบต่อส่วนอื่นของโปรแกรม

อย่างไรก็ตาม บางภาษารวมถึงภาษาจาวาออปเจ็ก เลือกที่จะให้ออปเจ็กอื่นเห็น ตัวแปรของมัน และบางครั้งสามารถเปลี่ยนแปลงได้ด้วย เช่นเดียวกับวิธี ออปเจ็กสามารถเลือกที่จะซ่อนบางเม็ท-ธอด ไม่ให้ออปเจ็กอื่นเห็น หรือเรียกใช้ได้

ประโยชน์ที่ได้รับจาก encapsulation

1. โค้ดของออปเจ็ก สามารถทำการเขียนแก้ไขได้อย่างอิสระจากโค้ดที่เขียนให้กับออปเจ็กอื่นๆ อีกทั้งออปเจ็ก ยังสามารถส่งผ่านไปยังระบบต่างๆ ได้ง่าย เหมือนกับการที่คุณให้จักรยานเพื่อใช้ และเพื่อคุณก็สามารถใช้งานจักรยานนั้นได้
2. ออปเจ็ก หนึ่งจะมีการติดต่อกับออปเจ็กอื่นได้มาก แต่ออปเจ็กสามารถจะแก้ไขตัวแปรหรือวิธี ของมัน โดยไม่ส่งผลกระทบต่อออปเจ็กอื่นที่ติดต่อกับมันอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

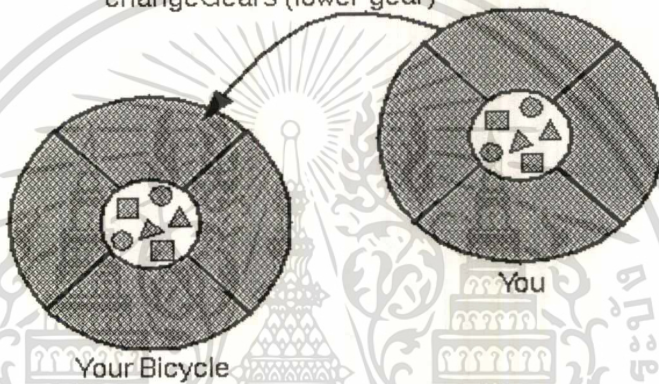
2.3.2 ข่าวสาร คืออะไร

ออปเจ็กต์ตัวเดียวโดดๆ โคนปรกติจะไม่ค่อยมีประโยชน์ โดยมากออปเจ็กต์มักจะเป็นส่วนประกอบของโปรแกรมที่ใหญ่กว่าหรือโปรแกรมสำเร็จรูปที่ประกอบด้วยออปเจ็กต์จำนวนมากในการติดต่อกันระหว่างออปเจ็กต์ ทำให้ผู้เขียนโปรแกรมสามารถทำงานที่ซับซ้อน

ซอฟต์แวร์ ออปเจ็กต์ จะมีการติดต่อกันระหว่างการส่ง ข่าวสาร ใ้กัน เมื่อออปเจ็กต์ A ต้องการให้ ออปเจ็กต์ B ทำ วิธี ที่อยู่ใน B ออปเจ็กต์ A จะส่ง ข่าวสาร ไปให้กับ ออปเจ็กต์ B ดังรูปที่ 2-4

A Message with Parameters

changeGears (lower gear)



รูปที่ 2-4 การส่ง ข่าวสาร ระหว่าง ออปเจ็กต์

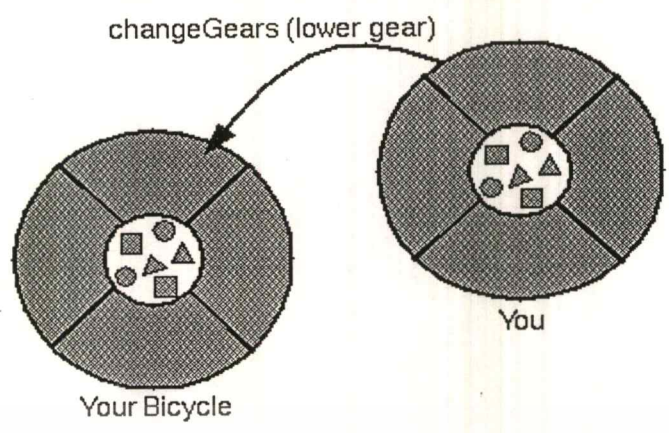
บางครั้งออปเจ็กต์ ที่เป็นฝ่ายรับ ข่าวสาร ต้องการข้อมูลเพิ่มเติม ซึ่งออปเจ็กต์เหล่านั้นรู้ว่าจะต้องทำอย่างไร เช่น เมื่อคุณต้องการเปลี่ยนเกียร์รถจักรยาน คุณต้องระบุว่าต้องการเปลี่ยนไปเกียร์ไหน ซึ่งข้อมูลเหล่านี้จะถูกส่งผ่านไปพร้อมกับ ข่าวสาร ในรูปของ พารามิเตอร์

องค์ประกอบ 3 อย่างที่รวมกันเป็น ข่าวสาร ได้แก่

1. ออปเจ็กต์ ที่เป็นผู้รับ ข่าวสาร
2. ชื่อของวิธี ที่ต้องการให้ทำ
3. พารามิเตอร์ ที่ วิธีต้องการ

องค์ประกอบทั้ง 3 นี้เพียงพอที่จะทำให้ ออปเจ็กต์ ที่รับ ข่าวสาร กระทำวิธีที่ต้องการ

A Message with Parameters



รูปที่ 2-5 การส่ง พารามิเตอร์ ไปพร้อมกับ ข่าวสาร

ประโยชน์ที่ได้รับจากข่าวสาร (message)

1. เนื่องจากกระทำของออปเจ็ค จะทำผ่านวิธีของมันเอง ฉะนั้นการส่งผ่าน ข่าวสาร ทำให้สามารถสนับสนุนการติดต่อทุกรูปแบบระหว่างออปเจ็ค
2. ออปเจ็คไม่จำเป็นต้องอยู่บนขบวนการเดียวกัน หรือแม้กระทั่งบนเครื่องคอมพิวเตอร์เดียวกัน โดยการใช้การส่ง ข่าวสาร ไปมาระหว่างกัน

2.3.3 class คืออะไร

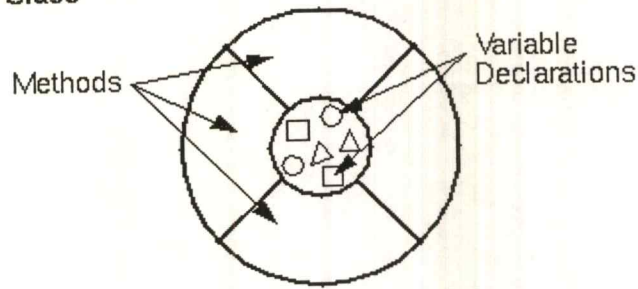
ในความเป็นจริง เรามักมีออปเจ็ค หลายๆตัวที่จัดเป็นชนิดเดียวกัน เช่น จักรยานที่คุณมีเป็นหนึ่งในจักรยานหลายคันที่มีอยู่ในโลก ด้วยแนวความคิดของ object-oriented กล่าวได้ว่า จักรยานของคุณ (object) เป็น instance หนึ่งของclassของ จักรยานจำนวนมาก จักรยานจะมีบางสถานะ (มีเกียร์ , มี 2 ล้อ) และ การทำงาน (การเปลี่ยนเกียร์ , การหยุดรถ) ที่เหมือนกัน อย่างไรก็ตามสถานะของจักรยานแต่ละคันก็เป็นอิสระต่อกัน และอาจมีส่วนที่แตกต่างกัน

ในการสร้างจักรยาน โรงงานจะใช้ต้นแบบของรถจักรยานเดียวในการผลิตจักรยานออกมาหลายๆคัน โดยใช้แนวความคิดที่ว่ารถจักรยานจะมีลักษณะพื้นฐานเหมือนกัน

ใน software object-oriented ก็เหมือนกัน เป็นไปได้ที่จะมีออปเจ็ค หลายๆตัวมีลักษณะบางประการเหมือนกัน ทำให้เราสามารถที่จะสร้างต้นแบบของออปเจ็คเหล่านั้นได้ ซึ่งซอฟต์แวร์ที่เป็นต้นแบบของออปเจ็ค เหล่านี้เรียกว่า class(class)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

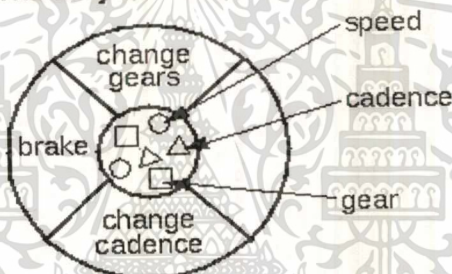
A Class



รูปที่ 2-6 ส่วนประกอบของ class

คุณสามารถสร้าง class ของ รถจักรยาน ซึ่งมีการประกาศ instance variable เพื่อใช้เก็บค่า เกียร์ และอื่นๆ ของจักรยานแต่ละคัน class ยังประกาศและมีเครื่องมือสำหรับ instance method ซึ่งทำให้ผู้ใช้สามารถเปลี่ยนเกียร์ หรือหยุดรถได้

The Bicycle Class



รูปที่ 2-7 ส่วนประกอบของ class ของรถจักรยาน

ค่าของ instance variable จะถูกกำหนดโดยแต่ละ instance ของ class ดังนั้น หลังจากคุณสร้าง class แล้ว คุณต้องสร้าง instance ของมันก่อนที่จะใช้งาน เมื่อคุณสร้าง instance ของ class นั้นหมายถึง คุณได้สร้างออบเจกต์ ที่มีชนิดเป็น class นั้น และระบบจะทำการจองเนื้อที่ไว้สำหรับ instance variable ซึ่งประกาศโดย class จากนั้นคุณสามารถอ้างถึง instance method ของออบเจกต์ นั้นเพื่อให้มันทำงานบางอย่าง ให้ instance ที่มาจาก class เดียวกันจะใช้ instance method เดียวกัน ซึ่งจะอยู่ใน class

ในการ instance variable และ method class สามารถกำหนด class variable และ class method ได้ คุณสามารถเข้าถึง class variable และ class method โดยผ่าน instance ของ class หรือจาก class โดยตรง class method จะทำงานกับ class variable เท่านั้น โดยจะไม่มีทางเข้าถึง instance variable และ instance method

ระบบจะสร้างสำเนาของ class variable ทั้งหมดสำหรับ class หนึ่งๆในครั้งแรกที่มันพบ class ในโปรแกรมเท่านั้น instance ของ class ทุกตัวจะแบ่งกันใช้ class variable นั้น เช่นรถจักรยานทุกคัน (จากต้นแบบเดียวกัน) ที่จำนวนเกียร์เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

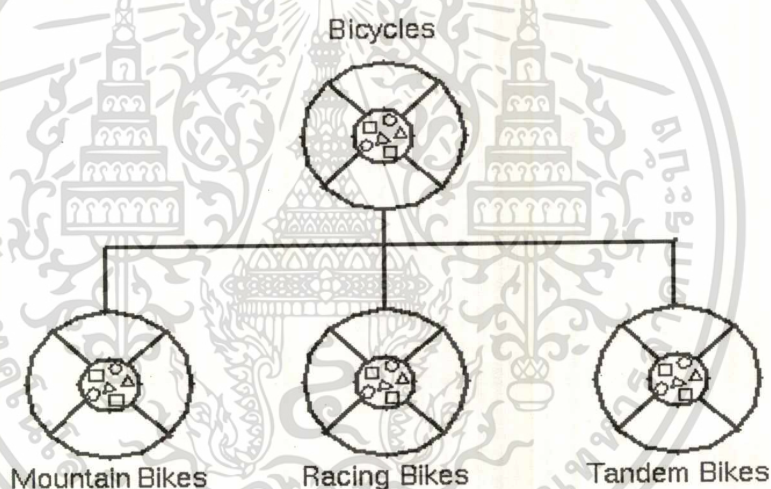
ประโยชน์ที่ได้รับจาก class

class ให้ประโยชน์ในแง่ของการทำงานไปใช้ซ้ำๆ กันได้ เช่นในโรงงานก็จะนำเอาต้นแบบมาใช้ซ้ำอีกเรื่อยๆ เพื่อผลิตจักรยานจำนวนมาก ผู้เขียนโปรแกรมก็จะใช้ class ซ้ำๆ (โค้ดเดิม) เพื่อใช้ในการผลิต ออปเจ็ค จำนวนมาก

2.3.4 Inheritance คืออะไร

ในระบบของ object-oriented คุณสามารถนิยาม class ในรูปของ class ได้ เช่น จักรยานภูเขา จักรยานเสือหมอบ จักรยานแบบ 2 คนถีบ ต่างก็เป็นจักรยานต่างชนิดกัน ใน object-oriented จักรยานภูเขา จักรยานเสือหมอบ จักรยานแบบ 2 คนถีบ เป็น subclass ของ class จักรยาน ในทางกลับกัน class จักรยาน ก็เป็น superclass ของจักรยานภูเขา จักรยานเสือหมอบ จักรยานแบบ 2 คนถีบ

Hierarchy of Classes



รูปที่ 2-8 ลำดับชั้นของ class

ในแต่ละ subclass จะรับช่วง ตัวแปร (ในรูปของการประกาศตัวแปร) จาก superclass และยังรับช่วงวิธี จาก superclass มากด้วย

อย่างไรก็ตาม subclass ไม่ได้ถูกจำกัดด้วย state และ behavior ของ superclass เท่านั้น subclass สามารถเพิ่มตัวแปร และวิธี เข้าไปยังออปเจ็ค ไหนก็ได้ที่รับช่วงมาจาก superclass เช่น รถจักรยานแบบ 2 คนถีบ นอกจากจะมีคุณสมบัติของจักรยานทั่วไปแล้ว ยังมีที่นั่งและมีที่จับเพิ่มขึ้นมาอีก เป็นต้น

subclass สามารถเขียนวิธี ที่รับช่วงต่อมาใหม่และเพิ่มการทำงานพิเศษเข้าไป เช่น จักรยานเสือภูเขาบางคนอาจมีการเพิ่มเกียร์พิเศษ ทำให้คุณต้องเขียนวิธีของการเปลี่ยนเกียร์ใหม่ เพื่อให้ผู้ขี่สามารถให้เกียร์ใหม่ที่เพิ่มขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับช่วงต่อนั้นสามารถทำได้เรื่อยๆเท่าที่ต้องการ วิธีและตัวแปร สามารถที่จะรับช่วงต่อได้เป็น ระเบียบๆ

ประโยชน์ที่ได้รับจาก inheritance

subclassสามารถเพิ่มเติม ตัวแปร และ วิธี ได้นอกเหนือที่ได้รับจาก superclass ด้วยการใช้ inheritance ทำให้เราสามารถใส่ โค้ดของ superclass ซ้ำได้หลายครั้ง

2.4 การเขียนโปรแกรมภาษาจาวา

เนื้อหาในส่วนนี้จะรวมความรู้ทั้งหมดที่จำเป็นมรกีเขียนโปรแกรม เช่น Data type, ชนิดของ Statement ที่ใช้ในการควบคุมการทำงาน, การสร้าง class และ วิธี เป็นต้น

2.4.1 ตัวแปร และ Data type

ในการประกาศใช้ ตัวแปร จะต้องประกอบด้วย 2 ส่วนคือ ชื่อของตัวแปร และ ชนิดของตัวแปร และอาจรวมถึงที่ที่ตัวแปร นั้นถูกประกาศใช้ เพื่อจะได้พิจารณาถึงขอบเขตของ ตัวแปร

ชนิดของ ตัวแปร

ตัวแปร ทุกตัวในภาษาจาวาจะต้องมีชนิดของข้อมูล ชนิดของตัวแปร จะบอกถึงชนิดของข้อมูลที่จะเก็บและการทำงานที่อาจเกิดขึ้นกับตัวแปร

ในภาษาจาวามีชนิดของตัวแปร อยู่ 2 ชนิด คือ primary type และ reference type primary type จะเก็บค่าได้ค่าเดียวและมีชนิดต่างๆดังนี้

Type	Size and Format	Description
	(whole numbers)	
byte	8-bit two's complement	Byte-length integer
short	16-bit two's complement	Short integer
int	32-bit two's complement	Integer
long	64-bit two's complement	Long Integer
	(real numbers)	
float	32-bit IEEE 754	Single-precision float point
double	64-bit IEEE 754	Double-precision float point
	(other types)	
char	16-bit Unicode character	A single character
boolean	true or false	A boolean value (true or false)

ตารางที่ 2-1 ชนิดของ ตัวแปร ในจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน reference type ค่าของ reference type (เป็นในลักษณะของ pointer) จะอ้างอิงค่า หรือกลุ่มของค่าของ ตัวแปร ไม่ได้เก็บค่าของ ตัวแปร นั้นไว้โดยตรง class และ interface ก็เป็น reference type อย่างหนึ่ง ฉะนั้นเมื่อคุณสร้าง class หรือ interface ใหม่ คุณจำเป็นต้องกำหนดชนิดของข้อมูลใหม่

ชื่อของ ตัวแปร

โปรแกรมจะอ้างอิงค่าของ ตัวแปร โดยผ่านทางชื่อของ ตัวแปร นั้นๆ ในที่ประชุมได้มีการตกลงกันว่า ชื่อของตัวแปรจะขึ้นต้นด้วยตัวเล็ก และชื่อ class ขึ้นต้นด้วยตัวใหญ่

ชื่อของตัวแปรในภาษาจาวา

1. จะต้องประกอบขึ้นด้วยตัวอักษรใน Unicode โดย Unicode จะเป็นระบบโค้ดของตัวอักษรที่ออกแบบมาเพื่อสนับสนุนข้อความในภาษา มนุษย์ที่มีแตกต่างกันมากมาย Unicode สามารถจะมีตัวอักษรได้ถึง 65,536 ตัว (ปัจจุบันมีเพียง 34,168 ตัว) ทำให้ผู้เขียนโปรแกรมสามารถเขียนโค้ดที่สื่อความหมายในภาษาของตนเอง

2. จะต้องไม่ใช่ชื่อซ้ำกับ keyword หรือ true หรือ false ของ boolean

*ไม่ได้ใช้ในปัจจุบันแล้ว

abstract	boolean	break	byte	case	catch	char	class
const *	continue	default	do	double	else	extends	final
finally	float	for	goto *	if	implement	import	instanceof
int	interface	long	native	new	null	package	private
protected	public	return	short	static	super	switch	synchronized
this	throw	throws	transient *	try	void	volatile	while

ตารางที่ 2-2 keyword ในจาวา

3. ชื่อจะต้องไม่ซ้ำกับ ตัวแปร อื่นที่ถูกประกาศภายในขอบเขต (scope) เดียวกัน

ในที่ประชุมได้ตกลงให้ชื่อของตัวแปร ขึ้นต้นด้วยตัวเล็ก ถ้าประกอบด้วยคำมากกว่า 1 คำในชื่อ เช่น isValueOf ตัวอักษรที่ขึ้นต้นของแต่ละคำยกเว้นคำแรก ให้ขึ้นต้นด้วยตัวใหญ่

Scope (ขอบเขต)

เป็นของเขตที่มีโค้ดอยู่ภายในซึ่งตัวแปรสามารถถูกเข้าถึงได้ ขอบเขตของ ตัวแปร จะถูกพิจารณาเมื่อตัวแปร ถูกสร้างและทำลาย ขอบเขตนี้จะทำให้ตัวแปร แบ่งออกเป็น 4 ประเภท

1. member variable
2. local variable
3. method parameter
4. exception handle parameters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

member variable จะเป็นสมาชิกของ class หรือ ออปเจ็ค และถูกประกาศอยู่ใน class (แต่ไม่ได้ อยู่ใน วิธี ของ class)

local variable จะถูกประกาศไว้ภายใน วิธี หรือ กลุ่มของโค้ด ภายในวิธี โดยทั่วไป local variable จะถูกเข้าถึงได้ตั้งแต่ ตัวแปร นั้นถูกประกาศจนจบขอบเขตของโค้ดที่ ตัวแปร ถูกประกาศไว้ภายใน

method parameter จะถูกใช้ในการผ่านค่าไปให้ วิธี และ constructor ขอบเขตของ วิธี พารามิเตอร์ คือภายใน วิธี หรือ ภายใน constructor ที่มันเป็น พารามิเตอร์ อยู่

exception handle parameters จะเหมือนกับ method parameter แต่จะถูกใช้ใน exception handle แทนที่จะเป็นใน วิธี หรือ constructor

การให้ค่าเริ่มต้นกับ ตัวแปร

local variable และ method variable สามารถกำหนดค่าเริ่มต้นได้เมื่อมันถูกประกาศเป็นครั้งแรก โดยค่าที่กำหนดให้ต้องสอดคล้องกับชนิดของ ตัวแปร นั้นๆ

method parameter และ exception handle parameter ไม่สามารถกำหนดค่าเริ่มต้นให้ได้ ค่าของมันจะถูกกำหนดโดย caller

2.4.2 ตัวดำเนินการ (Operator)

ตัวดำเนินการ จะทำงานได้ทั้งบนตัวถูกดำเนินการตัวเดียวและ 2 ตัว ตัวดำเนินการ ที่ต้องการ ตัวถูกดำเนินการ 1 ตัวเรียกว่า unary operators โคนสามารถเป็นได้ทั้ง prefix (ตัวดำเนินการ อยู่หน้า ตัวถูกดำเนินการ) และ postfix (ตัว ตัวดำเนินการ อยู่หลัง ตัวถูกดำเนินการ) ส่วน ตัวดำเนินการ ที่ต้องการ ตัวถูกดำเนินการ 2 ตัวเรียกว่า binary operators ซึ่งเป็นแบบ prefix อย่างเดียว (ตัวดำเนินการ อยู่ระหว่าง ตัวถูกดำเนินการ) ในการทำงานของ ตัวดำเนินการ จะมีการให้ค่ากลับมา ซึ่งค่าและชนิดของค่าที่ส่งกลับมา จะขึ้นอยู่กับ ตัวดำเนินการ และ ชนิดของ ตัวถูกดำเนินการ

ตัวดำเนินการ แบ่งออกเป็น

1. arithmetic
2. relational and conditional
3. bitwise logical
4. assignment

Arithmetic operator

ในจาวามีการสนับสนุน arithmetic operator จำนวนมาก ซึ่งมี ตัวดำเนินการ ต่างๆดังนี้

Operator	Use	Description
+	op1 + op2	Adds op1 and op2
-	op1 - op2	Subtracts op2 from op1
*	op1 * op2	Multiplies op1 by op2
/	op1 / op2	Divides op1 by op2
%	op1 % op2	Computes the remainder of dividing op1 by op2

ตารางที่ 2-3 binary arithmetic operation ของจาวา

หมายเหตุ เครื่องหมาย + ยังใช้ในการเชื่อ String เข้าด้วยกัน

เครื่องหมาย + และ - ยังใช้โคดในการบอกถึงค่าของ ตัวถูกดำเนินการ ว่ามากกว่าหรือน้อยกว่า 0

Operator	Use	Description
+	+ op	Indicates a positive values
-	- op	Arithmetically negates op

ตารางที่ 2-4 เครื่องหมายที่ใช้บอกค่าของ ตัวถูกดำเนินการ

นอกจากนี้เรายังใช้เครื่องหมาย ++ และ -- แทนการเพิ่มค่าและลดค่าที่ละ 1 ของ ตัวถูกดำเนินการ

ได้

Operator	Use	Description
++	op++	Increments op by 1; evaluates to value before incrementing
++	++op	Increments op by 1; evaluates to value after incrementing
--	op--	Decrements op by 1; evaluates to value before decrementing
--	--op	Decrements op by 1; evaluates to value after decrementing

ตารางที่ 2-5 การใช้ prefix และ postfix ในการเพิ่มและลดค่า

Relational and Conditional Operators

relational operator จะใช้ในการเปรียบเทียบค่าของ ตัวถูกดำเนินการ 2 ตัว และ พิจารณาความสัมพันธ์ระหว่าง ตัวถูกดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operator	Use	Return true if
>	op1 > op2	op1 is greater than op2
>=	op1 >= op2	op1 is greater than or equal to op2
<	op1 < op2	op1 is less than op2
<=	op1 <= op2	op1 is less than or equal to op2
==	op1 == op2	op1 and op2 are equal
!=	op1 != op2	op1 and op2 are not equal

ตารางที่ 2-6 relational operator ในจาวา

บ่อยครั้งที่ relational operator จะถูกใช้ร่วมกับ condition ตัวดำเนินการ เพื่อใช้ในการพิจารณา หรือ ตัดสินใจที่ซับซ้อน

Operator	Use	Return true if
&&	op1 && op2	op1 and op2 are both true
	op1 op2	either op1 or op2 is true
!	! op	op is false

ตารางที่ 2-7 condition operator ในจาวา

เครื่องหมาย & สามารถใช้แทน && เช่นเดียวกับ | ซึ่งใช้แทน || ได้ ถ้าตัวถูกดำเนินการ ทั้ง 2 ตัวเป็นboolean

Bitwise operators

bitwise operator อนุญาตให้คุณทำงานกับบิต ของข้อมูลได้ bitwise operator แสดงในตารางต่อไปนี้

Operator	Use	Operation
>>	op1 >> op2	shift bit of op1 right by distance op2
<<	op1 << op2	shift bit of op1 left by distance op2
>>>	op1 >>> op3	shift bit of op1 right by distance op2 (unsigned)
&	op1 & op2	bitwise and
	op1 op2	bitwise or
^	op1 ^ op2	bitwise xor
~	~ op	bitwise complement

ตารางที่ 2-8 Bitwise operator ในจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Assignment operators

คุณสามารถใช้เครื่องหมาย “=” ในการกำหนดค่าให้กับตัวอื่นๆ นอกจากการกำหนดค่าโดยใช้เครื่องหมาย = แล้ว จาวายังเพิ่ม assignment operator เพื่อให้คุณสามารถทำ arithmetic , logical หรือ bitwise operator ได้ 1 ครั้งก่อนที่จะให้ค่านั้นกับ ตัวถูกดำเนินการ

Operator	Use	Equivalent to
+=	op1 += op2	op1 = op1 += op2
-=	op1 -= op2	op1 = op1 -= op2
*=	op1 *= op2	op1 = op1 *= op2
/=	op1 /= op2	op1 = op1 /= op2
%=	op1 %= op2	op1 = op1 %= op2
&=	op1 &= op2	op1 = op1 &= op2
=	op1 = op2	op1 = op1 = op2
^=	op1 ^= op2	op1 = op1 ^= op2
<<=	op1 <<= op2	op1 = op1 <<= op2
>>=	op1 >>= op2	op1 = op1 >>= op2
>>>=	op1 >>>= op2	op1 = op1 >>>= op2

ตารางที่ 2-9 assignment operator เพิ่มเติม

2.4.3 Statement ที่ใช้ในการควบคุมการทำงาน

ในจาวามี statement ที่ใช้ในการควบคุม หรือ ลำดับการทำงานอยู่หลายประเภท ดังตาราง

Statement	Keyword
making decision	if - else , switch - case
loop	for , while do - while
exception	try - catch - finally , throw
miscellaneous	break , continue , label: , return

ตารางที่ 2-10 ประเภทของ statement ที่ใช้ควบคุมการทำงาน

if - else Statement

if - else statement ทำให้โปรแกรมของคุณสามารถเลือกทำงานตามเหตุการณ์ที่เข้ามา โดยรูปแบบที่ง่ายที่สุดอยู่ในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (expression)
    statement

```

โปรแกรมจะทำ statement ถ้า expression เป็นจริง โดยจะไม่สนใจถ้า expression เป็นเท็จ แต่ถ้าต้องการพิจารณาในกรณีที่ expression เป็นเท็จด้วยเราจะเพิ่ม statement *else* เข้าไปเป็น

```

if (expression) {
    // codes to perform when expression is true --
} else {
    // codes to perform when expression is false --
}

```

else statement ยังมีอีกรูปแบบหนึ่งคือ *else if* ซึ่งจะทำการ statement โดยขึ้นอยู่กับ expression อื่น มีรูปแบบดังนี้

```

if (expression) {
    statement
} else if (expression1) {
    statement1
} else if (expression2) {
    statement2
} else {statement n}

```

ในกรณีนี้ถ้า expression เป็นจริง โปรแกรมก็จะทำการ statement แต่ถ้า expression เป็นเท็จก็จะไปตรวจสอบ expression1 ต่อว่าเป็นจริงหรือเท็จ ถ้าเป็นจริงจึงทำการ statement1 ถ้าเป็นเท็จจะไปตรวจสอบ expression2 ต่อ แต่ถ้าไม่มี expression ใดที่เป็นจริงเลย โปรแกรมจะทำการ statement n if statement สามารถมี else if statement ได้หลายตัว แต่มี else statement ได้เพียงตัวเดียว

switch Statement

switch statement จะทำการพิจารณา expression ของมัน และจะเลือกทำ case statement ที่เหมาะสม ถ้าสังเกตเราจะเห็นว่า เราสามารถใช้ else if statement เขียนแทน switch statement ได้

switch statement ยังสามารถมี break statement หลัง case statement แต่ละตัว break statement จะทำให้การทำงานของ switch statement หยุดและไปทำ statement แรกที่ตามหลัง switch statement break statement มีความจำเป็นเนื่องจาก case statement จำทำงานแบบเรียงตามลำดับไปเรื่อยๆจนหมด case statement

```

int month;
int numDays;
...

