

การให้บริการผ่านเว็บเพจ

Service Web Page



โดย

นาย เจษฎาพร ศรีทับทิม

นาย ชานนท์ ภูริพันธุ์ภิญโญ

อาจารย์ที่ปรึกษา

ผศ. บรรจง ปิยะธำรง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมึก.....

เลขทะเบียน 37085

วัน, เดือน, ปี 30 ส.ค. 2542

ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2542

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การให้บริการผ่านเว็บเพจ

Service Web Page

ผู้จัดทำ

- | | | |
|----------------|-----------------|----------|
| 1. นาย เจษฎาพร | ศิริทัตติม | 39014099 |
| 2. นาย ชานนท์ | ภูริพันธุ์ภิญโญ | 39014131 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การให้บริการผ่านเว็บเพจ

นาย เจษฎาพร ศรีทับทิม
นาย ชานนท์ ฐิริพันธุ์ภิญโญ

ศส. บรรจง ปิยะธำรง
ปีการศึกษา 2542

บทคัดย่อ

ปัจจุบันเครือข่ายอินเทอร์เน็ตได้แพร่ขยายไปทั่วโลก และมีเว็บไซต์ต่าง ๆ เกิดขึ้นมากมายทำให้ การที่จะหาเว็บไซต์ที่ตรงกับที่เราต้องการนั้นจึงไม่ใช่เรื่องที่ยั่งยืน ดังนั้นจึงได้ทำโครงการนี้ขึ้นมาเพื่อ เป็นเว็บที่ช่วยให้ผู้ใช้ที่เข้ามาขอใช้บริการช่วยในการหาข้อมูลจากแหล่งต่าง ๆ ได้ง่ายขึ้น ซึ่งในโครงการนี้ ได้ทำเครื่องมือที่ช่วยในการค้นหาเว็บเพจต่าง ๆ (Search Engine) ซึ่งรองรับการค้นหาเว็บเพจให้ตรงความ ต้องการจากผู้ใช้งานได้โดยมีทั้งการค้นหาแบบธรรมดาซึ่งจะเป็นการใส่คำที่ต้องการเพียงอย่างเดียวและ การค้นหาโดยใช้เครื่องหมายซึ่งในที่นี้ได้รองรับกับคำสั่งแบบบูลีน (and,or) โดยข้อมูลที่ใช้ในการค้นหา นั้นได้มาจากโปรแกรมซึ่งจะท่องไปตามเว็บเพจต่าง ๆ เพื่อเก็บคีย์เวิร์ด และนอกจากที่เว็บของเราจะ สามารถเป็นเครื่องมือที่ช่วยในการค้นหาได้แล้ว ในโครงการนี้ยังมีส่วนที่เป็นเกมเชิร์ฟเวอร์ เพื่อให้ความ บันเทิงกับผู้ที่เข้ามาเยี่ยมชมเว็บไซต์ และเหตุที่ทำให้เกมเชิร์ฟเวอร์ขึ้นมาก็เพราะต้องการให้เป็นจุดเริ่มต้นให้เว็บ ไซต์อื่น ๆ ภายในประเทศไทยได้มีการทำเกมเชิร์ฟเวอร์ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Service Web Page

Mr. Chetsadaporn Sreethapthim

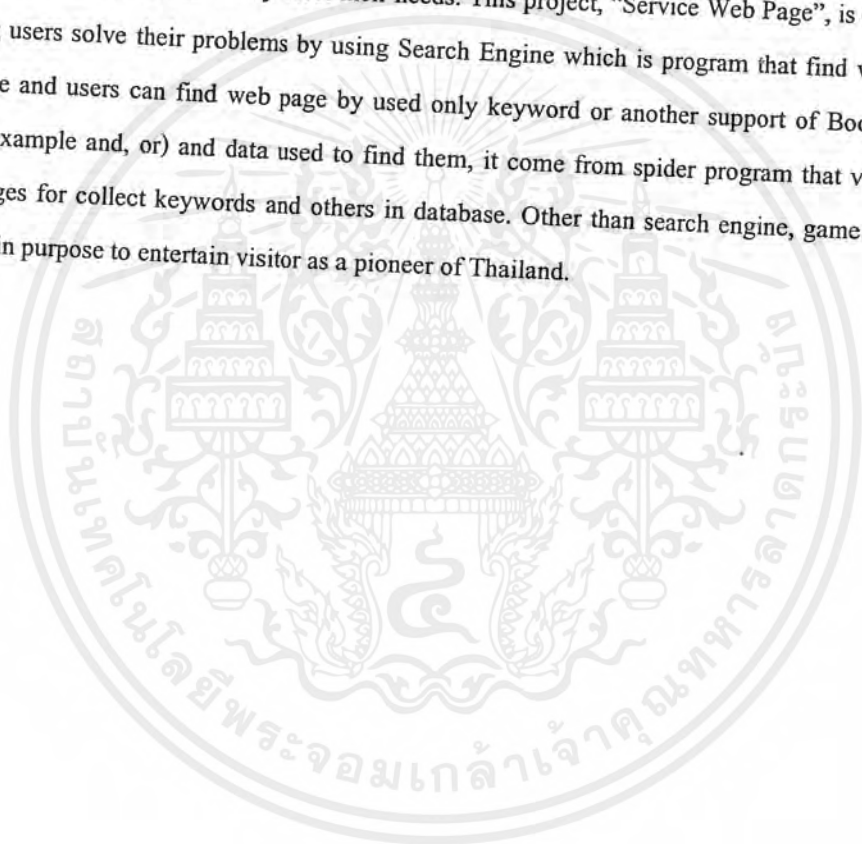
Mr. Chanon Pooripunpinyo

Assoc. Prof. Banjong Piyatumrong Advisor

1999

Abstract

At present, the Internet is growing by leaps and bounds, increasing amount of web sites makes users hard to find ones that really meet their needs. This project, "Service Web Page", is constructed for helping users solve their problems by using Search Engine which is program that find web page from database and users can find web page by used only keyword or another support of Boolean by using mark (example and, or) and data used to find them, it come from spider program that visit amount of web pages for collect keywords and others in database. Other than search engine, game server is also created in purpose to entertain visitor as a pioneer of Thailand.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นบุคคลสำคัญที่ทำให้ปริญญาบัตรนี้สำเร็จได้ด้วยดีคือ อาจารย์ บรรจง ปิยะธำรง อาจารย์ที่ปรึกษาที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณคณะอาจารย์ทุกท่านที่ได้ให้การอบรมสั่งสอนและให้ความรู้ความสามารถนำความรู้ที่ได้รับมาทั้งหมดมาใช้ได้

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้อำนวยความสะดวกในการทำโครงการนี้

ขอขอบคุณพี่ ๆ ที่มีรายงานดี ๆ ให้ศึกษา

ขอขอบคุณเพื่อน ๆ ชาว 4D ทุกคนที่ช่วยเป็นกำลังใจและกำลังความคิดที่ดีเสมอมา

ขอบคุณหมวย, สายสุดา, คิม, อ้อม ที่ให้ยืมใช้เครื่องทำโปรเจคที่ห้องภาคไม่จั่นก็คงไม่มีเครื่องใช้

ขอบคุณหมวย, สายสุดา, อ้อบในฐานะที่ให้คำปรึกษาที่ดีมาตลอด

ขอบคุณเพื่อน ๆ ทุกคนที่มีส่วนร่วมช่วยกันทำรายงานฉบับนี้

และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดในชีวิตนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้จัดทำ พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และคอยให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ผู้จัดทำขอระลึกในพระคุณอันสุดประมาท และขอกราบขอบพระคุณมา ณ ที่นี้

เจษฎาพร ศรีทับทิม

ชานนท์ ภูริพันธุ์ภิญโญ

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	3
1.5 ประโยชน์ที่ได้รับ	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 เซิร์ชเอนจิน (search engine)	4
2.2 เกมออนไลน์	7
2.3 ความรู้เบื้องต้นเกี่ยวกับเครือข่าย	13
2.3.1 ระบบเครือข่าย และการโปรแกรมบนระบบเครือข่าย	13
2.3.2 อินเทอร์เน็ต	18
2.4 ภาษาจาวา	24
2.4.1 Java Applet	26
2.4.2 Java Application	37
2.4.3 Java Servlet	39
2.4.4 Java Database Connectivity (JDBC)	43
2.4.5 การทำมัลติเทรด (Multithreading program)	50
2.4.6 ซ็อกเก็ต (socket)	53
บทที่ 3 การสร้างและการออกแบบ	57
3.1 การออกแบบและการสร้างในส่วนของเซิร์ชเอนจิน	57
3.1.1 การออกแบบและการสร้างในส่วนของการจัดเก็บข้อมูล	57
3.1.2 การออกแบบและการสร้างในส่วนของ การดึงข้อมูลจากเว็บเพจ เก็บในฐานข้อมูล	58
3.1.3 การออกแบบในส่วนของ การติดต่อระหว่างผู้ใช้บริการกับฐานข้อมูล	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบและการสร้างในส่วนของเกมออนไลน์	หน้าที่
3.2.1 ลักษณะของเกม	63
3.2.2 แนวการออกแบบเกม (Game Design)	64
บทที่ 4 การทดสอบและผลการทดสอบ	64
4.1 การทดสอบและผลการทดสอบในส่วนของเซิร์ฟเวอร์	68
4.1.1 ส่วนที่ทำการจัดการด้านฐานข้อมูล	68
4.1.2 ส่วนที่ทำการค้นหาข้อมูลจากเว็บเพจมาเก็บในฐานข้อมูล	70
4.1.3 ส่วนของการติดต่อระหว่างผู้ใช้บริการกับฐานข้อมูล	76
4.2 การทดสอบและผลการทดสอบในส่วนของเกมออนไลน์	80
4.2.1 Client – side	80
4.2.2 Server – side	81
บทที่ 5 บทสรุปและวิจารณ์	82
5.1 สรุปผลการดำเนินงาน	82
5.2 แนวทางในการพัฒนาต่อ	83
ภาคผนวก	
ภาคผนวก ก ตัวอย่างโค้ดของเว็บเพจ www.peugeot-th.com	ก-1
ภาคผนวก ข Stop Words	ข-1
ภาคผนวก ค แผนภาพการทำงานต่าง ๆ	ค-1
ภาคผนวก ง คู่มือการใช้งาน	ง-1
ภาคผนวก จ Source code	จ-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2-1 แสดงถึงเว็บเกมที่สามารถเล่นได้ทั้งแบบคนเดียวและเล่นบนเครือข่ายอินเทอร์เน็ต	8
รูปที่ 2-2 แสดงถึงเกมที่ไม่ได้ใช้จาวาเป็นหลักในการพัฒนา	8
รูปที่ 2-3 แสดงเกมที่เล่นสองคนบนเครื่องที่ไม่ได้ต่อกับระบบเครือข่าย	10
รูปที่ 2-4 แสดงเกมที่เล่นสองคนบนระบบเครือข่าย	10
รูปที่ 2-5 แสดงเกมแบบหมุนรอบกรณีมีผู้เล่น 4 คน	11
รูปที่ 2-6 แสดงเกมแบบเหตุการณ์ที่มีผู้เล่น 4 คน	12
รูปที่ 2-7 แสดงวิธีสเตทซิง โคร โนซ์เซชันที่มั่งผ่านข้อมูลระหว่างผู้เล่นสองคนในเกมแบบเครือข่าย	12
รูปที่ 2-8 แสดงวิธีอินพุทซิง โคร โนซ์เซชันในเกมแบบเครือข่ายที่เล่นสองคน	13
รูปที่ 2-9 ระบบเครือข่ายคอมพิวเตอร์เบื้องต้น	14
รูปที่ 2-10 OSI โมเดล ทั้ง 7 เลเยอร์	16
รูปที่ 2-11 การเอ็นแคปซูลเลขชั้น	16
รูปที่ 2-12 ลำดับการเอ็นแคปซูลเลขชั้น	17
รูปที่ 2-13 เลขอร์ของโปรโตคอล TCP/IP	18
รูปที่ 2-14 หมายเลข IP แอดเดรส	20
รูปที่ 2-15 แสดงคลาส จำนวนเครือข่าย และจำนวนโฮสต์ของแต่ละคลาส	20
รูปที่ 2-16 แสดงรูปแบบของ IP แอดเดรสในคลาสต่าง ๆ	21
รูปที่ 2-17 ตัวอย่างหมายเลข IP แอดเดรสทั้งสองแบบ	21
รูปที่ 2-18 การทำซบเน็ตติ้ง	22
รูปที่ 2-19 ความสัมพันธ์ระหว่างชุด โปรโตคอล	23
รูปที่ 2-20 การทำงานของเว็บเบราว์เซอร์ที่ควบคุมการทำงานของเมธอดต่างๆ ในแอปเพล็ต	26
รูปที่ 2-21 ระบบติดต่อผู้ใช้ของ AWT ภายใต้ Window 9x/NT	31
รูปที่ 2-22 แสดงโครงสร้างแบบรากต้นไม้ของระบบเชื่อมต่อตามรูปที่ 2	31
รูปที่ 2-23 ความสัมพันธ์แบบถ่ายทอดของคลาส AWT ต่างๆ	32
รูปที่ 2-24 แสดงอินสแตนซ์ของแต่ละคลาส (หน้าต่างของคอมโปเนนท์พื้นฐาน)	33
รูปที่ 2-25 แสดงประเภทของคอนเทนเนอร์	33
รูปที่ 2-26 แสดงการเชื่อมต่อแบบ Two-Tier model	44
รูปที่ 2-27 แสดงโครงสร้างของ Three-Tier model	45
รูปที่ 2-28 โครงสร้างการเชื่อมต่อภายใน JDBC	45
รูปที่ 2-29 แสดงส่วนประกอบหลัก ๆ ของ API	46
รูปที่ 2-30 แสดงลักษณะของการเชื่อมต่อไดร์ฟเวอร์ชนิดที่ 1 และ 2	47
รูปที่ 2-31 แสดงลักษณะของการเชื่อมต่อไดร์ฟเวอร์ชนิดที่ 3 และ 4	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-32 แสดงถึงการทำงานของเทรดทั้งแบบที่มีโปรเซสเซอร์ตัวเดียวและแบบที่มีโปรเซสเซอร์หลายตัว	50
รูปที่ 2-33 เทรดสามารถอยู่ได้ในหนึ่งใน 5 สถานะต่อไปนี้ new, ready, running, inactive หรือ finished	51
รูปที่ 2-34 แสดงถึงการทำงานของ ServerSocket	55
รูปที่ 3-1 แสดงอีอาร์โมเดล	57
รูปที่ 3-2 แสดงถึงบอร์ดของเกม Othello โดยทั่วไปที่อยู่ในสถานะเริ่มต้น	63
รูปที่ 3-3 ลูกศรแสดงถึงตำแหน่งที่สามารถจะวางได้เมื่อเราเป็นผู้เล่นสีดำ	64
รูปที่ 3-4 แสดงหน้าจอการทำงานของเซิร์ฟเวอร์ที่กำลังรอการติดต่อจากไคลเอนท์	66
รูปที่ 3-5 แสดงหน้าจอการทำงานของเซิร์ฟเวอร์หลังจากทำการติดต่อจากไคลเอนท์เรียบร้อยแล้ว	67
รูปที่ 3-6 แสดงผลการทำงานที่ฝั่งไคลเอนท์	67
รูปที่ 4-1 แสดงการรันคำสั่ง javac.exe และ java.exe	69
รูปที่ 4-2 แสดงผลจากการรันคำสั่งในฐานข้อมูล	69
รูปที่ 4-3 แสดงผลที่ได้จากการรันโปรแกรม	70
รูปที่ 4-4 แสดงผลลัพธ์จากการรันโปรแกรมในฐานข้อมูล	70
รูปที่ 4-5 ผลจากการรันโปรแกรมการดึงข้อมูลในส่วนบน	71
รูปที่ 4-6 ผลจากการรันโปรแกรมการดึงข้อมูลในส่วนล่าง	72
รูปที่ 4-7 ผลจากการวิเคราะห์ข้อมูลในส่วนของคีย์เวิร์ดที่ได้	73
รูปที่ 4-8 ผลจากการวิเคราะห์ข้อมูลในส่วนของชื่อลิงค์ที่ได้	73
รูปที่ 4-9 แสดงข้อมูลส่วนหนึ่งในตาราง search ที่ได้จากการวิเคราะห์เว็บเพจ	74
รูปที่ 4-10 แสดงข้อมูลในตาราง detail ที่ได้จากการวิเคราะห์เว็บเพจ	74
รูปที่ 4-11 แสดงผลการวิเคราะห์ข้อมูลให้เว็บเพจสุดท้ายของเว็บไซท์แรก	75
รูปที่ 4-12 ผลที่ได้จากการค้นหาในการค้นหาแบบธรรมดา (ส่วนบน)	76
รูปที่ 4-13 ผลที่ได้จากการค้นหาในการค้นหาแบบธรรมดา (ส่วนล่าง)	77
รูปที่ 4-14 ผลที่ได้จากการค้นหาในการค้นหาแบบใช้เครื่องหมาย (ส่วนบน)	78
รูปที่ 4-15 ผลที่ได้จากการค้นหาในการค้นหาแบบใช้เครื่องหมาย (ส่วนล่าง)	78
รูปที่ 4-16 ผลจากการตรวจสอบรายชื่อเว็บไซท์	79
รูปที่ 4-17 ผลจากการตรวจสอบรายชื่อเว็บไซท์	79
รูปที่ 4-18 ผลจากการ submit ชื่อเว็บไซท์	80