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch (month) {
case 1: System.out.println("January");
case 3: System.out.println("March");
case 5: System.out.println("May");
case 7: System.out.println("July");
case 8: System.out.println("August");
case 10: System.out.println("October");
case 12: System.out.println("December");
        numDays = 31; break;
case 4: System.out.println("April");
case 6: System.out.println("June");
case 9: System.out.println("September");
case 11: System.out.println("November");
        numDays = 30; break;
case 2: System.out.println("February");
if ( ((year % 4 == 0) && !(year % 100 == 0)) || (year % 400 == 0) )
    numDays = 29;
else numDays = 28; break;
default : System.out.println("Hay! That's is not a valid month.");
}

```

ท้ายสุดคุณสามารถใช้ default statement ที่ท้ายสุดของ switch เพื่อจะรองรับค่าที่ไม่ปรากฏใน case ตัวใดเลย ดังตัวอย่าง ถ้าค่าของเดือนที่รับมา ไม่ใช่ 1-12 ก็จะปรากฏข้อความ Hay! That's not a valid month.

loop Statement

สำหรับ loop statement จะเป็นการทำ statement หรือ กลุ่มของ statement ซ้ำเรื่อยๆ ในขณะที่ expression เป็นจริง แบ่งออกเป็น 3 แบบคือ

1. while statement มีรูปแบบดังนี้

```

while (expression)
    statement

```

โดย ถ้า expression ของ while statement เป็นจริง ถึงจะทำ statement โดนที่ statement สามารถเป็น statement เดียว หรือเป็นกลุ่มของ statement ซึ่งตามกฎของจาวาจะต้องอยู่ภายในเครื่องหมายปีกกา (“{” และ “}”)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **for loop** จะใช้เมื่อคุณทราบถึงข้อบังคับที่แน่นอนของ loop (ค่าเริ่มต้น , ค่าสิ้นสุด , ค่าที่เพิ่มขึ้นในแต่ละครั้ง) for loop มักใช้กับการทำงานกับข้อมูลใน array หรือตัวอักษรใน string โดย for loop มีรูปแบบดังนี้

```
for (initialization; termination; increment;)
```

```
statement
```

initialization เป็น expression ที่ใช้ในการเริ่มต้น loop ซึ่งจะถูกระบุ execute ครั้งเดียวเมื่อเริ่มต้นทำ loop termination เป็น expression ที่ใช้ในการพิจารณาเมื่อไหร่ถึงจะสิ้นสุด loop expression นี้จะถูกพิจารณาก่อนที่จะเริ่มทำงานในแต่ละ loop เมื่อพิจารณาว่าเป็นเท็จ loop ก็จะสิ้นสุด increment เป็น statement ที่จะทำในทุกๆ loop หลังจากที่ได้พิจารณา terminate statement แล้วว่าเป็นจริง

องค์ประกอบของ for loop ที่กล่าวมาแต่ละตัว หรือ ทั้งหมดสามารถละได้ โดยใส่เครื่องหมาย Semi-colon (;) แทน statement แต่ละตัวที่จะละ

3. **do - while loop** จะคล้ายกับ while loop ต่างกันที่ do - while loop จะทำการพิจารณา expression หลังการทำงานจบแต่ละ loop โดยมีรูปแบบดังนี้

```
do {
    statements
} while (booleanExpression);
```

Exception handle Statements

เมื่อเกิดข้อผิดพลาดขึ้นใน วิธี ของจาวา วิธี สามารถที่จะส่ง exception ไปยังตัวที่เรียกใช้ วิธี นั้น เพื่อบอกให้รู้ว่าเกิดข้อผิดพลาดขึ้น และ ชนิดของข้อผิดพลาดที่เกิด วิธี ที่เป็นตัวเรียกสามารถใช้ try , catch และ finally statement ในการจับและแก้ไขข้อผิดพลาดที่เกิด

Branching Statements

1. **break statement** ในตัวอย่างของ switch statement break statement ทำให้คุณสามารถกระโดดการทำงานจาก statement ปัจจุบัน ไปยัง statement ที่อยู่ถัดจาก statement ปัจจุบันทันที

ยังมี break statement รูปแบบอื่นที่จะกระโดดการทำงานไปยัง label statement โดยคุณสามารถกำหนด label ให้กับ statement ได้ โดยใส่ label ตามด้วยเครื่องหมาย “ : “ แล้วตามด้วย statement

```
breakToHere : someJavaStatement
```

การกระโดดการทำงานจาก break statement ไปยัง label statement ทำได้โดยใช้ break statement ตามด้วยชื่อของ label statement ที่ต้องการจะกระโดดไปทำงาน ดังนี้

```
break breakToHere ;
```

2. continue statement จะอยู่ใน loop เพื่อใช้กระโดดจาก statement ปัจจุบัน ไปยังต้นของ loop หรือไปยัง label statement ถ้าต้องการกระโดดไปยัง label statement จะต้องใช้ continue statement ที่ตามด้วย label ของ statement ที่จะทำ

3. return statement ใช้เพื่อออกจาก วิธี ปัจจุบัน กลับไปยัง statement ที่อยู่ใน วิธี ที่เป็นตัวเรียก การ return มี 2 แบบคือ มีการส่งค่ากลับไปด้วย กับ ไม่มีการส่งค่ากลับ การส่งค่ากลับไปด้วยทำได้โดยเติมค่า หรือ expression ที่ใช้หาค่า หลัง return

2.4.4 Arrays and Strings

Array

เหมือนกับตัวแปรอื่น ก่อนที่จะใช้ array ได้จะต้องมีการประกาศก่อน ซึ่งการประกาศต้องประกอบด้วย ชนิดของ array และ ชื่อของ array ชนิดของ array คือชนิดของข้อมูลที่จะเก็บใน array นั้นๆ โดยจะต้องกำหนดชนิดของ array เพื่อประกาศใช้ array การประกาศ array ไม่ได้มีการจองเนื้อที่เพื่อใช้เก็บข้อมูลของ array ถ้าโปรแกรมพยายามเข้าถึงหรือเก็บค่าลงใน array ก่อนมีการจองเนื้อที่ compiler จะบอกถึงความผิดพลาด ในการจองเนื้อที่ให้กับ array คุณต้องทำการ instance array ก่อน (ขั้นตอนเหมือนกับการ instance object จาก class : declaration , instantiation , initialization)

ปรกติในการสร้าง array เราใช้ operator new ตามด้วยชนิดของข้อมูลใน array และจำนวนของข้อมูลที่จะเก็บซึ่งอยู่ภายในเครื่องหมาย “[” และ “]”

```
elementType [] arrayName = new elementType [arraySize]
```

ขณะนี้ได้มีการจองเนื้อที่บางส่วนให้กับให้กับ array ของคุณแล้ว คุณสามารถเก็บค่าหรือเข้าถึงข้อมูลภายใน array ได้ ในจาวา array จะเริ่มที่ 0 และ สิ้นสุดที่ ขนาดของ array ลบด้วย 1 ข้อมูลใน array ของจาวานั้น สามารถเก็บ reference type ได้ เมื่อเราทำการสร้าง array ของ reference type เนื้อที่ที่จองไว้จะเก็บค่าที่ใช้ในการ reference ไปยังข้อมูลอีกที่ แต่ไม่ได้จองเนื้อที่สำหรับข้อมูลไว้ ฉะนั้นหากพยายามเข้าถึงข้อมูลโดยตรงจะเกิดข้อผิดพลาดขึ้น

String

ตัวอักษรที่นำมาเรียงต่อกันเราเรียกว่า string โดยถูกนำมาใช้ในจาวาในรูปแบบ string class (อยู่ใน package java.lang)

การเชื่อมต่อ string สำหรับจาวาอนุญาตให้ใช้เครื่องหมาย “+” โดยมากมักใช้ในการพิมพ์ output ดังตัวอย่างข้างล่าง

```
“Input has “ + count + “ chars.”
```

เป็นการนำ string 2 ตัวมาเชื่อมต่อกันคือ “Input has “ กับ “ chars.” (string จะอยู่ในเครื่องหมาย “ “) string ตัวที่ 3 (ตัวที่อยู่ตรงกลาง) ซึ่งเป็นตัวแปรอาจเป็นชนิดอื่นนอกจาก string ก็ได้ จะถูกแปลงเป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

string และถูกเชื่อมเข้ากับตัวอื่น (String ไม่สามารถแปลงไปเป็นชนิดอื่นได้หลังจากที่มันถูกสร้างขึ้นแล้ว)

2.4.5 ออปเจ็ค และ class

วงจรชีวิตของออปเจ็ค

เราสามารถแบ่งวงจรชีวิตของจาวาออกเป็น 3 ช่วงด้วยกันคือ

1. สร้างออปเจ็ค หรือเรียกอีกอย่างว่าสร้าง instance ของ class ซึ่งจะประกอบด้วยขบวนการ 3 อย่างคือ declaration , instantiation และ initialization ดังตัวอย่างข้างล่าง

```
Rectangle rect = new Rectangle (0 , 0 , 100 , 200);
```

statement ข้างบนจะสร้าง Rectangle object ขึ้นมาใหม่ (Rectangle เป็น class หนึ่งใน package java.awt)

Declaration การ declare จะเหมือนกับการ declare variable ทั่วไป โดยประกอบด้วย ชนิดของออปเจ็ค และ ชื่อของ ออปเจ็ค

```
type name ( Rectangle rect )
```

การ declare ไม่ได้สร้าง ออปเจ็ค ขึ้นมา เป็นเพียงการบอกให้ compiler รู้ว่าเราจะให้ name ในการอ้างถึง ตัวแปร ที่มีชนิดเป็น type

Instantiation new operator จะสร้าง instance ของ class โดยการจองเนื้อที่ในหน่วยความจำ new ต้องการ argument ตัวเดียวคือ ตัวเรียก constructor method constructor method เป็น วิธี พิเศษที่มีให้ในแต่ละ class ของจาวา โดยจะตอบสนองในการ initialization object ใหม่ ดังนั้น new จะสร้าง ออปเจ็ค ใหม่ และ constructor method จะทำ initialization

Rectangle (0, 0 , 100 , 200) จะไปเรียก constructor สำหรับ Rectangle new operator จะส่งค่าที่อ้างไปยัง ออปเจ็ค ใหม่กลับมา ซึ่งจะถูกกำหนดให้กับ ตัวแปร ที่มี type ที่เหมาะสม

Initialization ดังที่กล่าวไว้ class จะมี constructor method ซึ่งใช้ในการ initialize object ใหม่ class หนึ่งๆสามารถมี constructor ได้หลายตัวเพื่อให้สามารถทำการ initialization ได้หลายแบบ constructor method นี้จะมีชื่อเหมือนกับ class โดยจะเป็นแบบไม่ส่งค่ากลับ

```
Rectangle ( 0 , 0 , 100 , 200 )
```

constructor ที่ไม่ต้องการ argument (ในตัวอย่างข้างบนต้องการ argument 4 ตัว) เรียกว่า default constructor ซึ่ง class ส่วนมากจะมี 1 constructor และเป็น default constructor

2. ใช้ออปเจ็ค โดยมีวิธีการในการใช้ดังนี้

การอ้างถึง ตัวแปรของออปเจ็ค ในการอ้างถึง ตัวแปร ของ ออปเจ็ค ทำได้โดย

```
objectReference . variable
```

เมื่อคุณเข้าถึง ตัวแปรของออปเจ็ค หมายความว่า คุณเข้าถึงตัวแปรของออปเจ็ค เฉพาะแต่ละตัว เนื่องจากในการอ้างถึงตัวแปรในออปเจ็ค ได้กล่าวแล้ว ต้องเป็นสิ่งที่อ้างอิง ไปถึงออปเจ็ค ดังนั้นเราสามารถ ใช้ new operator ซึ่งจะให้ค่าที่อ้างถึงออปเจ็คกลับมาได้

```
height = new Rectangle (0 , 0 , 100 , 200).height
```

การเรียกใช้วิธีของออปเจ็ค การเรียกใช้วิธีของออปเจ็คเหมือนกับการเรียกใช้ ตัวแปร ของ ออปเจ็ค โดยการเอาชื่อ วิธี ที่ต้องการตามหลัง ออปเจ็ค reference โดยมี จุด (".") ขึ้นกลาง และส่ง argument ไปยัง วิธี ภายในเครื่องหมายวงเล็บ ถ้าต้องการ argument ก็ให้ใส่วงเล็บเปล่า

```
objectReference . methodName (argumentList); หรือ
```

```
objectReference . methodName ();
```

การเรียกใช้ วิธี เราอาจเรียกอีกอย่างว่า ข่าวสารs ซึ่งมีองค์ประกอบ 3 อย่าง (ชื่อ ออปเจ็ค ผู้รับ , ชื่อ วิธี , พารามิเตอร์ ที่วิธีต้องการ) ดังตัวอย่างข้างล่าง

```
rect . move (36 , 56);
```

เป็น ข่าวสาร ที่ส่งไปให้ object rect บอกให้ทำ method move โดยส่ง พารามิเตอร์ 36 กับ 56 ให้กับ วิธี

3. ทำลายออปเจ็คที่ไม่ได้ใช้ ในจาวาอนุญาตให้เราสร้างออปเจ็ค ได้มากเท่าที่ต้องการ โดยที่ระบบยังสามารถรองรับได้ โดยจะไม่ทำลาย JRE (Java Runtime Environment) จะลบออปเจ็ค ออกถ้ามันเห็นว่าไม่ได้ใช้อีกต่อไปแล้ว ซึ่งขบวนการนี้เรียกว่า garbage collection ออปเจ็ค ที่ถูกเลือกเพื่อทำลายคือ ออปเจ็ค ที่ไม่มีตัวใดอ้างถึง ออปเจ็ค นั้นแล้ว ประกติกการอ้างถึงใน ตัวแปร มักจะถูกทิ้งไปเมื่อออกนอก scope หรือคุณสามารถทิ้ง object reference ได้โดยการให้ค่ากับ ตัวแปร ที่มีชนิดของข้อมูลเป็น reference type นั้น

ก่อนที่ออปเจ็ค จะถูกทำลาย garbage collector (JRE) จะให้โอกาส ออปเจ็ค ที่จะปล่อย resource ที่ใช้ หรือ drop reference เพื่อให้ออปเจ็คอื่นมีโอกาสได้ใช้ ซึ่งขบวนการนี้เรียกว่า finalization โดย finalize method จะอยู่ใน object class class จะต้องทำการเขียน finalize method ใหม่เพื่อให้เหมาะกับ object ของ type นั้นๆ

การสร้าง class

ประกอบด้วย 2 อย่างคือ class declaration และ class body

```
classDeclaration {
    classBody
}
```

Class declaration

ในการ declare class อย่างน้อยที่สุดต้องมี class ซึ่งเป็น keyword และ ชื่อ class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
class NameOfClass {
```

```
...
```

```
}
```

ชื่อ class ต้องขึ้นต้นด้วยตัวใหญ่ อย่างไรก็ตาม class declaration สามารถบอกรายละเอียดเกี่ยวกับ class ได้อีก ได้แก่

1. ประกาศว่า class ใดเป็น superclass
2. แสดง interface ที่ implement โดย class
3. บอกว่า class เป็นแบบ-public , abstract หรือ final

การประกาศว่า class ใดเป็น superclass

ในจาวาทุกๆ class จะต้องมี superclass ถ้าคุณไม่ได้กำหนด มันจะถือว่าเป็น object class (เป็น class ที่เป็น superclass ของทุก class ในจาวาทั้งทางตรงและทางอ้อม ประกาศอยู่ใน java.lang) ในการกำหนด superclass ให้ใช้ keyword extends ตามด้วยชื่อของ superclass โดยจะอยู่ระหว่าง ชื่อ class และ “{“

```
class NameOfClass extends SuperClassName {
```

```
...
```

```
}
```

แสดง interface ที่ implement โดย class

interface จะประกาศกลุ่มของวิธี และข้อกำหนด ที่ไม่ได้มีการ implement ให้กับ วิธี ใด เมื่อ class ต้องการจะ implement interface ก็จะมีหมายถึง class ต้องการที่จะ implement method ทุกตัวที่ประกาศใน interface

class สามารถที่จะ implement interface ได้มากกว่าหนึ่ง โดยใช้ keyword interfaces ตามด้วย ชื่อ interface (ถ้ามีมากกว่า 1 ให้ใช้เครื่องหมาย “,” ขึ้นระหว่างชื่อ) ถ้า implement ไม่ครบทุกวิธี ที่มีใน interface compiler จะฟ้องถึงข้อผิดพลาด

public, abstract และ final class

คุณสามารถเลือก class modifier (public , abstract , final) ตัวใดตัวหนึ่งสำหรับ class declaration เพื่อประกาศว่า class ของคุณเป็นแบบ public , abstract หรือ final modifier จะมาก่อน keyword class

public modifier จะประกาศว่า class ของคุณสามารถถูกใช้โดย object หรือ class ที่อยู่นอก package ปัจจุบันได้ โดยปรกติแล้ว class จะถูกใช้โดย class อื่นที่อยู่ใน package เดียวกับที่มันถูกประกาศไว้เท่านั้น โดยมรการตกลงว่า ถ้าใช้ public keyword ใน class declaration จะพยายามให้มันอยู่เป็นตัวแรกในการ ประกาศ

abstract modifier จะประกาศว่า class ของคุณเป็น abstract class ซึ่ง abstract class จะมี abstract method (เป็น วิธี ที่ยังไม่มีการ implement) อยู่ใน abstract class มีแนวโน้มที่จะเป็น subclass และ ไม่สามารถสร้าง instance ได้

final modifier จะประกาศว่า class ของคุณ นั้นคือ class ของคุณไม่สามารถเป็น subclass ได้ เนื่องจากเหตุผล 2 ประการคือ เพื่อความปลอดภัย และ เพื่อการออกแบบ

สรุปรูปแบบของ class declaration ได้ดังนี้

```
[modifiers] class ClassName [extends SuperClassName] [implements InterfaceName] {
...
}
```

Class Body

ตัว class จะแบ่งออกเป็น 2 ส่วนคือ variables declaration และ วิธี โดยมากเราจะทำการประกาศ ตัวแปร ก่อนตามด้วย method declaration และ implement

```
ClassDeclaration {
    memberVariableDeclarations
    methodDeclarations
}
```

Declaration member variables

การประกาศ ตัวแปร จะประกอบด้วย 2 ส่วนคือ ชนิดของ ตัวแปร และ ชื่อของ ตัวแปร

```
type variableName ;
```

การ declare นี้จะเหมือนกับ local variables และ method parameters ทุกอย่างแม้แต่การตั้งชื่อ โดยชื่อของ ตัวแปร สามารถซ้ำกับชื่อของ วิธี ใน class เดียวกันได้

นอกจากนี้เรายังสามารถกำหนดคุณสมบัติอื่นๆของ ตัวแปร ได้ดังนี้

```
[accessSpecification] [static] [final] [transient] [volatile] type variableName ;
```

accessSpecification กำหนดการเข้าถึง ตัวแปร จาก class อื่น การควบคุมการเข้าถึง วิธี ก็ใช้ specification เดียวกันดังตาราง

Specifier	class	subclass	package	world
private	X			
protected	X	X *	X	
public	X	X	X	X
package	X		X	

ตารางที่ 2-11 ระดับการเข้าถึงของแต่ละ specifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. `static` ใช้เพื่อแสดงว่า ตัวแปรเป็นสมาชิกของ class variable (ไม่ใช่ของ instance variable) เราสามารถใช้ `static` ในการประกาศ class method ได้
2. `final` ใช้เพื่อแสดงว่าตัวแปร เป็นค่าคงที่
3. `transient` ใช้เพื่อแสดงว่าตัวแปร ไม่ใช่ส่วนหนึ่งของสถานะที่เป็นอยู่ของ ออปเจ็กต์
4. `volatile` ใช้แสดงว่าตัวแปรสามารถเปลี่ยนแปลงแบบ asynchronous สำหรับการทำงานของ thread พร้อมกัน

Implementing methods

เหมือนกับการ `implement class` มีองค์ประกอบ 2 ส่วนคือ การประกาศวิธี และตัววิธี

```
methodDeclaration {
    methodName
    methodNameBody
}
```

Method declaration อย่างน้อยต้องประกอบด้วย ชื่อ วิธี และ ชนิดของข้อมูลที่ วิธี ส่งกลับ

```
returnType methodName () {
    ...
}
```

ในการส่งค่ากลับ ชนิดของข้อมูลที่ส่งกลับเป็นได้ทั้ง primitive type และ reference type (ถ้าไม่มีการส่งค่ากลับให้ใส่ keyword `void`) วิธี จะใช้ `return operator` ในการส่งค่ากลับ วิธี ใดที่ไม่ได้ประกาศ `void` ได้ ต้องมี `return statement` ชนิดของข้อมูลที่ส่งกลับมาต้องเหมือนกับที่ วิธี ได้ประกาศไว้ ชื่อของ วิธี มีวิธีการตั้งชื่อดังนี้

1. วิธี สามารถใช้ชื่อซ้ำกันได้ แต่มีการส่งค่า พารามิเตอร์ ต่างกัน (ชนิดของ พารามิเตอร์ ต่างกัน)
2. วิธี ที่มีชื่อเดียวกับ class จะเป็น constructor ซึ่งจะทำหน้าที่ในการ initial object ที่สร้างขึ้นใหม่ของ class type นั้น
3. class สามารถ override method ใน superclass ของมันได้ การ override method จะต้องมีชื่อ , return type , parameter list เหมือนกับ วิธี ที่จะ override ลักษณะเพิ่มเติมของ method declaration ซึ่งจะบอกถึง argument ที่ วิธี ยอมรับ
[accessSpecifier] [static] [abstract] [final] [native] [synchronized]

```
returnType methodName ([paramList]) [throws exceptionsList]
```

Passing Information into a Method เมื่อคุณเขียน วิธี คุณประกาศจำนวนและชนิดของ argument ที่ วิธี นั้นต้องการ โดยจะต้องบอกชื่อและชนิดของ argument แต่ละตัวใน method signature ถ้ามี argument มากกว่า 1 ตัว แต่ละตัวให้ขึ้นด้วยเครื่องหมาย “,”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

methodDeclaration ( argumentType argumentName ) {
    methodBody
}

```

ชนิดของ argument สามารถเป็นได้ทั้ง primitive type และ reference type ส่วนชื่อของ argument สามารถซ้ำกับชื่อของ class variable ได้ ซึ่งมักใช้ใน constructor การใช้ชื่อของ argument ที่ซ้ำกับ ชื่อของ class variable ทำให้การอ้างชื่อใน วิธี หมายถึง argument ถ้าต้องการอ้างถึง class variable (member variable) ให้ใช้ keyword this ตามด้วยจุด (",") และชื่อของ class variable ที่ซ้ำกับ argument

Method Body ในตัวของ วิธี คุณสามารถใช้ keyword this ในการอ้างถึงสมาชิกในออปเจ็กต์ ปัจจุบัน (ออปเจ็กต์ ที่ วิธี ถูกเรียก) ใช้ keyword super อ้างถึงสมาชิกใน superclass ซึ่ง ออปเจ็กต์ ปัจจุบันถูกซ่อน หรือ ถูก override นอกจากนั้นภายใน วิธี ยังสามารถทำการ declare variable ซึ่งเป็น local variable ได้ด้วย

Instance and Class member

เมื่อคุณประกาศ member variable เช่น

```

Class MyClass {
    float aFloat ;
}

```

หมายถึงคุณได้ประกาศ instance variable ทุกครั้งที่คุณสร้าง instance ของ class ระบบจะสร้างสำเนา instance variable ของ class ให้กับแต่ละ instance variable จะต่างกับ class variable (ซึ่งคุณประกาศโดยใช้ static modifier) โดยระบบจะทำการจองเนื้อที่ให้กับ class variable เพียงครั้งเดียวเมื่อพบ class นั้นครั้งแรก instance อื่นของ class นั้นจะให้ class variable นี้ร่วมกัน คุณสามารถเข้าถึง class variable โดยผ่าน instance หรือ class โดยตรง

วิธี ก็เช่นเดียวกัน มีได้ทั้ง instance method และ class method (ทำได้โดยประกาศใน method declaration โดยใช้ static modifier เช่นเดียวกับ class variable) โดยที่ instance method จะทำงานกับ instance variable ของ ออปเจ็กต์ ปัจจุบัน แต่ก็สามารถเข้าถึง class variable ได้ ส่วน class method จะทำงานกับ class variable เท่านั้นไม่สามารถเข้าถึง instance variable ที่ประกาศอยู่ใน class เดียวกับมันได้ อย่างไรก็ตามสมาชิกที่ถูกประกาศอยู่ใน class เราเรียกว่า instance member

instance ของ class ทุกตัวจะทำการ implement class method ร่วมกัน instance method จะอ้างถึง instance variable ของ ออปเจ็กต์ โดยใช้ชื่อของ instance variable นั้นได้เลย เนื่องจากออปเจ็กต์ จะซ่อน instance variable ไว้โดย พารามิเตอร์ ของวิธี

ข้อแตกต่างอีกข้อของ instance member กับ class member คือ class member สามารถเข้าถึงจาก class ได้เลย ไม่ต้อง instantiate class เพื่อเข้าถึง class member

Life Cycle Methods

วงจรชีวิตของ วิธี แบ่งออกเป็น 2 ส่วนด้วยกันคือ

Constructor เป็นช่วงเริ่มต้นของวงจรชีวิต ทุก class ในจาวาจะมีวิธีพิเศษที่เรียกว่า constructor ซึ่งใช้ในการเริ่มออปเจ็กต์ใหม่ของ type นั้น constructor จะมีชื่อเดียวกับ class และ class หนึ่งๆจะมี constructor ได้หลายตัว เราแยกแยะ constructor เหล่านี้ได้โดยจำนวน หรือ ชนิดของ argument

constructor จะใช้ argument ของมันในการ initial สถานะของออปเจ็กต์ เมื่อคุณสร้าง ออปเจ็กต์คุณต้องเลือก constructor ที่มี argument ตรงตามที่คุณต้องการมากที่สุด

เมื่อคุณเขียน class ของคุณขึ้นมาเอง คุณไม่จำเป็นต้องเขียน constructor สำหรับมัน constructor ที่ไม่ต้องการ argument จะถูกจัดให้ระบบโดยอัตโนมัติ (สำหรับทุก class) การ declare และ implement จะเหมือนกับ วิธี ทั่วไป แต่ใน constructor จะไม่มีการส่งค่ากลับ

ในบางครั้งเราต้องการ initial โดยใช้ constructor ของ superclass ซึ่งทำได้โดย

```
super (" ClassName ");
```

statement super นี้จะอยู่เป็น statement แรกในตัว constructor

ในการประกาศ constructor คุณสามารถใช้ specifier ในการกำหนดว่า ออปเจ็กต์ ใดบ้างที่สามารถสร้าง instance ของ class คุณได้

private ไม่มี class ใดสามารถ instantiate class ของคุณเป็น ออปเจ็กต์ ได้

protected เฉพาะ subclass ของ class คุณเท่านั้นที่สามารถสร้าง instance ของมันได้

public ใครก็ได้สามารถสร้าง instance ของ class คุณได้

package ไม่มี class ใดที่อยู่นอก package (ที่คุณอยู่) ที่จะสร้าง instance ของ class คุณได้

Writing a finalize Method ก่อนที่วิธีจะถูกทำลาย ระบบจะเรียก finalize method เพื่อให้มันปล่อย resource ที่มีอยู่ก่อนถูกทำลาย คุณสามารถสร้าง finalize method ได้โดยกำหนดและ implement method ของคุณให้มีชื่อ finalize การประกาศ finalize method ทำได้ดังนี้

```
protected void finalize() throws Throwable
```

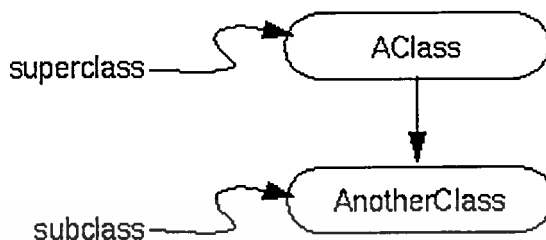
finalize method ถูกประกาศไว้ใน class java.lang.Object ดังนั้นเมื่อคุณเขียน finalize ของ class คุณก็เป็นการ override ใน superclass ตัวหนึ่ง

ถ้า superclass ของ class คุณมี finalize method finalize method ของ class คุณควรเรียก finalize ของ superclass หลังจากที่คุณทำความสะอาดเสร็จแล้ว

```
protected void finalize() throws Throwable {
    ... // Clean up code for this class here
    super.finalize();
}
```

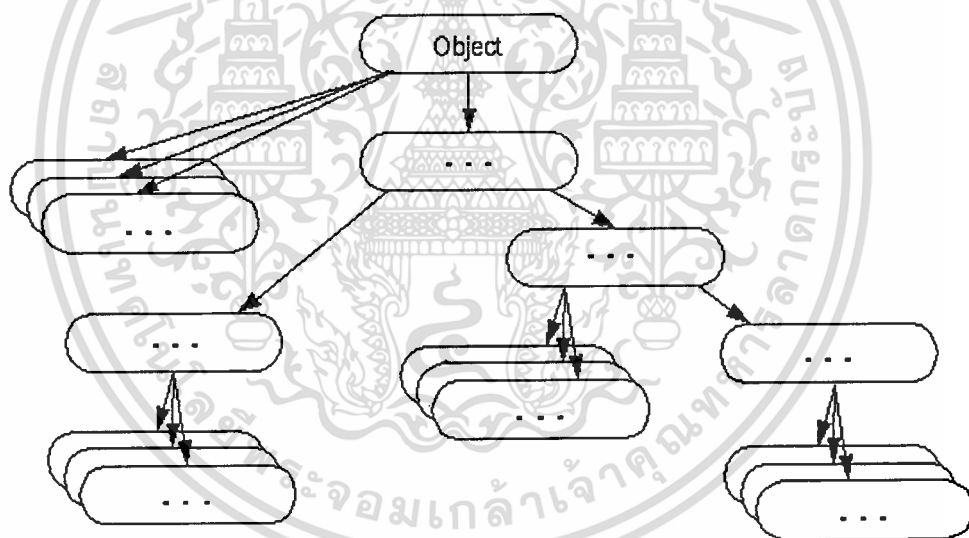
Subclass , Superclass and Interface

จากที่เราได้รามาแล้วว่า class อาจได้มาจาก class อื่น class ที่มีกำเนิดจาก class อื่นเรียกว่า subclass ส่วน class ที่ให้กำเนิด class อื่นเรียกว่า superclass



รูปที่ 2-9 superclass และ subclass

ความเป็นจริงแล้ว ในจาวาทุก class จะได้มาจาก class อื่นทั้งสิ้น โดย class ที่เป็นต้นแบบหรือให้กำเนิด class อื่นๆทั้งหมดในจาวาคือ class Object ซึ่งอยู่ใน java.lang



รูปที่ 2-10 class heirarchy

subclass จะรับช่วง state และ behavior ในรูปของตัวแปรและวิธี จาก superclass ของมัน subclass สามารถใช้สิ่งที่ได้รับช่วงมาจาก superclass เหมือนกับเป็นของตนเอง และสามารถที่จะเปลี่ยนแปลงหรือ override มันได้ ทำให้ยังมีการสืบทอดหลายครั้งยังทำให้ class ยังมีคุณสมบัติพิเศษมากขึ้น

การสร้าง subclass

คุณประกาศว่า class นี้เป็น subclass ของ class อื่นได้ในช่วง class declaration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
class SubclassName extends SuperclassName {
    ...
}
```

ในจาวาแต่ละ class สามารถมี superclass ได้เพียงตัวเดียว (จาวาไม่รองรับ multiple inheritance)

What Member Variable does a Subclass inherit

กฎ : subclass จะรับช่วงต่อ member variable ทั้งหมดที่อยู่ใน superclass ที่สามารถเข้าถึงได้จาก subclass (ยกเว้น member variable ที่ซ่อนโดย subclass)

ข้อตกลงในการรับช่วงต่อ member variable

1. subclass จะรับช่วงต่อ member variable ที่ประกาศให้เป็น public หรือ protected
2. subclass จะรับช่วงต่อ member variable ที่ประกาศโดยไม่มี access specifier ในขณะที่ subclass อยู่ใน package
3. เกี่ยวกับ superclass
4. subclass จะไม่รับช่วงต่อ member variable ของ superclass ถ้า subclass ประกาศใช้ member variable ของมันที่มีชื่อซ้ำกับของ superclass (เนื่องจาก member variable ของ subclass จะซ่อน member variable ใน superclass ไว้)
5. subclass จะไม่รับช่วงต่อ member variable ของ superclass ที่เป็น private

Hiding Member Variable

จากที่ได้กล่าวไปแล้วว่า member variable ใน subclass จะซ่อน member variable ที่มีชื่อเดียวกันใน superclass โดยเราสามารถเข้าถึง member variable ที่ซ่อนอยู่โดยผ่าน superclass ของมัน โดยใช้

```
super.memberVariable
```

super เป็น keyword ในจาวาที่ให้ วิธี สามารถเข้าถึง member variable ที่ซ่อนอยู่ และให้ override method ใน superclass ได้

What Method does a Subclass inherit

กฎ : subclass จะรับช่วงต่อ member method ทั้งหมดที่อยู่ใน superclass ซึ่งสามารถเข้าถึงได้จาก subclass (ยกเว้น วิธี ที่ override โดย subclass)

ข้อตกลงในการรับช่วงต่อ member method

1. subclass จะรับช่วงต่อ member method ที่ประกาศให้เป็น public หรือ protected
2. subclass จะรับช่วงต่อ member method ที่ประกาศโดยไม่มี access specifier ในขณะที่ subclass อยู่ใน package
3. subclass จะไม่รับช่วงต่อ member method ของ superclass ถ้า subclass ประกาศใช้วิธี ที่มีชื่อซ้ำกับของ superclass method ใน subclass จะถูกเรียกว่า override method ใน superclass
4. subclass จะไม่รับช่วงต่อ member method ของ superclass ที่ประกาศเป็น private
5. subclass สามารถ override การ implement หรือเพิ่มการทำงานให้กับวิธี ที่รับช่วงมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Method ที่ subclass ไม่สามารถจะ override ได้

- 1 subclass ไม่สามารถ override method ที่ประกาศเป็นแบบ final ใน superclass
- 2 subclass ไม่สามารถ override method ที่ประกาศเป็นแบบ static ใน superclass (subclass ไม่สามารถ override class method ได้)

วิธี ที่ subclass ต้อง override

subclass จะต้อง override method ที่ประกาศเป็นแบบ abstract ใน superclass หรือตัว subclass จะต้องเป็นแบบ abstract

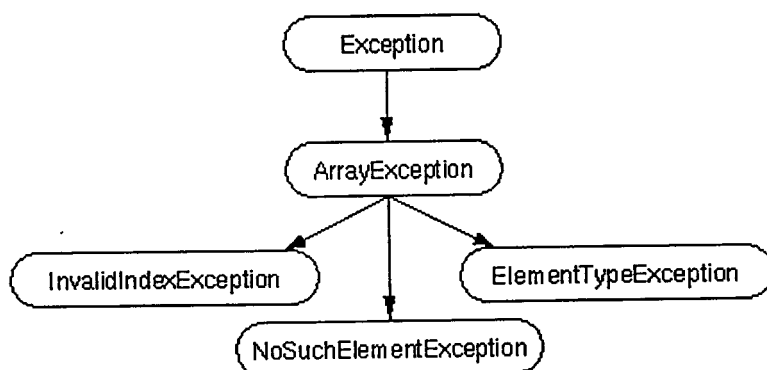
2.4.6 Handling Errors Using Exceptions

นิยาม exception เป็นเหตุการณ์ที่เกิดขึ้นระหว่างการ execute โปรแกรม ซึ่งทำลายการไหลของชุดคำสั่ง

exception object จะเก็บข้อมูลเกี่ยวกับ exception รวมทั้งชนิดและสถานะของโปรแกรมเมื่อเกิดข้อผิดพลาดขึ้น ในจาวาการสร้าง exception object และส่งไปยัง runtime system เรียกว่า throwing an exception

หลังจาก วิธี ได้ส่ง exception แล้ว runtime system จะทำการหาบางอย่างมารองรับ exception ที่เกิดขึ้น ซึ่งบางสิ่งที่ว่าคือกลุ่มของ วิธี ใน call stack ของ วิธี ที่เกิดข้อผิดพลาดขึ้น runtime system จะทำการค้นหาข้อย่อยหลังใน call stack เริ่มจาก วิธี ที่เกิดข้อผิดพลาด จนกระทั่งพบ วิธี ที่เก็บ exception handler ที่เหมาะสม exception handler จะถูกพิจารณาว่าเหมาะสมถ้าชนิดของ exception throw เป็นชนิดเดียวกับชนิดของ exception ที่ exception handler มี โดย exception ที่ถูกเลือกเรียกว่า catch the exception

ถ้า runtime system ทำการหาจนหมด call stack แล้วยังไม่เจอ exception handler ที่เหมาะสม runtime system และตามด้วย โปรแกรมจาวาก็จะสิ้นสุดลง ด้วยการใช้ exception เพื่อจัดการกับข้อผิดพลาดที่เกิดขึ้น โปรแกรมจาวาจะได้ประโยชน์ดังนี้แยกโค้ดที่ใช้จัดการกับข้อผิดพลาดออกจากโค้ดทั่วไป สามารถตรวจสอบข้อผิดพลาดจาก call stack ได้ รวบรวมชนิดของข้อผิดพลาดและแสดงความแตกต่างของข้อผิดพลาด ดังนี้



Java's Catch or Specify Requirement

จาวาต้องการ method catch หรือ specify ในการตรวจ exception ที่อาจถูกส่งมาใน scope ของ วิธี ความต้องการนี้มีองค์ประกอบซึ่งจะอธิบายดังนี้ catch , specify , checked exception และ exception ซึ่ง อาจถูกส่งเข้ามาใน scope ของ วิธี

catch method สามารถ "catch" exception ได้โดยจัดหา exception handler สำหรับ exception ชนิดนั้นๆ specify ถ้า วิธี เลือกที่จะไม่ "catch" exception method จะต้อง specify ว่ามันสามารถ throw exception นั้นได้ เนื่องจาก exception ที่สามารถ throw ได้จะเป็นส่วนหนึ่งของการเขียนโปรแกรมแบบ public interface ของ วิธี นั่นคือตัวที่เรียกใช้ วิธี จะต้องรู้เกี่ยวกับ exception ซึ่ง วิธี สามารถ throw ได้ ดังนั้นใน method signature คุณ specify exception ซึ่ง วิธี สามารถ throw ได้

checked exceptions ในจาวาจะมีชนิดของ exception ต่างกัน ที่เราจะพูดถึงในขณะนี้คือ runtime exception ซึ่งจะเกิดขึ้นในช่วง runtime system runtime exception สามารถเกิดขึ้นที่ใดก็ได้ในโปรแกรม สิ่งที่ต้องดูเกี่ยวกับการตรวจสอบสำหรับ runtime exception มักจะมากกว่า catching และ specifying มัน ดังนั้น compiler จึงไม่ต้องการ catch หรือ specify runtime exception checked exceptions เป็น exception ซึ่งไม่ใช่ runtime exception และถูกตรวจสอบโดย compiler compiler จะตรวจสอบว่า exception เหล่านี้เป็น catch หรือ specify

exception ที่สามารถส่งเข้ามาใน scope ของวิธี ได้ ซึ่งในที่นี้หมายความว่ารวมถึง exception ใดๆที่สามารถ ส่งในขณะที่การควบคุมยังคงดำเนินอยู่ภายใน วิธีรวมถึง exception ที่ถูกส่งโดยตรงจากวิธี ด้วย throw statement ของจาวา และexception ที่ถูกส่งโดยทางอ้อมโดยวิธี โดยการเรียกไปยังวิธี อื่น

Catching and Handling with Exceptions

ในหัวข้อต่อไปนี้จะพูดถึงองค์ประกอบหลัก 3 อย่างของ exception handler คือ try , catch , final block ซึ่งทั้งหมดจะแสดงถึงการเขียน exception handler

Try block อันดับแรกเราควรรู้ถึง statement ที่อาจส่ง exception เข้ามาใน try block ได้ โดย try block มี โครงสร้างดังนี้

```
try {
    Java Statements
}
```

Java Statements ประกอบด้วย statement ของจาวาที่สามารถส่ง exception ได้ try statement ครอบคลุม statement ที่อยู่ในนั้นและกำหนดขอบเขตของ exception handler ใดๆที่เกี่ยวข้องกับมัน หรือพูดอีกอย่างว่าถ้าเกิด exception ขึ้นภายใน try statement exception นั้นจะถูกจัดการโดย exception handler ที่เหมาะสมที่เกี่ยวข้องกับ try statement นั้น try statement จะต้องมาพร้อมกับ catch block หรือ finally block อย่างน้อย 1 ตัว

Catch block คุณเกี่ยวข้องกับ exception handler ด้วย try block โดยจัดหา catch block หนึ่งหรือมากกว่าให้โดยตรงหลัง try block (ไม่มีโค้ดใดระหว่างท้าย try block กับ ต้น catch block)

```
try {
    ...
} catch (...) {
    ...
} catch (...) {
    ...
}
```

รูปแบบทั่วไปของ catch statement มีลักษณะดังนี้

```
catch (SomeThrowableObject variableName) {
    Java Statements
}
```

จะเห็นว่า catch statement ต้องการ argument หนึ่งตัวชนิดของ argument (*SomeThrowableObject*) ประกาศชนิดของ exception ที่ exception handler สามารถจัดการและเป็นชื่อของ class ที่รับช่วงต่อจาก Throwable class (อยู่ใน package java.lang) เมื่อโปรแกรมจาวา throw exception ก็คือการ throw object และ ออปเจ็ค ที่รับช่วงต่อจาก Throwable class เท่านั้นที่สามารถ throw ได้

variableName เป็นชื่อที่ exception handler สามารถใช้อ้างถึง exception ซึ่ง catch โดย handler การเข้าถึง instance variable และวิธีเหมือนกับการเข้าถึง instance variable และ วิธีของออปเจ็ค อื่น

catch block จะเก็บกลุ่มของ statement ไว้ ซึ่ง statement เหล่านี้จะถูก execute เมื่อ exception handler ทำงาน โดย runtime system ให้ exception handler ทำงานเมื่อ handler เป็นตัวแรกใน call stack ซึ่งมีชนิดตรงกับ exception ที่ส่งมา

Finally Block

สุดท้ายเป็นการจัดขั้นตอนในการจัดการล้างสถานะของ วิธี ก่อนที่จะผ่านการควบคุมไปยังส่วนอื่นของโปรแกรม ซึ่งทำได้โดยการใส่โค้ดที่ใช้ทำความสะอาดนี้ลงไป ใน finally block runtime system จะ execute โค้ดใน finally block โดยขึ้นอยู่กับเหตุการณ์ที่เกิดขึ้นใน try block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

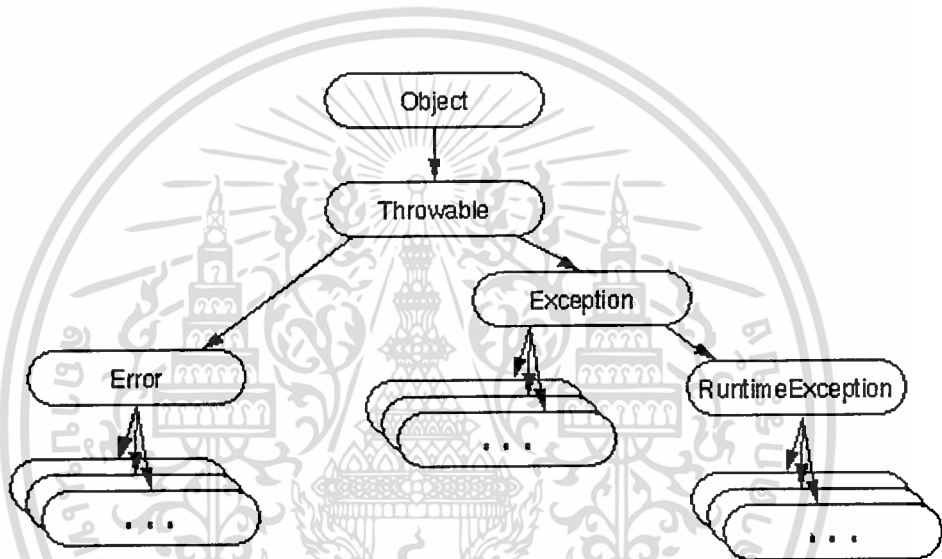
How to throw an Exception

Throw statement วิธี ของจาวาทุกตัวจะใช้ throw ในการส่ง exception throw statement ต้องการ argument ตัวเดียวคือ : *throwable object* ในระบบของจาวา throwable object เป็น instance ของ subclass ใดๆของ Throwable class ตัวอย่างของ throwable statement แสดงดังต่อไปนี้

```
throw someThrowableObject ;
```

throw statement จะอยู่หลังชื่อของ วิธี

Throwable class และ subclass ของมัน รูปต่อไปนี้แสดงถึง class heirarchy ของ Throwable class และ subclass ที่สำคัญ



รูปที่ 2-12 class heirarchy ของ Throwable class

subclass ที่สำคัญของ throwable มี 3 ตัวด้วยกันคือ

1. **Errors** เมื่อเกิดข้อผิดพลาดขึ้นใน virtual machine virtual machine จะส่ง Error ซึ่งโปรแกรมโดยทั่วไปไม่สามารถ catch Error นี้ได้
2. **Exception** โปรแกรมส่วนมากจะ throw และ catch object ที่ได้มาจาก Exception class exception จะบอกว่าเกิดปัญหาขึ้นแต่ไม่ได้เป็นปัญหาของระบบที่สำคัญ subclass ของ Exception ที่สำคัญตัวหนึ่งคือ
3. **RuntimeException** เป็น class ซึ่งจะแสดงถึง exception ที่เกิดขึ้นใน จาวา virtual machine (ในช่วง runtime)

2.4.7 Threads of Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thread คืออะไร

thread มีลักษณะเหมือนกับโปรแกรมทั่วไป thread 1 ตัวจะมีจุดเริ่มต้น จุดสิ้นสุด โปรแกรม (เป็นเพียงคอนหรือช่วงหนึ่ง) และเวลาใดๆขณะที่ thread กำลังทำงานจะมีจุดประสงค์ของการ execute เพียงอย่างเดียว อย่างไรก็ตาม thread โดยตัวมันเองแล้วไม่ใช่โปรแกรม เพราะไม่สามารถทำงานด้วยตัวเองได้ แต่ต้องทำงานกับโปรแกรม

นิยาม thread เป็นการทำงานอย่างใดอย่างหนึ่งในโปรแกรม

การทำงานของ thread ที่จะพูดถึงไม่ใช่การทำงานของ thread ตัวเดียว แต่เป็นการทำงานแบบ multiple threads ภายในโปรแกรมเดียวกัน โดย thread จะทำงานพร้อมๆกันแต่จะทำหน้าที่คนละอย่าง

ในหนังสือบางเล่มจะใช้คำว่า lightweight process แทนคำว่า thread (เนื่องจาก thread มีลักษณะคล้าย process และที่เรียกว่า lightweight เนื่องจาก thread จะทำงานภายใต้สิ่งแวดล้อมเดียวกับโปรแกรม และใช้ resource ที่ถูกจองให้กับโปรแกรม) แต่อย่างไรก็ตาม thread จะต้องใช้ resource บางอย่างที่เป็นของตัวเอง เช่น stack, program counter เป็นต้น โค้ดที่อยู่ใน thread จะต้องทำงานภายใต้สิ่งแวดล้อมที่กล่าวมาเท่านั้น ฉะนั้นเราอาจเรียก thread อีกอย่างว่า execution context

Thread Attributes

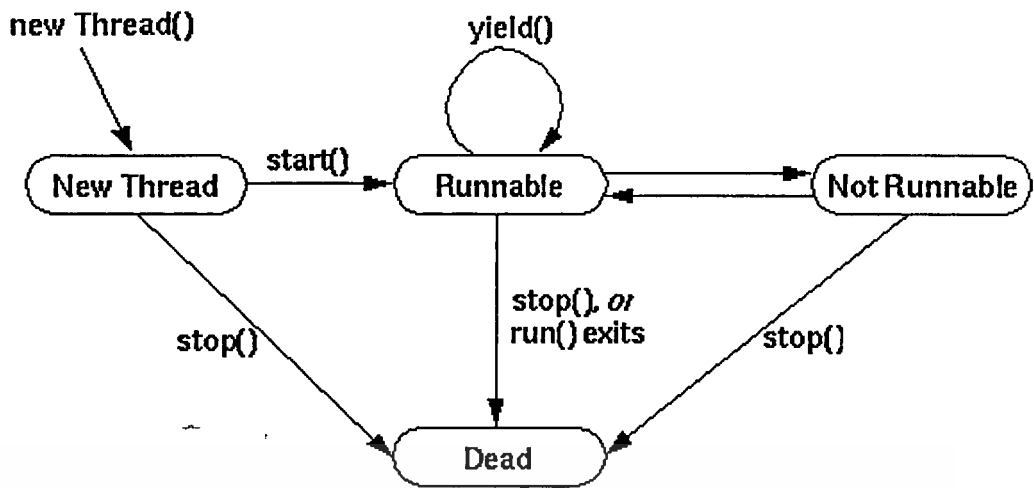
thread ของจาวาถูก implement โดย Thread class ซึ่งอยู่ใน package java.lang thread class จะ implement ข้อกำหนดที่เป็นอิสระจากระบบของ thread

Thread Body การทำงานทุกอย่างเกิดขึ้นใน thread body (ใน run method) หลังจากที่ thread ถูกสร้างและ initial แล้ว runtime system จะเรียก run method ของ thread นั้น โค้ดใน run method จะ implement การทำงานของ thread ที่สร้างขึ้น

บ่อยครั้งที่การทำงานของ thread จะเป็นการทำงานแบบ loop ทำให้บางครั้งการทำงานใช้เวลานาน คุณสามารถเลือกหนึ่งในสองวิธีต่อไปนี้ เพื่อลดขนาดของ run method สำหรับ thread ของจาวา

1. สร้าง subclass ของ Thread class ใน java.lang และ override run method
2. สร้างซึ่ง implement Runnable interface ซึ่งอยู่ใน package java.lang เช่นกัน ดังนั้นถ้าคุณสร้าง instance ของ thread (ทั้งทางตรงจาก Thread class และทางอ้อมจาก subclass ของ Thread) ให้สร้าง handle ของ instance ของ Runnable class ของคุณให้กับ thread ที่สร้างใหม่ runnable object นี้จะเตรียม run method ให้กับ thread

Thread State รูปข้างล่างนี้แสดงสถานะของ thread ในจาวา



รูปที่ 2-13 สถานะของ Thread ในจาวา

New Thread : statement ต่อไปนี้ใช้ในการสร้าง thread ใหม่ แต่ยังไม่มีการทำงานเกิดขึ้น

```
Thread myThread = new myThreadclass ();
```

เมื่อเป็น thread ใหม่ มันจะเป็น thread object ที่ว่างเปล่า ยังไม่มีการจอง resource ของระบบให้ เมื่อ thread อยู่ในสถานะ new thread มันสามารถจะเริ่มหรือหยุด thread ได้เท่านั้น

Runnable : พิจารณาโค้ดต่อไปนี้

```
Thread myThread = new myThreadclass ();
```

```
myThread . start ();
```

start method จะทำการจอง resource ที่จำเป็นในการทำงานของ thread ให้ จัดตารางการทำงานของ thread และเรียก run method ของ thread เมื่อถึงจุดนี้ thread จะอยู่ในสถานะ "Runnable" ที่ไม่เรียก running เนื่องจากยังไม่มีการทำงานเกิดขึ้น เนื่องจากคอมพิวเตอร์ส่วนใหญ่มี processor ตัวเดียว ทำให้ไม่สามารถ runnable thread ทุกตัวทำงานพร้อมกันได้ runtime system ของจาวาจะต้อง implement หลักการจัดการ ซึ่งจะแบ่งการใช้งาน processor ให้กับ runnable thread ทุกตัว

Not Runnable : thread จะมาอยู่ที่สถานะ not runnable เมื่อเหตุการณ์ใดเหตุการณ์หนึ่งใน 4 เหตุการณ์นี้เกิดขึ้น

1. เมื่อมีการอ้าง sleep method ของมัน
2. เมื่อมีการอ้าง suspend method ของมัน
3. thread ใช้ wait method ของมันเอง เพื่อรอสถานะของ variable
4. thread ติดอยู่กับ I/O

การออกจากสถานะ not runnable (ไม่ว่าจะเข้ามาทางไหน) เกิดขึ้นเมื่อ

1. ถ้า thread เข้าสู่สถานะ sleep จะออกได้เมื่อเวลาที่กำหนดไว้หมดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ถ้า thread เข้าสู่สถานะ suspend จะออกได้เมื่อมีการเรียก resume method
3. ถ้า thread เข้าสู่สถานะ wait เพื่อรอสถานะ variable จะออกได้เมื่อ อยปเจ็ท ที่เป็นเจ้าของ variable ปลออยมันโดยการเรียก notify หรือ notifyAll
4. ถ้า thread ติดอยู่กับ I/O จะออกได้เมื่อ I/O ทำงานเสร็จ

Dead : thread สามารถสิ้นสุดได้ 2 วิธี คือ สิ้นสุดด้วยตัวเอง (ออกจาก run method ตามปรกติ) หรือถูกทำให้สิ้นสุด(หยุด) ซึ่ง thread สามารถถูกทำให้หยุดได้ทุกเวลาโดยการเรียก stop method stop method จะส่ง ThreadDeath object thread จะสิ้นสุดเมื่อมันได้รับ ThreadDeath exception

Thread Priority การจะให้ multithread ทำงานบน CPU ตัวเดียวนั้น จะต้องมีการทำ scheduling โดย runtime system ของจาวาจะมีขั้นตอนในการทำ scheduling เรียกว่า fixed priority scheduling ซึ่งขั้นตอนเหล่านี้จะจัด thread ตาม priority (ความสำคัญ) ซึ่งจะสัมพันธ์กับ thread อื่น

thread แต่ละตัวในจาวาจะถูกกำหนดค่าความสำคัญให้ซึ่งจะอยู่ระหว่าง MIN_PRIORITY และ MAX_PRIORITY (เป็นค่าคงที่ซึ่งกำหนดใน Thread class) ณ.เวลาใดๆเมื่อ multithread พร้อมที่จะทำงาน thread ที่มีความสำคัญมากที่สุดจะถูก execute ก่อน เมื่อ thread ที่ทำงานอยู่สิ้นสุดหรือ พักการทำงานเท่านั้น thread ที่มีความสำคัญน้อยกว่าถึงจะสามารถทำงานได้

scheduling ของ CPU จะมีสิทธิในการเลือก thread เดิมที ถ้า thread ที่มีความสำคัญมากกว่า thread ที่ทำงานอยู่ในขณะนั้น thread ที่มีความสำคัญมากกว่าจะถูกทำ schedule ในทันที

runtime system จะไม่ใช่สิทธิในการเรียกการทำงานคือเพื่อให้ thread อื่นที่มีความสำคัญเท่ากันทำงานในขณะใดๆ thread อาจยกเลิกสิทธิในการทำงานโดยการเรียก yield method thread สามารถมอบ CPU ให้กับ thread อื่นที่มีความสำคัญเท่ากัน (ไม่สามารถให้กับ thread ที่มีความสำคัญต่ำกว่าได้)

เมื่อ runnable thread ในระบบทุกๆตัวมีความสำคัญเท่ากัน runtime system จะทำ scheduling โดยการเลือก thread ที่อยู่ถัดไป

thread ที่มี while loop อยู่ภายใน run method เมื่อทำงานมันจะไม่ปล่อยการควบคุม CPU ให้กับ thread อื่นจนกว่ามันจะสิ้นสุดด้วยตัวมันเอง (while loop สิ้นสุด) หรือจนกระทั่งถูกอ้างสิทธิ์จาก thread ที่มีความสำคัญมากกว่า thread เช่นนี้เรียกว่า selfish thread ในบางระบบหาวิธีแก้ selfish thread ด้วยการทำ time-slicing time-slicing จะมีบทบาทเมื่อมี runnable thread หลายตัวที่มีความสำคัญเท่ากัน และ thread เหล่านั้นซึ่งเป็น thread ที่มีความสำคัญสูงที่สุดแข่งกันกันเพื่อแย่ง CPU โดยระบบ time-sliced จะแบ่ง CPU ออกเป็นเวลาออกเป็นช่วงๆและจะแบ่งให้กับ thread ทุกๆตัวที่มีความสำคัญเท่ากันและมีความสำคัญสูงสุดเป็นรอบๆ ในแต่ละรอบจะให้ช่วงเวลา thread ละ 1 ช่วงเพื่อให้ทำงานของมัน จะสังเกตได้ว่า time-sliced จะไม่รับประกันความถี่หรือลำดับการทำ schedule

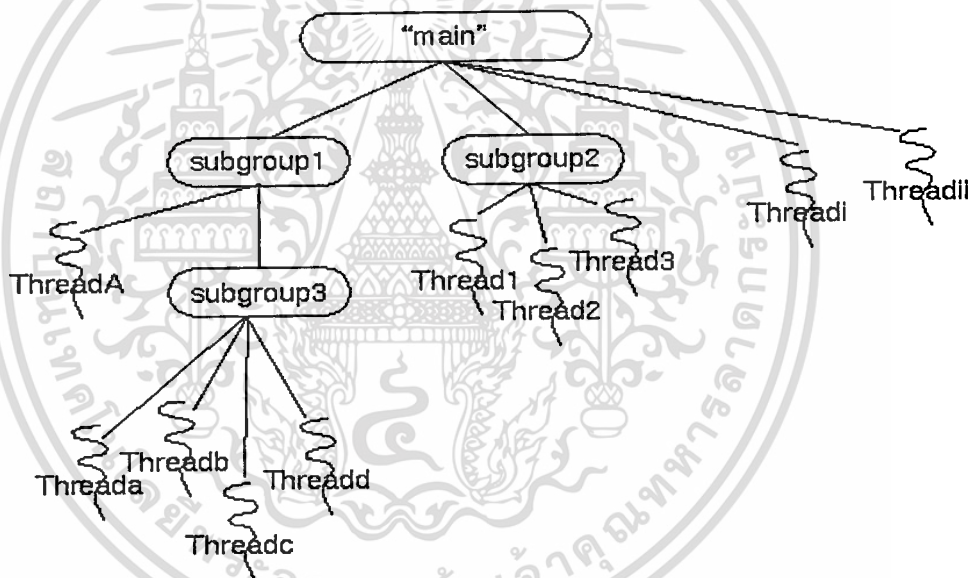
Daemon Thread thread ในจาวาสามารถเป็น daemon thread ได้ daemon thread เป็น thread ที่ให้บริการ thread อื่นที่กำลังทำงานอยู่ใน process เดียวกัน run method ของ daemon thread ทำซ้ำไม่มีที่สิ้นสุดเพื่อรอการขอบริการจาก thread อื่น เมื่อ thread เดียวที่เหลือใน process เป็น daemon thread intprepter จะออกจากการทำงาน (เนื่องจากไม่มี thread อื่นที่ต้องบริการ)

ในการกำหนดค่าให้ thread เป็น daemon thread ทำได้โดยการเรียก setDaemon method พร้อม argument "true" ส่วนการตรวจสอบว่า thread ใดเป็น daemon thread ทำได้โดยใช้ isDaemon method

Thread Group ทุก thread ในจาวาเป็นสมาชิกของ thread group thread group เป็นเครื่องมือในการรวม multithread เข้าด้วยกันเป็น ออปเจ็ค เดียวและใช้ thread เหล่านั้นทั้งหมดพร้อมกัน thread group ถูก imlement ใน ThreadGroup class ใน package java.lang

runtime system จะทำรวม thread เข้า thread group ระหว่างที่ thread ถูกสร้าง เมื่อคุณสร้าง thread ใหม่ คุณสามารถให้ runtime system รวม thread ของคุณเข้ากับ thread group ใดๆที่มีอยู่ หรือจะตั้ง thread group ใหม่ก็ได้ เมื่อ thread เป็นสมาชิกของ thread group ใดแล้วไม่สามารถจะย้าย thread group ได้

เมื่อโปรแกรมจาวาเริ่มทำงาน runtime system จะสร้าง thread group ชื่อว่า main ถ้าไม่ได้เจาะจงไว้ thread จะถูกรวมเข้าไว้ใน main thread group (แต่ถ้าคุณสร้าง thread ใน แอปเพล็ต thread group ใหม่อาจใช้ชื่ออื่นที่ไม่ใช่ main)



รูปที่ 2-14 Thread Group class

Multithreaded Programs

Synchronizing Threads บ่อยครั้งที่ thread ต้องการใช้ข้อมูลร่วมกันและต้องพิจารณาถึงสถานะและกิจกรรมที่ thread อื่นทำอยู่ โดยกลุ่มของโปรแกรมที่อยู่ในสถานะการณ์นี้เรียกว่า producer / consumer โดย producer จะสร้างข้อมูล ซึ่งจะถูกร consumer นำไปใช้ ซึ่งทำให้การทำงานของทั้ง producer และ consumer เข้าจ้งหวะกัน (synchronize) ปัญหาของการทำงานไม่เข้าจ้งหวะกันเรียกว่า race conditions เกิดขึ้นเมื่อ multithread ที่ทำงานไม่เข้าจ้งหวะกันเข้าถึง ออปเจ็ค ตัวเดียวกันในเวลาเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กัน ทำให้ได้ผลลัพธ์ที่ผิดพลาด เพื่อป้องกัน race conditions ไม่ให้เกิดขึ้นสำหรับการทำงานแบบ producer / consumer เรามีวิธีแก้ 2 วิธีคือ monitors และ ใช้ notifyAll กับ wait method

1.Monitors : ออปเจ็ค ที่ถูกใช้ร่วมกันระหว่าง thread สองตัวซึ่งการเข้าถึงจะต้องเข้าจังหวะกันเรียกว่า condition variable monitors จะเกี่ยวกับการกำหนดข้อมูล (condition variable) และการทำงานเช่น lock ในข้อมูลเหล่านั้น เมื่อ thread ทำ monitor บนข้อมูลตัวใด แล้วมี thread อื่นมาทำการ lock บนข้อมูลตัว จะทำให้ thread ที่ทำ monitor บนข้อมูลนั้น (thread ที่ไม่ได้เป็นผู้ lock) ไม่สามารถใช้ข้อมูลตัวนั้นได้ ส่วนของโค้ดที่อยู่ในโปรแกรมที่เข้าถึงข้อมูลตัวเดียวกันใน thread เรียกว่า critical sections ในจาวาคุณสามารถกำหนด critical sections ในโปรแกรมของคุณได้โดยใช้ synchronized keyword

2.notifyAll และ wait methods : วิธี ทั้ง 2 ตัวนี้เป็นสมาชิกของ java.lang.Object class 3.Note : notifyAll และ wait method ถูกอ้าง โดย thread ที่ถือ lock เท่านั้น

4.notifyAll method : ใช้ในการบอกให้กับ thread ตัวอื่นๆที่รอ บน monitor โดย thread ปัจจุบันให้ตื่นขึ้น หนึ่งใน thread ที่รออยู่ก็จะทำ monitor และทำงานต่อ

5.wait method : จะทำให้ thread ปัจจุบันหยุดการทำงานจนกว่า thread อื่นจะบอกมันถึงสถานะที่เปลี่ยนไป

2.5 ข้อแตกต่างระหว่าง Application กับ แอปเพล็ต

ในการเขียนโปรแกรมด้วยภาษาจาวา สามารถเขียนได้ 2 แบบคือ เขียนเป็น Application หรือเขียนเป็น แอปเพล็ต ซึ่งทั้งคู่ก็เป็น โปรแกรมภาษาจาวาด้วยกันสามารถทำงานบนเครือข่ายได้เหมือนกัน แต่ข้อแตกต่างหลักก็คือ แอปเพล็ต ต้องทำการโหลดมาจากเครื่อง Server และดำเนินการบน โปรแกรมบราวเซอร์ (Browser) ที่สนับสนุนการทำงานของจาวา ข้อแตกต่างอื่นๆได้แสดง ไว้ดังตารางที่ 2-12

Application	Applet
สามารถรันบนเครื่อง โคลน์เอ็นท์, server หรือ ทั้งคู่ได้	สามารถรันบนเครื่อง โคลน์เอ็นท์ ได้อย่างเดียว
ขึ้นอยู่กับ Java runtime environment	ต้องอาศัย Java-enable Web browser
สามารถ access ไฟล์บนเครื่อง โคลน์เอ็นท์ ได้	ไม่สามารถ access ไฟล์บนเครื่อง โคลน์เอ็นท์ ได้ถ้าถูก download มาจาก Web server
ซอฟต์แวร์ ต้องถูก install บนเครื่อง โคลน์เอ็นท์	ซอฟต์แวร์ ถูก download ขณะ runtime มายัง Web browser ซึ่งวิ่งอยู่บนเครื่อง โคลน์เอ็นท์

ตารางที่ 2-12 ข้อแตกต่างระหว่าง Application กับ Applet

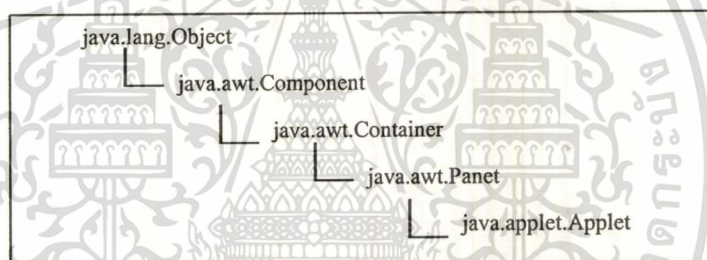
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การเขียน แอปเพล็ต

แอปเพล็ต เป็นโปรแกรมที่ได้รับอนุญาตให้ทำงานได้ในโปรแกรมบราวเซอร์ ที่รองรับจาวา โดยมีขั้นตอนในการสร้าง แอปเพล็ต อย่างคร่าวๆดังนี้

1. สร้างโค้ดของจาวา แอปเพล็ต
 - 1.1 นำเอา class และ package เข้ามา
 - 1.2 กำหนด Applet subclass
 - 1.3 ทำการสร้าง วิธีs ให้กับ แอปเพล็ต
2. ทำการ compile โค้ดจาวาที่เขียนเสร็จแล้ว
3. สร้างไฟล์ HTML ที่รวมเอา แอปเพล็ต ที่สร้างขึ้นไว้ข้างใน
4. นำแอปเพล็ต ไปใช้งาน โดยการเรียกใช้ไฟล์ HTML

แอปเพล็ต ทุกตัวจะถูก implement โดยการสร้าง subclass ของ Applet class รูปต่อไปนี้จะแสดงถึง heirarchy ของ Applet class



รูปที่ 2-15 heirarchy ของ Applet class

2.6.1 วงจรชีวิตของ แอปเพล็ต

1. Loading and applet เมื่อเราทำการ load แอปเพล็ต จะเกิดเหตุการณ์ต่อไปนี้
 - subclass ของ Applet class จะถูกสร้างขึ้น
 - แอปเพล็ต จะ initial ตัวเอง
 - แอปเพล็ต จะเริ่มทำงาน
2. เข้าและออกจากหน้าของ แอปเพล็ต เมื่อเราออกจากหน้าที่ แอปเพล็ต ทำงานอยู่ไปยังหน้าอื่นแอปเพล็ต จะหยุดทำงาน และเมื่อเรากลับมายังหน้าของ แอปเพล็ต อีกครั้ง แอปเพล็ต ก็จะเริ่มทำงานใหม่
3. ทำการ load แอปเพล็ต ใหม่ โปรแกรมบราวเซอร์บางตัวให้ผู้ใช้สามารถ reload แอปเพล็ต ได้ โดยจะ unload แอปเพล็ต แล้วค่อยทำการ load ใหม่ ก่อนที่ แอปเพล็ต จะ unload มันจะหยุดตัวเองก่อน แล้วทำการ final cleanup ซึ่ง แอปเพล็ต จะทำการปล่อย resource ที่มันถืออยู่ทั้งหมด จากนั้นจึงทำการ load แอปเพล็ต อีกครั้ง
4. ออกจากโปรแกรมบราวเซอร์ เมื่อผู้ใช้ออกจากโปรแกรมบราวเซอร์ แอปเพล็ต จะหยุดการทำงาน และทำการ final cleanup ก่อนที่จะออกจากโปรแกรมบราวเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 วิธีที่ใช้สำหรับตรวจสอบ

```
public class Simple extends Applet {
    . . .
    public void init() { . . . }
    public void start() { . . . }
    public void stop () { . . . }
    public void destroy () { . . . }
    . . .
}
```

ใน แอปเพล็ต ได้มีการ override method อยู่ 4 ตัวคือ ซึ่งมันจะตอบสนองต่อเหตุการณ์ต่างกัน

init : ใช้ในการ initial applet ในแต่ละครั้งที่มันถูก load (หรือ reload)

start : จะถูกใช้เมื่อแอปเพล็ต ถูก load แล้ว หรือเมื่อผู้ใช้กลับมาเข้ามายังหน้าที่มี แอปเพล็ต อยู่อีกครั้ง

stop : จะถูกใช้เมื่อผู้ใช้ออกจากหน้าที่มี แอปเพล็ต อยู่ หรือออกจากโปรแกรมบราวเซอร์

destroy : ใช้เพื่อทำการ final cleanup สำหรับ unload

ในการเขียนแอปเพล็ต ไม่จำเป็นต้อง override method ที่กล่าวมาก็ได้ init method จะมีประโยชน์สำหรับการ initial ครั้งแรกที่ไม่ต้องใช้เวลาาน โดยทั่วไป init method จะเก็บโค้ดที่คุณจะได้เข้าไปใน constructor เพราะแอปเพล็ต ที่ไม่มี constructor ในแอปเพล็ต จะไม่รับประกันว่าจะมีสภาพแวดล้อมที่พร้อมจนกว่า init method จะถูกเรียก

แอปเพล็ต ทุกตัวที่ทำงานบางอย่างหลังจาก initial แล้ว (ยกเว้นรอรับการตอบสนองจากผู้ใช้) จะต้อง override start method start method จะทำงานของแอปเพล็ต หรือเริ่มการทำงานของ thread

แอปเพล็ตที่ override start method มักจะ override stop method ด้วย stop method จะรวมอยู่ในการทำงานของแอปเพล็ต ด้วย มันจะไม่ใช้ resource ของแอปเพล็ต ถ้าผู้ใช้ไม่ได้ดูหน้าที่มีแอปเพล็ต อยู่

ส่วน destroy method ไม่จำเป็นจะต้อง override ถ้า stop method ทำสิ่งที่จำเป็นในการหยุดการทำงานของแอปเพล็ต แล้ว อย่างไรก็ตาม destroy method ก็ยังจำเป็นสำหรับแอปเพล็ต ที่ต้องการปล่อย resource เพิ่มเติม