สารบัญตาราง

ตารางที่ 2-1 แสดงคุณสมบัติคลาสคอนเทนเนอร์ทั้ง 4 ประเภท

หน้าที่

34



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

เมื่อมีการนำคอมพิวเตอร์หลายๆ เครื่องเข้ามาเชื่อมต่อกัน เพื่อเป็นการแบ่งปันทรัพยากรที่มีอยู่ให้สามารถใช้งานร่วมกัน และเกิดความคุ้มค่ามากขึ้นกลายเป็นระบบเครือข่ายคอมพิวเตอร์ท้องถิ่น (LAN : Local Area Network) และระบบเครือข่ายคอมพิวเตอร์ขนาดใหญ่ (WAN : Wide Area Network) จนกระทั่งในปัจจุบันนี้ระบบเครือข่ายคอมพิวเตอร์ได้ถูกพัฒนาขึ้นให้สามารถติดต่อสื่อสารกันได้ทั่วโลก กลายเป็นระบบเครือข่ายอินเทอร์เน็ต (Internet) ซึ่งกลายเป็นที่นิยมกันอย่างกว้างขวางมีการเชื่อมต่อเครื่องคอมพิวเตอร์หลากหลายชนิดที่อยู่ในสถานที่ต่างๆ ซึ่งห่างไกลกันได้ทั่วโลก ทำให้สามารถใช้งานกับแหล่งข้อมูลต่างๆ เช่น ระบบฐานข้อมูลซึ่งช่วยให้เกิดการพัฒนาในด้านต่างๆ อย่างรวดเร็ว เช่น การสืบค้นข้อมูลในการทำวิจัยของนักวิชาการ นักวิทยาศาสตร์ การสืบค้นข้อมูลสำหรับการทำธุรกิจ การสืบค้นข้อมูลของนักศึกษาในการเรียนการศึกษาหรือแม้กระทั่งการค้นหาข้อมูลต่างๆ เพื่อความบันเทิงของผู้ใช้งานทั่วไป ดังนั้นแนวโน้มของการใช้งานระบบเครือข่ายอินเทอร์เน็ตจะมีมากมายและเพิ่มขึ้นอย่างต่อเนื่อง

ประโยชน์ที่ได้รับในการสื่อสารโดยใช้ระบบเครือข่ายอินเทอร์เน็ต

- เป็นแหล่งที่ทำให้เกิดการพัฒนาเทคโนโลยีใหม่ๆ
- เกิดการติดต่อสื่อสารระหว่างกันและให้บริการข่าวสาร
- ทำให้เกิดการแลกเปลี่ยนความรู้ แนวความคิดและเกิดการสื่อสาร
- เป็นการติดต่อสื่อสารข้อมูลที่มีความรวดเร็ว
- ใช้สื่อสารทางด้านธุรกิจทางเครือข่ายคอมพิวเตอร์

การใช้งานระบบเครือข่ายอินเทอร์เน็ตที่เป็นที่นิยมมากในปัจจุบันนี้ ได้แก่การใช้งานบริการเว็ลด์ไวด์เว็บ (WWW : World Wide Web) ในช่วงแรกของการเริ่มมีการบริการเว็ลด์ไวด์เว็บนั้นเว็บเบราว์เซอร์สามารถอ่านข้อมูลจากเว็บเซิร์ฟเวอร์ และสามารถแสดงเฉพาะตัวอักษรได้เพียงอย่างเดียวซึ่งในยุคนี้บริการเว็บยังไม่ได้รับความนิยมมากเท่าใดนัก ต่อมามีการพัฒนาให้เว็บเบราว์เซอร์สามารถแสดงข้อมูลในโหมคกราฟฟิกได้ กล่าวคือในหน้าเว็บเพจสามารถที่จะใส่ทั้งข้อความ รูปภาพและไฟล์แบบต่างๆ เช่น เสียงหรือวีดิโอลงไปได้ด้วย จุดนี้ทำให้การบริการเว็ลด์ไวด์เว็บเป็นที่นิยมอย่างท่วมท้น และได้แข่งบริการตัวอื่นๆ ในอินเทอร์เน็ตไปอย่างรวดเร็ว จนบางครั้งหลายคนเข้าใจว่าเว็ลด์ไวด์เว็บ คือ อินเทอร์เน็ตไปเลยทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบริการเว็ลด์ไวด์เว็บสามารถช่วยให้ผู้ใช้งานสามารถค้นหาข้อมูลต่างๆ ได้ง่ายและรวดเร็วซึ่งข้อมูลที่ได้จากบริการเว็ลด์ไวด์เว็บนี้ มีทั้งข้อมูลที่เป็นตัวอักษร รูปภาพ ภาพเคลื่อนไหวหรือแม้กระทั่งข้อมูลเสียงก็มี นอกจากนี้รูปแบบการใช้งานมีการติดต่อกับผู้ใช้แบบกราฟฟิค (GUI : Graphic User Interface) ซึ่งทำให้ใช้งานได้ง่ายขึ้นและน่าสนใจยิ่งขึ้น การใช้บริการเว็ลด์ไวด์เว็บนี้นอกจากจะเรียกข้อมูลที่ต้องการขึ้นมาดูได้แล้ว ยังสามารถเผยแพร่ข้อมูลและสามารถทำงานกับระบบฐานข้อมูลที่มีอยู่ในฝั่งเซิร์ฟเวอร์ได้ด้วย โดยสามารถเพิ่มเติม แก้ไขข้อมูลในฐานข้อมูลก็ได้หรือจะสั่งให้แอปพลิเคชัน (Application) ที่อยู่ที่ฝั่งเซิร์ฟเวอร์ทำการประมวลผลหรือให้ทำงานอื่นใดก็ได้ตามต้องการ เช่น ระบบการเล่นเกมแบบมัลติเพลย์เยอร์ (Multi Player) ผ่านทางอินเทอร์เน็ตซึ่งต้องมีเซิร์ฟเวอร์คอยประมวลผลและส่งข้อมูลถึงกันระหว่างผู้เล่น เป็นต้น

จากข้อความข้างต้นจะเห็นว่ารูปแบบการนำเสนอข้อมูลแบบเสียง ภาพเคลื่อนไหวบนเว็บ การเชื่อมต่อกับระบบเครือข่ายจะสามารถดึงดูดให้ผู้ใช้งานบริการเว็ลด์ไวด์เว็บสนใจในเว็บเพจเหล่านั้น ซึ่งถ้าหากกล่าวในภาษาโปรแกรมมิ่งที่สามารถทำงานได้บนเว็บเบราว์เซอร์และกำลังที่จะเป็นที่สนใจของผู้เขียนโปรแกรมนั้นก็คือ ภาษาจาวา (Java) ซึ่งจาวานี้มีความสามารถมากกว่าภาษาอื่นที่ใช้งานบนเว็บ เช่น HTML ซึ่งแสดงข้อมูลได้เพียงแต่ข้อความและรูปภาพเท่านั้น ต่างจากภาษาจาวาซึ่งมีความสามารถในการตอบสนองความต้องการด้านมัลติมีเดียต่างๆ ไม่ว่าจะเป็นภาพเคลื่อนไหว วิดีโอ เสียงและการติดต่อสื่อสารได้ต่อกันระหว่างระบบเครือข่ายได้ดีกว่า

1.2 วัตถุประสงค์ของงานวิจัย

- ศึกษาทฤษฎีของภาษาจาวา ซึ่งเป็นภาษาที่สามารถทำงานบนระบบเครือข่ายหรือเรียกใช้บริการทางเว็ลด์ไวด์เว็บได้
- ศึกษาการสร้างแอปพลิเคชันที่สามารถเรียกใช้งานผ่านทางบริการเว็ลด์ไวด์เว็บได้และมีรูปแบบของการนำเสนอที่น่าสนใจ
- ศึกษาการสร้างแอปพลิเคชันที่สามารถปฏิบัติงานร่วมกันซึ่งมีหลายผู้ใช้งานผ่านทางบริการเว็ลด์ไวด์เว็บได้
- ศึกษาการสร้างแอปพลิเคชันที่สามารถเข้าใช้งานกับระบบฐานข้อมูลโดยผ่านการเรียกใช้ทางบริการเว็ลด์ไวด์เว็บได้

1.3 ขอบเขตของงานวิจัย

ในงานวิจัยนี้ได้แบ่งขอบเขตของการศึกษาและพัฒนาออกเป็น 2 ส่วนหลักคือ

- เกมออนไลน์ (Game - Online)
- เซิร์ชเอนจิน (Search Engine)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในงานวิจัยนี้ใช้ภาษาจาวาเป็นหลักในการพัฒนาทั้งในส่วนที่เป็นเว็บเซิร์ฟเวอร์และส่วนที่เป็นไคลเอนท์ นอกจากนี้ในส่วนที่มีการเข้าใช้งานกับระบบฐานข้อมูลก็ทำการติดต่อโดยใช้ JDBC อีกด้วย และด้วยการนำจาวามาใช้ในการพัฒนานี้เอง ทำให้ผู้ใช้งานสามารถใช้ระบบงานที่ทำขึ้นนี้ได้โดยง่าย การติดต่อกับระบบเครือข่ายและแอปพลิเคชันที่สร้างขึ้นก็ทำงานได้โดยอิสระ ไม่ขึ้นกับระบบปฏิบัติการที่ผู้ใช้งานด้วย

1.4 วิธีการดำเนินงาน

การดำเนินงานของโครงการมีขั้นตอนดังนี้

1. ศึกษาค้นคว้าความรู้เกี่ยวกับทฤษฎีและวิธีการในการสร้างเซิร์ฟเวอร์และเกมออนไลน์
2. ศึกษาค้นคว้าภาษาจาวาในส่วนที่ต้องนำมาประยุกต์ใช้งาน
3. ออกแบบและสร้างโปรแกรม
4. ทดสอบโปรแกรมเพื่อนำผลการทดสอบมาวิเคราะห์ถึงการนำทฤษฎีและความรู้ที่ได้จากการศึกษามาปฏิบัติจริง

1.5 ประโยชน์ที่ได้รับ

1. ได้รับความรู้ในทฤษฎีและวิธีการในการสร้างเซิร์ฟเวอร์และเกมออนไลน์
2. เข้าใจการเชื่อมต่อและการทำงานของระบบเครือข่ายได้ดียิ่งขึ้น
3. ได้รับความรู้ในการเขียนภาษาจาวามากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 เซิร์ชเอนจิน (search engine)

เนื่องจากว่าอินเทอร์เน็ตเป็นแหล่งข้อมูลที่มีความหลากหลายและมีขนาดใหญ่มาก ทำให้การที่เราจะหาข้อมูลใด ๆ ที่เราสนใจบนอินเทอร์เน็ตจึงไม่ใช่เรื่องง่ายนัก ดังนั้นเราจำเป็นจะต้องมีเซิร์ชเอนจินซึ่งจะเป็นเครื่องมือที่จะช่วยหาสิ่งที่เราต้องการ เช่น การค้นหาเว็บเพจ การค้นหารูปภาพ การค้นหาไฟล์ เป็นต้น ซึ่งในที่จะเป็นการค้นหาเว็บเพจให้ตรงกับความต้องการของผู้ใช้

เซิร์ชเอนจินมีหลายประเภทดังนี้

1. เซิร์ชเอนจิน ซึ่งจะมีโปรแกรม spider คอยวิ่งไปตามเว็บเพจต่างๆ เพื่อเก็บคีย์เวิร์ดที่สำคัญที่แสดงถึงข้อมูลในเว็บเพจนั้น
2. ไคเร็คทอรี เป็นการแบ่งกลุ่มของเว็บไซต์โดยใช้นุชย์เป็นคนจัดการ
3. ไฮบริดเซิร์ชเอนจิน (Hybrid search engines) เป็นเซิร์ชเอนจินที่มีการจัดการแบบไคเร็คทอรี

โครงการนี้ได้ทำเซิร์ชเอนจินที่มีคุณสมบัติเป็นประเภทที่ 1 คือ จะมีโปรแกรมที่เรียกว่า spider คอยเก็บคีย์เวิร์ดตามเว็บเพจต่าง ซึ่งเซิร์ชเอนจินประกอบไปด้วย 3 ส่วนหลัก ๆ ดังนี้

1. Spider (Crawler)

เป็นโปรแกรมซึ่งจะวิ่งไปตามเว็บเพจต่าง ๆ เพื่อเก็บคีย์เวิร์ดที่สำคัญมาเก็บไว้ในฐานข้อมูล และเมื่อเจ้าของเว็บไซต์ใด ๆ ทำการลงทะเบียนการ submit มาเราจะส่งตัว spider ไปยังเว็บไซต์นั้นเพื่อไปค้นหาว่าเว็บไซต์นั้นมีคีย์เวิร์ดอะไรบ้างที่สำคัญและนำมาเก็บไว้ในฐานข้อมูลเพื่อใช้ในการค้นหาคำให้ตรงกับที่ผู้ใช้ต้องการต่อไป ซึ่งปกติแล้วจะมีการส่งโปรแกรม spider ไปยังเว็บเพจต่าง ๆ สมมติเพื่อหาการเปลี่ยนแปลงซึ่งอาจจะกำหนดเป็นทุก ๆ เดือนหรือเท่าไรก็ได้ตามที่เหมาะสม

Spider มีอยู่ด้วยกัน 2 แบบคือ

- Deep เมื่อรับ url ที่ได้มาจากการลงทะเบียนการ submit แล้ว spider จะเข้าไปที่ทุก ๆ เว็บเพจในเว็บไซท์นั้น เช่น เมื่อเจ้าของเว็บไซต์ www.java.sun.com ได้ submit เข้ามา เราจะส่งตัว spider ไปค้นหาคีย์เวิร์ดในทุก ๆ เว็บเพจ ของ www.java.sun.com เช่น www.java.sun.com , www.java.sun.com/jdbc.html , www.java.sun.com/product.html ฯลฯ
- Shallow เมื่อรับ url ที่ได้มาจากการลงทะเบียนการ submit เข้ามาแล้ว spider จะเข้าไปที่เว็บเพจเฉพาะ url ที่ได้มานั้น โดยที่จะไม่ลงลึกไปในเว็บเพจ อื่น ๆ ของ เว็บไซต์ นั้น

2. Index

เก็บข้อมูลต่าง ๆ ที่ได้มาจาก spider และถ้า page มีการเปลี่ยนแปลงข้อมูลในส่วนนี้ก็จะมีการเปลี่ยนแปลงไปด้วย

3. Search engine software

เป็นโปรแกรมที่จะใช้ในการค้นหาในส่วนของ index เพื่อให้ตรงกับคำที่ผู้ค้นหาต้องการ

- Stemming เป็นการค้นหาคำต่าง ๆ ที่มีพื้นฐานบนคำเดียวกัน เช่น ถ้าค้นหาคำว่า “swim” ก็จะพบใน “swims” และ “swimming” ด้วย
- Search engine math เป็นการใช้เครื่องหมายทางคณิตศาสตร์ช่วยในการค้นหาเพื่อให้มีประสิทธิภาพในการค้นหาสูงขึ้น เครื่องหมายที่ใช้มีดังนี้
 - “+” เป็นความต้องการให้แสดง page ที่มีคำตามเครื่องหมาย “+” ครอบคลุมคำ
 - “-” บอกว่าให้ค้นหา page ที่ไม่มีคำที่ตามหลังเครื่องหมายนี้
 - “” ใช้ค้นหา page ที่มีคำเหมือนในเครื่องหมายนี้ เช่น หาคำว่า +song+mp3 มันจะหา page ที่มีคำครบทั้ง 2 คำ แต่อาจไม่ตรงกับความหมายที่เราต้องการ เพราะตำแหน่งที่พบคำแต่ละคำอาจจะอยู่ห่างกันได้ ดังนั้นถ้าเราต้องการให้หา page ที่มีคำที่มีความหมายตรงกับที่เราต้องการจริง ๆ ให้ใส่ “ “ ครอบไว้ เช่น “song mp3”
 - เซิร์ชเอนจินบางตัวสามารถให้เราค้นหาเฉพาะในส่วนที่ต้องการได้ เช่น ค้นหาเฉพาะใน title ก็จะทำการค้นหาคำที่ต้องการเฉพาะใน title เท่านั้น
 - Boolean มีการใช้ทฤษฎีทางด้านตรรกวิทยาในการค้นหา เช่น AND , OR
 - Wildcards (*) เช่น ค้นหาคำว่า “sing*” ก็จะพบได้ในคำที่ขึ้นต้นด้วย sing เช่น พบในคำว่า “singing” และ “sings”

ลักษณะที่สำคัญของเซิร์ชเอนจินมีดังนี้

1. Crawling

จะแสดงถึงวิธีที่เซิร์ชเอนจินวิ่งไปค้นหาข้อมูลตามเว็บเพจต่าง ๆ ประกอบไปด้วย

- Deep Crawl คือการวิ่งไปยังเว็บเพจทุกเพจที่เจอในเพจใด ๆ
- Instant Indexing เซิร์ชเอนจินที่ใช้วิธีนี้จะแสดงเว็บเพจที่ได้ submit แล้วประมาณ 1-2 วัน หลังจากการ submit
- Frame support แสดงว่า search engine สามารถติดตาม frame link ได้หรือไม่ (ถ้าใน page มีหลาย frame) แต่ spider ทั่ว ๆ ไปจะเห็นเพียงแค่หน้าหลัก ๆ (master page) เท่านั้น
- Meta tag support สามารถค้นหา keyword จาก meta tag ได้หรือไม่ เช่น ค้นหา keyword จาก meta keyword หรือ meta description
- Link popularity เซิร์ชเอนจินสามารถกำหนดความนิยมของเว็บเพจโดยการวิเคราะห์หว่ามีการลิงก์ไปที่เพจนั้นจากเพจอื่น ๆ ได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Learn Frequency เซิร์ฟเวอร์สามารถรู้ได้หรือไม่ว่าเว็บเพจที่เคย submit เข้ามา มีการเปลี่ยนแปลง
- Image map support เซิร์ฟเวอร์สามารถอ่าน url ที่อยู่ใน imagemap ได้หรือไม่
- Password Protected Sites เซิร์ฟเวอร์บางตัวสามารถเข้า password ของ password protected site ได้ เนื่องจากเว็บมาสเตอร์ได้ทำ user name และ password ไว้ให้เพื่อต้องการให้รู้ว่าที่ใดที่ใดที่มีข้อมูลอยู่
- Robot.txt เป็นวิธีที่เว็บมาสเตอร์ใช้สำหรับกันไม่ให้เซิร์ฟเวอร์เข้ามาดูข้อมูลที่เว็บไซท์ของคุณ
- Meta Robots Tag ใช้สำหรับเว็บมาสเตอร์ที่ไม่ได้ทำไฟล์ robot.txt

2. Indexing

เป็นการกล่าถึงสิ่งที่ได้มาจากส่วน Crawling เพื่อจะทำกลับมาทำเป็นอินเด็กซ์ ซึ่งประกอบไปด้วย

- Full body text เก็บทุกคำที่ได้มาจาก spider
- Stop words อาจจะมีการเว้นคำบางคำในการค้นหาหรือเก็บข้อมูล เพื่อประหยัดพื้นที่ (saving space) รวมทั้งเพิ่มความเร็วในการค้นหาด้วย (speeding search)

การประหยัดพื้นที่ เช่น พิจารณาประโยค *"The way to the school is long and hard when walking in the rain"* ซึ่งเราพบว่ามาคำว่า *"the"* ปรากฏอยู่ 3 ครั้ง ดังนั้นเพื่อเป็นการประหยัดข้อมูลเราจะแทน *"the"* ด้วย *"*"* ดังนี้ *"* way to * school is long and hard when walking in * rain"*

การเพิ่มความเร็วในการค้นหา บางครั้งเซิร์ฟเวอร์จะเก็บทุกคำที่ได้มาแต่จะไม่ค้นหาบางคำเพื่อประหยัดเวลา เช่น พิจารณาการค้นหาคำว่า *"the piano player"* ซึ่งปกติในการค้นหาจะมีการค้นหา 3 ครั้ง คือ คำว่า *"the"* , *"piano"* และ *"player"* แต่วิธีการในการเพิ่มความเร็วในการค้นหาจะมีวิธีที่ดีกว่าคือ ค้นหาแค่คำว่า *"piano"* และ *"player"* เท่านั้น เพราะในความเป็นจริงแล้วคำว่า *"the"* นั้นไม่มีความหมาย ซึ่งมันเป็นเพียง article เท่านั้น

- Meta Description & Meta Keywords ใช้เพื่อแสดงว่าเซิร์ฟเวอร์อินโดสแน็บสนุน meta description และ meta keywords tags แต่ไม่ได้หมายความว่าแท็กเหล่านี้จะช่วยให้เพจนั้นมีอันดับที่สูงขึ้น
- ALT Text/Comments ใช้เพื่อแสดงว่าเซิร์ฟเวอร์อินโดสแน็บสนุนของ ALT Text ซึ่งสัมพันธ์กับรูปภาพหรือข้อความใน comment tags

3. Ranking

โดยส่วนมากแล้วเซิร์ฟเวอร์จะใช้บริเวณและความบอที่พบคีย์เวิร์ดบนเว็บเพจ เป็นพื้นฐานในการจัดลำดับเพื่อสนองต่อการค้นหาที่เข้ามา โดยวิธีการก็จะแตกต่างกันออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Meta Tags Boost Ranking เซิร์ชเอนจินบางตัวที่สนับสนุน meta description และ keywords tag จะให้สิทธิพิเศษแก่เพจนั้น ๆ ในกรณีที่ search terms อยู่ในบริเวณนั้นไม่ได้หมายความว่าเซิร์ชเอนจินทุกตัวที่สนับสนุนแท็กนี้ก็จะได้รับการจัดลำดับไว้ในตำแหน่งที่สูง
- Reviewed Status Boost Rankings เซิร์ชเอนจินบางตัวมีการรีวิว (review) หรือนำเว็บไซต์นั้นมาเก็บในลิสต์ซึ่งสัมพันธ์กับโดเมนเรกทอรี โดยอาจจะมีการให้สิทธิพิเศษแก่เว็บไซต์ที่ถูกรวบรวมในลิสต์ด้วยวิธีการนี้ก็ได้
- Link Popularity Boost Rankings เป็นการจัดอันดับตามจำนวนลิงก์ที่มีมายังเพจนั้น
- Direct Hit Boosts Rankings จัดอันดับจากจำนวนการคลิกหลังจากที่ได้แสดงผลลัพธ์หลังจากการค้นหา