2.6.3 Method for Drawing and Event Handling

Simple applet ในตัวอย่างข้างล่างนี้เป็นการ override paint method :

```
class Simple extends Applet {
    . . .
    public void Paint(Graphics g) { . . . }
    . . .
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

paint method เป็นหนึ่งในสองวิธีที่ใช้ในการแสดงผล ซึ่งแอปพลิเคชัน สามารถ override ได้

paint : เป็นวิธีที่ใช้แสดงผลพื้นฐานแอปพลิเคชัน หลายตัว implement paint method เพื่อใช้เพื่อใช้แสดงผลของ แอปพลิเคชัน ในหน้าของโปรแกรมบราวเซอร์

update : เป็นวิธีที่คุณสามารถใช้พร้อมกับ paint method เพื่อปรับปรุงการแสดงผล

แอปพลิเคชัน จะรับช่วงต่อ paint และ update method มาจากแอปพลิเคชัน class ซึ่งรับช่วงต่อจาก Abstract Windows Toolkit (AWT) component class อื่นๆ

แอปพลิเคชัน จะรับช่วงต่อกลุ่มของ event-handling methods มาจาก Component class ซึ่งจะมี วิธี ที่ใช้รองรับเหตุการณ์ต่างๆที่จะเกิดขึ้นเช่น mouseDown , mouseUp , action เป็นต้น หรือ handleEvent method ซึ่งจะใช้ในการรองรับเหตุการณ์ที่เกิดขึ้นทั้งหมดด้วย วิธี เดียว ในการตอบสนองต่อเหตุการณ์ที่เกิดขึ้น แอปพลิเคชัน จะต้อง override method ที่ใช้รองรับเหตุการณ์ที่เหมาะสม หรือ handleEvent method

2.6.4 วิธี ที่ใช้ในการเพิ่ม UI Component

เป็นการช่วยเพิ่มส่วนแสดงผลที่มีประสิทธิภาพมากขึ้นให้กับแอปพลิเคชัน

UI Components ที่มีมาให้

AWT จะจัดหา UI Components ต่อไปนี้ไว้ให้ (class ซึ่งสร้างส่วนประกอบแต่ละตัวจะแสดงไว้ในวงเล็บ)

- Buttons (java.awt.Button)
- Checkboxes (java.awt.Checkbox)
- Single-line text fields (java.awt.TextField)
- Larger text display and editing areas (java.awt.TextArea)
- Labels (java.awt.Label)
- Lists (java.awt.List)
- Pop-up list of choices (java.awt.Choice)
- Sliders and Scrollbars (java.awt.Scrollbar)
- Drawing areas (java.awt.Canvas)
- Menus (java.awt.Menu , java.awt.MenuItem , java.awt.CheckboxMenuItem)
- Containers (java.awt.Panel , java.awt.Window และ subclass ของมัน)

Mehod สำหรับใช้ UI Components ใน แอปพลิเคชัน

เนื่องจาก Applet class รับช่วงต่อจาก AWT Container class ทำให้ง่ายในการเพิ่มส่วนประกอบลงไป ใน แอปพลิเคชัน และใช้ layout manager ในการจัดการตำแหน่งของ ส่วนประกอบ บนจอภาพ ต่อไปนี้เป็น วิธี ของ Container ที่ แอปพลิเคชัน สามารถใช้ได้

add : ใช้เพิ่ม ส่วนประกอบ ที่กำหนดไว้

remove : ใช้ย้าย ส่วนประกอบ ที่กำหนดไว้
 setLayout : ใช้กำหนด layout manager

2.6.5 ข้อจำกัดและความสามารถของ แอปเพล็ต

ข้อจำกัดทางด้านความปลอดภัย

โปรแกรมบราวเซอร์ทุกตัวจะมีการ implement ถึงความปลอดภัย เพื่อให้ความปลอดภัยกับ แอปเพล็ต อย่างไรก็ตามศักยภาพความปลอดภัยของโปรแกรมบราวเซอร์แต่ละตัวจะแตกต่างกันไป โปรแกรมบราวเซอร์ในปัจจุบันได้มีการกำหนดข้อจำกัดสำหรับ แอปเพล็ต ใดๆที่ load ผ่านเครือข่ายมา

- แอปเพล็ต ไม่สามารถ load library หรือ กำหนด native method ได้
- แอปเพล็ต ไม่สามารถอ่านหรือเขียนไฟล์บน host ที่กำลัง execute มันได้
- แอปเพล็ต ไม่สามารถสร้างการเชื่อมต่อผ่านเครือข่ายยกเว้น ไปยัง host ที่มันมา
- แอปเพล็ต ไม่สามารถเรียกใช้งานโปรแกรมอื่นๆบน host ที่กำลัง execute มันได้
- แอปเพล็ต ไม่สามารถอ่านคุณสมบัติของระบบได้
- window ที่แอปเพล็ตสร้างจะต่างกับ window ที่โปรแกรมสร้างขึ้น

โปรแกรมบราวเซอร์แต่ละตัวจะมี SecurityManager object ซึ่ง implement อยู่ในศักยภาพความปลอดภัยเมื่อ SecurityManager object จับได้ว่าการละเมิดข้อบังคับ มันจะส่งต่อไปยัง SecurityException แอปเพล็ต ของคุณสามารถตรวจจับ SecurityException นี้ได้และจะตอบสนองไปอย่างเหมาะสม

ความสามารถของ แอปเพล็ต

java.applet package จัดหา API ซึ่งจะเพิ่มความสามารถของ แอปเพล็ต ที่โปรแกรมสำเร็จรูปทั่วไปไม่มี เช่น แอปเพล็ต สามารถส่งเสียงได้ โปรแกรมบราวเซอร์และ แอปเพล็ต viewer อื่นๆในปัจจุบันจะอนุญาตให้ แอปเพล็ต ทำสิ่งต่างๆต่อไปนี้ได้

- แอปเพล็ต สามารถสร้างการติดต่อข้างเครือข่ายไปยัง host ที่มันมาได้
- แอปเพล็ต สามารถอ้างถึง public method ของ แอปเพล็ต อื่นๆในหน้าเดียวกันได้
- แอปเพล็ต ที่กำลังทำงานอยู่ในโปรแกรมบราวเซอร์ สามารถแสดงผลได้ง่ายโดย HTML ไฟล์

-แอปเพล็ต ที่ load จาก local file system (จากไดเรกทอรีใน classPATH ของผู้ใช้) จะไม่มีข้อจำกัดเหมือนกับ แอปเพล็ต ที่ถูก load ข้ามเครือข่ายมา

2.6.6 การเพิ่ม แอปเพล็ต เข้าไปในหน้าของ HTML

การทำให้ แอปเพล็ต ทำงานได้ในโปรแกรมบราวเซอร์หรือใน JDL แอปเพล็ต viewer แอปเพล็ต จำเป็นจะต้องรวมเข้ากับหน้าของ HTML โดยใช้ tag <APPLET> จากนั้นก็กำหนด URL ของหน้าของ HTML ให้กับโปรแกรมบราวเซอร์ของคุณหรือ แอปเพล็ต viewer

<APPLET> Tag

ข้างล่างนี้จะแสดงถึงการกำหนดลักษณะที่ แอปเพล็ต จะถูกแสดงออกทางหน้าของ HTML โดยใช้ <APPLET> tag ตัวหนาเป็นคำที่คุณจะต้องพิมพ์ลงไป (ไม่จำเป็นต้องเป็นตัวพิมพ์ใหญ่). ส่วนตัวเอียง เป็นสิ่งที่คุณจะต้องเอาคำมาใส่เอง เครื่องหมาย “[]” บอกว่าคุณจะพิมพ์ลงไปหรือไม่ก็ได้ ส่วนที่ไม่มีเครื่องหมาย “[]” หมายความว่า คุณจำเป็นต้องพิมพ์ลงไปด้วย

```
< APPLET
    [CODEBASE = codebaseURL]
    CODE = appletFile
    [ALT = alternateText]
    [NAME = appletInstanceName]
    WIDTH = pixels
    HEIGHT = pixels
    [ALIGN = alignment]
    [VSPACE = pixels]
    [HSPACE = pixels]
>
[< PARAM NAME = appletParameter1 VALUE = value >]
[< PARAM NAME = appletParameter2 VALUE = value >]
...
[alternateHTML]
</APPLET>
```

CODEBASE = codebaseURL

กำหนด URL พื้นฐานของ แอปเพล็ต (เป็นไคเรกทอรีที่เก็บโค้ดของ แอปเพล็ต ไว้) ถ้าไม่กำหนดไว้จะใช้ไคเรกทอรีที่ HTML ไฟล์อยู่

CODE = appletFile

เป็นคุณสมบัติที่จำเป็นซึ่งจะเก็บชื่อไฟล์ของ แอปเพล็ต ที่ compile แล้ว ไฟล์นี้จะต้องสัมพันธ์กับ URL พื้นฐานของ แอปเพล็ต

ALT = alternateText

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะกำหนดข้อความที่จะถูกแสดงเมื่อโปรแกรมบราวเซอร์เข้าใจ `<APPLET>` tag แต่ไม่สามารถให้ แอปเพล็ต ทำงานได้

`NAME = appletInstanceName`

จะกำหนดชื่อของ applet instance เพื่อให้ แอปเพล็ต อื่นในหน้าเดียวกันสามารถหาหรือติดต่อกันได้

`WIDTH = pixels`

`HEIGHT = pixels`

เป็นคุณสมบัติที่จำเป็นต้องกำหนดเพื่อกำหนดขนาดที่จะใช้ในการแสดง แอปเพล็ต

`ALIGN = alignment`

ใช้ในการกำหนดตำแหน่งของ แอปเพล็ต โดยค่าที่เป็นไปได้คือ left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom

`VSPACE = pixels`

`HSPACE = pixels`

กำหนดจำนวนจุด (pixel) ทั้งบนและล่าง (VSPACE) ของ แอปเพล็ต และทั้งซ้ายและขวา (HSPACE) ของ แอปเพล็ต

`<PARAM NAME = appletParameter1 VALUE = value >`

`<PARAM>` tag เป็นวิธีเดียวในการกำหนด พารามิเตอร์ ของ แอปเพล็ต แอปเพล็ต จะอ่านค่าพารามิเตอร์ ที่ผู้ใช้กำหนดโดยใช้ `getParameter()` method

altetnateHTML

ถ้าหน้า HTML มี `<APPLET>` tag ซึ่งแสดงโดยโปรแกรมบราวเซอร์ที่ไม่เข้าใจ `<APPLET>` tag นี้ โปรแกรมบราวเซอร์จะไม่สนใจ `<APPLET>` และ `<PARAM>` tag และจะแสดงข้อความที่อยู่ระหว่าง `<APPLET>` และ `</APPELT>` tag แทน

2.6.7 การสิ้นสุดการทำ แอปเพล็ต

ก่อนนำ แอปเพล็ต ออกใช้งานจริง

ก่อนที่จะคุณจะทำ แอปเพล็ต ที่คุณสร้างขึ้นไปใช้งานจริง คุณควรแน่ใจก่อนว่าคุณสามารถทำคำถามเหล่านี้ให้เป็นจริงได้

1. คุณได้นำเอาจุดบกพร่องเกี่ยวกับการแสดงผลออกไปหมดหรือยัง ?

ข้อผิดพลาดเกี่ยวกับการแสดงผลเช่น การใช้ `System.out.println` ซึ่งสร้างสะดวกให้กับคุณแต่อาจสร้างความสับสนหรือรบกวนกับผู้ใช้ ถ้าคุณต้องการแสดงข้อความให้กับผู้ใช้ทราบ พยายามแสดงอยู่ภายในส่วนแสดงผลของแอปเพล็ตหรือในแถบแสดงสถานะบริเวณด้านล่างของหน้าต่างโดยใช้ `showStatus` method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แอปพลิเคชันของคุณหยุดการทำงานเมื่อออกนอกจอภาพหรือไม่ ?

แอปพลิเคชันไม่ควรใช้ resource เมื่อโปรแกรมบราวเซอร์แสดงหน้าต่างที่ไม่ได้มี แอปพลิเคชัน นั้นบรรจุอยู่ ถ้าโค้ด แอปพลิเคชัน ของคุณมี thread อยู่ คุณควรจะ implement stop method (ซึ่งจะหยุดการทำงาน ของ thread โดยการกำหนดให้เป็น null)

3. ถ้าแอปพลิเคชันของคุณทำสิ่งที่อาจเป็นการรบกวนผู้ใช้ คุณให้ผู้ใช้สามารถหยุดสิ่งรบกวนเหล่านั้นได้หรือไม่?

คุณควรให้ทางเลือกกับผู้ใช้ในการหยุดการทำงานสิ่งที่อาจรบกวนผู้ใช้ได้ โดยผู้ใช้ไม่จำเป็นต้องออกจากหน้านั้น

การจบ แอปพลิเคชัน ที่สมบูรณ์

สิ่งที่จะพูดถึงต่อไปนี้เป็นข้อเสนอแนะในการทำให้ แอปพลิเคชัน ของคุณน่าสนใจมากยิ่งขึ้น

1. พยายามให้ แอปพลิเคชัน ของคุณมีความยืดหยุ่นมากที่สุด

โดยการกำหนด พารามิเตอร์ เพื่อให้ แอปพลิเคชัน ของคุณสามารถใช้ได้ทุกสถานการณ์ โดยไม่ต้องทำการเขียนใหม่

2. ทำการ implement getParameterInfo method

การ implement method นี้จะทำให้ แอปพลิเคชัน ของคุณสามารถพัฒนาได้ง่ายในอนาคต ในปัจจุบันนี้ไม่มีโปรแกรมบราวเซอร์ตัวใดใช้วิธีนี้ ถึงอย่างไรก็ตามในไม่ช้านี้อาจมีการใช้เพิ่มขึ้น เพื่อช่วยสร้าง GUI ซึ่งอนุญาตให้ผู้ใช้สามารถติดต่อกับ พารามิเตอร์ เพื่อกำหนดค่าให้

3. ทำการ implement getMethodInfo method

วิธี นี้จะส่งข้อมูลสั้นๆเกี่ยวกับ แอปพลิเคชัน มาให้ ซึ่งปัจจุบันนี้ก็ยังไม่มีโปรแกรมบราวเซอร์ตัวใดใช้วิธี นี้

บทที่ 3

การพัฒนาจาวาด้วย Java Bean

3.1 บีนคืออะไร

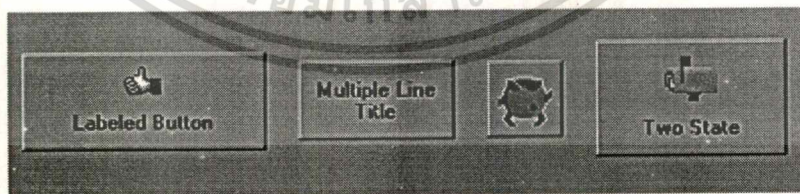
จาวาบีนเป็นส่วนประกอบของซอฟต์แวร์ที่สามารถนำกลับมาใช้งานใหม่ และทำงานร่วมกับกับภาษาจาวา แต่ที่พิเศษก็คือจาวาบีนเป็นส่วนประกอบของซอฟต์แวร์ ที่สามารถเห็นเป็นภาพ (Visual Manipulation) ได้ในเครื่องมือที่ใช้สร้าง(Builder tool) เป้าหมายของจาวาบีนคือการกำหนด ”แบบจำลองส่วนประกอบของซอฟต์แวร์” (Software Component Model) ซึ่งส่งผลให้ผู้ผลิตซอฟต์แวร์อิสระ (Independent Software Vendors) สามารถสร้างและเชื่อมต่อซอฟต์แวร์ที่ถูกพัฒนาขึ้นมาจากภาษาจาวาให้ทำงานร่วมกันได้ ซึ่งการที่จะทำความเข้าใจกับข้อความที่ได้อธิบายไปจะต้องเข้าไปในคำต่อไปนี้

- ส่วนประกอบของซอฟต์แวร์ (Software component)
- เครื่องมือที่ใช้ในการสร้าง (Builder tool)
- การมองเห็นภาพในขณะที่สร้างได้ (Visual manipulation)

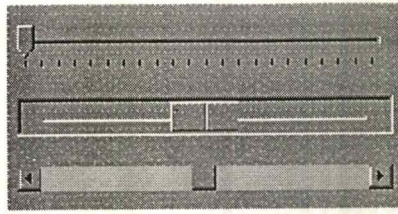
ส่วนประกอบของซอฟต์แวร์

ส่วนประกอบของซอฟต์แวร์ถูกออกแบบมาให้สามารถนำมากลับมาใช้ใหม่ได้ ทำให้สามารถถอดส่วนประกอบบางส่วนออกจากซอฟต์แวร์ที่เก่าหรือล้าสมัย หรือมีความสามารถไม่พอลอก แล้วแทรกหรือเพิ่มส่วนประกอบของซอฟต์แวร์ใหม่ที่ทันสมัย หรือเหมาะสมกว่าเข้าไปแทน(อาจมาจากผู้ผลิตรายอื่น) ทำให้ผู้ใช้สามารถเลือกใช้ผลิตภัณฑ์ที่มีความหลากหลาย มีคุณภาพ และราคาเหมาะสม

ส่วนประกอบของซอฟต์แวร์ที่เป็นมาตรฐานสามารถนำมาใช้ใหม่เป็นรูปแบบง่ายๆเช่น button , text field , scrollbar ตัวอย่างเช่น

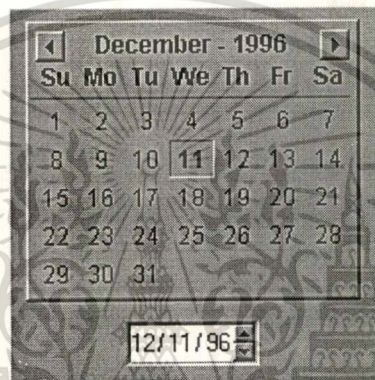


รูปที่ 3-1 Button Bean

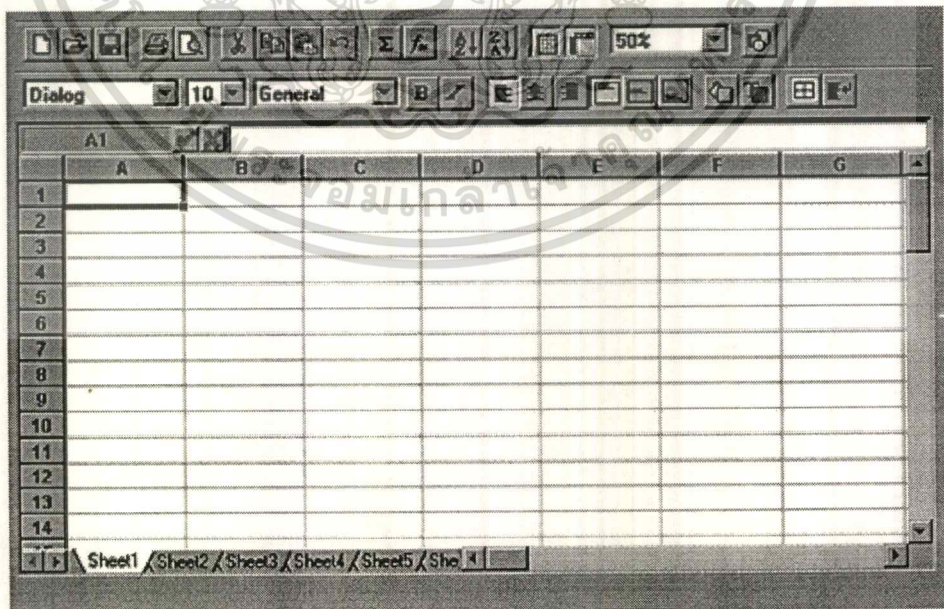


รูปที่ 3-2 Sliders Bean

ทั้งหมดนี้เป็นชนิดของกลุ่มของส่วนประกอบ , toolkits , widget library ซึ่งขายโดยบุคคลที่สาม ภายหลังจากเราจะเห็นผู้ขายนำส่วนประกอบของซอฟต์แวร์ที่ซับซ้อนมากขึ้นออกมาขาย เช่น ปฏิทิน และ spreadsheet



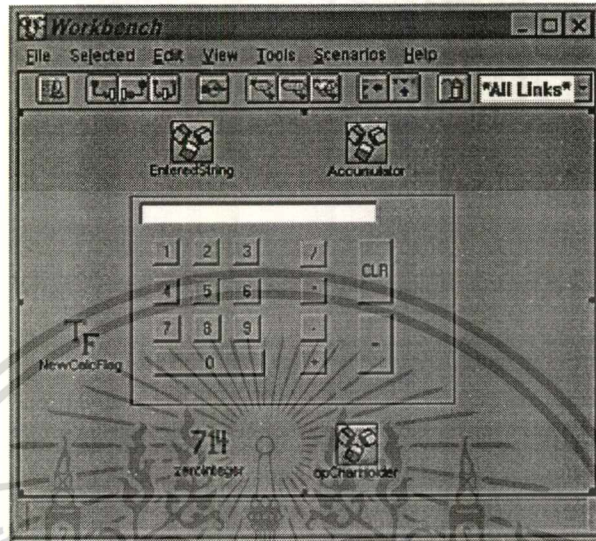
รูปที่ 3-3 Calendar Bean



รูปที่ 3-4 Spreadsheets Bean

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

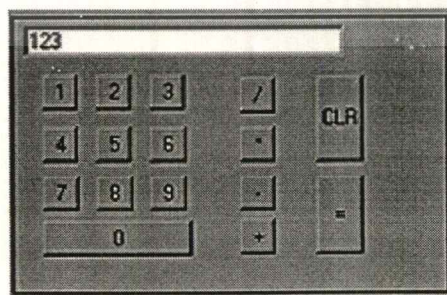
มาตรฐานของการเชื่อมต่อส่วนประกอบและอ็อบเจ็กต์ต่างๆ ทำให้ผู้พัฒนาสามารถกำหนดคุณสมบัติและการทำงานของส่วนประกอบเหล่านั้นได้ ส่วนประกอบของซอฟต์แวร์บางอันไม่จำเป็นต้องปรากฏอยู่ในแอปพลิเคชันนั้นขณะที่ปฏิบัติงานอยู่ แต่จะปรากฏให้เห็นเมื่อเวลาที่สร้างแอปพลิเคชันเท่านั้น



รูปที่ 3-5 เครื่องคิดเลขขณะสร้างในเครื่องมือที่ใช้สร้าง

ตัวอย่างเช่น เครื่องมือที่ใช้ในการสร้างข้างบนแสดงถึงส่วนประกอบของเครื่องคิดเลข ซึ่งสร้างจากส่วนประกอบของปุ่ม 16 ปุ่ม ส่วนประกอบของ text field และ panel ที่ใช้วางปุ่มและ text field ในรูปจะเห็นส่วนประกอบที่ไม่สามารถมองเห็นได้อีก 5 ตัว ซึ่งส่วนประกอบเหล่านี้จะเก็บค่าหรือข้อความที่ใช้โดยเครื่องคิดเลข

ผู้เขียนโปรแกรมสามารถเคลื่อนย้าย ถาม หรือ เห็นความเกี่ยวข้องระหว่างส่วนประกอบทั้งหมดในขณะที่ใช้งานเครื่องมือที่ใช้สร้างอยู่ อย่างไรก็ตามส่วนประกอบเหล่านี้จะทำงานของมันซึ่งไม่สามารถมองเห็นได้เมื่อโปรแกรมทำงาน



รูปที่ 3-6 เครื่องคิดเลขขณะกำลังทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

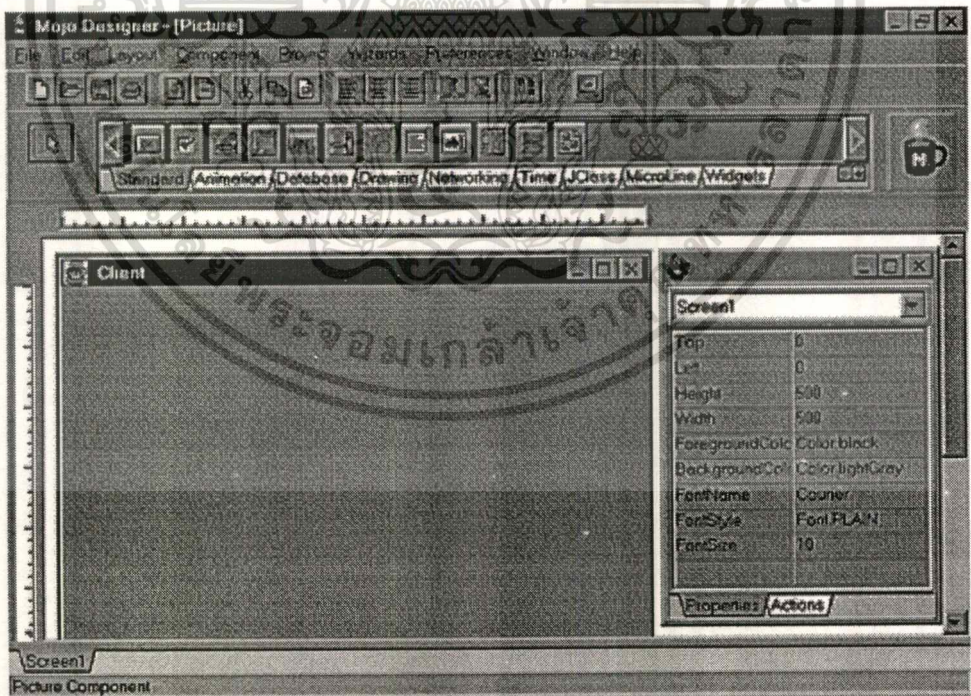
ด้วยบีนคุณสามารถซื้อส่วนประกอบสำหรับจาวาจากบุคคลที่สามได้ คุณยังสามารถซื้อเครื่องมือที่ใช้สร้างหรือโปรแกรมสำเร็จรูปที่ใช้สร้างโปรแกรมที่รองรับบีน โปรแกรมสำเร็จรูปที่เป็นตัวสร้างทำให้คุณสามารถสร้างโปรแกรมจาวาสำเร็จรูป โดยการเลือกมีส่วนประกอบที่เห็นมาจัดวางลงฟอร์มหรือบนหน้าต่างของเครื่องขอรับบริการ และทำการเชื่อมต่อเหตุการณ์ไปยังตัวรับเหตุการณ์

ส่วนประกอบสามารถเชื่อมต่อกันและซับซ้อนได้ตามต้องการ เช่นส่วนประกอบหลายๆตัว (button , text field , panel)รวมกันเป็นเครื่องคิดเลข ซึ่งเครื่องคิดเลขนี้ก็อาจถูกนำไปใช้เป็นส่วนประกอบในโปรแกรมอื่น

3.2 เครื่องมือที่ใช้สร้างแอปพลิเคชัน

จุดประสงค์เริ่มแรกของบีนคือ เพื่อให้เกิดการพัฒนาแอปพลิเคชันในรูปแบบของการมองเห็นได้ (Visual Programming) เช่น Visual Basic , Delphi เป็นต้น เครื่องมือเหล่านี้เรียกว่า visual application builders หรือ builder tools

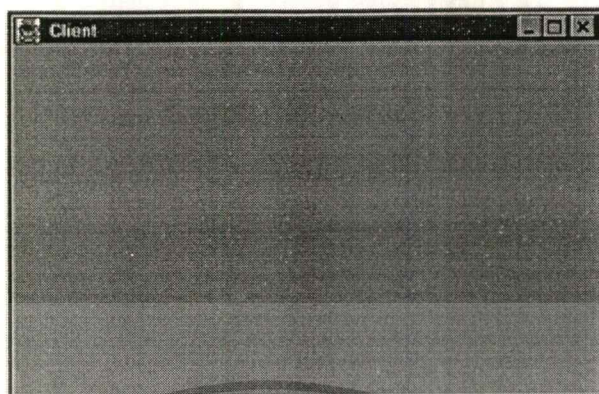
เครื่องมือเหล่านี้เป็นต่างก็เป็นแอปพลิเคชันแบบกราฟิก (GUI หรือ Graphic User Interface) โดยมักจะมีแถบของส่วนประกอบที่ใช้งานได้ ซึ่งผู้ออกแบบโปรแกรมสามารถดึงหรือใช้เมาส์ลากเอามาวางลงในฟอร์มหรือ โคลอนที่วินโดวส์ได้ เช่นใน Mojo



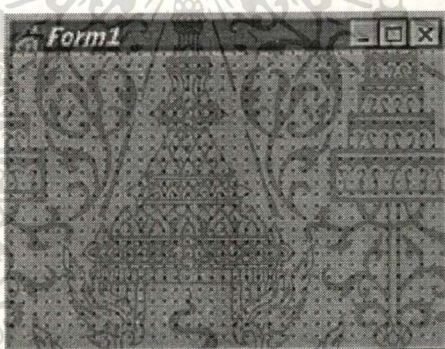
รูปที่ 3-7 Mojo ตัวอย่าง Builder tool

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในสิ่งแวดล้อมของรูปแบบวินโดวส์มักจะเรียก ไคลเอนท์วินโดวส์

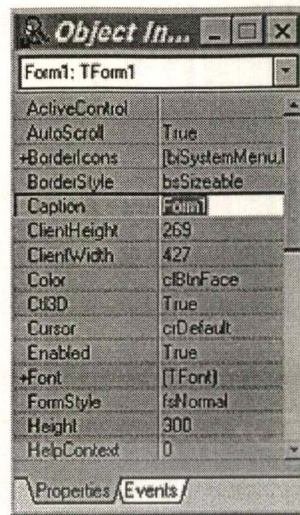


รูปที่ 3-8 หน้าต่างของเครื่องรับบริการ



รูปที่ 3-9 ฟอรัม

ไคลเอนท์วินโดวส์หรือฟอรัมจะแสดงผลของโปรแกรมขณะกำลังออกแบบ และเมื่อนำโปรแกรมไปใช้งานจริง นอกจากนั้นโปรแกรมที่ใช้สร้างส่วนมากนิยมใช้หน้าต่างที่เรียกว่า property sheet , property editor หรือ properties window ซึ่งใช้ในการแก้ไขคุณสมบัติหรือเหตุการณ์ (บางครั้งเรียกการทำงาน) ที่เกี่ยวข้องกับส่วนประกอบ



รูปที่ 3-10 property sheet

3.1.3 แนวความคิดพื้นฐานของบีน

ในจาวาบีนแต่ละตัวจะมีการทำงานที่แตกต่างกัน แต่ส่วนมากจะมีลักษณะที่เหมือนกันดังนี้

- สนับสนุนการใคร่ครวญทำให้เครื่องมือที่ใช้สร้างสามารถวิเคราะห์การทำงานของบีนได้
- สนับสนุนลูกค้ำทำให้ผู้ใช้สามารถเปลี่ยนแปลงการทำงานของบีนได้
 - สนับสนุนเหตุการณ์ทำให้บีนสามารถตั้งเหตุการณ์ และบอกให้กับเครื่องมือที่ใช้สร้างถึงเกี่ยวกับเหตุการณ์ที่ทิ้งไป และเหตุการณ์ที่สามารถรองรับได้
- สนับสนุนคุณสมบัติ ทำให้บีนสามารถปรับเปลี่ยนคุณสมบัติได้ เช่นเดียวกับการเปลี่ยนแปลงการทำงาน
 - สนับสนุนการยีนกราน โดยให้บีนที่ได้รับการแก้ไขในเครื่องมือที่ใช้สร้างสามารถเก็บสถานะ และสามารถนำมาซ่อมแซมได้

บทที่ 4

JDBC (Java Database Connectivity)

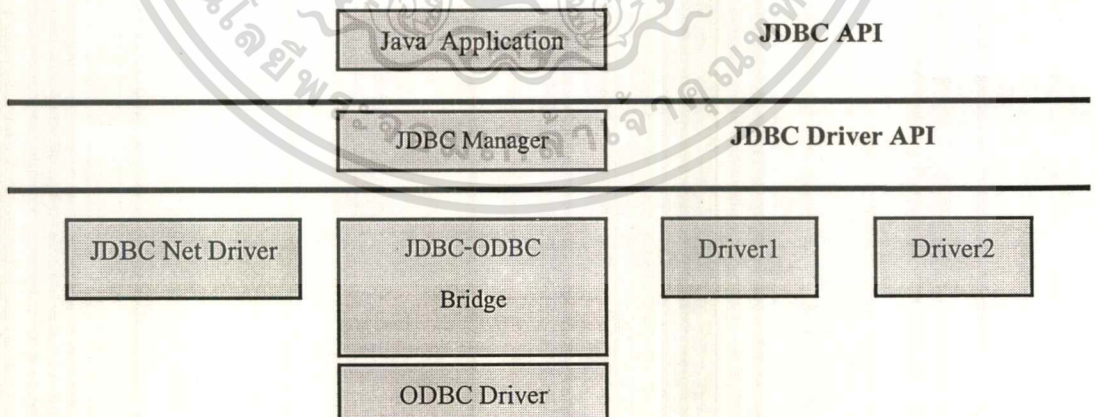
4.1 เจดีบีซี คืออะไร

เจดีบีซี หรือ Java Database Connection เป็นมาตรฐานที่ช่วยในการติดต่อ ระหว่างตัวจาวากับระบบจัดการฐานข้อมูล (DBMS : Database Management System) โดยตัวที่ใช้ทำงานจริง ๆ ระหว่างจาวากับระบบจัดการฐานข้อมูลก็คือ เจดีบีซีเอพีไอ (JDBC API : Java Database Connectivity Application Programming Interface) สำหรับใช้ในการ ปฏิบัติตามคำสั่งเอสคิวแอล ซึ่งประกอบไปด้วยคลาสและอินเตอร์เฟซที่เขียนด้วยภาษาจาวาเจดีบีซีจะจัดหาเอพีไอ มาตรฐานสำหรับผู้ทำการพัฒนา เครื่องมือ(tool) หรือฐานข้อมูล และทำให้เป็นไปได้ในการเขียนแอปพลิเคชันเกี่ยวกับฐานข้อมูลโดยใช้ภาษาจาวา

4.2 เลเยอร์ของเจดีบีซี

การทำงานจะอยู่ที่ 2 เลเยอร์หลักได้แก่

1. เจดีบีซีเอพีไอ ซึ่งจะทำการติดต่อระหว่างแอปพลิเคชันหรือแอปเพล็ตใดๆ ที่เขียนขึ้นด้วยภาษาจาวากับเจดีบีซีเมนเนเจอร์
2. เจดีบีซีไดรฟ์เวอร์เอพีไอ จะทำการติดต่อระหว่างเจดีบีซีเมนเนเจอร์กับไดรฟ์เวอร์ของระบบฐานข้อมูลที่ติดต่อด้วยดั่งรูป



รูปที่ 4-1 โมเดลชั้นการทำงานของเจดีบีซี

เราจะเห็นได้ว่า 2 เลเยอร์หลัก จะมีองค์ประกอบที่จำเป็นของการทำงานของเจดีบีซี คือชั้นเจดีบีซีเอพีไอ จะมีแอปพลิเคชันหรือแอปเพล็ต และชั้นเจดีบีซีไดรฟ์เวอร์เอพีไอก็มีเจดีบีซีเมนเนเจอร์ ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำหน้าที่ในการเป็นเจดีบีซีไคร่ฟเวอร์เอพีไอ เพื่อให้แอปพลิเคชันหรือแอปเพล็ตในชั้นบนสามารถติดต่อกับไคร่ฟเวอร์ของระบบฐานข้อมูลได้

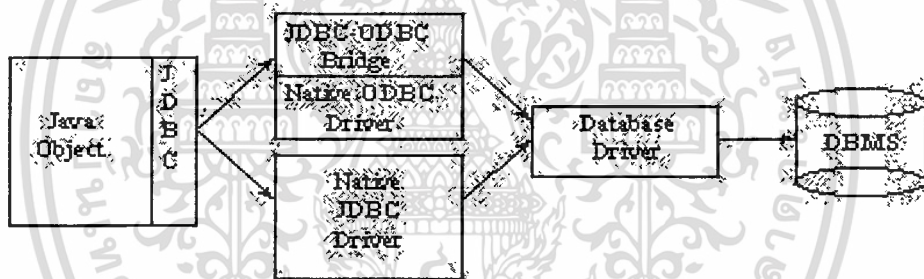
ในชั้นล่างสุดเราจะเห็นไคร่ฟเวอร์ต่างๆที่เจดีบีซี จะอาศัยเจดีบีซีเมเนเจอร์ในการติดต่อเข้ากับแอปพลิเคชันหรือแอปเพล็ต และยังมีเจดีบีซี-โอดีบีซีบริดจ์ (JDBC - ODBC Bridge) ซึ่งเป็นตัวเชื่อมระหว่างแอปพลิเคชันหรือแอปเพล็ตใดๆที่ต้องการติดต่อกับฐานข้อมูลที่สนับสนุนมาตรฐานของไคร่ฟเวอร์โอดีบีซี โดยการทำงานก็เช่นเดียวกับการติดต่อไคร่ฟเวอร์ต่างๆไป

4.3 การทำงานของเจดีบีซี

เจดีบีซีจะทำหน้าที่หลัก 3 อย่างด้วยกันคือ

1. สร้างจุดเชื่อมต่อกับฐานข้อมูลที่ต้องการติดต่อกับ
2. ส่งคำสั่งเอสคิวแอล (SQL Statement) ไปยังฐานข้อมูลนั้น
3. รับผลจากคำสั่งที่ส่งไปในข้อ 2.

รูปแบบที่ใช้ในเจดีบีซี จะเป็นดังรูปข้างล่างนี้



รูปที่ 4-2 รูปแบบที่ใช้ในเจดีบีซี

ในรูปแบบนี้จะสังเกตเห็นได้ว่า จาวาออบเจ็กต์ (Java object) จะร้องขอไปยัง เจดีบีซีคลาส ซึ่งจะเรียกว่าไปยังตัว เจดีบีซีไคร่ฟเวอร์ หรือตัวเจดีบีซี-โอดีบีซีบริดจ์ ที่กำหนด จากนั้นตัวไคร่ฟเวอร์ จะเรียกตัวไคร่ฟเวอร์ ของฐานข้อมูลจริงๆ เพื่อที่จะทำการเชื่อมต่อไปยังฐานข้อมูลให้สมบูรณ์

ขั้นตอนในการเข้าถึง relational database โดยผ่าน เจดีบีซี มีดังนี้

1. ทำการติดตั้งเจดีบีซีแพ็คเกจ (package) และตัวไคร่ฟเวอร์ของฐานข้อมูลที่จำเป็น
2. ทำการเปิดการเชื่อมต่อกับฐานข้อมูล
3. ทำการสร้าง คำสั่งเอสคิวแอล หรือกระบวนการเรียกใช้
4. ทำการปฏิบัติตาม คำสั่งเอสคิวแอล หรือกระบวนการเรียกใช้
5. ทำการประมวลผลที่ได้จาก คำสั่งเอสคิวแอล หรือกระบวนการเรียกใช้
6. ทำการ commit หรือ rollback งาน transaction (ถ้าจำเป็น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 การเชื่อมต่อไปยังฐานข้อมูล

การเชื่อมต่อไปยังฐานข้อมูลจะผ่าน DriverManager class และการเชื่อมต่อ อินเทอร์เน็ต โดยทำตามขั้นตอนดังต่อไปนี้

การร้องขอ เจดีบีซี package:

```
import java.sql.*;
```

ทำการ load ตัว ไดรฟ์เวอร์ ของฐานข้อมูลที่เหมาะสม:

```
// Use the JDBC-ODBC bridge as an example
Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
(Throws a ClassNotFoundException error.)
```

การสร้างฐานข้อมูล URL:

```
// The database URL takes the form jdbc:subprotocol:subname
String url = "jdbc:odbc:dsn"
```

การกำหนด USER ID และ Password:

```
String user = "guest";
String password = "guest";
```

การเปิดการเชื่อมต่อ:

```
// Call the Driver Manager to open a database connection
Connection con = DriverManager.getConnection (url, user, password);
(Throws a SQLException error.)
```

การ Settings อื่น ๆ:

```
// Transactions are automatically committed when executed
con.setAutoCommit (true) ;

// Put connection in read-only mode
con.setReadOnly (true).