4. Spam

เซิร์ชเอนจินหลัก ๆ ทุกตัวจะมีการกันเว็บไซต์ที่พยายามจะอวดอ้างตนเองเพื่อให้เว็บเพจของตัวเองนั้นอยู่ในอันดับที่สูง ๆ ออกไป

- Meta Refresh เจ้าของเว็บไซต์บางที่ได้สร้างเพจเป้าหมายซึ่งจะนำผู้ที่เข้าชมเว็บนั้น ไปยังเพจที่ต่างไปโดยอัตโนมัติซึ่งอยู่ภายในเว็บไซต์นั่นเอง ซึ่ง meta refresh tag นี้เป็นวิธีหนึ่งที่จะทำให้เพจที่กล่าวมา ซึ่งเซิร์ชเอนจินบางตัวจะไม่ทำอินเด็กซ์เพจที่มีอัตรา meta refresh สูง
- Invisible Text เป็นวิธีการในการพราง text ในเว็บเพจด้วยการทำให้ text นั้นมีสีเหมือนกับสีของพื้นหลังทำให้เราไม่สามารถมองเห็นได้ ซึ่งเซิร์ชเอนจินบางตัวจะไม่ทำอินเด็กซ์ของ text ประเภทนี้
- Tiny Text คือการที่ text นั้นมีขนาดเล็กมาก เว็บเพจที่มี text ที่มีขนาดเล็กมาก ๆ อาจจะไม่ได้ถือว่าเป็น span หรืออาจจะไม่ทำอินเด็กซ์ของ tiny text

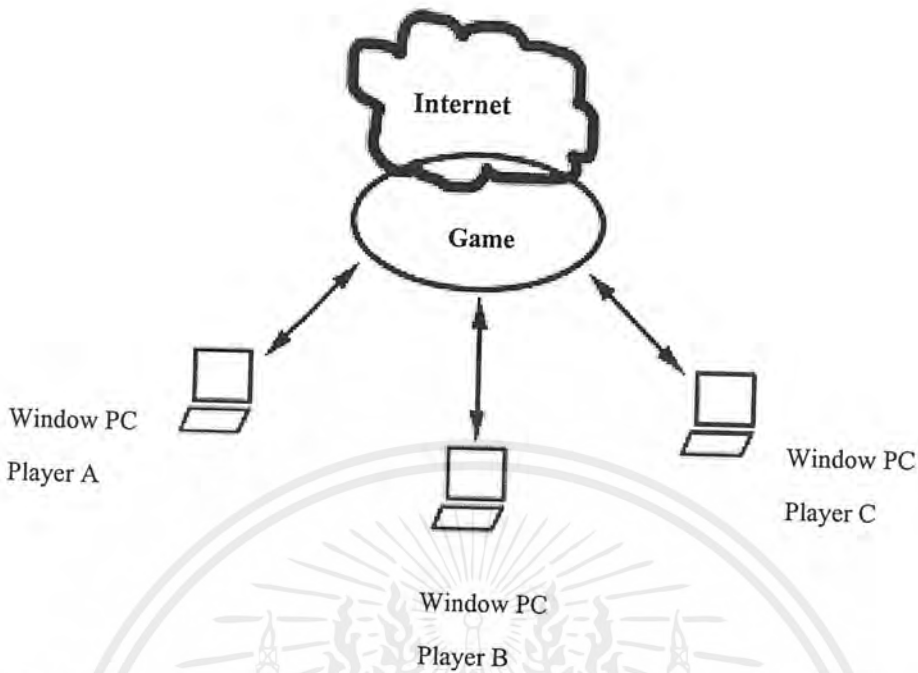
2.2 เกมออนไลน์

เกมออนไลน์เป็นเกมที่ถูกพัฒนาขึ้นมาเพื่อสร้างความบันเทิงให้กับผู้เล่นบนระบบเครือข่ายอินเทอร์เน็ต ซึ่งสามารถเล่นได้กับทุกคนที่กำลังใช้งานบนอินเทอร์เน็ตอยู่

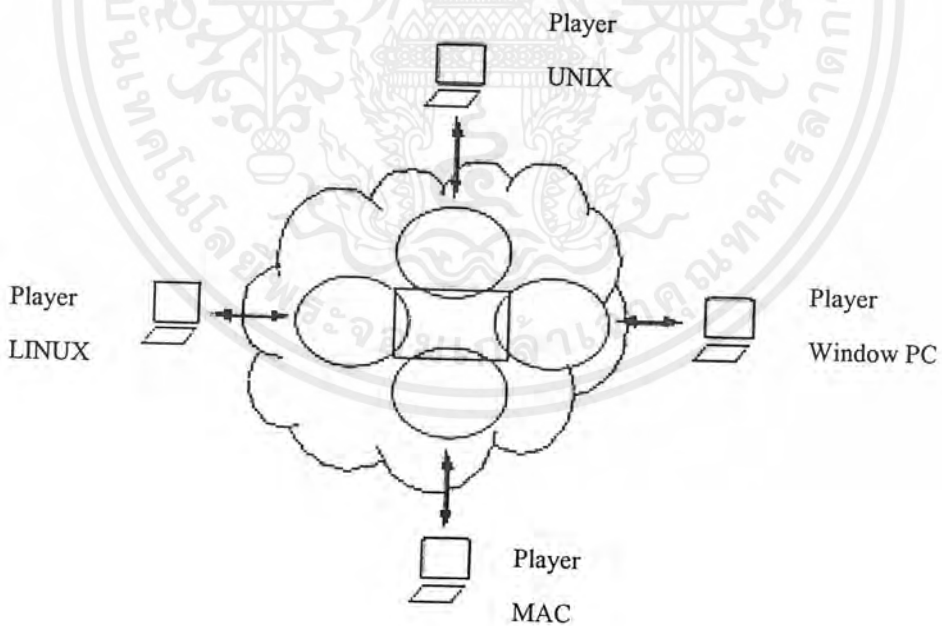
ประเภทของเกมออนไลน์

สามารถทำงานบนอินเทอร์เน็ตและเล่นอยู่บนเว็บเพจโดยไม่ขึ้นกับแพลตฟอร์มหรือระบบปฏิบัติการใด ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 แสดงถึงเว็บเกมที่เล่นได้ทั้งแบบคนเดียวและเล่นบนเครือข่ายอินเทอร์เน็ต



รูปที่ 2-2 แสดงถึงเกมที่ไม่ได้ใช้จาวาเป็นหลักในการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่ควรพิจารณาเมื่อเขียนเกมด้วยจาวา

ในการเขียนเกมด้วยจาวาจำเป็นต้องพิจารณาถึงองค์ประกอบเหล่านี้ คือ

1. กราฟฟิกและอนิเมชันหรือภาพเคลื่อนไหว
2. การป้อนอินพุท
3. เสียง
4. ระบบเครือข่าย
5. การจัดการกับมีเดีย

กราฟฟิกและอนิเมชันหรือภาพเคลื่อนไหว

เนื่องจากกราฟฟิกและภาพเคลื่อนไหวถือเป็นจุดสำคัญที่จะทำให้ตัวเกมน่าเล่นหรือน่าเบื่อ ดังนั้นจึงถือเป็นจุดสำคัญที่ควรพิจารณาเป็นอันดับแรกในการเขียนเกม จาวามีเอพีไอ (API) ที่สนับสนุนการทำงานเกี่ยวกับกราฟฟิกต่าง ๆ เช่น รูปภาพ กราฟฟิก 2 มิติ เป็นต้น ในส่วนของภาพเคลื่อนไหวนั้น ถึงแม้ว่าจาวาจะไม่มีเอพีไอที่สามารถจัดการได้โดยตรงแต่ก็มีแนวทางในการเขียนที่สามารถทำให้ง่ายขึ้น

การป้อนอินพุท

การป้อนอินพุทหรือการรับอินพุทก็นับเป็นจุดสำคัญอีกจุดหนึ่งในการเล่นเกมนเพราะเป็นสิ่งที่ใช้บ่งบอกความรู้สึกระหว่างเกมกับผู้เล่น จาวาได้สนับสนุนการทำงานของอุปกรณ์ที่ใช้รับอินพุทจากผู้เล่นไว้สองแบบ คือ คีย์บอร์ดและเมาส์ โดยในขณะที่เขียนโปรแกรมอยู่ ผู้เขียนสามารถทำการตรวจสอบการทำงานของอุปกรณ์เหล่านี้ได้ โดยการตรวจสอบเหตุการณ์ที่เกิดขึ้นจากอุปกรณ์เหล่านี้

เสียง

เสียงก็นับว่าเป็นจุดสำคัญในเกมเหมือนกัน จาวาสนับสนุนการทำงานกับเสียงได้ไม่ดีเท่าที่ควรเนื่องจากทำงานได้กับรูปแบบของเสียงเพียงสองรูปแบบ คือ *.au และ *.midi

ระบบเครือข่าย

ระบบเครือข่ายเป็นส่วนที่สำคัญที่ทำให้จาวาได้รับความนิยมอย่างมาก ระบบเครือข่ายของจาวาเป็นรูปแบบที่สนับสนุนการใช้งานด้วยเอพีไออย่างง่ายคายนมาก และจาวาก็คือภาษาที่ไม่ขึ้นกับระบบปฏิบัติการใดๆ ซึ่งทำให้สามารถเล่นเกมกับผู้อื่นที่ใช้เครื่องชนิดอื่น เช่น Sun หรือ Macintoshes ได้อย่างไม่เป็นปัญหา

การจัดการกับมีเดีย

การจัดการกับมีเดียต่าง ๆ คือ การคอยตรวจสอบมีเดียต่าง ๆ เช่น กราฟฟิก เสียงและอื่น ๆ ว่าถ่ายโอนข้อมูลสมบูรณ์หรือยัง เพื่อป้องกันปัญหารูปภาพแสดงไม่เต็มจอขณะที่เล่นเกมอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบเกม

เป็นการพิจารณาว่าเกมจะออกมาในรูปแบบไหน ซึ่งมีหัวข้อที่ต้องนำมาพิจารณาเป็นหัวข้อย่อย ดังนี้

- แนวความคิดพื้นฐาน (Basic Idea)
- แนวการดำเนินเกม (Story Line)
- โหมดการเล่นเกม (Play Mode)

แนวความคิดพื้นฐาน

เป็นแนวความคิดเกี่ยวกับรูปแบบของเกมว่าจะเป็นแบบเกมผจญภัย เกมแอ็คชั่น เกมยิงหรือรวมกันหลาย ๆ อย่าง

แนวการดำเนินเกม

เป็นสิ่งที่ช่วยให้ขอบเขตของเกมแคบขึ้น ทำให้มุมมองภาพรวมของเกมทั้งหมดได้ชัดเจนขึ้น และแนวทางดำเนินเกมจะช่วยในการเขียนหรือพัฒนาโปรแกรมได้ง่ายขึ้น

โหมดการเล่นเกม

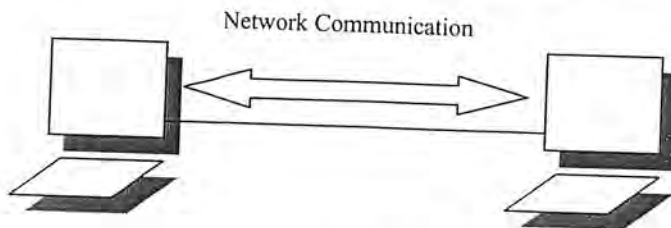
เป็นรายละเอียดของเกมในลักษณะที่ว่ามีจำนวนผู้เล่นเท่าไร เล่นคนเดียวหรือสองคน เล่นคนเดียวต่อหน้าจอหรือเล่นสองคนต่อหน้าจอ เป็นต้น

แนวความคิดพื้นฐานสำหรับการเล่นเกมแบบหลายคน

สำหรับการพัฒนาเกมบนเครือข่ายสามารถทำได้ทั้งที่เป็นแบบที่สามารถเล่นได้พร้อมกันได้หลายคนบนเครือข่าย (Network Game) หรืออาจจะไม่ต้องการติดต่อกับเครือข่ายแต่ก็สามารถเล่นเกมแบบหลายคนได้ (Non-Network Game)



รูปที่ 2-3 แสดงเกมที่เล่นสองคนบนเครื่องที่ไม่ได้ต่อกับระบบเครือข่าย



รูปที่ 2-4 แสดงเกมที่เล่นสองคนบนระบบเครือข่าย

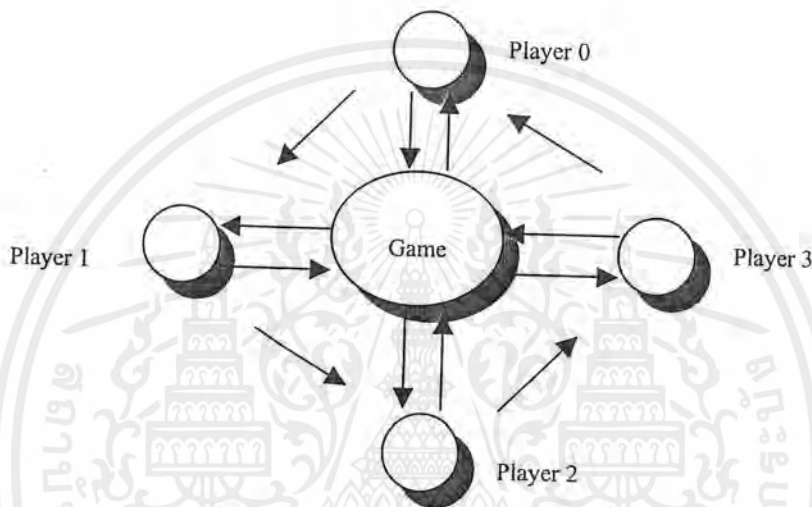
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทของเกมแบบหลายคนเล่น

ประเภทของเกมที่จะนำมาพิจารณาในตอนนี้ คือ เกมที่เล่นบนเครือข่าย (Network Games) ซึ่งแบ่งออกเป็นสองประเภท คือ

เกมแบบหมุนรอบ

เป็นเกมที่มีรูปแบบการเล่นที่ผู้เล่นจะมีการสับเปลี่ยนกันเล่นตามรูปด้านล่าง โดยที่เกมนี้สามารถทำงานบนเน็ตเวิร์คได้ไม่ยากนัก เพราะมีเพียงผู้เล่นคนเดียวที่สามารถติดต่อกับเกมในเวลานั้น และจะผลัดเปลี่ยนผู้เล่นทุกช่วงเวลาค่าหนึ่ง เกมประเภทนี้จะออกแบบให้ผู้เล่นส่วนใหญ่ต้องรองอนกว่าจะถึงตาที่ตัวเองเป็นผู้เล่น

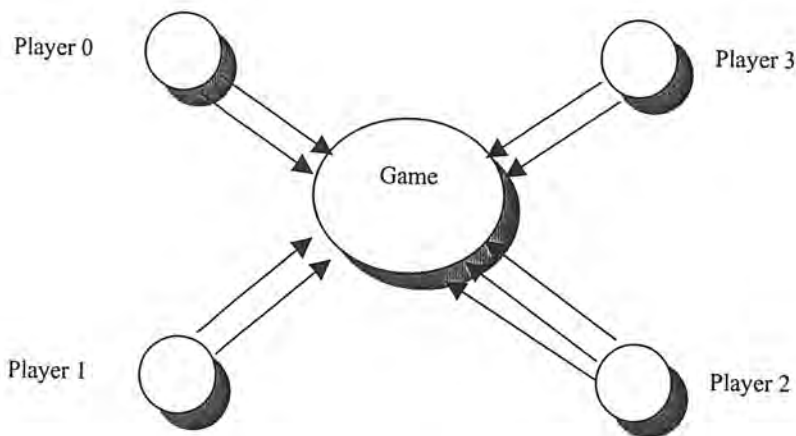


รูปที่ 2-5 แสดงเกมแบบหมุนรอบกรณีมีผู้เล่น 4 คน

เกมแบบเหตุการณ์

เป็นเกมประเภทที่ถูกบังคับโดยเหตุการณ์ที่ป้อนเข้ามาในเวลาใด ๆ ซึ่งจะเห็นได้ว่ารูปแบบของเกมนี้ จะไม่มีการหมุนหรือสับเปลี่ยนผู้เล่นเหมือนกับเกมแบบหมุนรอบ แต่เกมจะดำเนินไปเมื่อมีการป้อนอินพุตมาเท่านั้น

รูปแบบของเกมประเภทนี้มีความยุ่งยากซับซ้อนกว่าเกมแบบหมุนรอบ เป็นต้นว่า การใช้ช่องสื่อสารของเกมต้องการแบนด์วิธที่กว้างกว่า เพราะจำเป็นต้องใช้ในการส่งข้อมูลจำนวนมาก และนอกจากนี้ เกมประเภทที่สร้างขึ้นมาโดยแบบนี้จะเจอกับปัญหาที่ต่างกันออกไป และในการเล่นเกมจะต้องมีการโต้ตอบระหว่างผู้เล่นด้วย

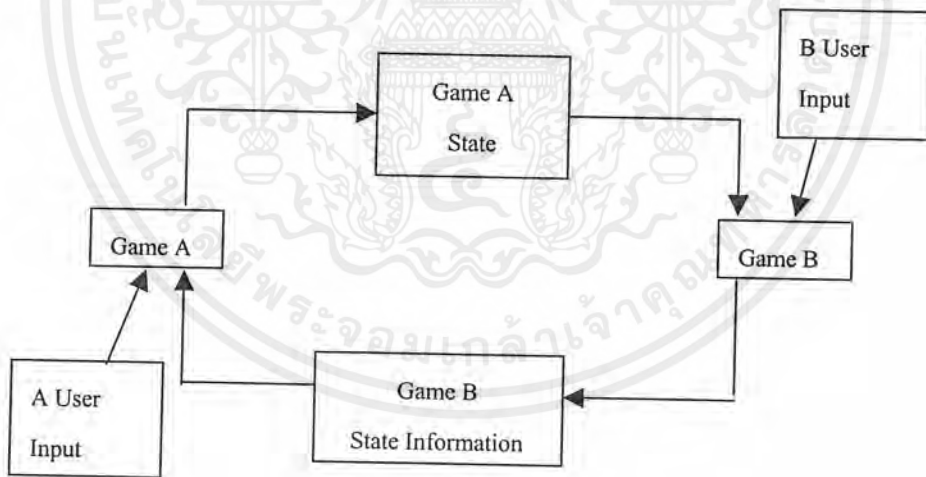


รูปที่ 2-6 แสดงเกมแบบเหตุการณ์ที่มีผู้เล่น 4 คน

ปัญหาที่อาจเกิดขึ้นในเกมแบบเครือข่ายและวิธีการแก้ปัญหา

ปัญหาที่เกิดขึ้นส่วนใหญ่จะเป็นปัญหาที่เกี่ยวกับการซิงโครไนซ์เซชัน (Synchronization) ซึ่งการซิงโครไนซ์เซชันเป็นการระบุเกี่ยวกับจำนวนของอินสแตนซ์ของเกมที่มีข้อมูลอยู่ในสถานะเดียวกัน เพื่อให้ผู้เล่นแต่ละคนทำงานได้เข้าจังหวะกันได้อย่างเหมาะสม

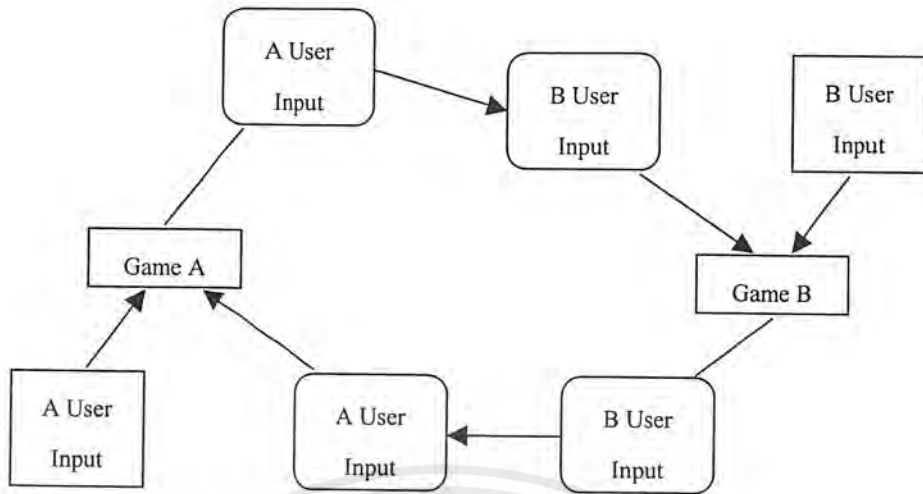
ส่วนการแก้ปัญหานั้น ใช้วิธีสเตทซิงโครไนซ์เซชัน (State Synchronization) ซึ่งจะเป็นวิธีการสื่อสารโดยที่แต่ละอินสแตนซ์จะสื่อสารสถานะปัจจุบันของตัวเองให้กับอินสแตนซ์อื่นตามรูป



รูปที่ 2-7 แสดงวิธีสเตทซิงโครไนซ์เซชันที่มั่งผ่านข้อมูลระหว่างผู้เล่นสองคนในเกมแบบเครือข่าย

หรือนอกจากนี้อาจจะใช้วิธีอินพุตซิงโครไนซ์เซชัน (Input Synchronization) ซึ่งเป็นวิธีการสื่อสารโดยที่แต่ละอินพุตของผู้เล่นจะสื่อสารไปยังอินสแตนซ์อื่นในเกม ตามรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-8 แสดงวิธีอินพุตซึ่งโครในซ์เซชันในเกมแบบเครือข่ายที่เล่นสองคน

2.3 ความรู้เบื้องต้นเกี่ยวกับเครือข่าย

2.3.1 ระบบเครือข่าย และการโปรแกรมบนระบบเครือข่าย

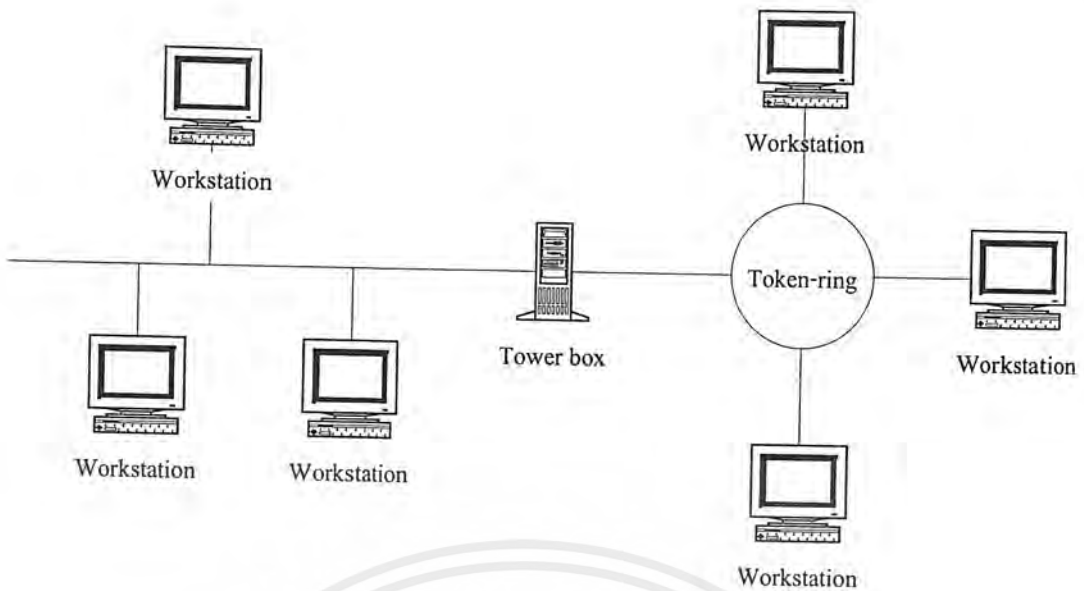
การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายจำเป็นต้องมีความรู้พื้นฐานทางด้านระบบเครือข่ายพอสมควร ในหัวข้อนี้จะอธิบายคำศัพท์พื้นฐานและหลักการของระบบเครือข่ายที่จำเป็นสำหรับการทำความเข้าใจในการพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายดังจะกล่าวต่อไป

2.3.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์ และอินเทอร์เน็ตเวิร์กกิง (Internet-working)

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) คือระบบการเชื่อมต่อระหว่างระบบปลายทาง (End-System) ซึ่งระบบปลายทางเป็นระบบที่เป็นอิสระจากกัน (Autonomous) ระบบปลายทางสามารถเป็นไปได้ตั้งแต่ไมโครคอมพิวเตอร์ไปจนกระทั่งซูเปอร์คอมพิวเตอร์ขนาดใหญ่ เพื่อจุดมุ่งหมายในการแลกเปลี่ยนข้อมูลและการแบ่งปันทรัพยากรของระบบ เช่น ไฟล์ (File) ข้อมูล , เครื่องพิมพ์ (Printer) , โมเด็ม (Modem) ตลอดจนการให้บริการฐาน ข้อมูลร่วม (Sharing database)

อินเทอร์เน็ตเวิร์กกิง หรือ อินเทอร์เน็ต คือ การเชื่อมต่อของระบบเครือข่าย 2 เครือข่ายขึ้นไป ดังนั้นคอมพิวเตอร์บนระบบเครือข่ายหนึ่งก็สามารถติดต่อกับคอมพิวเตอร์บนระบบเครือข่ายอื่น ๆ ได้ เช่น เครือข่ายโทเคนริง (Tokenring network) เชื่อมกับ เครือข่ายอีเทอร์เน็ต (Ethernet network) โดยมีเกตเวย์ (Gateway) เป็นตัวเชื่อมดังรูปที่ 2-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-9 ระบบเครือข่ายคอมพิวเตอร์เบื้องต้น

2.3.1.2 รูปแบบของโปรแกรมบนระบบเครือข่าย (Network Programming Models)