(Throws a SQLException error.)
```

การปิดการเชื่อมต่อ:

```
con.close ();
```

(Throws a SQLException error.)

4.3.2 คำสั่งเอสคิวแอล

เอสคิวแอล queries ถูกแสดงผ่านการอินเทอร์เฟซ Statement และ ResultSet

การสร้าง Statement Object:

```
Statement stmt = con.createStatement ();
```

(Throws a SQLException error.)

การ Set Query String:

```
String query = "SELECT * FROM TABLE_NAME";
```

การปฏิบัติตาม Query และการสร้าง Result Set Object:

```
ResultSet rs = stmt.executeQuery (query);
```

(Throws a SQLException error.)

4.3.3 กระบวนการทำ Result Set

ผลของ เอสคิวแอล query ถูกนำออกผ่าน ResultSet object โดยในแอปพลิเคชัน จะมีวิธีการจัดการกับ rows และ columns จาก query

การอ่าน row ถัดไป:

```
boolean found = rs.next (); // returns true if row is read, false if end
of result set
```

(Throws a SQLException error.)

การ extract column values:

```
/** Issue a getXxxx() call for each column, base on its data type.
```

```
* Common data types are:
```

```
* INTEGER           getInt () or getBigDecimal (,0)
* NUMBER            getBigDecimal ()
* VARCHAR           getString ()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
* LONGVARCHAR          getAsciiStream ()
* TIMESTAMP            getDate (), getTime (), or getTimestamp ()
```

*

```
* Columns may be accessed through a relative position within the row (e.g.
* column 1, 2, 3, ...) or through its actual column name.
```

*

```
* SQL NULL values will result in the Java null value being assigned to the
* Java variable.
```

*/

```
int    personKey = rs.getInt (1);
String lastName = rs.getString ("LAST_NAME");
String firstName = rs.getString ("FIRST_NAME");
Date   birthDate = rs.getDate (4);
(Throws a SQLException error.)
```

การปิด Result Set:

```
// Close the result set object after row processing is complete
rs.close ();
(Throws a SQLException error.)
```

การปิด Statement:

```
// Close the statement object when row processing is complete
stmt.close ();
(Throws a SQLException error.)
```

4.3.4 การ Update Transaction

SQL INSERT, UPDATE, หรือ DELETE คล้ายคลึงกับ QUERY ทั่ว ๆ ไป แตกต่างก็เพียงแค่ชื่อวิธีเรียก และชนิดของผลที่จะได้รับ

การสร้าง Statement Object:

```
Statement stmt = con.createStatement ();
(Throws a SQLException error.)
```

การ Set คำสั่งที่ Update:

```
String query = "INSERT INTO TABLE_NAME VALUES
(1, 'SMITH', 'JOHN', NULL)";
```

การ Execute Update:

```
int rowsUpdated = stmt.executeUpdate (query);
// returns number of rows updated
(Throws a SQLException error.)
```

การ Commit Transaction:

```
con.commit ();
(Throws a SQLException error.)
```

การ Rollback Transaction:

```
con.rollback ();
(Throws a SQLException error.)
```

การปิด Statement Object:

```
// Close the statement object after update is complete
stmt.close ();
(Throws a SQLException error.)
```

4.3.5 การเรียกกระบวนการเก็บ

การเรียกกระบวนการเก็บและความสัมพันธ์ต่าง ๆ จะเป็น บิตมากกว่า เอสคิวแอล queries และการ updates โดยใช้ CallableStatement อินเตอร์เฟส ซึ่งจะต่อจาก PreparedStatement อินเตอร์เฟส โดยตัวแปรที่ใช้จะเป็น placeholder symbols เช่น INOUT ใช้ ชนิดข้อมูลจากชนิดของคลาส

การสร้างคำสั่งด้วย placeholders:

```
// Stored procedure call
String command = "{call PROCEDURENAME (?, ?, ?, ?)}";
// 4 placeholders
// Stored function call
String command = "{? = call FUNCTIONNAME (?, ?, ?, ?)}";
// 4 placeholders + 1 return value
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้าง Callable Statement Object:

```
CallableStatement cstmt = con.prepareCall (command);\n\n(Throws a SQLException error.)
```

Register the IN and OUT Placeholders:

```
// Assign In parameters (use positional placement)\n\n cstmt.setInt (1, 50);           // first placeholder has an integer value of 50\n\n cstmt.setString (2, "Smith");\n\n     // second placeholder has a string value of "Smith"\n\n cstmt.registerOutParameter (3, Types.NUMERIC);\n\n     // NUMERIC is preferred over INTEGER with some databases\n\n cstmt.registerOutParameter (4, Types.VARCHAR);\n\n     // fourth placeholder is an OUT field of type VARCHAR.\n\n(Throws a SQLException error.)
```

หมายเหตุ: INOUT fields may be registered using the setXXXX() and registerOutParameter() methods for the same placeholder position. Array parameters (i.e. PL/SQL Tables) used in Oracle stored procedure and function calls are not supported by the JDBC.

การ Execute กระบวนการและ function การเรียก:

```
cstmt.execute () ;\n\n(Throws a SQLException error.)
```

กระบวนการ OUT Placeholders:

```
/** Use proper getXXXX() method based on parameter data types.\n\n * Placeholder 3 will receive a NUMERIC value as a BigDecimal object with\n\n * 0 decimal places.\n\n * Placeholder 4 will receive a VARCHAR value as a String object.\n\n */\n\nBigDecimal num = cstmt.getBigDecimal (3,0);\n\nString str = cstmt.getString(4);\n\n(Throws a SQLException error.)
```

การปิด CallableStatement Object:

```
// Close the call object after update is complete
cstmt.close ();
(Throws a SQLException error.)
```

4.3.6 การเข้าถึงเมตาดาต้า (Metadata)

เมตาดาต้า เป็นข้อมูลที่อธิบายเกี่ยวกับตัวโครงสร้างข้อมูล และการเชื่อมต่อภายในฐานข้อมูล ตัวอย่างของ เมตาดาต้า จะรวมถึง:

- ใครเป็นผู้ติดต่อ ไปยังฐานข้อมูล
- ฐานข้อมูลนี้เป็น read-only หรือ updatable
- ชนิดข้อมูล ที่เหมาะสมสำหรับฐานข้อมูลนี้คืออะไร
- ชนิดข้อมูล ของ column นี้คืออะไร
- vendor ของฐานข้อมูลเกี่ยวข้องกับกระบวนการเก็บอย่างไร
- database handle cursors ทำ rollbacks และ commits อย่างไรเพิ่มเติมคือเมตาดาต้า จะจัดหาโดยผลจาก query ตัวอย่างของเมตาดาต้าที่สัมพันธ์กับ query จะรวมถึง
- มี columns ที่ columns ที่ถูกนำออกในแต่ละ row ของผลที่ได้
- ชื่อของ column และชนิดของ เอสคิวแอล คืออะไร
- ชื่อของ ตาราง ของ column และ ตาราง schema คืออะไร
- ใน 1 column มีตัวเลขของฐานสิบอยู่เท่าไร

ในการเข้าถึง เมตาดาต้า จะทำผ่านอินเทอร์เฟส 2 อีอบเจ็กต์คือ DatabaseMetaData และ

ResultSetMetaData

```
DatabaseMetaData dmeta = con.getMetaData ();
ResultSetMetaData rsmeta = rs.getMetaData ();
```

ตัวอย่าง

ตัวอย่างโปรแกรม Sample.java จะสาธิตการใช้เจดีบีซี โดยใช้ จาวาแอปพลิเคชัน เพียงอย่างเดียว