จากหัวข้อ 2.3.1.1 ได้ให้ความหมายของระบบเครือข่าย แสดงถึงวิธีที่ระบบคอมพิวเตอร์ใดๆ จะทำการเชื่อมต่อกับระบบเครือข่าย แต่ว่าคอมพิวเตอร์ที่เชื่อมต่ออยู่กับระบบเครือข่ายนั้น มีวิธีการอย่างไรในการแลกเปลี่ยนข้อมูล และแบ่งปันทรัพยากรเหล่านั้นได้ ทำให้ต้องมีโปรแกรมประยุกต์ ซึ่งสามารถที่จะจัดการในสิ่งที่กล่าวมาแล้วอย่างเหมาะสม ซึ่งในหัวข้อนี้จะกล่าวถึงรูปแบบของโปรแกรมบนระบบเครือข่าย 2 รูปแบบ คือ ไคลเอนต์เซิร์ฟเวอร์ (Client/Server Computing) และการประมวลผลแบบกระจาย (Distributed Computing)

2.3.1.2.1 การประมวลผลแบบไคลเอนต์เซิร์ฟเวอร์ (Client/Server Computing)

การประมวลผลแบบไคลเอนต์เซิร์ฟเวอร์นี้ การประมวลผลของโปรแกรมประยุกต์จะแบ่งออกเป็น 2 ส่วน คือ ส่วนฟรอนต์เอนด์ (front-end) ที่ทำงานบนไคลเอนต์ ส่วนนี้จะทำหน้าที่แสดงผลที่ได้จากการประมวลผลและรับข้อมูลจากผู้ใช้ อีกส่วนหนึ่งจากแบ็คเอนด์ (back-end) ทำงานบนเซิร์ฟเวอร์มีหน้าที่ในการเก็บรวบรวมและจัดการข้อมูลจากฟรอนต์เอนด์ในรูปแบบการประมวลผลไคลเอนต์เซิร์ฟเวอร์นี้เครื่องเซิร์ฟเวอร์มักจะเป็นเครื่องที่มีความสามารถสูงกว่าเครื่องไคลเอนต์ โดยปกติเครื่องเซิร์ฟเวอร์มักจะเป็นเครื่องเมนเฟรมหรือมินิคอมพิวเตอร์และเครื่องไคลเอนต์มักจะเป็นเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งการติดต่อกันระหว่างส่วนฟรอนต์เอนด์และแบ็คเอนด์ทำโดยผ่านระบบเครือข่าย ซึ่งในทางปฏิบัติแล้วในส่วนแบ็คเอนด์ที่อยู่บนเซิร์ฟเวอร์จะเป็นฝ่ายให้บริการแก่งานของไคลเอนต์หลายงานในเวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.2.2 การประมวลผลแบบกระจาย (Distributed Computing)

การประมวลผลของโปรแกรมประยุกต์มีรูปแบบการประมวลผล 2 แบบ คือ

1. ฟรีคอลลีเกชัน เป็นลักษณะการทำงานที่ข้อมูลที่ต้องการในการจัดเก็บและส่งต่อไปที่ระบบเครือข่ายตลอดเวลาอย่างสม่ำเสมอ การทำงานในลักษณะนี้จะเหมาะสมกับงานบางงาน เช่น ต้องการเก็บสถานะของคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายหนึ่งๆ ทุกเครื่อง
2. การประมวลผลแบบขนาน การประมวลผลในลักษณะนี้งานใด ๆ จะถูกประมวลผลด้วยคอมพิวเตอร์หลายๆ เครื่อง โดยเครื่องคอมพิวเตอร์เหล่านั้นสามารถจะติดต่อกันโดยระบบเครือข่าย เช่น การพัฒนาโปรแกรมประยุกต์ขนาดใหญ่โดยทีมพัฒนาที่มีผู้พัฒนาหลายคนสามารถลดเวลาของการแปลและการรวมโมดูล (module) ต่างๆ เข้าเป็นโปรแกรมเดียวกันโดยการแบ่งงานการแปลโมดูลเหล่านั้นแก่คอมพิวเตอร์ในระบบเครือข่ายทำการแปลในเวลาเดียวกัน

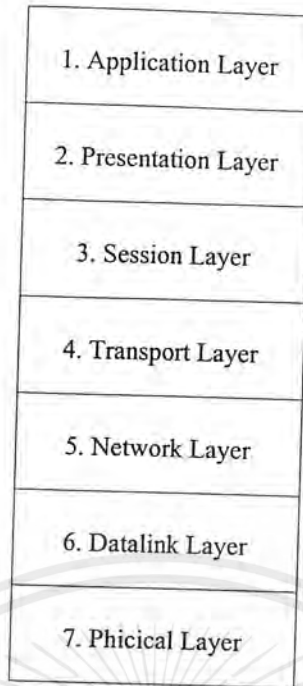
2.3.1.3 โมดูล OSI

OSI เป็นคำย่อที่มาจากคำว่า Open Systems Interconnection โดยที่เป็นมาตรฐานที่ถูกเสนอขึ้นโดย International Standards Organization ซึ่งเป็นองค์กรที่จัดตั้งขึ้นมาเพื่อดูแลและส่งเสริม ตลอดจนกำหนดมาตรฐานของการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์ โดยโมเดล OSI นี้มีลักษณะเป็นสถาปัตยกรรมแบบระบบเปิดเพราะมุ่งที่จะให้ระบบคอมพิวเตอร์ในหลายๆ รูปแบบที่แตกต่างกันสามารถเชื่อมต่อกันได้ OSI โมเดลได้แบ่งโปรโตคอลในการสื่อสารออกเป็น 7 เลเยอร์ ซึ่งโปรโตคอลคือชุดของกฎหรือข้อตกลงในการติดต่อ ข้อสังเกตโมเดล OSI เป็นเพียงข้อเสนอแนะมิใช่ข้อกำหนดและควรรู้ว่ายังไม่มีการเชื่อมต่อใดที่สร้างเหมือนกัน โมเดล OSI จริงๆ

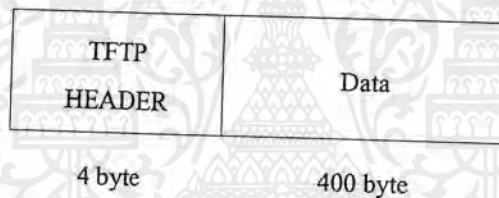
ในหนึ่งชั้นของเลเยอร์ไม่ได้กำหนดว่าจะต้องมีเพียงหนึ่งโปรโตคอลเท่านั้นที่อยู่ในระดับเลเยอร์เดียวกันและในทางตรงข้ามชุดของโปรโตคอลใด ๆ อาจจะมีมากกว่าหนึ่งเลเยอร์ประกอบกันเป็นข้อกำหนดของระบบเครือข่ายเรียกว่าชุดโปรโตคอล เช่น ชุดโปรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นต้น ประโยชน์ในการแบ่งเป็นเลเยอร์ คือกำหนดการติดต่อระหว่างเลเยอร์ทำได้โดยไม่ต้องคำนึงถึงการเปลี่ยนเลเยอร์ใด ๆ ที่ติดกัน

2.3.1.4 เอ็นแคปซูลชัน (Encapsulation)

พิจารณาโปรแกรมประยุกต์ TFTP (File Transfer Protocol) ซึ่งใช้ในโปรโตคอล UDP (User Datagram Protocol) ระหว่างสองระบบซึ่งเชื่อมต่อกันด้วยอีเทอร์เน็ต ถ้าโปรแกรมไคลเอนต์ TFTP มีข้อมูล 400 ไบต์ต้องการส่งไปที่โปรแกรมเซิร์ฟเวอร์โปรแกรมไคลเอนต์ TFTP จะเพิ่มข่าวสารควบคุม 4 ไบต์เป็นส่วนหัวของข้อมูลก่อนที่จะผ่านข้อมูลไปสู่เลเยอร์ UDP การเพิ่มของข่าวสารควบคุมไปที่ข้อมูลเรียกว่า เอ็นแคปซูลชัน ดังแสดงในรูปที่ 2-11



รูปที่ 2-10 OSI โมเดล ทั้ง 7 เลเยอร์



รูปที่ 2-11 การเอ็นแคปซูลชั้น

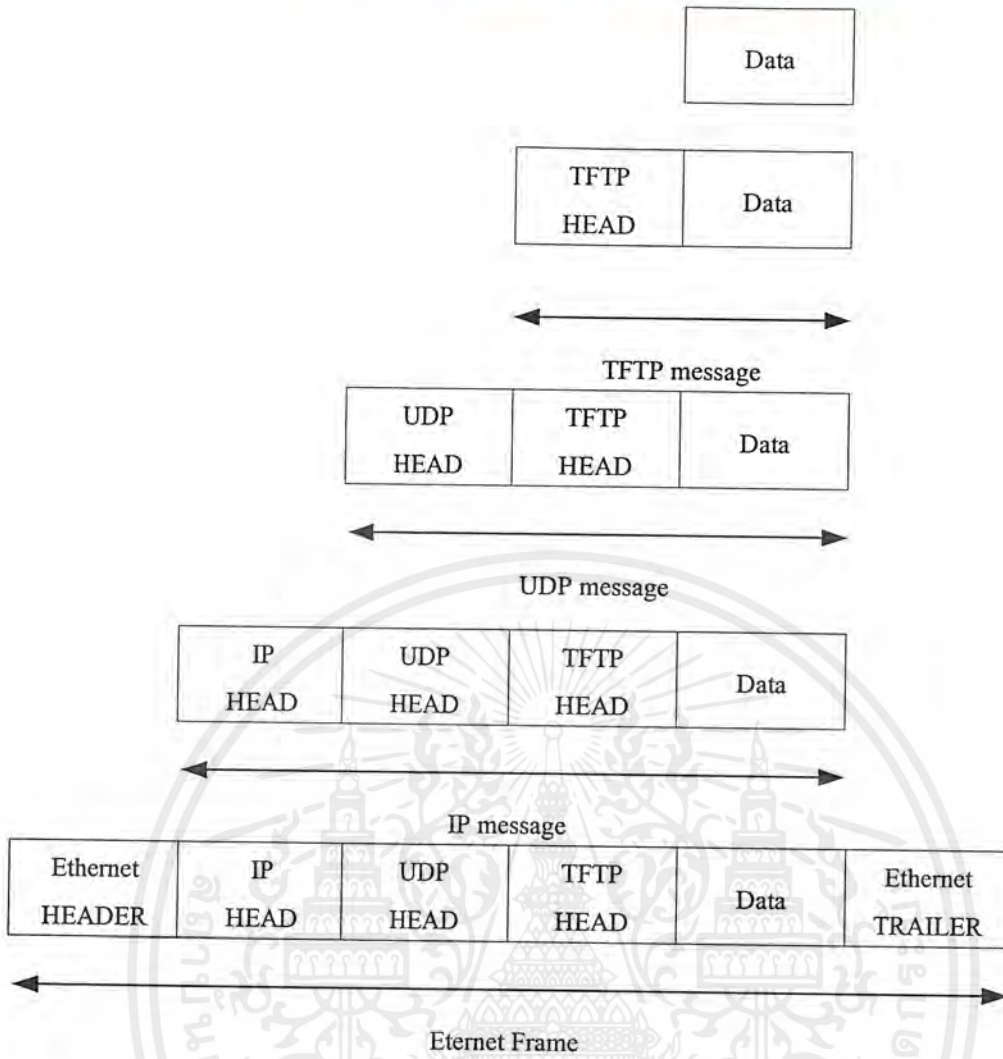
เลเยอร์ UDP จะไม่มีการตีความส่วนหัว TFTP 4 ไบต์ งานของเลเยอร์ UDP คือส่งข้อมูล 404 ไบต์ไปสู่เลเยอร์ UDP ของโปรแกรมอีกด้านหนึ่ง จากนั้นเลเยอร์ UDP จะทำการเพิ่มส่วนหัว 8 ไบต์ แล้วส่งข้อมูล 432 ไบต์ไปยังเลเยอร์ค่าคำศัพท์ที่เลเยอร์นี้มีการเพิ่มส่วนหัวอีก 14 ไบต์ ดังรูปที่ 2-12

2.3.1.5 ลักษณะของการติดต่อ แบ่งออกเป็น 2 ชนิดคือ

2.3.1.5.1 **Connection-oriented** คือการติดต่อที่ต้องมีการเชื่อมโปรเซสที่จะทำการติดต่อก่อนที่จะมีการส่งหรือรับข้อมูลซึ่งสามารถใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่างโปรเซส ถึงแม้ว่าข้อมูลนี้อาจจะผ่าน Packet-Switching Network บริการชนิดนี้ส่วนมากจะใช้ในกรณีที่มีข่าวสารต้องการมากกว่าหนึ่งข่าวสารดังนั้นสามารถแบ่งชิ้นการทำงานออกเป็น

- ขั้นตอนการสร้างการติดต่อ (Connection establishment)
- ขั้นตอนการส่งผ่านข้อมูล (data transfer)
- ขั้นตอนยกเลิกการติดต่อ (connection termination)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-12 ลำดับการเ็นแคปซูลชั้น

2.3.1.5.2 Connectionless หรือดาต้าแกรม (Datagram) คือจะไม่มีชั้นการสร้างการติดต่อและชั้นยกเลิกการติดต่อแต่จะมีชั้นการส่งผ่านข้อมูลอย่างเดียว โดยข้อมูลเรียกว่าดาต้าแกรมจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นกับดาต้าแกรมอื่น

1.3.1.6 แอดเดรส (Address)

การที่ระบบในระบบเครือข่ายสามารถติดต่อกันได้จำเป็นต้องมีแอดเดรสไว้คล้ายกับหมายเลขประจำตัวซึ่งลำดับของแอดเดรสสามารถพิจารณาได้ คือ

- แต่ละเครือข่ายจะต้องมีแอดเดรสสำหรับเครือข่าย
- คอมพิวเตอร์โฮสต์แต่ละเครื่องในเครือข่ายจะต้องมีแอดเดรส
- แต่ละโปรเซสในโฮสต์จะต้องมีหมายเลขประจำตัว

โดยทั่วไปแอดเดรสของโฮสต์จะประกอบด้วยหมายเลข(Network ID) และหมายเลขของโฮสต์ (Host ID)

ส่วนแอดเดรสของโปรเซสของผู้ใช้จะอยู่ในรูปจำนวนเต็มซึ่งกำหนดโดยโปรโตคอล เช่น โปรโตคอล TCP/IP จะใช้เลขจำนวนเต็มขนาด 32 บิตในการกำหนดหมายเลขเครือข่ายและหมายเลขโฮสต์ และทั้ง TCP และ UDP ใช้เลขจำนวนเต็มขนาด 16 บิตเป็นหมายเลขพอร์ตหรือหมายเลขโปรเซส

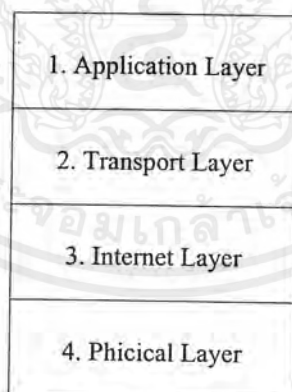
ชุดของโปรโตคอลส่วนใหญ่จะมีการกำหนดชุดของแอดเดรสสำหรับการบริการที่เป็นที่รู้จักกันโดยทั่วกัน เช่น คอมพิวเตอร์ที่โปรโตคอล TCP/IP ส่วนใหญ่จะมี FTP ซึ่งไคลเอนต์สามารถติดต่อได้โดยใช้หมายเลขพอร์ตคือ 21

2.3.2 อินเทอร์เน็ต

อินเทอร์เน็ตเป็นที่รวมของระบบเน็ตเวิร์คที่มากกว่า 1 เน็ตเวิร์คโดยมีการเชื่อมต่อพื้นฐานด้วยเน็ตเวิร์คโปรโตคอล อินเทอร์เน็ตมีโปรโตคอล TCP/IP เป็นลักษณะเฉพาะในการติดต่อของระบบเครือข่ายที่กว้างโดยประกอบด้วยไอพีแอดเดรสและโปรโตคอลที่เป็นพื้นฐานในการติดต่อระบบอินเทอร์เน็ต

2.3.2.1 เปรียบเทียบระหว่างโปรโตคอล TCP/IP และ OSI โมเดล

การออกแบบโปรโตคอล TCP/IP นั้นไม่ได้เป็นไปตามรูปแบบของ OSI โมเดลเนื่องจากถูกออกแบบโดยองค์กรขนาดใหญ่ซึ่งใช้เวลาในการออกแบบตลอดจนการรับรองมาตรฐานต่างกับโปรโตคอล TCP/IP ที่ถูกออกแบบด้วยความต้องการอันเร่งด่วนของรัฐบาลสหรัฐจึงทำให้การพัฒนาโปรโตคอล TCP/IP มีเงื่อนไขในด้านความต้องการที่ต่างจาก OSI โมเดล คือ มี 4 ชั้นเท่านั้น ดังรูปที่ 2-13 โดยแบ่งเป็น



รูปที่ 2-13 เลเยอร์ของโปรโตคอล TCP/IP

2.3.2.1.1 แอปพลิเคชันเลเยอร์

ในเลเยอร์นี้ประกอบด้วยโปรแกรมประยุกต์ที่ใช้เครือข่าย เช่น โปรแกรมส่งถ่ายข้อมูล (file-transfer program) และอาจกล่าวได้ว่าเลเยอร์นี้โปรโตคอล TCP/IP ก็คือเลเยอร์ในชั้นแอปพลิเคชันเลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวมกับชั้นพีริเซนเดชั่นเลเยอร์ใน OSI โมเดลนั่นเองและในเลเยอร์ชั้นนี้ของโปรโตคอล TCP/IP จะกลืนอยู่ในตัวโปรแกรมประยุกต์

2.3.2.1.2 ทรานสปอร์ตเลเยอร์

ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายถึงจุดปลายหากเปรียบเทียบ กับ OSI โมเดลก็สามารถเทียบได้กับชั้นเซสชันเลเยอร์ร่วมกับทรานสปอร์ตเลเยอร์นั่นเอง โดยโปรโตคอล TCP/IP มีชื่อเรียกเป็นจุดปลาย (end-point) ในการสื่อสาร ซึ่งชื่อเรียกนี้ประกอบไปด้วยหมายเลขของคอมพิวเตอร์และหมายเลขพอร์ต (port ID) ของเครื่องที่ต้องการส่งข้อมูลไปถึงในชั้นนี้มีการรับรองการถึงที่หมายและลำดับของข้อมูลที่ส่งโดยไม่ซ้ำและความผิดพลาดของข้อมูล

2.3.2.1.3 อินเทอร์เน็ตเลเยอร์

เลเยอร์นี้มีการกำหนดค่าแอดเดรสและทำการหาเส้นทางการส่ง โดยหน้าที่ของเลเยอร์นี้เทียบเท่ากับเน็ตเวิร์กเลเยอร์และดาต้าลิงก์เลเยอร์ของ OSI โมเดล

2.3.2.1.4 ฟิสิคอลลเยอร์

โปรโตคอล TCP/IP ไม่ได้กำหนดรูปแบบของการเชื่อมต่อในระดับนี้ไว้ใหม่แต่ได้ใช้มาตรฐานที่มีอยู่เดิมที่กำหนดไว้ก่อน เช่น RS232, อีเทอร์เน็ต เป็นต้น

2.3.2.2 รูปแบบการกำหนดแอดเดรสของโปรโตคอล TCP/IP

ในหัวข้อนี้จะได้อธิบายรูปแบบการกำหนดแอดเดรสของโปรโตคอล TCP/IP ซึ่งลักษณะแอดเดรสของโปรโตคอลนี้ค่าของแอดเดรสของเครื่องคอมพิวเตอร์ในระบบเครือข่ายจะไม่ซ้ำกันเลข โดยเลขนี้เรียกว่า IP แอดเดรส เป็นเลข 32 บิต IP แอดเดรสจะอยู่ในรูปของ x.x.x.x. ซึ่งแต่ละ x จะเป็น 1 ไบต์ เช่น 152.2.254.81 ทุกๆ IP แอดเดรสจะสามารถแบ่งออกเป็นเครือข่ายได้โดยแบ่งเป็นบิตของกุ่ม IP แอดเดรสเพื่อจุดประสงค์ของการดูแลและปรับปรุง ซึ่งแบ่งเป็นคลาส ตามหลักในการพิจารณาที่จะได้กล่าวต่อไป

2.3.2.2.1 การแบ่งเน็ตเวิร์กคลาส

เนื่องจากหมายเลขแอดเดรสของคอมพิวเตอร์เครื่องใด ๆ นั้นจะต้องสามารถบอกถึงความแตกต่างระหว่างตัวเครื่องเองตลอดจนถึงเครือข่ายที่คอมพิวเตอร์นั้นเชื่อมต่อด้วย หมายเลข IP แอดเดรสจึงแยกออกเป็น 2 ส่วน ได้แก่ ส่วนที่แสดงหมายเลขของคอมพิวเตอร์โฮสต์และส่วนที่เป็นหมายเลขของเครือข่าย

การแบ่งคลาสของแอดเดรสทำได้โดยพิจารณาจำนวนบิตของ 2 ส่วนประกอบข้างต้น ซึ่งมีการแบ่งออกเป็น 5 คลาสแต่มีการใช้เพียง 3 คลาสแรก คือ คลาส A, คลาส B, คลาส C, คลาส D และ คลาส E ถูกสงวนไว้สำหรับจุดประสงค์พิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส A : 1-126 (16M โฮสต์ในแต่ละอัน)

คลาส B : 128-191 (65536 โฮสต์ในแต่ละอัน)

คลาส C : 192-223 (256 โฮสต์ในแต่ละอัน)

คลาส D : 224-239 (โหมด multicast)

คลาส E : 240-255 (สำรองไว้สำหรับใช้ในอนาคต)

CLASS ID	NETWORK ID	HOST ID
----------	------------	---------

IP Adress Format

รูปที่ 2-14 หมายเลข IP แอดเดรส

Network Class	Networks	Hosts / Network
A	126	16,777,214
B	16,328	65,564
C	2,097,150	254

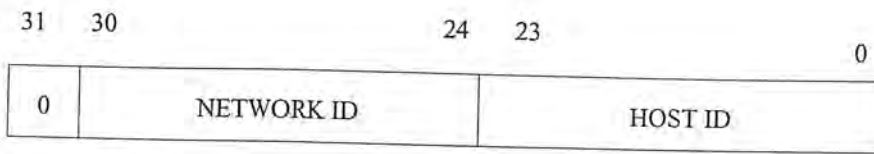
รูปที่ 2-15 แสดงคลาส จำนวนเครือข่าย และจำนวนโฮสต์ของแต่ละคลาส

โดยปกติแล้วผู้พัฒนาโปรแกรมไม่ต้องสนใจความแตกต่างระหว่างคลาสของ IP แอดเดรส

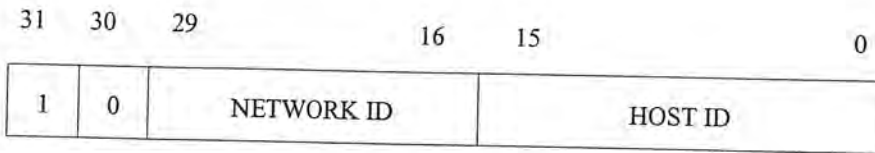
2.3.2.2.2 การแทนด้วยเลขฐานสิบ และจุด (Dotted Decimal Notation)

เนื่องจากการแทนหมายเลข IP แอดเดรสเป็นเลขฐาน 2 ซึ่งค่อนข้างอ่านไม่สะดวกจึงมีการแทนเลขฐานสองเหล่านั้นอยู่ในรูปเลขฐานสิบและจุด โดยเลขฐานสิบแต่ละตัวจะแทนเลขฐานสอง จำนวน 8 บิต โดยระหว่างเลขฐานสิบแต่ละตัวจะแทรกด้วยจุด ดังนั้นจะต้องใช้เลขฐานสิบ 4 ตัวในการแทนเลข 32 บิตที่เป็น IP แอดเดรส ดังตัวอย่างตามรูปที่ 2-17 จะสังเกตว่าหมายเลข IP แอดเดรสนี้จัดอยู่ในคลาส B โดยหมายเลขของเครือข่าย 166.78 และหมายเลขประจำตัวเครื่อง 4.139

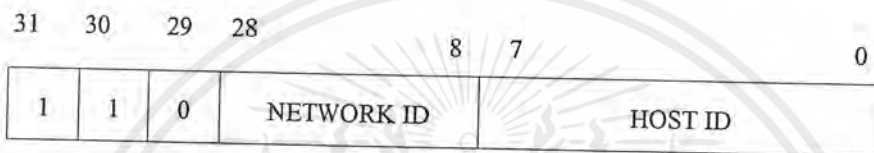
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Class A IP Address Format

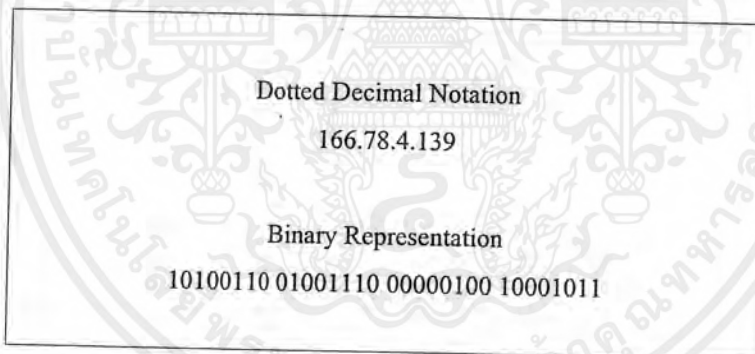


Class B IP Address Format



Class C IP Address Format

รูปที่ 2-16 แสดงรูปแบบของ IP แอดเดรสในคลาสต่าง ๆ



รูปที่ 2-17 ตัวอย่างหมายเลข IP แอดเดรสทั้งสองแบบ

2.3.2.2.3 การทำซับเน็ตติ้ง (Subnetting)

IP จะใช้ประโยชน์ในซับเน็ต ซึ่งจะเป็นการแบ่งทางลอจิกของเครือข่ายให้เป็นเครือข่ายขนาดเล็ก ๆ ซึ่งแต่ละเครือข่ายจะประกอบด้วยขอบเขตของแอดเดรสจาก IP ต้นแบบ ซับเน็ตเป็นสิ่งจำเป็นเพื่อที่จะออกแบบเครือข่าย IP ซึ่งมันจะทำมาจากมากกว่า 1 broadcast base network เช่นอีเทอร์เน็ตแต่ละซับเน็ตจะบรรจุ broadcast แอดเดรสของมัน