ในการ compile โปรแกรมให้ทำดังนี้:

```
javac Sample.java
```

ทำการ run โปรแกรม และสังเกตผลที่ได้:

```
java Sample
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 เปรียบเทียบซีจีไอ (CGI) และ เจดีบีซี

ความพิเศษของการเลือกใช้ภาษาจาวา

การใช้ ภาษา จาวา สามารถทำการเข้าถึงข้อมูลได้ เนื่องจากภาษา จาวา เป็นภาษาที่มีคุณลักษณะ ต่างๆดังนี้

1. มี performance
2. เป็นแบบ database and platform independence
3. มี scalability
4. มีความ simplicity
5. มีความ compatibility

ในการเข้าถึงฐานข้อมูลบนเว็บสามารถทำได้ 2 ทางคือ

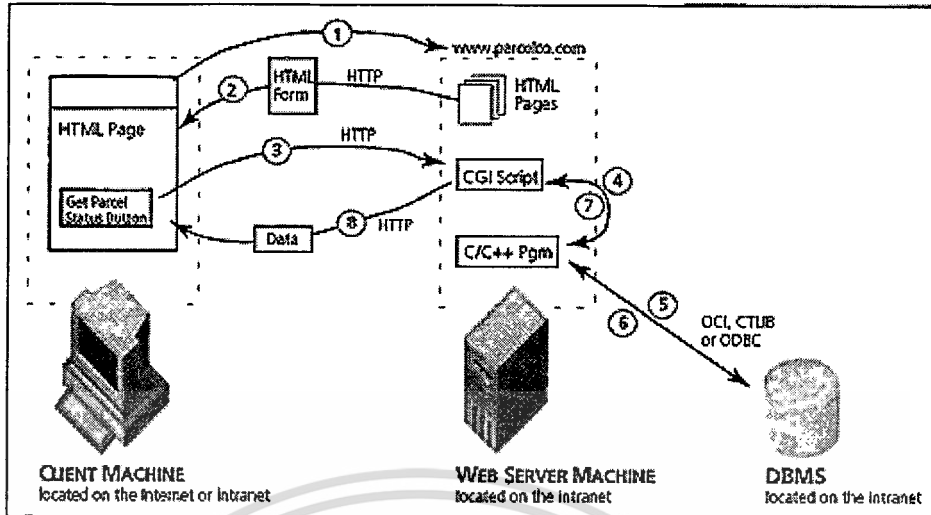
1. HTML pages โดยใช้ Common Gateway Interface (CGI) scripts บน server

ในการใช้ CGI script จะต้องใช้โอทีบีซีไครฟ์เวอร์ ในการเข้าถึงข้อมูลของระบบฐานข้อมูล โดยเว็บเซิร์ฟเวอร์ จะส่งรูปแบบของ เอกซีเอ็มแอล ไปยังผู้ใช้ เว็บเบราว์เซอร์ (Web browser) นั้น ๆ อยู่ จากนั้นรูปแบบ เอกซีเอ็มแอล จะทำการเข้าถึงข้อมูลที่อยู่ใน DBMS ซึ่งมันจะร้องขอ CGI script บน เว็บเซิร์ฟเวอร์ โดย CGI script ก็จะร้องขอต่อไปยัง C/C++ program เพื่อให้ส่งค่ากลับมาอีกที

ข้อเสีย :

1. Low Performance เนื่องจากในการร้องขอไปยังฐานข้อมูลทุกครั้งจะต้องทำการ connection ไปยัง server machine ทุกครั้ง เกิด overhead เพิ่มขึ้น
2. Complex Coding ในการใช้ CGI script บ่อยครั้งที่ต้องเขียนด้วย PERL ซึ่งเป็นภาษาที่ทำความเข้าใจได้ยาก ทำการสร้าง หรือ maintain ได้ยากเช่นกัน
3. Platform-specific Deployments ในการใช้ CGI script จะต้องมี การเขียน code ทำการสร้าง และทดสอบเฉพาะของแต่ละแพลตฟอร์มไปเลย
4. Poor scalability ใน HTTP จะจำกัดความสามารถในการใช้ application แบบ multi-threading เพราะการใช้งานของแต่ละคนกับฐานข้อมูลมักแยกคนละ process กัน
5. Reduced Interactivity เพราะมันจะทำได้ที่เครื่อง เซิร์ฟเวอร์ เท่านั้น
6. Unsophisticated Data Handling

Internet DBMS access using HTML and CGI scripts



รูปที่ 4-3 การติดต่อไปยังฐานข้อมูลโดยใช้ CGI

2. Java programs โดยใช้เจดีบีซี

ในการใช้แบบที่ 2 นี้ ในเว็บเบราว์เซอร์จะให้ผู้ใช้ได้ URL และ เว็บเซิร์ฟเวอร์ จะรับ การร้องขอ จากผู้ใช้และใช้ HTTP return Web page ไปให้ เว็บเบราว์เซอร์ของ โคลน์เอ็นท์ ซึ่งมี จาวาแอฟเพล็ต อยู่ ด้วย จากนั้นจะรับการร้องขอจากเจดีบีซีเซิร์ฟเวอร์ และส่งผลไปยังเจดีบีซี โคลน์เอ็นท์ จากนั้น แอฟเพล็ต จะได้รับผล และแสดงผลออกมาบน เว็บเบราว์เซอร์ของผู้ใช้

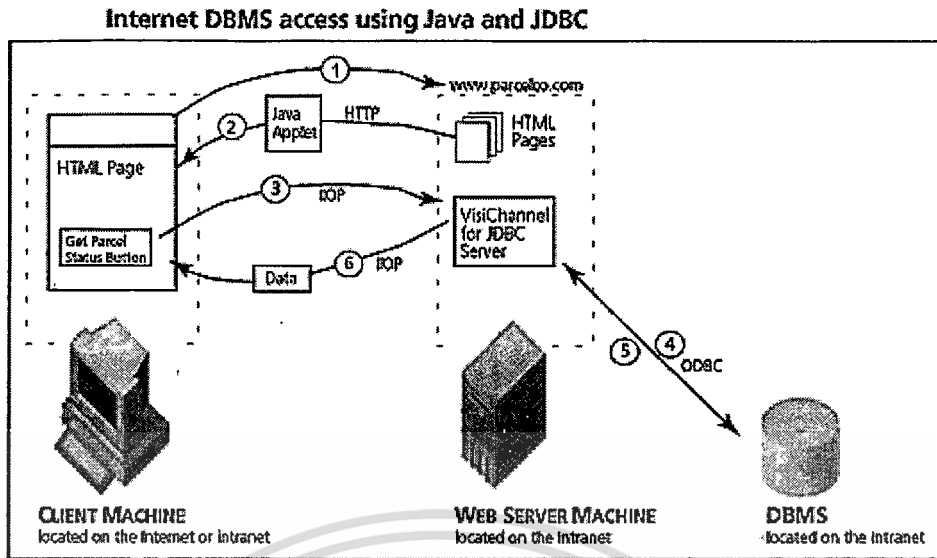
การใช้ตามแบบที่ 2 นี้จะช่วยแก้ปัญหาของข้อแรกได้:

1. More Efficient Resource Usage
2. Improved Reliability and Flexibility ภาษา จาวาเป็นภาษาที่มีความน่าเชื่อถือ
3. Complete Interoperability สามารถ run บนเครื่องใดก็ได้
4. Increase Scalability มีความสามารถในการทำ multithreading

และประโยชน์อื่น ๆ อีกคือ

1. อนุญาตให้ใช้ ส่วนประกอบของซอฟต์แวร์และสามารถนำกลับมาใช้ใหม่ได้ เนื่องจากเป็น object-oriented architecture
2. มีการจัดหา web-based services ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



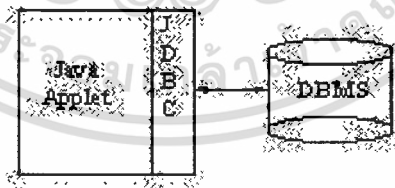
รูปที่ 4-4 การติดต่อไปยังฐานข้อมูลโดยผ่าน เจดีบีซี

4.5 การพิจารณาการออกแบบ Client-Server

รูปแบบของ client-server แบบดั้งเดิมจะถูกสร้างโดยใช้ two หรือ three tiers ดังนี้

Two-Tier Model

รูปแบบนี้ถูกพบเมื่อ ไลอ้อนเอ็นท์ ทำการเรียกแบบตรง ๆ ไปยังฐานข้อมูลในโลกของจาวา จะถูกแสดง โดยจาวาแอปเพล็ต แสดงการเรียกเจดีบีซี



รูปที่ 4-5 Two-Tier model

Java Applet → Database:

จาวาแอปเพล็ต ถูกโหลดเข้าไปในบราวเซอร์ของไลอ้อนเอ็นท์ และเรียกฐานข้อมูลผ่านเจดีบีซี

ข้อดี:

1. ช่วยบำรุงการเชื่อมต่อของฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โคลนเอ็นท์ที่ไม่ต้องการการบำรุงเพราะการเข้าถึงฐานข้อมูลทั้งหมดถูกทำบน เว็บเซิร์ฟเวอร์
3. จัดหาส่วนควบคุมส่วนกลาง และ administration ของ logic ของฐานข้อมูลบน เว็บเซิร์ฟเวอร์

ข้อเสีย:

1. มีบาง performance หายไป
2. ไม่สามารถบำรุงการติดต่อฐานข้อมูล
3. ความต้องการ resource บน เว็บเซิร์ฟเวอร์ เพิ่มขึ้น

4.6 ชนิดของ เจดีบีซี ไดรฟ์เวอร์ (Driver)

เจดีบีซี ไดรฟ์เวอร์หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชันเอพีไอ ต่างๆของเจดีบีซี ถูกจัดแบ่งออกเป็นทั้งหมด 4 ชนิดคือ

1. JDBC-ODBC Bridge
2. Native-API (PARTLY-JAVA) Drivers
3. Net-Protocol (All-Java) Drivers
4. Native-Protocol (ALL-Java) Drivers

โดยมีรายละเอียดการทำงานของแต่ละชนิดอธิบายดังนี้

1. JDBC-ODBC Bridge

ทำหน้าที่เป็นตัวกลางในการเข้าถึงฐานข้อมูลโดยแปลงฟังก์ชันของเจดีบีซี ให้อยู่ในรูปแบบฟังก์ชันของ โอดีบีซี ทำให้เจดีบีซีสามารถติดต่อกับฐานข้อมูลได้โดยผ่านการทำงานของโอดีบีซีรูปแบบการเชื่อมต่อกับฐานข้อมูลของ เจดีบีซี-โอดีบีซีบริดจ์ แสดงดังรูป4-7

ข้อดี:

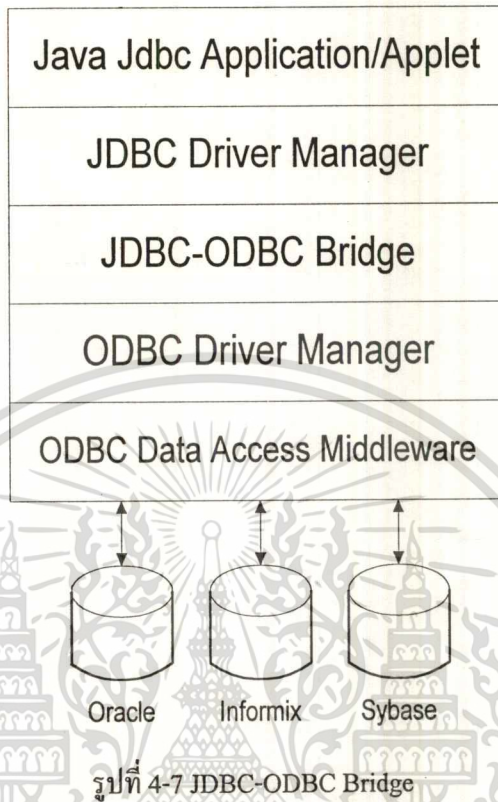
-ไม่ต้องติดตั้งและปรับเปลี่ยนไดรฟ์เวอร์ใหม่ทั้งหมด เพราะ โอดีบีซีได้ถูกติดตั้งอยู่บนเครื่องคอมพิวเตอร์ส่วนมากที่ใช้งานในปัจจุบันอยู่แล้ว (Windows 95/NT, Macintosh, และ UNIX บางระบบ) ทำเพียงแค่นำ เจดีบีซี-โอดีบีซีบริดจ์ มาต่อเชื่อมเพิ่มประสิทธิภาพให้กับโอดีบีซีเท่านั้น

ข้อเสีย:

-เจดีบีซี-โอดีบีซีบริดจ์ ไดรฟ์เวอร์ถูกเขียนขึ้นจากภาษา C/C++ ในบางส่วน จึงทำให้ไม่สามารถดาวน์โหลด (Download) ข้ามเครือข่ายอินเทอร์เน็ตและแปล (Interpret) บนเว็บเบราว์เซอร์(Webbrowser) ได้ เพราะภาษา C/C++ ต้องรันเฉพาะแต่ละระบบเท่านั้น ดังนั้นโอดีบีซี ไดรฟ์เวอร์ในรูปแบบของไลบรารีไฟล์หรือ DLL (Dynamic Link Library) และ เจดีบีซี-โอดีบีซีบริดจ์ ไดรฟ์เวอร์ต้องได้รับการติดตั้งบน

โคลนเอ็นท์แต่ละเครื่องเพื่อให้สามารถติดต่อกับฐานข้อมูลที่รู้จักได้สุดท้าย เจดีบีซี-โอดีบีซีบริดจ์ จึงเหมาะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการใช้งานกับระบบ Three-Tier โดยไคลฟ์เวอร์ทั้งสองจะได้รับการติดตั้งอยู่ที่ Middle-Tier หรือคอมพิวเตอร์ตัวกลาง เช่น เว็บเซิร์ฟเวอร์ เป็นต้น



2. Native-API (Partly-Java) Drivers

ทำงานโดยแปลงฟังก์ชันของเจดบีซีให้เป็นฟังก์ชัน เอพีไอ เฉพาะของแต่ละฐานข้อมูลเป้าหมาย Native-API Drivers มีหลักการทำงานคล้ายกับ โอดีบีซี ที่ต้องมีไคลฟ์เวอร์ (DLL) ติดตั้งอยู่บน โคลน์เอ็นท์ แต่ละเครื่องยกตัวอย่างเช่น การเชื่อมต่อกับฐานข้อมูล Oracle ดังรูป ก็ต้องใช้ เจดบีซี ไคลฟ์เวอร์และ DLL เฉพาะสำหรับติดต่อผ่าน โปรโตคอล (Protocol) SQL-NET ของ Oracle เท่านั้น ฉะนั้น Native-API Drivers จึงเหมาะกับการทำงานแบบ Two-Tier คือ โคลน์เอ็นท์ที่ติดต่อกับฐานข้อมูลบนเซิร์ฟเวอร์โดยตรง

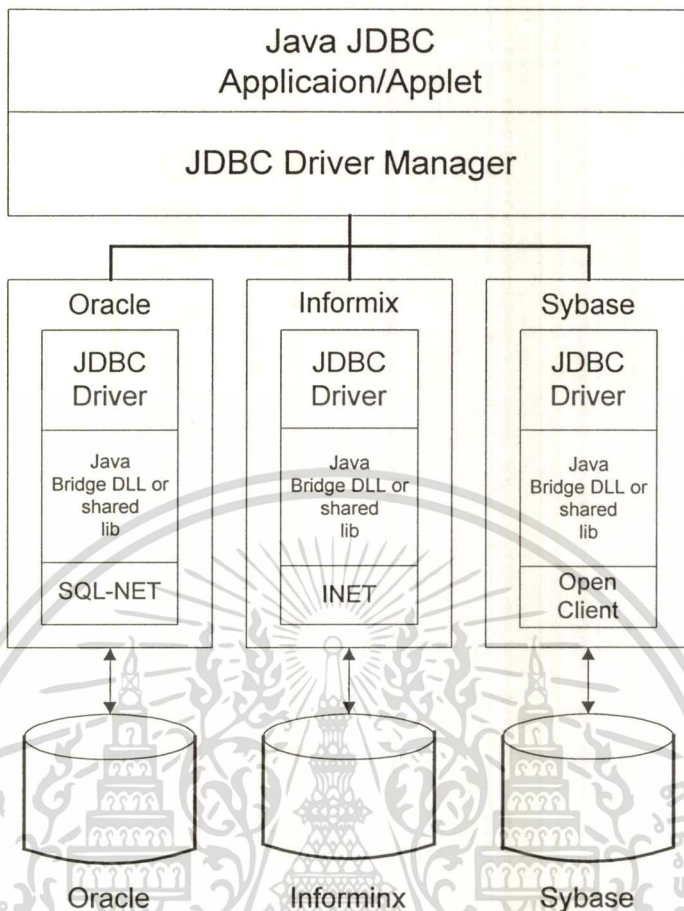
ข้อดี:

- สามารถใช้ Native-API Drivers ติดต่อกับฐานข้อมูลนั้นๆ โดยตรงโดยผ่าน โปรโตคอลเดิมที่ใช้อยู่ก่อนแล้ว ทำให้เหมาะสมกับการเชื่อมต่อกับฐานข้อมูลแบบ Two-Tier Client-Server

ข้อเสีย:

- คล้ายคลึงกับเจดบีซี-โอดีบีซีบริดจ์ คือต้องติดตั้งไคลฟ์เวอร์ (DLL) และโปรโตคอลไคลฟ์เวอร์ (เช่น SQL-NET, INET, Open Client, ฯลฯ) บนเครื่องคอมพิวเตอร์นั้นๆ ที่สำคัญส่วนหนึ่งของ Native-API Drivers ถูกเขียนขึ้นจากภาษา C++ (Partly-Java) จึงไม่สามารถดาวน์โหลดผ่านเครือข่ายอินเทอร์เน็ตได้ และไม่สามารถเชื่อมต่อฐานข้อมูลข้ามชนิดกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 Native (Partly-Java) Drivers

3. Net-Protocol (All-Java) Drivers

ทำหน้าที่แปลงฟังก์ชันของ เจดีบีซี ให้อยู่ในรูปแบบของเน็ตเวิร์กโปรโตคอลร่วม (DBMS-Independent network Protocol) ซึ่งหลังจากนั้นจะถูกแปลงให้อยู่ในรูปแบบเฉพาะของแต่ละฐานข้อมูลบนเซิร์ฟเวอร์นั้นๆ รูปแบบการเชื่อมต่อข้อมูลแสดงดังรูป สำหรับ เจดีบีซี ไดรฟ์เวอร์ชนิดนี้เป็นไดรฟ์เวอร์ที่ใช้งานได้กว้างขวาง เพราะเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งทำให้สามารถเชื่อมต่อกับฐานข้อมูลหลายๆตัวที่แตกต่างกัน

ข้อดี:

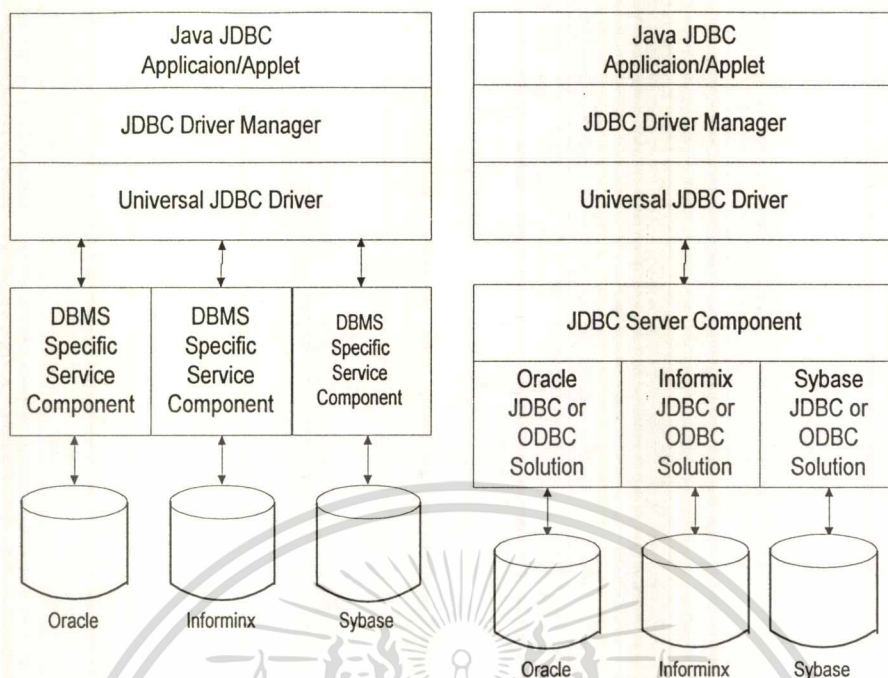
-เป็นไดรฟ์เวอร์ที่ยืดหยุ่นคล่องตัวที่สุด เพราะว่าเขียนขึ้นจากภาษาจาวาทั้งหมด ทำให้ไม่จำเป็นต้องมีไดรฟ์เวอร์ร่วมที่เขียนที่เขียนขึ้นจากภาษาอื่นซึ่งต้องติดตั้งบน โคลน์เอ็นทีนั้น และสามารถรันบนระบบใดก็ได้ที่สนับสนุนสภาพแวดล้อมเสมือนของจาวาหรือ JVM (Java Virtual Machine) การปฏิบัติงานทุกรูปแบบ ทั้งดาวน์โหลดข้ามเครือข่ายหรือรันบนเครื่องเดี่ยวๆสามารถกระทำได้หมด

ข้อเสีย:

-ในเรื่องความปลอดภัยเพราะทางเลือกในการใช้งานต่างๆจะขึ้นอยู่กับผู้ผลิตไดรฟ์เวอร์ว่าจะ

อนุญาตให้ใช้งานส่วนใดได้บ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-9 Net-Protocol (All-Java) Drivers

4. Native-Protocol (All-Java) Drivers

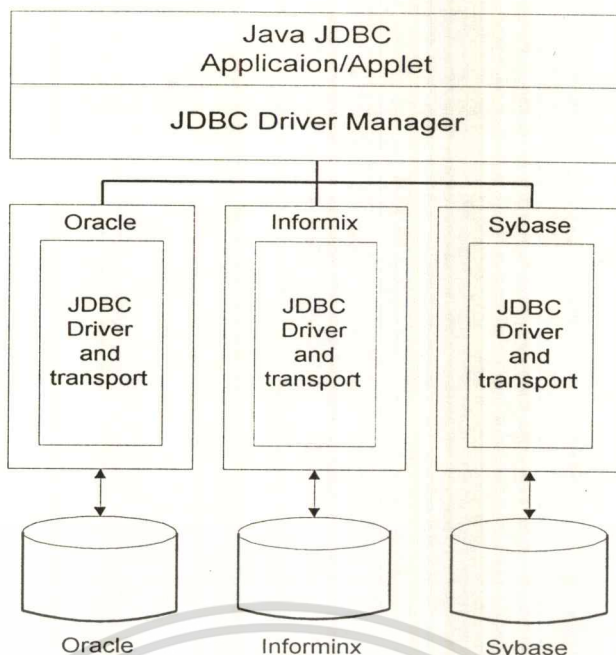
ทำหน้าที่แปลงคำสั่งของ เจดีบีซี ให้อยู่ในรูปแบบของเน็ตเวิร์คโปรโตคอลเฉพาะของฐานข้อมูลนั้นโดยตรง การร้องขอข้อมูลจากโคลนเอ็นทีจึงกระทำโดยตรงต่อเซิร์ฟเวอร์ดังแสดงในรูป ไดรฟ์เวอร์ชนิดนี้ถูกเขียนขึ้นจากภาษาจาวาใหม่ทั้งหมดจากผู้ผลิตระบบฐานข้อมูลนั้นๆ และไดรฟ์เวอร์ใหม่นี้จะทำงานที่แทนเน็ตเวิร์คโปรโตคอลเดิม (Oracle SQL-NET, Informix I-Connect, Sybase Open Client, ฯลฯ)

ข้อดี:

-ไดรฟ์เวอร์ชนิดนี้คือไม่ต้องมีการปรับเปลี่ยนระบบฐานข้อมูลเดิมที่ใช้งานอยู่แล้วในแต่ละองค์กร และไม่มีควมจำเป็นต้องติดตั้งไดรฟ์เวอร์ตัวกลาง (เช่น โอดีบีซี) บนโคลนเอ็นทีแต่ละเครื่อง จึงเหมาะสำหรับการพัฒนาการเชื่อมต่อฐานข้อมูลแบบ Two-Tier สำหรับเครือข่ายอินทราเน็ตภายในองค์กร

ข้อเสีย:

-ไดรฟ์เวอร์ชนิดนี้ไม่เหมาะที่จะนำไปใช้บนเครือข่ายอินเทอร์เน็ต เพราะปัญหาเรื่องความปลอดภัย การเชื่อมต่อข้อมูลไม่สามารถกระทำข้ามระบบที่แตกต่างกันได้ ต้องใช้ไดรฟ์เวอร์เฉพาะสำหรับฐานข้อมูลตัวนั้นๆ



รูปที่ 4-10 Native-Protocol (All-Java) Drivers

โดยทั่วไปแล้วแบบที่ 1 จะช้าที่สุด ส่วนแบบที่ 2 จะเร็วที่สุด แบบที่ 3 จะมีการโหลดแต่จะใช้เวลาน้อยส่วนการปฏิบัติการจะไม่เร็วเท่ากับแบบที่ 2 แบบที่ 4 จะมีประสิทธิภาพที่สุด

เจดีบีซี ไดรฟ์เวอร์ ชนิดที่ 3 และ 4 คือ ไดรฟ์เวอร์ ที่คาดว่าจะเป็นสิ่งที่ต้องการสำหรับการเชื่อมต่อกับฐานข้อมูลในอนาคตเพราะถูกเขียนขึ้นจากภาษาจาวาทั้งหมดซึ่งมีความปลอดภัยและความคล่องตัวมากกว่า (เช่น มีการติดตั้ง ไดรฟ์เวอร์ โดยอัตโนมัติเมื่อมีการ download จาวาแอปเพล็ต) สำหรับ ไดรฟ์เวอร์ ชนิดที่ 1 และ 2 คือ ไดรฟ์เวอร์ ขัดตาทัพซึ่งถูกพัฒนาขึ้นมาเพื่อใช้งานร่วมกับ ไดรฟ์เวอร์ ดั้งเดิม (โอดีบีซี หรือ Network protocol เฉพาะฐานข้อมูล) ที่มีใช้อยู่ในปัจจุบันไปพลาง ๆ ก่อน ซึ่งแน่นอนว่า ไดรฟ์เวอร์ ทั้ง 2 ชนิดนี้มีข้อจำกัดในการใช้งานมากกว่า 2 ชนิดแรก โดยเฉพาะเรื่องความปลอดภัยและความไม่เป็นอิสระจากระบบคอมพิวเตอร์

รายชื่อในตารางข้างล่างต่อไปนี้เป็นผู้ผลิต (vendor) ที่ได้พัฒนาซอฟต์แวร์ของตนเพื่อช่วยในการสนับสนุนการติดต่อกับฐานข้อมูล ซึ่งจะแบ่งตามชนิดประเภท เจดีบีซี ไดรฟ์เวอร์ ที่ได้กล่าวไป

ผู้ขาย (Vendor)	ชนิด	สนับสนุนฐานข้อมูล
Agave Software Design	3	Oracle, Sybase, Informix, others via ODBC
Asgard Software	3	Unisys A series DMSII database
Borland	4	InterBase 4.0
Caribou Lake Software	3	Ingres
Connect Software	4	Sybase, MS SQL Server
DataRamp	3	Several dozen through ODBC Driver.*
Ensodex, Inc.	3	Several through ODBC
IBM	2 / 3	IBM DB2 Version 2
IBM	4	DB2 for OS/400
GWE Technologies	4	MySQL
IDS Software	3	Oracle, Sybase, MS SQL Server, Informix, MS Access, Watcom and others via ODBC.
I-Kinetics, Inc.	3	Oracle, Informix, Sybase and others via ODBC.
Imaginary	4	MSQL
InterSoft	3	Essentia
Intersolv	3	DB2, Ingres, Informix, Oracle, MS SQL Server, Sybase 10/11
JavaSoft	1	Several dozen through ODBC drivers.*
KonaSoft, Inc.	3 / 4	Sybase, Oracle, Informix
NetAway	3	Oracle, Informix, Sybase, MS SQL Server, DB2, others via ODBC
OpenLink	3	Oracle, Informix, Sybase, MS SQL Server, CA-Ingres, Progress, Unify, PostgreSQL, Solid, others via ODBC
Oracle Corporation	2 / 4	Oracle
SAS Institute, Inc.	3 / 4	SAS, and via SAS/Access, Oracle, Informix, Ingres and ADABAS
SCO	3	Oracle, Informix, Sybase, Ingres, Interbase
StormCloud Development	3	Any DBMS that is accessible via ODBC
Sybase, Inc.	3 / 4	Sybase SQL Server, SQL Anywhere, Sybase IQ, Replication server and more than 25 enterprise and legacy database servers via Sybase OmniCONNECT
Symantec	3	Oracle, Informix, Sybase, MS SQL Server, MS Access, Watcom and others via ODBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ขาย (Vendor)	ชนิด	สนับสนุนฐานข้อมูล
Trifox, Inc.	3	ADABAS, DB2, Informix, Ingres, Oracle, SQL Server, Rdb, Sybase and legacy systems via GENESIS
Visigenic	3	Several dozen through ODBC drivers.*
Weblogic	2	Oracle, Sybase, MS SQL Server
Weblogic	3	Several dozen through ODBC drivers.*
XDB System, Inc.	1 / 3	Many databases through ODBC.*
Yard Software GmbH	4	YARD-SQL Database

ตารางที่ 4-1 รายชื่อ ผู้ผลิตซอฟต์แวร์สำหรับเจดีบีซี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบและการสร้าง

ในการออกแบบและการสร้างโฮมเพจของสำนักทะเบียนและประมวลผลกลาง จะประกอบไปด้วยขั้นตอนต่างๆ ซึ่งจะทำการแยกออกเป็นหัวข้อหลักๆ 2 หัวข้อคือ การออกแบบ และ การสร้าง

5.1 การออกแบบโฮมเพจ

5.1.1 การกำหนดขอบเขตการทำงาน

ในการออกแบบโฮมเพจนี้จะเริ่มการกำหนดขอบเขตการทำงาน โดยขั้นแรกจะทำการศึกษาและสอบถามความต้องการของนักศึกษา, อาจารย์ และ เจ้าหน้าที่ที่เกี่ยวข้อง ว่าต้องการให้มีการบริการแบบใดบ้างในโฮมเพจซึ่งความต้องการเหล่านี้จะถูกนำมาคัดเลือกและตรวจสอบถึงความเหมาะสม ความเป็นไปได้ต่างๆ เช่น ในการตรวจสอบเกรด ซึ่งจะต้องทำเรื่องของอนุญาติในการตัดออกตารางเกรดของนักศึกษา รวมถึงการให้พิจารณาถึงความเหมาะสมและความปลอดภัยในการนำข้อมูลมาเสนอ โดยข้อสรุปของการให้บริการในโฮมเพจ จะประกอบด้วยบริการดังต่อไปนี้

1. การให้บริการรายชื่อนักศึกษา

1.1. รายชื่อนักศึกษาแบ่งตามหลักสูตร, ภาควิชา, ชั้นปี, ภาค และ ห้อง พร้อมทั้งสามารถทำการจัดเก็บลงบนเครื่องของผู้ใช้ได้

1.2. รายชื่อนักศึกษาแบ่งตามวิชาที่นักศึกษาแต่ละคนได้ลงทะเบียนเรียนไว้ พร้อมทั้งสามารถทำการจัดเก็บลงบนเครื่องของผู้ใช้ได้

2. การให้บริการตรวจสอบการลงทะเบียน (เฉพาะ ในเทอมที่เรียนอยู่)

3. การให้บริการตรวจสอบผลการเรียนทั้งหมดทุกเทอม ตั้งแต่เริ่มเข้าศึกษา (สำหรับการให้บริการนี้ไม่สามารถใช้งานได้จริง เนื่องจากไม่มีข้อมูลเกรดของนักศึกษาจริงๆ เป็นเพียงข้อมูลที่สมมุติขึ้นมาเพื่อให้เห็นแนวทางในการพัฒนาเท่านั้น เนื่องมาจากต้องทำเรื่องขออย่างเป็นทางการ)

4. การให้บริการบอร์ดประกาศข่าว

4.1. บอร์ดประกาศข่าวถึงนักศึกษา จากอาจารย์และจากทางสำนักทะเบียน โดยเป็นข่าวสารที่จะประกาศให้นักศึกษาทั่วไปรับทราบ (ข่าวสารที่มาจากทางสำนักทะเบียนจะคงอยู่เป็นเวลา 30 วัน ส่วนข่าวสารที่มาจากอาจารย์จะคงอยู่เป็นเวลา 7 วัน)

4.2. บอร์ดประกาศข่าวระหว่างอาจารย์และทางสำนักทะเบียน (เป็นบอร์ดข่าวที่ใช้คิดต่อระหว่างอาจารย์ และ สำนักทะเบียน ข่าวสารบนบอร์ดนี้จะคงอยู่เป็นเวลา 7 วัน สำหรับบอร์ดนี้จะมีรหัสผ่านเพื่อป้องกันนักศึกษาหรือผู้ไม่เกี่ยวข้องเข้ามาดู)

4.3. การส่งข่าวขึ้นบอร์ด มีไว้ให้สำหรับ นักศึกษา, อาจารย์ และ เจ้าหน้าที่ ส่งข่าวที่ต้องการประกาศไปยังบอร์ดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การให้บริการปฏิทินประจำปี (ไม่มีส่วนของการเขียนโปรแกรม)
6. การให้บริการด้านใบเกรด (ไม่มีส่วนของการเขียนโปรแกรม)

5.1.2 ทำการศึกษาฐานข้อมูลเดิม

สำหรับฐานข้อมูลของสำนักทะเบียนและประมวลผลกลางนี้ ทางสำนักวิจัยเป็นผู้ดูแลอยู่ และมีโครงการที่จะโอนให้กับสำนักทะเบียนและประมวลผลกลางในอีกไม่ช้านี้ โดยฐานข้อมูลที่ใช้อยู่เป็นฐานข้อมูล FoxPro 2.0 จากการตรวจสอบพบว่าฐานข้อมูลมีความซับซ้อน เนื่องจากได้มีการพัฒนามาหลายครั้ง ประกอบกับไม่มีเอกสารอ้างอิงต่าง ทำให้ยากต่อการเข้าใจว่าแต่ละตารางแต่ละเรคคอร์ดใช้เก็บข้อมูลอะไร มีความสัมพันธ์กันอย่างไร ทำให้การศึกษาต้องใช้เวลาในการคัดเลือกตารางที่ต้องการ รวมถึงตารางที่เกี่ยวข้องทั้งหมด

5.1.3 ศึกษาและเลือกฐานข้อมูลใหม่ที่จะใช้

หลังจากได้ทราบถึงชนิด และจำนวนข้อมูลจากฐานข้อมูลเดิมแล้วก็ทำการศึกษา และเลือกฐานข้อมูลใหม่ที่จะใช้ในโครงการ เนื่องจากได้มีโครงการที่จะย้ายฐานข้อมูลใหม่ทำให้มีแนวความคิดที่จะเลือกให้ฐานข้อมูลชนิดเดียวกับของสำนักทะเบียน แต่เนื่องจากทางสำนักทะเบียนยังไม่ได้มีการระบุถึงชนิดของฐานข้อมูลที่จะใช้ ดังนั้นจึงได้เลือกฐานข้อมูลเป็นไมโครซอฟแอคเซส (Microsoft Access) เนื่องจากง่ายต่อการ ใช้ การศึกษาทำได้ง่าย สามารถแปลงฐานข้อมูลจาก FoxPro มาอยู่ในรูปของไมโครซอฟแอคเซส ได้ง่ายไม่ยุ่งยาก

5.1.4 ออกแบบหน้าจอ

การออกแบบหน้าจอ (User Interface) ในที่นี้ รวมทั้งหน้าจออินพุต และ หน้าจอเอาต์พุต การออกแบบหน้าจอนี้มีความสำคัญเป็นอย่างมาก จำเป็นที่จะต้องสามารถสื่อสารกับผู้ใช้ให้เข้าใจ และผู้ใช้สามารถเข้าใจและปฏิบัติตามได้ง่าย (User friendly) รูปแบบหน้าจอดึงดูดให้ผู้ใช้งานเกิดความรู้อยากใช้ ไม่ซับซ้อน มีคำแนะนำให้ผู้ใช้งานที่จำเป็น ไม่ตกแต่มาจนทำให้ดูรก การออกแบบหน้าจอนี้จะต้องออกแบบในแนวทางเดียวกัน และต้องออกแบบสำหรับทุกๆฟังก์ชันการทำงาน รูปที่ 5-1 แสดงตัวอย่างหน้าจอที่ออกแบบเรียบร้อยแล้ว

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG FACULTY OF ENGINEERING				
ID: 20012517		NAME: น.ส. อรุณีภา ธรรมสิงห์		
ร.ร.	วิชา/วิชา	ชื่อวิชา	หน่วยกิต	เกรด
1	01070001	Mathematics for Computer Engineering	3	B+
2	01072105	Microprocessor Interfacing	3	A
3	01073101	Computer Project I	2	A
4	01073103	Computer Laboratory II	2	A
5	01073105	Information System Analysis & Design	3	B+
6	01073107	Communication Engineering	3	A
7	03150003	Anthropology	2	B+
			SEMESTER	19.0 3.77
CUMULATED		CG	GPA	GPA
100.0		285.0	3.88	

รูปที่ 5-1 ตัวอย่างการออกแบบหน้าจอ

หลังจากออกแบบหน้าจอครบหมดแล้ว ให้ทำการจัดเรียงลำดับแต่ละหน้าจอเข้าด้วยกันตามความเหมาะสม

5.1.5 เลือกรูปแบบการติดต่อกับฐานข้อมูลและโปรแกรมช่วยอื่นๆ

ในการใช้ เจดีบีซี ติดต่อกับฐานข้อมูลนั้นสามารถเลือกใช้ได้ 4 แบบ ดังที่ได้กล่าวไว้ในตอนที่ 3.2 ซึ่งจำเป็นต้องคำนึงถึงข้อจำกัดและความเหมาะสมในด้านต่างๆ ซึ่งในที่นี้ได้เลือก ชนิด-3 เนื่องจากประสิทธิภาพ ความสะดวกในการเขียนโปรแกรม และการใช้ โอดีบีซี ซึ่งทำให้สามารถปรับเปลี่ยนการใช้ฐานข้อมูลได้ง่าย ยิ่งในสถานะกรณีที่ยังไม่แน่ว่าทางสำนักทะเบียนจะใช้ฐานข้อมูลใด

จากนั้นก็เลือกดูจากรายชื่อ vender และ software จากตารางของ เจดีบีซี ไครฟ์เวอร์ ในตอนที่ 3.2 ให้ตรงกับความต้องการมากที่สุด นอกจากนี้ยังต้องพิจารณาถึงคุณสมบัติ ประสิทธิภาพ ราคา รวมทั้งท้ายสุดจึงได้เลือกใช้ IDS Server ของบริษัท IDS Software (www.idsoftware.com) เป็นซอฟต์แวร์ที่จะใช้ในการพัฒนาเนื่องจากมีคุณสมบัติตรงตามต้องการ รวมถึงข้อเสนอของทางบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การสร้าง

5.2.1 เขียนโปรแกรมในส่วนของ แอปพลิเคชัน

โดยเริ่มจากจากเขียนในส่วนของจียูไอ (GUI :Graphic User Interface) ซึ่งเป็นส่วนที่จะใช้ในการติดต่อกับผู้ใช้ จากนั้นทำการเขียนในส่วนที่เป็นการประมวลผล ซึ่งอาจเป็นการประมวลผลบนข้อมูลที่ได้รับจากผู้ใช้ผ่านทางจียูไอซึ่งการประมวลผลนี้จะขึ้นอยู่กับเหตุการณ์ (event) ที่เกิดขึ้นกับแอปพลิเคชัน ซึ่งในที่นี้รวมถึงการเขียนโค้ดในส่วนที่ติดต่อกับฐานข้อมูล การส่ง คำสั่งเอสคิวแอล ไปยังฐานข้อมูล การรับผลที่ได้จาก คำสั่งเอสคิวแอล และการประมวลผลและการแสดงผลที่ได้จากฐานข้อมูลออกทาง แอปพลิเคชัน

ทำการเขียนแอปพลิเคชัน และทดสอบการทำงานบน Development Tool จนเป็นที่น่าพอใจ จากนั้นก็ทำการเขียน แอปพลิเคชัน ตัวอื่นๆทั้งหมดที่ต้องใช้ในโฮมเพจ

5.2.2 เขียนโปรแกรมในส่วนของโฮมเพจด้วย เอชทีเอ็มแอล

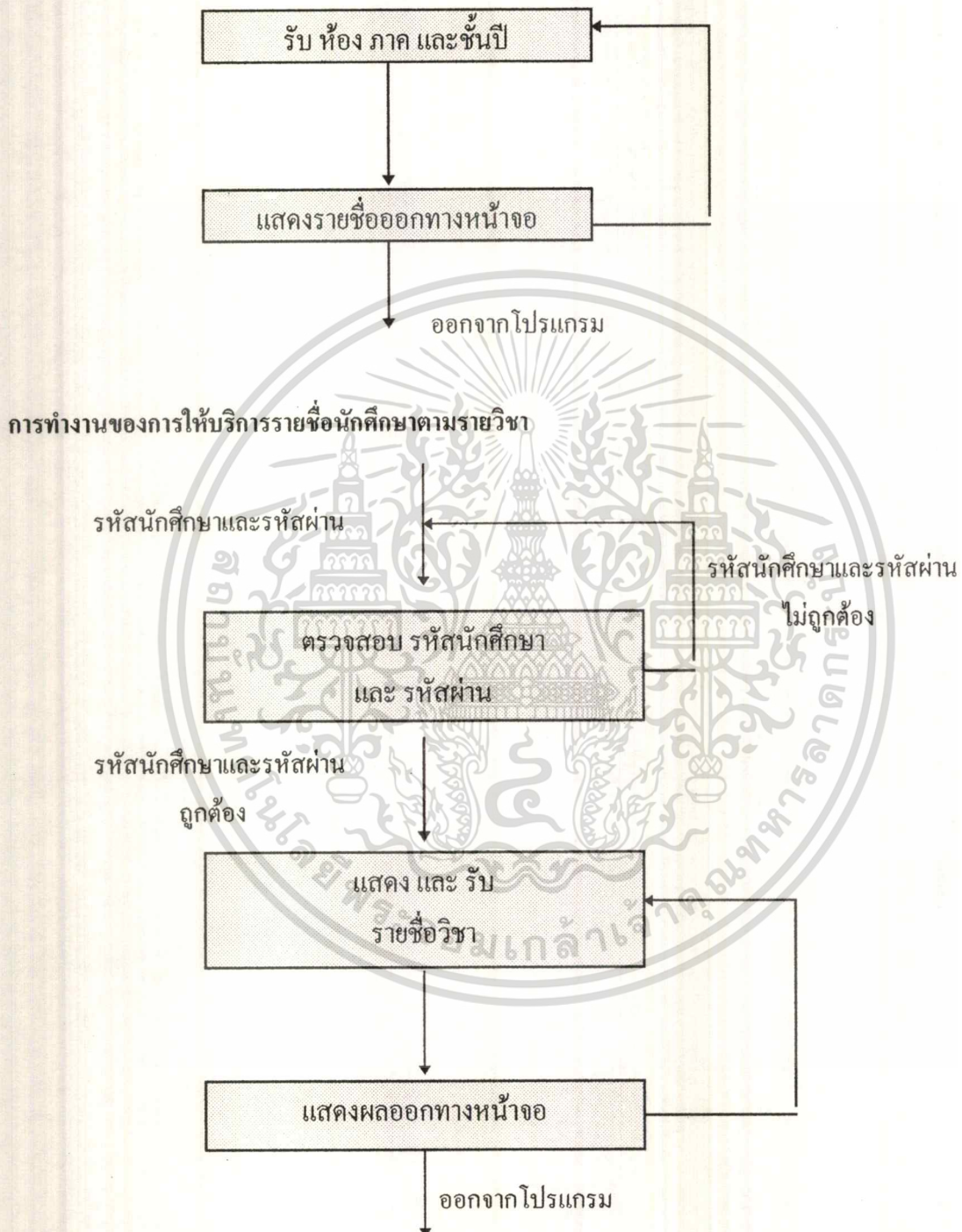
เมื่อทำการเขียน แอปพลิเคชัน เสร็จเรียบร้อยแล้วก็ทำการเขียนโฮมเพจตามที่ได้ออกแบบไว้ จากนั้นนำเอา แอปพลิเคชัน ที่ได้เขียนไว้มารวมเข้าไว้ในเว็บเพจ (web page) แต่ละหน้า (วิธีตามหัวข้อ 2.6.6) นำเอาเว็บเพจ แต่ละหน้าที่ได้มาเชื่อมต่อกันเป็นโฮมเพจที่สมบูรณ์ และตรวจสอบการทำงาน การเรียงลำดับ ให้เรียบร้อย

5.2.3 ตรวจสอบการทำงานและแก้ไข

หลังจากที่ได้โฮมเพจที่เสร็จเรียบร้อยแล้ว ให้ให้นำเอาโฮมเพจไปทดลองใช้ในสภาพแวดล้อมจริง เพื่อดูการทำงานและประสิทธิภาพ จากนั้นประเมินผลและแก้ไขในส่วนที่บกพร่องจนกว่าจะได้โฮมเพจที่ทำงานได้สมบูรณ์เรียบร้อย

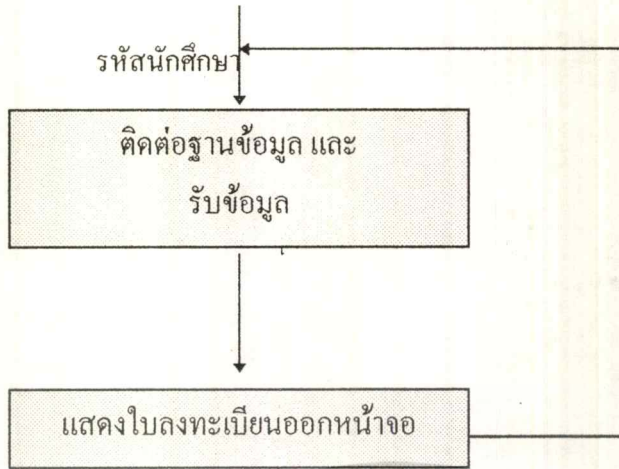
ไต่ถามกรรมการออกแบบโปรแกรม

การทำงานของกรให้บริการรายชื่อนักศึกษาตามห้อง ภาค ชั้นปี



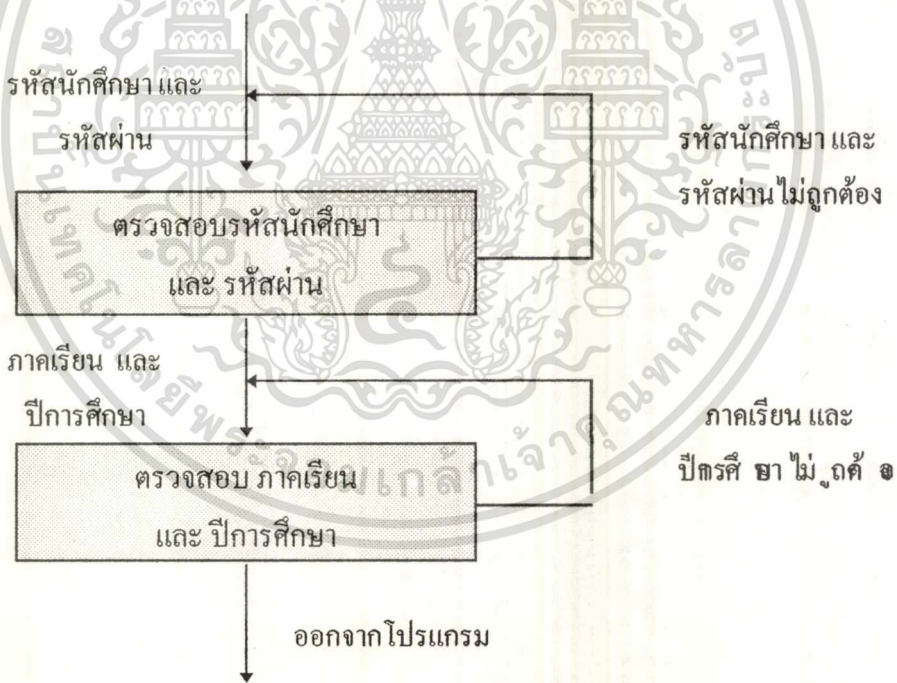
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานการตรวจสอบการลงทะเบียน



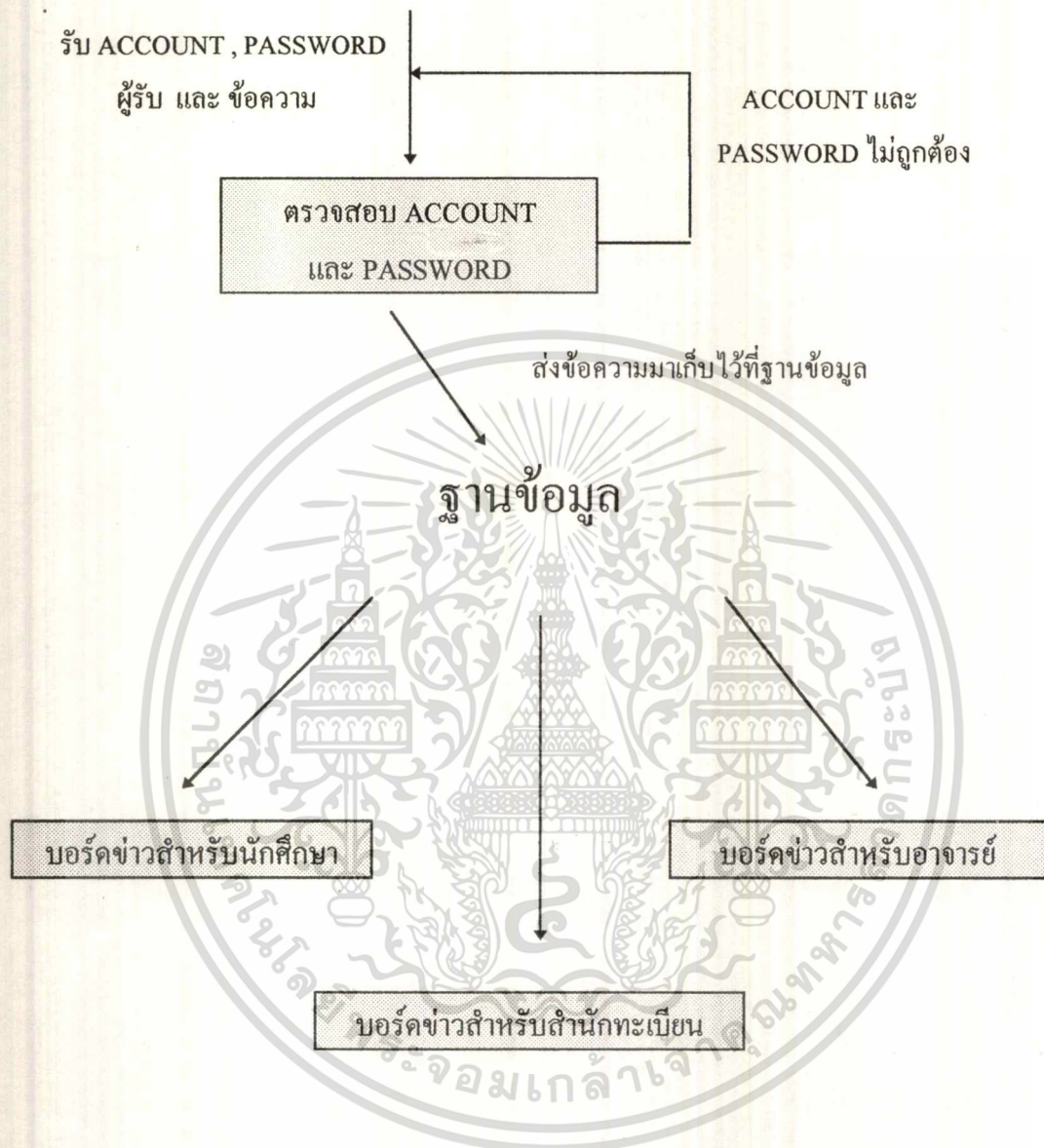
ออกจากโปรแกรม

การให้บริการการตรวจสอบผลการเรียน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การให้บริการบอร์ดประกาศข่าว

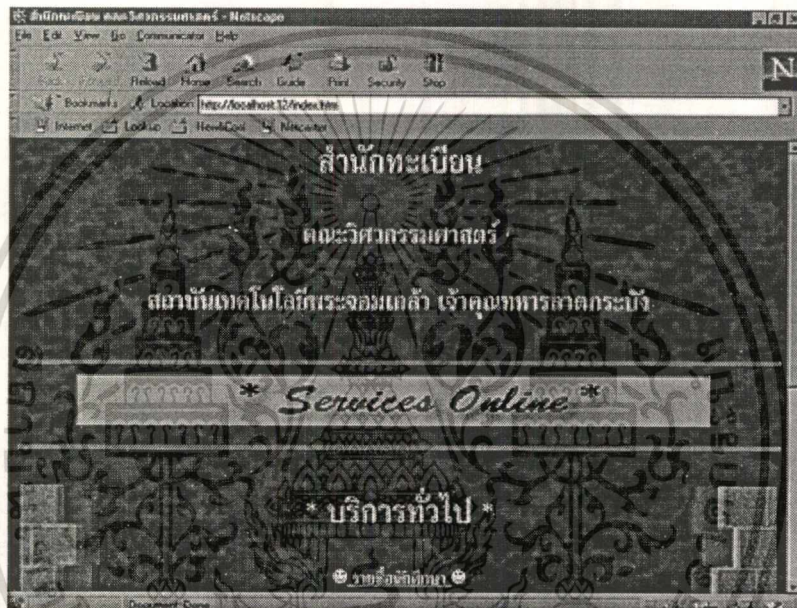


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

ผลงาน

สำหรับผลงานในโครงการนี้ก็คือ โสมเพจของสำนักทะเบียน คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งจะสามารถเข้าใช้งานได้ที่ 161.246.6.95 ซึ่งรูปร่างหน้าตาของโสมเพจในหน้าแรกจะเป็นดังรูปที่ 6-1



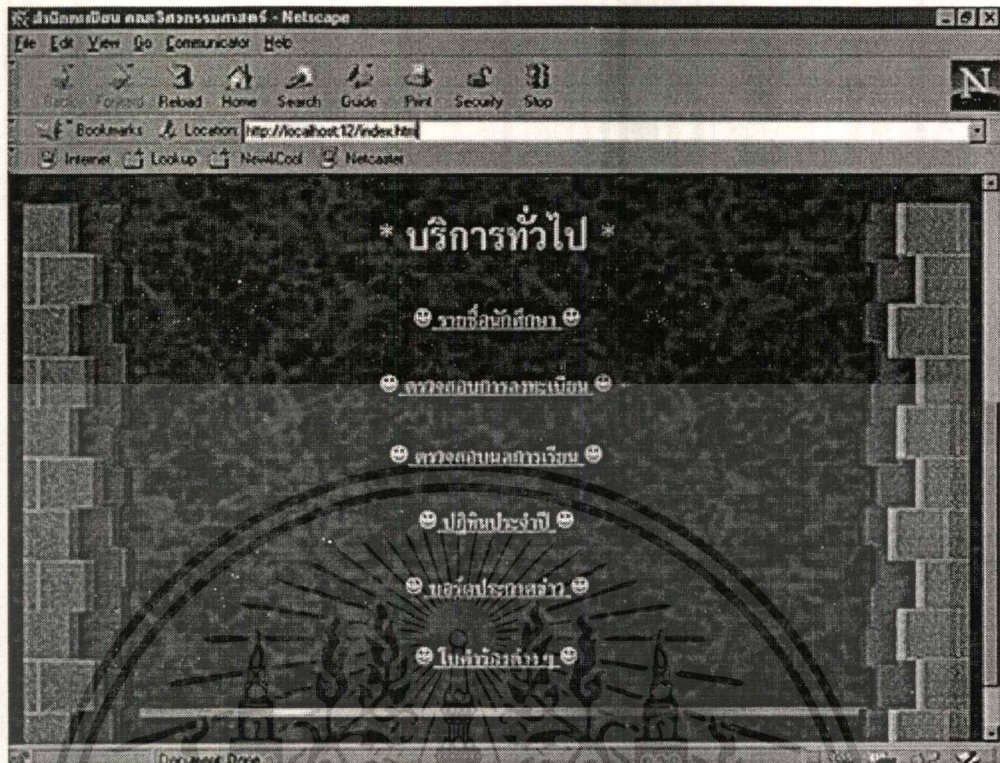
รูปที่ 6-1 หน้าจอแรกของโสมเพจสำนักทะเบียน

ในหน้าแรกของโสมเพจนี้จะประกอบไปด้วยรายการของการให้บริการที่ทางสำนักทะเบียนจัดไว้ให้ ซึ่งประกอบไปด้วยบริการหลักๆ 6 ประการด้วยกันคือ

1. การให้บริการรายชื่อนักศึกษา ซึ่งแบ่งออกเป็น 2 แบบ คือ
 - 1.1 รายชื่อนักศึกษาที่แบ่งตามหลักสูตร ห้อง ภาค ชั้นปี
 - 1.2 รายชื่อนักศึกษาที่แบ่งตามรายวิชาที่ลงทะเบียนไว้
2. การให้บริการตรวจสอบการลงทะเบียน (เฉพาะในเทอมนั้นๆ)
3. การให้บริการตรวจสอบผลการเรียน (สามารถตรวจสอบได้ทุกเทอมที่ผ่านมา)
4. การให้บริการบอर्डประกาศข่าว
5. การให้บริการปฏิทินประจำปี
6. การให้บริการใบคำร้องต่างๆ

สำหรับหน้าจอแสดงรายการให้บริการแสดงดังรูป 6-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-2 หน้าจอแสดงรายการที่ให้บริการ

สำหรับหรับบริการ 2 อย่างสุดท้ายคือ การให้บริการปฏิทินประจำปี กับ การให้บริการใบคำร้อง จะไม่มีส่วนของโปรแกรมอยู่ ส่วนบริการ 4 อย่างแรกจะประกอบด้วยโปรแกรมที่เขียนด้วยจาวา ซึ่งจะอยู่ในรูปของ แอปเพล็ต เพื่อคอยให้บริการอยู่ ซึ่งจะมีรายละเอียดดังต่อไปนี้

1. การให้บริการรายชื่อนักศึกษา

1.1 รายชื่อนักศึกษาตาม หลักสูตร ภาค ชั้นปี ห้อง ซึ่งรายชื่อลักษณะนี้ตามปรกติทางสำนักทะเบียนจะมีให้อยู่ชั้นหน้าสำนักทะเบียน ชั้น 5 ตึก 12 ผู้ที่ต้องการสามารถไปเอารายชื่อที่ต้องการได้ สำหรับในโครงการนี้ก็จะมีลักษณะเช่นเดียวกัน โดยผู้ใช้จะต้องเลือก หลักสูตร ภาค ห้อง ชั้นปีที่ต้องการจากเมนูที่มีไว้ให้ จากนั้นก็กดปุ่มเพื่อยืนยันการทำงานรายชื่อก็จะปรากฏที่หน้าจอของแอปเพล็ต จากนั้นผู้ให้จะสามารถเลือกได้ว่าจะทำการจัดเก็บรายชื่อดังกล่าวเพื่อนำไปใช้ต่อไปหรือไม่ โดยการคลิกปุ่ม SAVE (สำหรับการทำการงานนี้จะทำได้เฉพาะในเน็ตสแต็บคอมมิวนิตีเตอร์ เท่านั้น)

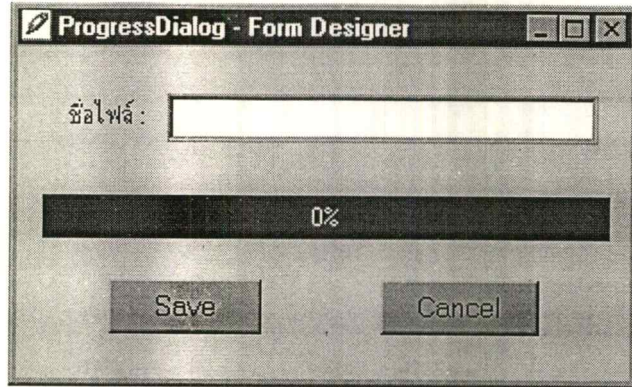
หน้าจอจะปรากฏ dialog เพื่อให้ผู้ใช้พิจารณาว่าจะอนุญาตให้มีการจัดเก็บไฟล์ลงเครื่องโคลนเอ็นท์หรือไม่ ถ้าผู้ใช้อนุญาตโดยการกดปุ่ม Grant ก็จะมีปรากฏหน้าจอ dialog ดังรูป 6-3 ซึ่งผู้ใช้จะต้องกรอกชื่อพร้อมทั้งไครเรททอรี่ที่ต้องการ เพื่อจัดเก็บข้อมูล

ข้อมูลที่จัดเก็บนี้จะอยู่ในรูปของเท็กซ์ไฟล์ (text file) ซึ่งผู้ใช้สามารถจะนำไปแปลงเป็นไฟล์-

ไมโครซอฟเอ็กเซล(Microsoft Excel) หรือ ไฟล์อื่นที่ต้องการเพื่อนำไปใช้งานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-3 dialog แสดงหน้าจอการจัดเก็บไฟล์

1.2 รายชื่อนักศึกษาตามวิชาที่ใดลงเรียน สำหรับรายชื่อนี้ เฉพาะอาจารย์ที่สอนแต่ละวิชาเท่านั้นที่จะสามารถเอารายชื่อนักศึกษาที่ลงเรียนวิชานั้นได้ โดยอาจารย์จะต้องกรอกรหัส (account) และรหัสลับ (password) ของตนลงใน dialog จากนั้น แอปพลิเคชัน จะส่งข้อมูลกลับไปเพื่อค้นหารายชื่อวิชาที่อาจารย์ท่านนั้นๆสอน แล้วจะทำการแสดงออกทางหน้าจอเพื่อให้อาจารย์เลือกว่าจะเอารายชื่อนักศึกษาวิชาอะไร ดังรูป 6-4 จากนั้นขั้นตอนอื่นๆก็จะเหมือนกับในรายชื่อที่แยกตาม 1.1 ซึ่งก็รวมถึงการจัดเก็บไฟล์เช่นกัน

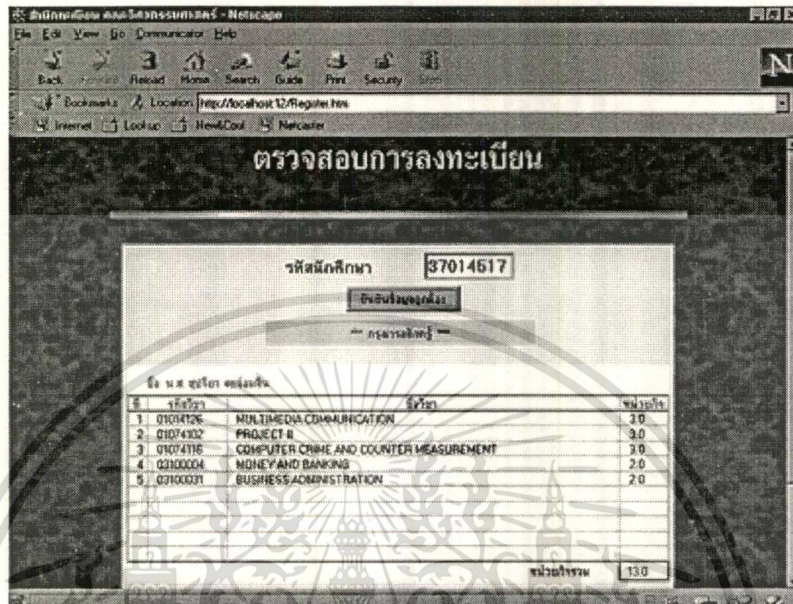
ลำดับ	ชื่อผู้ลงทะเบียน	ชื่อ-นามสกุล
1	39014001	น.ส.ศุภมาส อิ่มละมัย
2	39014006	นางสาว ศุภมาส ไชยวงษ์
3	39014010	น.ส.ศุภมาส สุทธิวงษ์
4	39014014	นางสาวศุภมาส อังศรี
5	39014031	น.ส.ศุภมาส วิมลประเสริฐ
6	39014033	น.ส.ศุภมาส วิมลประเสริฐ
7	39014043	นายเกียรติยศ วิมลประเสริฐ
8	39014045	นายเกียรติยศ วิมลประเสริฐ
9	39014052	นายเกียรติยศ วิมลประเสริฐ
10	39014056	นายเกียรติยศ วิมลประเสริฐ
11	39014057	นายเกียรติยศ วิมลประเสริฐ
12	39014058	นายเกียรติยศ วิมลประเสริฐ
13	39014077	นายเกียรติยศ วิมลประเสริฐ
14	39014094	นายเกียรติยศ วิมลประเสริฐ
15	39014099	นายเกียรติยศ วิมลประเสริฐ
16	39014102	นายเกียรติยศ วิมลประเสริฐ
17	39014130	น.ส.ศุภมาส วิมลประเสริฐ
18	39014132	นายเกียรติยศ วิมลประเสริฐ
19	39014133	นายเกียรติยศ วิมลประเสริฐ
20	39014136	น.ส.ศุภมาส วิมลประเสริฐ
21	39014120	นายเกียรติยศ วิมลประเสริฐ
22	39014131	นายเกียรติยศ วิมลประเสริฐ
23	39014132	นายเกียรติยศ วิมลประเสริฐ
24	39014134	น.ส.ศุภมาส วิมลประเสริฐ
25	39014146	นายเกียรติยศ วิมลประเสริฐ
26	39014148	น.ส.ศุภมาส วิมลประเสริฐ
27	39014200	นายเกียรติยศ วิมลประเสริฐ
28	39014303	นายเกียรติยศ วิมลประเสริฐ
29	39014311	นายเกียรติยศ วิมลประเสริฐ
30	39014312	นายเกียรติยศ วิมลประเสริฐ

รูปที่ 6-4 dialog แสดงรายชื่อผู้ลงทะเบียนตามรายวิชา

2. การให้บริการตรวจสอบการลงทะเบียน การให้บริการนี้จะเหมือนกับที่ทางสำนักทะเบียนจัดให้มีการตรวจสอบได้จากที่ห้องสมุด หรือที่อื่นๆที่จัดไว้ให้ สำหรับการให้บริการให้เลือกหัวข้อบริการการตรวจสอบการลงทะเบียน จากนั้นจะปรากฏหน้าจอ ดังรูป 6-5 ซึ่งจะคล้ายกับในลงทะเบียนที่ใช้ใน

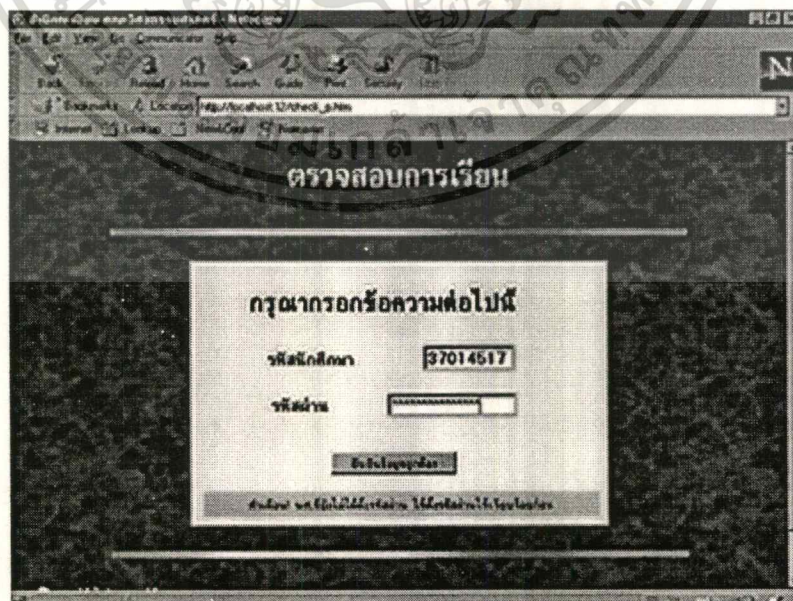
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนลงทะเบียน ซึ่งจะทำให้ผู้ใช้สามารถเข้าใจถึงรายละเอียดที่แสดงได้ง่าย ผู้ใช้เพียงแค่กรอกรหัสนักศึกษา ยาลงไปในช่องที่กำหนดไว้ แล้วกดปุ่มเพื่อยืนยัน จากนั้นหน้าจอจะปรากฏผลดังแสดงในรูป 6-5 เช่นกัน



รูปที่ 6-5 แสดงหน้าจอการตรวจสอบการลงทะเบียน

3. การให้บริการตรวจสอบผลการเรียน สำหรับการตรวจสอบผลการเรียนนั้นจะต้องมีการตั้งรหัสลับ เพื่อป้องกันผู้อื่นมาแอบดูผลการเรียนของคนอื่นๆ ดังรูป 6-6



รูปที่ 6-6 แสดงหน้าจอการเข้าใช้บริการตรวจสอบผลการเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการแจกรหัสลับจำสามารถทำได้ 2 วิธี คือ

1. ให้ผู้ใช้ทำการตั้งในครั้งแรกที่เข้ามา แต่วิธีนี้มีข้อเสียคือผู้ที่เข้ามาก่อนสามารถมาตั้งรหัสลับของคนอื่นได้

2. ทางสำนักทะเบียนตั้งรหัสลับให้ก่อน แล้วแจกให้กับนักศึกษา นักศึกษาที่เข้ามาใช้งานทำการเปลี่ยนรหัสลับในภายหลังเอง ซึ่งจะมี dialog สำหรับผู้ใช้ในการเปลี่ยนรหัสผ่านได้

เมื่อผู้ใช้กรอกรหัสนักศึกษาและรหัสลับเรียบร้อยแล้ว จะปรากฏ dialog ให้ผู้ใช้เลือกภาคการศึกษาและปีการศึกษาที่ต้องการตรวจสอบผลการเรียน กดปุ่มเพื่อยืนยัน จากนั้นหน้าจอจะแสดงผลคล้ายกับใบเกรดที่นักศึกษาได้รับ ซึ่งจะเป็นการเข้าใจได้ง่าย ดังรูป 6-7

ที่	รหัสวิชา	ชื่อวิชา	หน่วยกิต	เกรด
1	01070001	Mathmatics for Computer Engineering	3	B+
2	01072105	Microprocessor Interfacing	3	A
3	01073101	Computer Project I	2	A
4	01073103	Computer Laboratory II	2	A
5	01073106	Information System Analysis & Design	3	B+
6	01073107	Communication Engineering	3	A
7	03150063	Anthropology	2	B+
			SEMESTER	18.0
			3.77	
			COMULATION	100.0
			GP	906.0
			GPA	3.06

รูปที่ 6-7 แสดงหน้าจอผลการเรียน

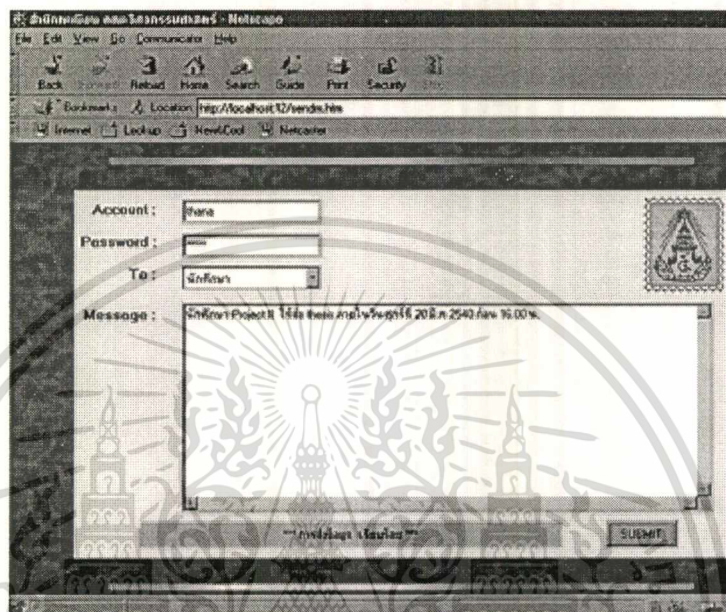
4. การให้บริการบอร์ดฝากข่าว ซึ่งแบ่งออกเป็น 3 ส่วนคือ

1. เป็นบอร์ดข่าวสำหรับนักศึกษา ซึ่งบอร์ดนี้จะแบ่งออกเป็นบอร์ดข่าวที่ส่งมาจากอาจารย์ (ข่าวจะอยู่บนบอร์ด 7 วัน) และบอร์ดข่าวที่มาจากสำนักทะเบียน (ข่าวจะอยู่บนบอร์ด 30 วัน) ทั้ง 2 บอร์ดนี้สามารถดูได้โดยไม่ต้องมีรหัสผ่านแต่อย่างใด

2. เป็นบอร์ดสำหรับอาจารย์และเจ้าหน้าที่ใช้ในการติดต่อกัน บอร์ดนี้จะมีรหัสลับ อยู่ 2 ตัวคือ รหัสลับของสำนักทะเบียน และรหัสลับของอาจารย์ (อาจารย์ทั้งหมดใช้รหัสลับตัวเดียวกัน) ข่าวบนบอร์ดนี้จะคงอยู่เป็นเวลา 7 วัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เป็นการฝากข่าวเพื่อแสดงบนบอร์ด โดยในการฝากข้อความนี้ ผู้ใช้จะต้องกรอกว่าจะส่งถึงใคร ซึ่งมีด้วยกัน 3 ฝ่ายคือ นักศึกษา อาจารย์ และสำนักทะเบียน จากนั้นจะเป็นรหัสลับโดยทางสำนักทะเบียนและอาจารย์แต่ละคนจะมีรหัสลับของตน (ในที่นี้จะป็นรหัสลับตัวเดียวกับที่ใช้ในการดูรายชื่อนักศึกษา เพื่อสะดวกในการจำ) จากนั้นก็กรอกข้อความลงในพื้นที่ที่กำหนด ดังรูป 6-8 โดยไม่ต้องระบุชื่อผู้ส่ง เนื่องจากชื่อผู้ส่งจะถูกเพิ่มโดยอัตโนมัติจากรหัสลับที่กรอกลงไป



รูปที่ 6-8 แสดงหน้าจอที่ใช้ในการฝากข่าว

เมื่อกรอกข้อความเรียบร้อยแล้วให้คลิกปุ่ม SUBMIT เพียงครั้งเดียว เพื่อส่งข้อมูลไปเก็บไว้ในฐานข้อมูลเพื่อแสดงบนบอร์ดที่เหมาะสมต่อไป

บทที่ 7

สรุปผลและวิเคราะห์

7.1 ปัญหาและแนวทางแก้ไข

1. สถานข้อมูล

ปัญหา

เนื่องจากฐานข้อมูลเดิมที่ใช้อยู่ ทางสำนักวิจัยเป็นผู้ดูแลอยู่ในขณะนี้ แต่จะได้มีการโอนมาเป็นของทางสำนักทะเบียนและประมวลผลกลางในอีกไม่ช้านี้ และการโอนย้ายอีกยังไม่แน่ว่าจะมีการเปลี่ยนจากฐานข้อมูลเดิมซึ่งเป็นฟ็อกซ์โปร (FoxPro) ไปเป็นฐานข้อมูลอื่นหรือไม่ เนื่องจากหลายฝ่ายเห็นว่าควรจะได้มีการเปลี่ยนแปลงฐานข้อมูลให้มีความทันสมัยขึ้น และจะได้เป็นการจัดระเบียบโครงสร้างของฐานข้อมูลเสียใหม่ เนื่องจากฐานข้อมูลในปัจจุบันได้มีการพัฒนามาหลายครั้งหลายหน และในการพัฒนาแต่ละครั้งก็ไม่ได้มีการจัดทำเอกสารอ้างอิง หรือเอกสารที่ใช้ในการอธิบายถึงโครงสร้างของฐานข้อมูลเดิมเลย จึงเป็นการยากในการศึกษาว่าข้อมูลใดเก็บอยู่ในตารางใด ข้อมูลแต่ละตัวหรือแต่ละตารางมีความสัมพันธ์กันอย่างไร

นอกจากปัญหาเรื่องความยุ่งยากของฐานข้อมูลและขาดเอกสารอ้างอิงแล้ว ฐานข้อมูลเดิมยังไม่เหมาะที่จะนำมาให้เพื่อเป็นฐานข้อมูลในเว็บคาด้าเบส จึงจำเป็นต้องมีการแปลงข้อมูลจากฐานข้อมูลเดิมมาเป็นฐานข้อมูลใหม่ ในขั้นแรกคาดว่าจะเลือกฐานข้อมูลแบบเดียวกับที่ทางสำนักทะเบียนจะเลือกใช้ใหม่ แต่ทางสำนักทะเบียนยังไม่มีการตกลงว่าจะเลือกฐานข้อมูลชนิดใด จึงได้สังเกตการทำงานของเจ้าหน้าที่ทางสำนักทะเบียน พบว่าฐานข้อมูลที่ทางเจ้าหน้าที่ที่มีความชำนาญในการใช้งานมี 2 ชนิดด้วยกันคือ ไมโครซอฟแอสเซส และเอสคิวเอลเซิร์ฟเวอร์ (SQL Server) แต่เมื่อสอบถามแล้วเจ้าหน้าที่พอใจที่จะใช้ ไมโครซอฟแอสเซส มากกว่าเนื่องจากสามารถศึกษาได้ง่าย มีหนังสือแนะนำมาก หาได้ง่าย และที่สำคัญคือสามารถแปลงฐานข้อมูลจากฟ็อกซ์โปรมาเป็นไมโครซอฟแอสเซสได้ง่ายดังนั้นท้ายสุดจึงได้เลือกใช้ไมโครซอฟแอสเซส เป็นฐานข้อมูลทดสอบในโครงการนี้

วิธีแก้ไข และ คำแนะนำ

สำหรับเรื่องฐานข้อมูลขอแนะนำสำหรับความซับซ้อนยุ่งยาก ควรจะได้มีการจัดระเบียบและจัดทำเอกสารอ้างอิงให้เรียบร้อย แต่ทำได้ยากเนื่องจากพบว่าฐานข้อมูลมีข้อมูลจำนวนมาก และไม่มีเจ้าหน้าที่ที่ทราบรายละเอียดทั้งหมดของฐานข้อมูลที่มีอยู่ ส่วนมากจะรู้ในส่วนที่ตนต้องเกี่ยวข้องด้วยเท่านั้น ดังนั้นจึงเป็นการยากสำหรับบุคคลภายนอกที่ต้องการจะพัฒนาหรือใช้ประโยชน์จากฐานข้อมูลนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาเลือกใช้เจดีย์ซีไครฟ์เวอร์ ของบริษัทที่ให้คานวโนลคมาทคลงใช้ก่อน ซึ่งโดยมากก็จะมีข้อจำกัด เช่น dbANYWHERE ของ บริษัทซีเมนส์คคสามารถคานวโนลคมาใช้ได้ แต่ละ reset ตัวเองหลังจาก เข้าถึง ไปแล้ว 50 ครั้ง หรือ IDS Server ซึ่งจะต้องทำการต่ออายุทุกเดือน และ จำกัดการ เข้าถึง ได้แค่ 16 คน หลังจากที่ได้พิจารณาแล้วก็ได้เลือกใช้ IDS Server ของบริษัท IDS Software โดยได้ติดต่อไปทางบริษัทให้ช่วยยืดเวลาของการต่ออายุ ซึ่งทางบริษัทก็ได้อนุเคราะห์ให้ใช้งานได้โดยไม่ต้องต่ออายุ จนถึงวันที่ 1 มิ.ย. 41 เนื่องจาก IDS Server ใช้การติดต่อกับฐานข้อมูลโดยผ่าน โอดีบีซีไครฟ์เวอร์ บนเครื่องเซิร์ฟเวอร์ซึ่งจะมีความยืดหยุ่นในการเปลี่ยนแปลงชนิดของฐานข้อมูล นอกจากนี้ยังมีการ Encrypt ข้อมูลเพื่อความปลอดภัย และยังสามารถทำงานเป็นเอชทีทีพีเซิร์ฟเวอร์ (http server) ได้อีกด้วย ทำให้ง่ายในการพัฒนาบนเครื่องพีซีที่ไม่ได้ต่อเข้าระบบเครือข่าย

3. การ เข้าถึง local file ของ appet

ปัญหา

เนื่องจากในโครงการนี้ได้มีบริการในการแสดงรายชื่อนักศึกษา ทั้งตามหลักสูตร ห้อง ภาค และตามวิชาที่นักศึกษาได้ลงทะเบียนไว้ ดังนั้นจึงเห็นว่า ถ้าผู้ใช้จะสามารถจัดเก็บรายชื่อเหล่านี้ได้จะเป็นประโยชน์ยิ่งขึ้น เช่น อาจารย์สามารถนำไปใช้ในการกรอกคะแนน หรือ นักศึกษาสามารถนำไปใช้ในกิจกรรมต่างได้ แต่เนื่องจากในโครงการนี้โปรแกรมจาวาอยู่ในรูปของ แอปเพล็ต ซึ่งต้องถูกแสดงโดยบราวเซอร์ และ บราวเซอร์ ที่เป็นที่ยอมรับมากที่สุดก็แค่เน็ตสเคป (Netscape) แต่ทางเน็ตสเคป และบราวเซอร์ ส่วนใหญ่จะไม่อนุญาตให้ แอปเพล็ต เข้าถึงหรือจัดการกับ local file ได้ ซึ่งข้อกำหนดนี้ไม่ได้เป็นสำหรับทุก บราวเซอร์ เช่นใน แอปเพล็ต viewer ของ Sun แอปเพล็ต จะสามารถ เข้าถึง local file ได้

วิธีแก้ปัญหา และ คำแนะนำ

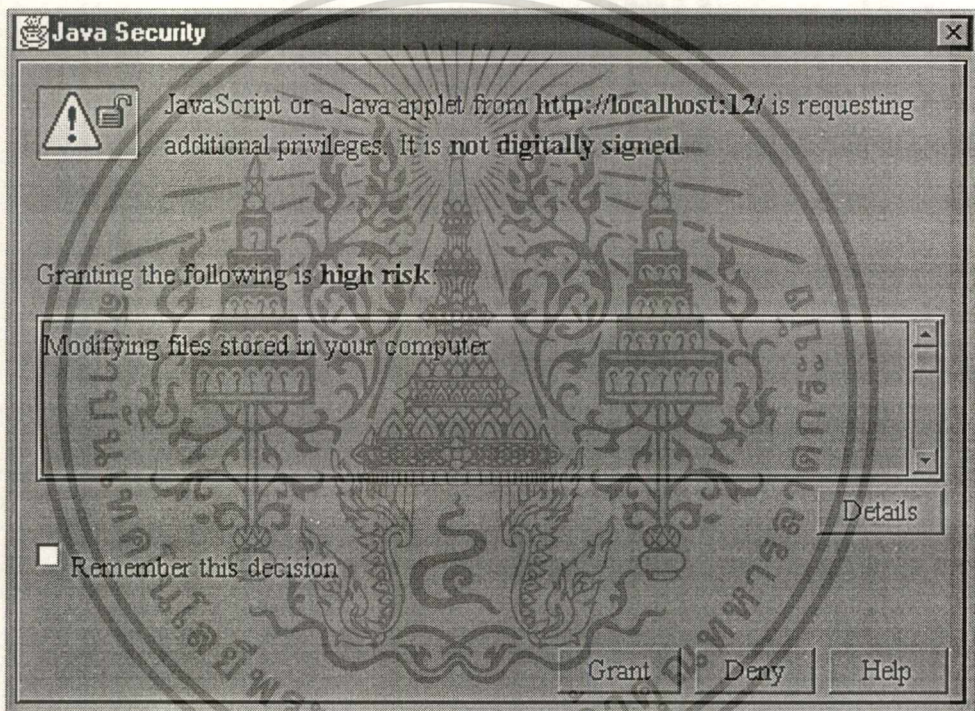
ในการแก้ปัญหานี้จะพิจารณาเฉพาะ บราวเซอร์ 2 ตัวคือ เน็ตสเคป และอินเทอร์เน็ตเอ็กซ์โพลเรอร์ (Internet Explorer) เท่านั้น จากการศึกษาคพบว่าวิธีการแก้ปัญหานี้ด้วยกันดังนี้

1. ไปทำการจ้างให้บริษัทที่ได้รับอนุญาตทำการออก Certificate ให้ ซึ่งโดยมากจะเสียค่าใช้จ่ายประมาณ 200-500ดอลลาร์สหรัฐซึ่งเป็นค่าใช้จ่ายที่สูง ดังนั้นวิธีนี้จึงเป็นวิธีที่ใช้กันมากในบริษัทขนาดใหญ่และเป็นที่ยอมรับ แต่วิธีนี้ก็สามารถจะทำให้แอปเพล็ตสามารถเข้าถึง local file ได้ทั้งในเน็ตสเคป และ ในอินเทอร์เน็ตเอ็กซ์โพลเรอร์
2. สำหรับใน Internet Explorer นอกจากจะใช้ certificate แล้วยังมีอีกวิธีคือ คลาสทั้งหมดของ แอปเพล็ตจะต้องถูกติดตั้งอยู่บนเครื่อง โคลนเอ็นท์ ที่ แอปเพล็ต ต้องการจะ เข้าถึง local file และต้องเป็นในไคเรกทอรีของ คลาไฟล์เท่านั้น ดังนั้นจึงต้องสร้าง link ให้ผู้ใช้สามารถคานวโนลคคลาไฟล์ และติดตั้งบนเครื่องได้ ซึ่งเป็นวิธีการที่ยุงยาก และไม่น่าทำ เนื่องจากในอินเทอร์เน็ตเอ็กซ์โพลเรอร์ ไม่ค่อยสนับสนุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของแอปพลิเคชันที่ควร และในอินเทอร์เน็ตเอ็กซ์พลอเรอร์ ก็ไม่สามารถแสดงตัวอักษรภาษาไทยได้

3. สำหรับในเน็ตสเค็บนอกจาก certificate แล้วยังมีอีกวิธี แต่วิธีนี้จะใช้ได้เฉพาะใน เน็ตสเค็บคอมมิวนิเคเตอร์ (Netscape Communicator) เป็นต้นไปเท่านั้น ซึ่งก็ทำได้โดยใช้ เอพีโอ ทางด้านความปลอดภัย ของทาง Netscape มาเขียนรวมในแอปพลิเคชัน ซึ่งจะทำได้สามารถแอปพลิเคชัน set security ตามที่จำเป็นได้ ซึ่งจะปรากฏเฟรมแสดงการ Grant ของ java security ให้ผู้ใช้ทราบและตัดสินใจก่อนดังรูปข้างล่าง ซึ่งวิธีนี้ถึงแม้ว่าจะจำกัดอยู่แค่ เน็ตสเค็บคอมมิวนิเคเตอร์ก็ตาม แต่ก็เป็นวิธีที่ดีเนื่องจากไม่ต้องเสียเวลา และไม่เสียค่าใช้จ่ายใดๆ และคาดว่าในอนาคตอันใกล้นี้ เน็ตสเค็บคอมมิวนิเคเตอร์จะเข้ามาแทน เน็ตสเค็บรุ่นเก่าๆ ในท้องตลาด



รูปที่ 7-1 dialog แสดงการ Grant java security ใน Netscape Communicator

7.2 สรุป

จากการศึกษาและทำโครงการมาทั้งหมด ทำให้เล็งเห็นถึงความสำคัญของเว็บคาด้าเบสในโลกปัจจุบันที่มีการแข่งขันกันทางเทคโนโลยีอย่างสูง ซึ่งข้อมูลข่าวสารเป็นสิ่งจำเป็น และได้เห็นถึงความสามารถการพัฒนาขีดความสามารถของจาวา ทำให้ตระหนักได้ว่า จาวาน่าจะเป็นภาษาที่เป็นที่นิยมและแพร่หลายต่อไปในอนาคต ส่วนที่สำคัญที่สุดก็เป็นในส่วนของการวิเคราะห์และการออกแบบเว็บคาด้าเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งต้องทำการวิเคราะห์รายละเอียดทั้งหมด ความเหมาะสม และเทคโนโลยีในขณะนั้น นอกจากนี้ สำหรับการทำเว็บบอร์ดที่ข้อมูลต้องการความปลอดภัยสูง ยังต้องมีความระมัดระวังในการออกแบบ มากยิ่งขึ้น ในการออกแบบ ปัจจัยรอบข้างจะมีส่วนสำคัญในการกำหนดคุณภาพของเว็บบอร์ด ตั้งแต่ ต้นทุน บุคลากร ความรู้ของเจ้าหน้าที่ที่เกี่ยวข้อง เพราะการทำเว็บบอร์ดจะต้องครอบคลุมการทำงาน อย่างใกล้ชิด ดังนั้นการตรวจสอบการทำงานจึงมีความจำเป็นอย่างมาก จึงต้องมีการทดสอบในสภาวะ การทำงานจริง จากนั้นก็วิเคราะห์ผลที่ได้จากการตอบสนองของผู้ใช้ ว่าพอใจมากน้อยแค่ไหน ต้องปรับปรุงอะไรอีกบ้าง จนกว่าจะเป็นที่พอใจของทุกฝ่าย

7.3 การพัฒนาต่อไปในอนาคต

โครงการนี้เป็นแนวทางให้การทำเว็บบอร์ด ซึ่งสามารถจะนำไปประยุกต์ใช้ในงานพัฒนาเว็บบอร์ดบนอินเทอร์เน็ตได้หลายๆด้าน เช่น การประกาศผลสอบเอ็นทรานส์บนอินเทอร์เน็ต, การลงทำเบียนผ่านเว็บ ซึ่งได้มีหลายที่ได้ทำไปบ้างแล้ว, การเผยแพร่ข่าวสารต่างๆ, รายงานตลาดหุ้น หรือการส่งข้อสินค้าทางเว็บ เป็นต้น ซึ่งจะทำให้ข้อมูลสามารถกระจายไปได้ทั่วโลก ไม่จำกัดอยู่ในวงแคบๆเหมือนในสมัยก่อน

ในส่วนของโฮมเพจของทางสำนักทะเบียนและประมวลผลกลางที่ได้ทำขึ้น สามารถนำไปใช้งานจริง จะมีก็ในส่วนของการตรวจสอบผลการเรียนที่ไม่ได้ใช้ฐานข้อมูลจริงทำให้มีแค่ข้อมูลที่มีสมมติขึ้นเท่านั้น ในส่วนอื่นเช่นรายชื่อนักศึกษาและการตรวจสอบการลงทะเบียนสามารถนำไปใช้งานได้จริง แต่ถ้าจะให้ใช้งานได้สมบูรณ์ควรมีการปรับปรุงฐานข้อมูลเดิมของทางสำนักทะเบียนให้มีระบบระเบียบมากกว่านี้เสียก่อน

สำหรับมิดเดิลแวร์(middleware)จะเป็นตัวที่ผู้พัฒนาสามารถที่จะเลือกใช้ได้ตามความเหมาะสม ซึ่งในอนาคตคงจะมีบริษัทผู้ผลิตมิดเดิลแวร์เหล่านี้เพิ่มขึ้น ซึ่งจะทำให้ราคาลดลงกว่านี้ เพราะปัจจุบันราคาของมิดเดิลแวร์ตัวนี้มีราคาค่อนข้างสูง เป็นอุปสรรคกับการศึกษาเพื่อพัฒนาแอปพลิเคชัน

ภาคผนวก ก.

ODBC

Open Database Connectivity

โอดีบีซี (ODBC) คืออะไร

โอดีบีซี หรือ Open Database Connectivity คือวิธีการติดต่อและเข้าถึงระบบฐานข้อมูลจากแอปพลิเคชันโดยใช้ภาษาเอสคิวเอล เป็นมาตรฐานในการเข้าถึงข้อมูล ความสามารถในการเชื่อมต่อแบบนี้ทำให้แอปพลิเคชันสามารถเข้าถึงฐานข้อมูลได้หลายรูปแบบ ซึ่งทำให้ผู้พัฒนาโปรแกรมสามารถพัฒนาโปรแกรมไปได้ โดยไม่ต้องทำการระบุชนิดของระบบจัดการฐานข้อมูล

เดิมการพัฒนาโปรแกรมประยุกต์ที่ใช้งานทางด้านฐานข้อมูลการจะเข้าใช้ฐานข้อมูลของโปรแกรมเหล่านี้จะทำโดยการเรียกใช้เอสคิวเอล แบบฝังตัว (Embedded SQL) ซึ่งในขณะนั้นก็ดูจะไปได้ดี เพราะว่าตัวโปรแกรมสามารถทำการเปลี่ยนรูปแบบของระบบ ไม่ว่าจะเป็นทางด้านฮาร์ดแวร์ หรือทางด้านซอฟต์แวร์ได้หลายรูปแบบ รวมทั้งระบบปฏิบัติการด้วย (โดยการคอมไพล์ใหม่ทุกครั้งที่มีการย้ายระบบ)

อย่างไรก็ตามในการพัฒนาโปรแกรมในระบบที่มีความแตกต่างกัน เช่นการเรียกใช้ข้อมูลของออราเคิล (Oracle) จากไมโครซอฟต์เอ็กเซล (Microsoft Excel) วิธีการเข้าถึงข้อมูลแบบเดิมนั้นจะต้องทำการพรีคอมไพล์โค้ดของเอ็กเซลและออราเคิล โดยใช้ IBM precompiler และ Oracle precompiler ตามลำดับ จะเห็นว่าเป็นการยุ่งยากมากทีเดียว

วิธีการต่อเชื่อมแบบโอดีบีซี จะให้ความสะดวกในการติดต่อข้อมูลมากกว่าวิธีแบบดั้งเดิม โดยการกำหนดมาตรฐานการเชื่อมต่อของข้อมูล (Data protocol, DBMS capability) และแนวทางนี้ได้ทำให้เกิดความคิดที่จะสร้างไครีโรวอร์สำหรับการติดต่อกับฐานข้อมูลขึ้นมา (DLL)

ทฤษฎีการทำงานของโอดีบีซี

ส่วนติดต่อของการเชื่อมต่อฐานข้อมูลแบบเปิด (ODBC : Open Database Connectivity) เป็นตัวทำให้โปรแกรมประยุกต์สามารถที่จะเข้าถึงระบบจัดการฐานข้อมูล (DBMS : Database Management System) โดยใช้ภาษา เอสคิวเอล (Structure Query Language) เป็นมาตรฐานหลักในการเข้าถึงข้อมูล

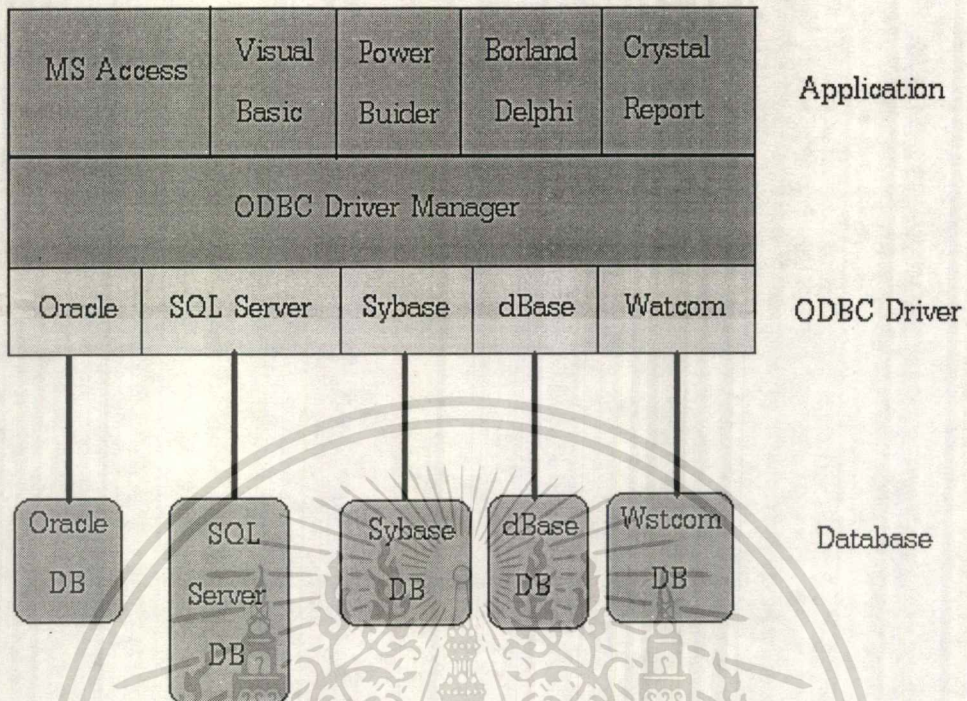
ข้อดีของการติดต่อโดยใช้โอดีบีซี

1. ฟังก์ชันของโอดีบีซีอนุญาตให้แอปพลิเคชันติดต่อกับระบบจัดการฐานข้อมูลได้โดยสะดวก (การทำคำสั่งเอสคิวแอล และการรับผลลัพธ์)
2. ใช้ภาษาเอสคิวแอล ตามมาตรฐาน SQL CAE, X/Open และ SQL Access Group (SAG)
3. มีการกำหนดการส่งกลับรหัสความผิดพลาด (Error Code) เป็นมาตรฐานเดียวกัน
4. เป็นวิธีมาตรฐานในการติดต่อกับระบบจัดการฐานข้อมูล
5. มีการกำหนดชนิดข้อมูล (Data Type) เป็นมาตรฐาน
6. ชุดคำสั่งเอสคิวแอล สามารถกำหนดได้แม้ในขณะที่ Runtime
7. สามารถเขียนโปรแกรมชุดเดียว แต่สามารถเข้าใช้ระบบจัดการฐานข้อมูลได้หลายตัว
8. ตัวโปรแกรมไม่ต้องรับผิดชอบในการดูแลการติดต่อข้อมูลกับระบบจัดการฐานข้อมูล
9. ค่าข้อมูลสามารถถูกส่งหรือรับได้ในรูปแบบที่สะดวกขึ้น

องค์ประกอบของโอดีบีซี

สถาปัตยกรรมของโอดีบีซี ประกอบด้วย 4 ส่วนสำคัญ

1. แอปพลิเคชัน ทำหน้าที่ประมวลผลและเรียกใช้ฟังก์ชันของโอดีบีซี ตามคำสั่งภาษาเอสคิวแอล พร้อมทั้งรับผลลัพธ์ด้วย
2. ตัวจัดการ ไดรฟ์เวอร์ (Driver Manager) ทำหน้าที่โหลดไดรฟ์เวอร์เชื่อมต่อกับแหล่งข้อมูล
3. ไดรฟ์เวอร์ ทำหน้าที่ประมวลผลการเรียกใช้ฟังก์ชันของโอดีบีซี และส่งคำสั่งเอสคิวแอลไปสู่แหล่งข้อมูลที่ต้องการ และทำการส่งผลลัพธ์กลับไปให้แอปพลิเคชัน และในบางครั้งไดรฟ์เวอร์จะทำหน้าที่แปลงคำสั่งที่ส่งมา ให้อยู่มาในรูปแบบที่สนับสนุนโดยระบบจัดการฐานข้อมูลแต่ละชนิดอีกด้วย
4. แหล่งข้อมูล (Data Source) เป็นแหล่งข้อมูลที่ใช้ต่อเข้าถึง



รูปที่ ก-1 องค์ประกอบของ ODBC

แอปพลิเคชัน

ตัวโปรแกรมจะเรียกใช้การเชื่อมโอดีบีซี ในการทำงานต่อไปนี้

1. ร้องขอการต่อเชื่อมกับแหล่งข้อมูล
2. ส่งคำสั่งเอสคิวแอล ตูแหล่งข้อมูล
3. กำหนดพื้นที่การจัดเก็บและรูปแบบของข้อมูลที่เป็นผลลัพธ์จากการร้องขอเอสคิวแอล
4. ร้องขอผลลัพธ์
5. ประมวลผลและจัดการกับข้อผิดพลาด
6. รายงานผลให้กับผู้ใช้ (ถ้าจำเป็น)
7. ร้องขอการ Commit หรือ Rollback สำหรับควบคุมการประมวล Transaction
8. ยกเลิกการติดต่อกับแหล่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวจัดการไครฟ์เวอร์

ตัวจัดการไครฟ์เวอร์ คือ DLL (Dynamic Link Library) และไลบรารีอื่นๆ หน้าที่หลักของตัวจัดการไครฟ์เวอร์ก็คือการโหลดไครฟ์เวอร์ ส่วนหน้าที่อื่นๆก็มีดังนี้

1. เรียกใช้ไฟล์ ODBC.INI เพื่อกำหนดชื่อของแหล่งข้อมูล (Data Source Name) ให้กับไครฟ์เวอร์ DLL
2. ทำการประมวลผลการเริ่มต้นการเชื่อมต่อของ โอดีบีซี
3. เป็นจุดต่อเชื่อมระหว่างฟังก์ชันของโอดีบีซี กับไครฟ์เวอร์แต่ละตัว
4. ทำการตรวจสอบพารามิเตอร์และลำดับการเรียกใช้โอดีบีซี

ไครฟ์เวอร์

ไครฟ์เวอร์คือ DLL ที่สร้างฟังก์ชันของโอดีบีซี และทำการโต้ตอบกับแหล่งข้อมูล ไครฟ์เวอร์ทำการตอบสนองการเรียกฟังก์ชันของโอดีบีซี โดยจะทำงานต่อไปนี้

1. สร้างการต่อเชื่อมกับแหล่งข้อมูล
2. ส่งคำร้องให้กับแหล่งข้อมูล
3. แปลงข้อมูลจากรูปแบบหนึ่งสู่อีกรูปแบบหนึ่ง
4. ส่งผลลัพธ์กลับให้แอปพลิเคชัน
5. จัดการส่งข้อมูลความผิดพลาดให้อยู่ในรูปแบบรหัสมาตรฐาน แล้วส่งกลับไปให้แอปพลิเคชัน
6. ทำหน้าที่จัดการและคูเรเตอร์เซอร์ (Cursor)

แหล่งข้อมูล

แหล่งข้อมูล หมายถึง การรวมกันของระบบจัดการฐานข้อมูล, ระบบปฏิบัติการ และระบบเครือข่ายโดยมีการชี้เฉพาะแต่ประเภทลงไป หรืออีกนัยหนึ่งหมายความว่า การที่แอปพลิเคชันทำการติดต่อกับระบบจัดการฐานข้อมูลยี่ห้อหนึ่งบนระบบปฏิบัติการหนึ่ง เข้าถึงโดยระบบเครือข่ายชนิดหนึ่ง

ภาคผนวก ข.

IDS Server

IDS Server คืออะไร

IDS Server เป็นเซิร์ฟเวอร์ สำหรับการเข้าถึง ไปยังฐานข้อมูลบน internet (internet database access server) ซึ่งทำให้เอชทีเอ็มแอล และ ผู้พัฒนา จาวา สามารถเขียนโฮมเพจ, จาวาแอปเพล็ตและจาวาแอปพลิเคชัน เพื่อติดต่อกับฐานข้อมูลได้ ดังนั้น IDS Server จึงเป็นหนทางในการแก้ปัญหาของ เอชทีเอ็มแอล และเจค็อบีซี ในการพัฒนาแอปพลิเคชันเกี่ยวกับฐานข้อมูลทั้งบน Internet และ Intranet IDS Server ในปัจจุบันสนับสนุนการเข้าถึงยังโอค็อบีซี ทุกตัวบนระบบของ Windows NT และ Windows 95 ซึ่งทำให้สามารถใช้งานกับระบบฐานข้อมูลต่างๆมากมาย เช่น Oracle, Sybase, SQL Server, MS Access และระบบฐานข้อมูลหลักอื่นๆอีกมาก

ใน IDS Server จะมี IDS JDBC Driver ซึ่ง จาวาแอปพลิเคชัน สามารถใช้ เจค็อบีซี เอพีไอ ในการเข้าถึง ไปยังฐานข้อมูลโดยทำการเชื่อมต่อไปยัง IDS Server IDS JDBC Driver มีประสิทธิภาพสูง กระทัดรัด และจัดอยู่ในประเภท ชนิด-3 เนื่องจากการ เข้าถึง ไปยังฐานข้อมูลส่วนมาก IDS Server สามารถทำได้อย่างมีประสิทธิภาพจึงทำให้ เจค็อบีซีไครฟ์เวอร์เหมาะสำหรับทั้งจาวาแอปเพล็ต และจาวาแอปพลิเคชัน IDS Server สามารถทำงานได้กับจาวาบราวเซอร์ ทุกตัว และ กับทุกรุ่นของ JDK, Microsoft SDK สำหรับจาวา และอื่นๆ

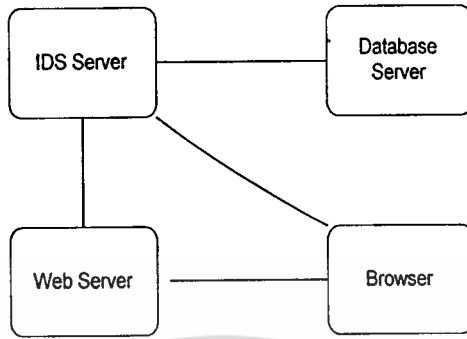
ถึงแม้ว่า IDS Server จะไม่ได้ออกแบบมาเพื่อทำหน้าที่ เว็บเซิร์ฟเวอร์ โดยตรง แต่ IDS Server ก็ สามารถทำหน้าที่ให้บริการง่ายๆ ที่ เว็บเซิร์ฟเวอร์ ทั่วไปสามารถทำได้ ดังนั้นจึงเป็นไปได้ที่ IDS Server จะทำหน้าที่เป็นระบบที่ทำการเข้าถึง ไปยังฐานข้อมูลผ่านทางอินเทอร์เน็ตได้อย่างสมบูรณ์

รูปแบบของ IDS Server

IDS Server สามารถเป็น self-contained HTTP สำหรับ internet application ได้ โดยมันจะทำการจอง IP address และ port number ซึ่งเหมือนกับโปรแกรม TCP/IP อื่นๆ เช่น FTP, Gopher, DNS และเว็บเซิร์ฟเวอร์

ในการ เข้าถึง ฐานข้อมูลในสภาพแวดล้อมของ Internet และ/หรือ Intranet มักจะประกอบด้วย องค์ประกอบหลัก 3, ประการคือ เว็บเซิร์ฟเวอร์, เซิร์ฟเวอร์ของฐานข้อมูลและเว็บเบราว์เซอร์ องค์ประกอบทั้ง 3 นี้สามารถอยู่บนเครื่องคอมพิวเตอร์เดียวกันได้ หรือแยกกันอยู่บนแต่ละเครื่อง โดยธรรมชาติของ IDS Server แล้วสามารถที่จะถูก install อยู่บนการรวมกันแบบใดก็ได้ของทั้ง 3 องค์ประกอบนี้ หรืออยู่บนเครื่องอื่นที่ไม่มีองค์ประกอบอันใดอันหนึ่งอยู่เลยก็ได้ ดังรูป (IDS Server เป็นโปรแกรมแบบเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

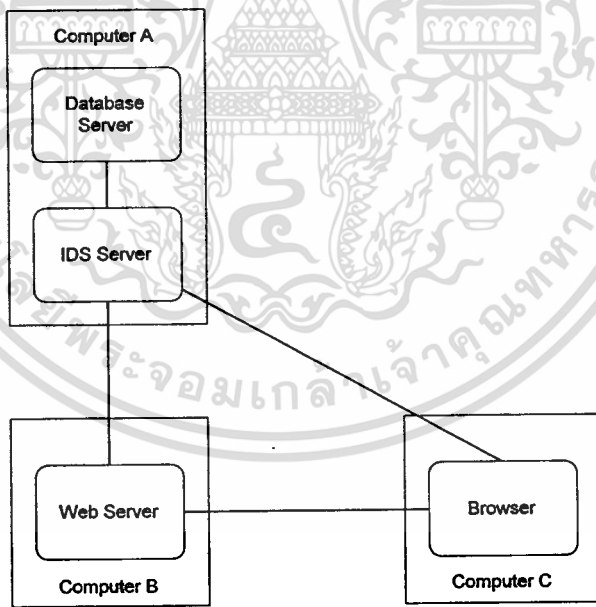
multi-threaded server ดังนั้นสำหรับ IDS Server หนึ่งๆจะสามารถทำงานกับ เว็บเซิร์ฟเวอร์หรือฐานข้อมูล ได้ไม่จำกัด)



Computer A

รูปที่ ข-1 การติดตั้ง IDS Server บน Personal Computer

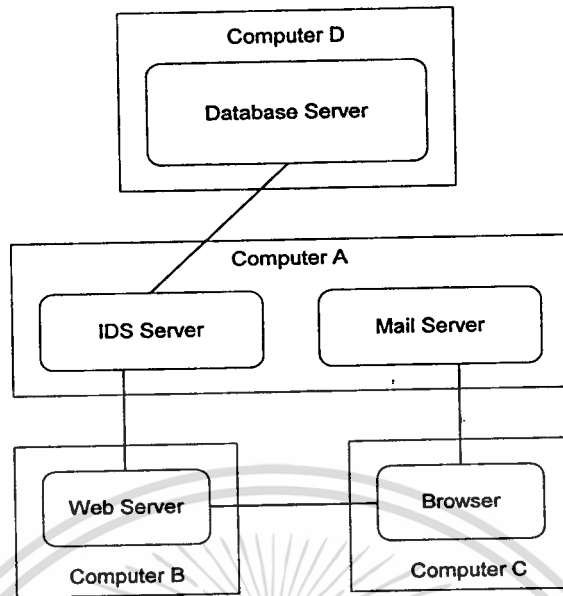
รูปที่ ข-1 แสดงการติดตั้ง IDS Server บนเครื่องพีซี ซึ่งองค์ประกอบทั้ง 3 จะทำงานอยู่บนเครื่องคอมพิวเตอร์เดียวกันหมด



รูปที่ ข-2 การติดตั้ง IDS Server แบบแยกองค์ประกอบบน 3 เครื่อง

ในรูปที่ ข-2 องค์ประกอบทั้ง 3 จะมีการกระจายอยู่บนเครื่องต่างๆ ในรูปแบบนี้ IDS Server จะถูกติดตั้งอยู่บนเครื่องเดียวกันกับเซิร์ฟเวอร์ของฐานข้อมูลซึ่งวิธีนี้ฐานข้อมูลจะมีความสัมพันธ์กับการจราจร (traffic) ในเครือข่ายไม่รวมถึงเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-3 การติดตั้ง IDS Server แบบแยกองค์ประกอบบน 4 เครื่อง

ในรูปที่ ข-3 เป็นการแสดงการกระจายองค์ประกอบทั้งหมดออกจากกัน ซึ่งจะมีการกระจายอยู่บน 4 เครื่องด้วยกัน IDS Server ถูกติดตั้งอยู่บนเครื่องซึ่งมี mail server แต่ไม่มีองค์ประกอบทั้ง 3 ประการที่ได้กล่าวมาแล้วเลย ข้อดีของรูปแบบนี้ก็คือ mail server มักจะมีกิจกรรมทางด้านเครือข่ายน้อย ดังนั้นจึงเป็นการช่วยลดการจราจรของเครือข่ายออกจาก เว็บเซิร์ฟเวอร์ หรือ เซิร์ฟเวอร์ของฐานข้อมูล โดยเฉพาะอย่างยิ่งเมื่อเซิร์ฟเวอร์ของฐานข้อมูล วิ่งอยู่บนเครื่องที่มีฮาร์ดแวร์และ Operating System จำกัดทำให้ไม่สามารถติดตั้ง IDS Server ได้ รูปแบบนี้จะช่วยได้มาก นอกจากนี้รูปแบบนี้ยังต้องการ โปรแกรม database network client ที่เหมาะสมติดตั้งอยู่บนเครื่อง Computer A สำหรับ IDS Server ติดต่อไปยัง remote database เช่น ถ้า remote database เป็น Oracle 7 Server SWL*Net จะต้องถูกติดตั้งบนเครื่อง Computer A ด้วย

ขอบเขตของ IDS Server ที่ใช้

ในที่นี้ได้ใช้ IDS Server version 2.5 Export เป็นโปรแกรมที่ให้ทดลองใช้ (ไม่ใช่โปรแกรมเต็ม สามารถ download ได้ที่ www.idssoftware.com) ซึ่งมีข้อจำกัดการใช้งานดังนี้

[General]

MaxSessionTime = 30 ;minutes

DefSessionTime = 15 ;minutes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DatabaseIdleTime = 10 ;minutes
DBLoginTimeout = 45 ;seconds, 0 < DBLoginTimeout < 600 (10 min)
Port = 12 ;default 80 if not specified
CheckRemoteHTX = 1 ;0 disable, 1 enable
Hosts = deny ;deny or grant
DefMaxFieldSize = 16 ;kbytes, 1 < DefMaxFieldSize <= 1024 (1 meg)

```

[Hosts]

```

;192.9.200.0 sun Sun
;192.9.200.1 mercury Mercury
;192.9.200.2 venus Venus
;192.9.200.3 earth Earth 3th_Rock_from_the_Sun
;192.9.200.4 mars Mars

```

[Mime-type]

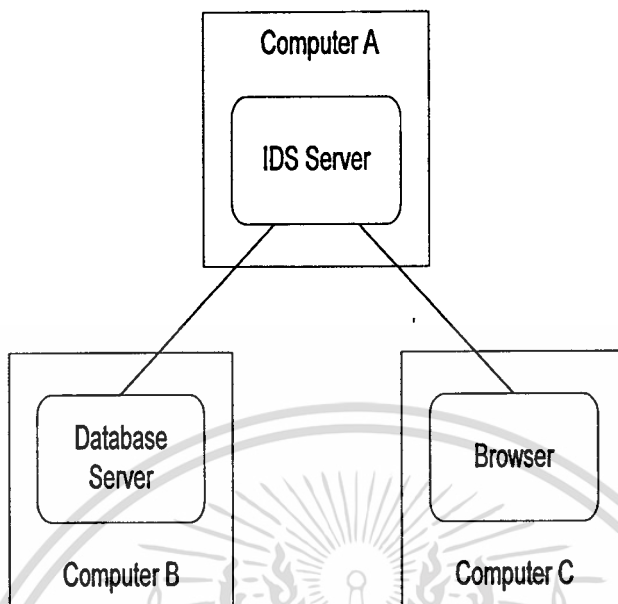
```

htm text/html
html text/html
gif image/gif
jpg image/jpeg
jpeg image/jpeg
txt text/plain

```

IDS JDBC Driver

IDS JDBC Driver เป็น เจดีบีซีไดรฟ์เวอร์ชนิดที่3 ซึ่งตัว IDS Server ก็คือ net sever middleware ที่ถูกใช้โดย IDS JDBC การเข้าถึง ไปยังฐานข้อมูลจะทำโดยไดรฟ์เวอร์ซึ่งจะถูกส่งไปและกลับระหว่าง ไดรฟ์เวอร์กับ IDS Server ซึ่งจะถูกทำโดย เจดีบีซีไดรฟ์เวอร์ ทำการติดต่อกลับไปยัง IDS Server โดยผ่าน socket ในจาวา ดังแสดงในรูป ข-4



รูปที่ ข-4 แบบแผนของ IDS Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.
ตารางและรายละเอียดในฐานข้อมูล
(ใช้ Microsoft Access 7.0)

1. ตาราง RENSJ	เก็บรายละเอียดเกี่ยวกับนักศึกษาทั้งหมดของคณะวิศวกรรมศาสตร์
† SDID	รหัสนักศึกษา
SDFAC	รหัสคณะของนักศึกษา (01 = คณะวิศวกรรมศาสตร์)
SDDEPT	รหัสของภาควิชาของนักศึกษาอยู่ (อ้างอิงตาราง RENDP)
SDPREF	เพศของนักศึกษา ภาษาไทย (1 = ชาย , 2 = หญิง)
SDPREFT	สถานะ (ภาษาไทย) นาย , น.ส. , นาง
SDNAME	ชื่อ - นามสกุลของนักศึกษา (ภาษาไทย)
SDPREFE	สถานะ (ภาษาอังกฤษ) MR. , MISS. , MRS.
SDNAMEE	ชื่อ ข นามสกุลของนักศึกษา (ภาษาอังกฤษ)
SDSEX	เพศของนักศึกษา ภาษาอังกฤษ (M = ชาย , F = หญิง)
SDSTAT	สถานะทางการศึกษาในภาคเรียนปัจจุบัน(1=กำลังศึกษาอยู่ , 2 = ลาออก , 3 = พักการศึกษา , 4 = จบการศึกษาแล้ว)
SDREGFLG	สถานะการลงทะเบียนแรกเข้า (= ไม่มาลงทะเบียน , 1 = ลงทะเบียน , 3 = ลาพัก)
SDLEVEL	กำลังศึกษาอยู่ชั้นปีที่
SDFTERM	เริ่มเข้าศึกษาครั้งแรกในภาคการศึกษาที่
SDFYEAR	เริ่มเข้าศึกษาครั้งแรกในปีการศึกษาที่
SDROOM	ห้องเรียนขณะที่อยู่ชั้นปีที่ 1 (ห้อง 1 - 10 เฉพาะภาคปรกติ)

2. ตาราง RENSJ	เก็บรายละเอียดเกี่ยวกับวิชาทั้งหมดที่เปิดสอนในปีการศึกษานั้น
† SJID	รหัสวิชา
SJFAC	รหัสคณะของวิชานั้นๆ
SJCRD	จำนวนหน่วยกิจของวิชา
SJLECCRD	จำนวนหน่วยกิจบรรยาย
SJPRACRD	จำนวนหน่วยกิจปฏิบัติ
SJOPEN	วิชานั้นๆเปิดสอนหรือไม่ (Y = เปิดสอน , N = ไม่เปิดสอน)
SJNSECT	วิชานั้นๆ แบ่งออกเป็นกี่ section
SJXDAY	วันที่สอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SJXTBEG	เวลาที่เริ่มสอบ
SJXTEND	เวลาที่สิ้นสุดการสอบ
SJTERM	ภาคเรียนที่เปิดสอน
SJCOMMENT	หมายเหตุเพิ่มเติม
SJTC1...SJTC7	รายชื่ออาจารย์ที่สอนวิชานี้

3. ตาราง **RENTD** เก็บรายละเอียดการลงทะเบียนของภาคการศึกษา และ ปีการศึกษา หนึ่งๆ

†	TDS DID	รหัสนักศึกษา
†	TDS JID	รหัสวิชา
	TDSECT	section ที่ได้ลงทะเบียนไว้
	TDCRB	จำนวนเครดิตของวิชาที่ได้ลงทะเบียน
	TDDEP	ภาควิชาที่นักศึกษากำลังศึกษาอยู่
	TDGRD	ผลการเรียนที่ได้

4. ตาราง **RENDP** เก็บรหัสและชื่อภาควิชาทั้งหมดของคณะวิศวกรรมศาสตร์

†	Dept_code	รหัสภาควิชา
	Dept_name	ชื่อภาควิชา
	Dept_char	อักษรที่ใช้แทนภาควิชา

5. ตาราง **REN RG** เก็บรายละเอียดการลงทะเบียนและเกรดที่ได้ทั้งหมดทุกภาคและปีการศึกษา

†	StudentId	รหัสนักศึกษา
	SubjectId	รหัสวิชา
	Year	ปีการศึกษาที่ลงทะเบียน
	Semester	ภาคการศึกษาที่ลงทะเบียน
	Credit	หน่วยกิต
	Grade	ผลการเรียนที่ได้

หมายเหตุ: ตารางนี้ใช้ในการตรวจสอบผลการเรียนของนักศึกษาที่สมมุติขึ้น เพราะไม่สามารถนำตารางจริงมาใช้ได้

6. ตาราง **REPAS** เก็บ password ของนักศึกษาทั้งหมดที่กำลังศึกษาอยู่ในขณะนี้

†	StudentId	รหัสนักศึกษา
	Passw	รหัสผ่านของนักศึกษา มีความยาวได้ไม่เกิน 20 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ : ที่แยกตารางนี้ออกจากตาราง RENS D เนื่องจากไม่ต้องการไปเปลี่ยนแปลงโครงสร้างของตารางเดิมซึ่งเอามาจากสำนักทะเบียน

7. ตาราง **RENTC** เก็บรายละเอียดของอาจารย์ทั้งหมดที่ทำการสอนในคณะวิศวกรรมศาสตร์

TCID	รหัสอาจารย์
TCFAC	คณะที่อาจารย์สังกัดอยู่ (01 = คณะวิศวกรรมศาสตร์)
TCPOS	ตำแหน่งของอาจารย์
TCNAME	ชื่อ - นามสกุลของอาจารย์
TCACC	account ของอาจารย์ (ใช้ในการตรวจสอบรายชื่อ น.ศ.)
TCPASS	password ของอาจารย์ (ใช้ในการตรวจสอบรายชื่อ น.ศ.)

8. ตาราง **Grade** เก็บค่าค่าของเกรดแต่ละตัว

Grade	เกรดที่มีทั้งหมด (A , B+ , B , C+ , C , D+ , D , F , S , U)
Gradevalue	ค่าของเกรดแต่ละตัว (0 - 4.00)

9. ตาราง **Subject** เก็บรายชื่อวิชาทั้งหมดของคณะวิศวกรรมศาสตร์

SubjectId	รหัสวิชา
Subjectname	รายชื่อวิชา
Credit	หน่วยกิต

รายชื่อตารางที่ใช้สำหรับบอร์ดข่าว

10. ตาราง **Board** เก็บรายละเอียดเกี่ยวกับ message

No	ลำดับที่
Date	วันที่ที่ได้รับ message
Time	เวลาที่ได้รับ message
Message	message ที่ได้รับ
Fr	ชื่อผู้ส่ง message
To	ชื่อผู้รับ message

11. ตาราง **Board_user** เก็บรายละเอียดของผู้ที่มีสิทธิ์ใช้บริการบอร์ดข่าว

No	ลำดับที่
Name	ชื่อ - นามสกุล ของผู้มีสิทธิ์ใช้บริการบอร์ดข่าว
Account	account ของผู้ใช้ ความยาวไม่เกิน 20 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการชิ้นนี้สามารถสำเร็จลุล่วงไปได้ด้วยดีนั้นก็เพราะได้รับความช่วยเหลือ และการสนับสนุนจากบุคคลหลาย ๆ ท่านดังนี้

- ขอขอบพระคุณคุณพ่อและคุณแม่ที่ให้การสนับสนุนในทุก ๆ เรื่อง
- ขอขอบพระคุณอาจารย์ที่ปรึกษา อ.ธนา หงษ์สุวรรณ ที่กรุณาเป็นอาจารย์ที่ปรึกษาให้พวกเรา
- ขอขอบพระคุณอาจารย์เกียรติวรรณที่ช่วยในเรื่องฐานข้อมูลต่าง ๆ
- ขอขอบพระคุณพี่แมว แผนกทะเบียนที่ให้คำแนะนำต่าง ๆ
- ขอขอบพระคุณ Michael Chen ที่กรุณาให้ความรู้และความร่วมมือในด้านต่าง ๆ
- ขอขอบพระคุณบริษัท IDS Software ที่ให้ download IDS Server มาใช้
- ขอขอบพระคุณ Web site ทุกตัวที่ให้ความรู้มาฟรี ๆ
- ขอขอบคุณเพื่อน ๆ ทุกคนที่ให้ความช่วยเหลือ (นะจ๊ะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- JDBC Developer's Resource
by Art Taylor
- The Java Language Environment (A White Paper)
by SUN
- JAVA Programming Explorer
By Steve Simkin, Neil Barlett, Alex Leslie
- สร้างสรรเว็บเพจสวย ๆ ด้วยภาษา JAVA
โดย Pual J. Perry เรียบเรียงโดย ณัฐวัฒน์ ลิ้มล้อมวงศ์
- Creating Cool Web Pages with HTML
By David Taylor, IDG Backs Worldwide, Inc.
- BCM Business Computer Magazine vol 9. NO. 107
January 1998 หัวข้อ WEB Database
- Internet-Intranet ปี 2 ฉบับ 8 ตุลาคม-พฤศจิกายน 2540