การทำซับเน็ตติ้งเป็นการเปลี่ยนแปลงการใช้หมายเลขของเครื่องโฮสต์ และหมายเลขของเครือข่ายในระดับท้องถิ่น โดยในทางตรรกคือการเส้นแบ่งที่แยกหมายเลขเครื่อง และหมายเลขของเน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวิร์กที่อยู่ในหมายเลข IP แอดเดรส โดยที่ปริมาณของหมายเลขเครื่องโฮสต์ และหมายเลขเครือข่ายจะแปรผกผันกัน เช่น หากมีปริมาณของเน็ตเวิร์กมาก ก็จะทำให้เครื่องใด ๆ ที่ต่อกับระบบเครือข่ายหนึ่ง ๆ จะน้อยลงเป็นต้น ในทางปฏิบัติการทำซับเน็ตติ้งทำโดยการนำซับเน็ตติ้งมาร์ค (subnet mark) คือตัวเลขจำนวน 32 บิต มาทำการกระทำตรรกและ (AND) กับหมายเลข IP แอดเดรส ดังตัวอย่างโดยกำหนดหมายเลข IP แอดเดรส คือ 166.78.4.139 และซับเน็ตมาร์ค คือ 255.255.255.0 ทำการกระทำตรรกและดังรูปที่ 2-18

166.78.4.139

AND

255.255.255.0

166.78.4.0

รูปที่ 2-18 การทำซับเน็ตติ้ง

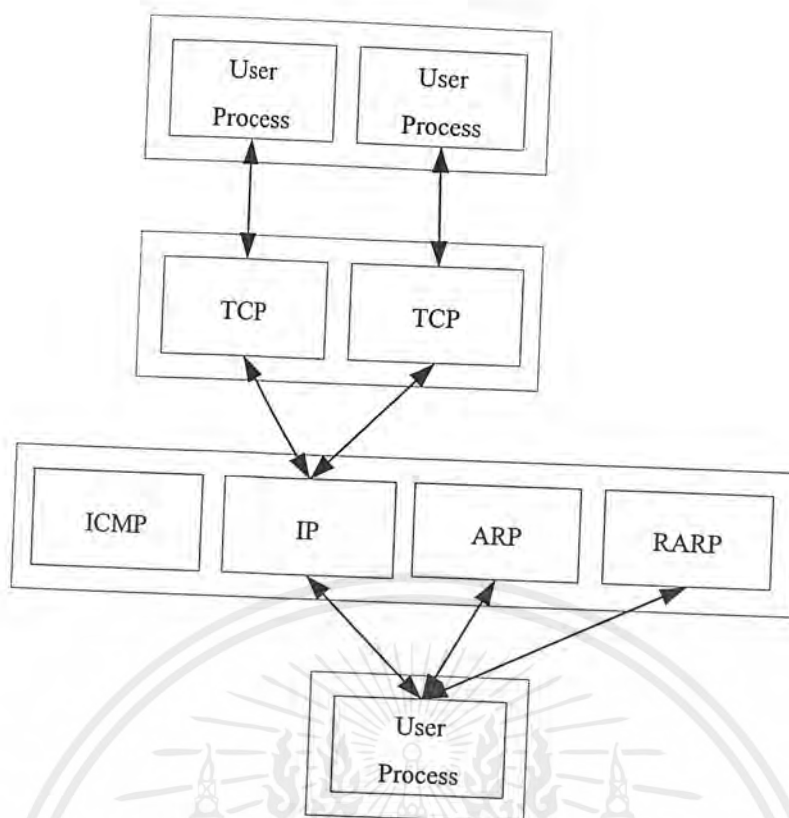
จะเห็นว่าหากพิจารณาโดยไม่มีการทำซับเน็ตติ้งแล้วจะได้หมายเลขของเน็ตเวิร์คคือ 166.78 และหมายเลขประจำเครื่องคือ 4.139 แต่ผลที่ได้จากการกระทำตรรกและในรูปข้างต้นผลลัพธ์นี้คือหมายเลขเน็ตเวิร์คคือ 166.78.4.0 และหมายเลขเครื่องคือ 139 หรืออาจพูดได้ว่าคอมพิวเตอร์เครื่องนี้มีหมายเลขเครื่องเท่ากับ 139 และอยู่บนเครือข่ายย่อยหมายเลข 166.78.4

2.3.2.3 ชุดโปรโตคอล TCP/IP

ชุดโปรโตคอล TCP/IP นอกจากมีโปรโตคอล TCP และ IP แล้วยังมีโปรโตคอลอย่างอื่นอีกดังรูปที่ 2-19 แสดงความสัมพันธ์ของชุดโปรโตคอลโดยแบ่งตามเลเยอร์

2.3.2.3.1 โปรโตคอล IP (Internet Protocol)

โปรโตคอล IP เป็นโปรโตคอลแบบคอนเนกชันเลส (Connectionless protocol) โดยที่โปรโตคอล IP ไม่รับประกันว่าข้อมูลที่ส่งจะไปถึงปลายทางซึ่งแพคเกจของข้อมูลอาจไปถึงในลักษณะที่ผิดลำดับซ้ำกัน หรือไม่ไปถึงเลย โดยความน่าเชื่อถือของการส่งจะถูกควบคุมในโปรโตคอลเลเยอร์ต่าง ๆ ไป การหาเส้นทางของข้อมูลจะทำในระดับของโปรโตคอล IP นี้ โดยพิจารณาแต่ละแพคเกจแยกออกจากกัน และยังมีหน้าที่ในการจัดเรียงข้อมูลใหม่ที่ปลายทางอีกด้วย



รูปที่ 2-19 ความสัมพันธ์ระหว่างชุดโปรโตคอล

2.3.2.3.2 โปรโตคอล ARP (Address Resolution Protocol)

โปรโตคอลนี้ทำหน้าที่จับคู่ระหว่างหมายเลข IP แอดเดรสเข้ากับหมายเลขแอดเดรสทางฮาร์ดแวร์ โดยโปรโตคอลนี้ทำการส่งข้อความไปทั่วเครือข่ายท้องถิ่น ซึ่งข้อความนี้เป็นลักษณะข้อความที่ตรวจสอบว่ามีคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกันที่ต้องการหรือไม่ หากคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกันนั้นได้รับข้อความนี้ก็จะตอบกลับและเป็นที่น่าสังเกตว่าโปรโตคอลนี้ทำงานได้กับระบบเครือข่ายท้องถิ่นเท่านั้นเพราะว่าโครงสร้างหมายเลขทางฮาร์ดแวร์ของเครื่องคอมพิวเตอร์จะขึ้นอยู่กับชนิดของระบบเครือข่าย

2.3.2.3.3 โปรโตคอล RARP (Reverse Address Resolution Protocol)

เป็นโปรโตคอลที่ทำหน้าที่จับคู่ระหว่างหมายเลขของฮาร์ดแวร์กับหมายเลข IP แอดเดรสหรือทำงานกลับกันเป็นโปรโตคอล ARP

2.2.3.4 IP Routing

ตามแนวความคิดเครือข่าย IP จะอยู่บนพื้นฐานของโฮสต์และ routers อุปกรณ์โฮสต์ IP จะรวบรวมและทำการส่งควบคุมการไหล การตรวจสอบความผิดพลาด (ถ้ามี) และกระบวนการข้อมูลอื่น ๆ router เป็นตัวหาเส้นทางที่ดีที่สุดเพื่อใช้ในการติดต่อของระบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎเกณฑ์นำไปสู่ IP คือความสามารถในการหาเส้นทางของมัน IP Router มีหลาย ๆ เครื่องช่วยเชื่อมต่อและตรวจสอบแต่ละแพ็กเกจ มันจะคำนวณเส้นทางที่มีราคาน้อยที่สุดที่จะไปยังปลายทางโดยวิธีโปรโตคอล routing ของมัน และส่งแพ็กเกจเส้นทางที่เหมาะสมไปสู่ปลายทาง router อย่างง่ายไม่ต้องการความรู้ทั้งหมดของเครือข่าย แต่ต้องรู้การเชื่อมต่อท้องถิ่นของมัน แพ็กเกจควรจะถูกส่งไปยังหน้าบนพื้นฐานแอดเดรสปลายทาง

2.3.2.3.5 โปรโตคอล UDP

โปรโตคอลนี้เป็นโปรโตคอลที่อยู่ในระดับทรานสปอร์ตเลเยอร์และมีความสำคัญเพราะว่าเป็นโปรโตคอลที่ผู้พัฒนาโปรแกรมสามารถใช้ได้โดยตรง โปรโตคอลนี้เป็นโปรโตคอลแบบคอนเนกชันเลสซึ่งมีความน่าเชื่อถือต่ำ ไม่รับรองว่าข้อมูลที่ส่งไปจะถึงปลายทางหรือไม่ และอาจซ้ำซ้อนหรือผิดพลาดได้ แต่ข้อดีของโปรโตคอลนี้คือค่าความสิ้นเปลือง (overhead) ที่ต่ำ

2.3.2.3.6 โปรโตคอล TCP

โปรโตคอลนี้อยู่ในระดับทรานสปอร์ตเลเยอร์เหมือนกับโปรโตคอล UDP แต่มีลักษณะที่ตรงข้ามกันคือเป็นโปรโตคอลแบบคอนเนกชันออเรียนเต็ด โดยที่มีความน่าเชื่อถือในการรับส่งข้อมูล และลำดับของข้อมูลจะมีลำดับเหมือนกับต้นทางและเนื้อหาข้อมูลไม่ผิดพลาดจึงทำให้เกิดความสิ้นเปลืองในการเชื่อมต่อของการส่งข้อมูลมากกว่าโปรโตคอล UDP

2.3.2.4 หมายเลขพอร์ต

เนื่องจากในเวลาใด ๆ สามารถมีโพรเซสของผู้ใช้สามารถใช้ UDP หรือ TCP ได้พร้อมกันดังนั้นจึงต้องมีวิธีแยกแยะว่าข้อมูลเป็นของผู้ใช้คนใด ซึ่งวิธีที่ TCP และ UDP ใช้คือการใช้หมายเลขพอร์ต

เมื่อโพรเซสไคลเอนต์ต้องการที่จะติดต่อกับเซิร์ฟเวอร์ ไคลเอนต์จะต้องเจาะจงที่เซิร์ฟเวอร์ที่ต้องการติดต่อกแต่ละฝั่งรูแอดเดรสอินเตอร์เน็ต 32 บิตเพียงอย่างเดียวนั้นไม่พอ เพราะสามารถติดต่อกับโฮสต์ได้เพียงอย่างเดียวแต่ไม่สามารถเจาะจงโพรเซสที่จะทำการติดต่อกได้ ดังนั้นเพื่อแก้ปัญหาที่ทั้ง TCP และ UDP ได้มีการกำหนดหมายเลขพอร์ตมาตรฐาน (well-know ports) ซึ่งไคลเอนต์เล็กใช้หมายเลขพอร์ตนี้แล้ว สามารถกำหนดหมายเลขพอร์ตนี้ให้ไคลเอนต์อื่นได้ โพรเซสที่รับมอบหมายเลขพอร์ตชั่วคราวนี้จะไม่สนใจว่ามีค่าเท่าไร แต่เป็นหน้าที่ของอีกโพรเซสหนึ่งที่ต่อกันที่ต้อสนใจ เพราะต้องส่งข้อมูลกลับมาที่พอร์ตนี้ใน TCP และ UDP นั้นหมายเลขพอร์ตตั้งแต่ 1-1023 เป็นพอร์ตที่สงวนไว้สำหรับหมายเลขพอร์ตมาตรฐาน

2.4 ภาษาจาวา

ทุกวันนี้ internet กำลังเติบโตเป็นอย่างมากซึ่งนำไปใช้งานในด้านต่าง ๆ ทั้งธุรกิจการค้า การเรียนการสอน การติดต่อสื่อสาร และอื่น ๆ อีกมาก ซึ่งในการพัฒนาแอปพลิเคชันบนอินเตอร์เน็ตนั้นก็มีโปรแกรมต่าง ๆ มากมายที่สนับสนุน แต่โปรแกรมที่มีความสามารถมากที่สุดตัวหนึ่งบนอินเตอร์เน็ตคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่สร้างขึ้นมาจากภาษาจาวา ซึ่งถูกพัฒนาโดยบริษัทซันไมโครซิสเต็มส์ซึ่งจุดประสงค์เริ่มแรกคือพัฒนาขึ้นมาเพื่อทำการควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้โดยไม่ขึ้นอยู่กับแพลตฟอร์ม (platform) ที่ใช้คือการพัฒนาซอฟต์แวร์ให้สามารถทำงานได้โดยไม่ต้องสนใจว่าฮาร์ดแวร์ที่ใช้มีลักษณะอย่างไร นอกจากนี้จาวายังสนับสนุนความสามารถในการเชื่อมต่อเครือข่าย การติดต่อสื่อสารผ่านทางระบบเครือข่ายคอมพิวเตอร์ด้วย

คุณสมบัติทั่วไปของภาษาจาวามีดังนี้

1. เขียนง่าย เนื่องจากจาวามีลักษณะเป็นภาษาคอมพิวเตอร์ระดับสูงแต่ได้ตัดบางคุณลักษณะที่ไม่จำเป็นในภาษาระดับสูงออกไป เช่น ไม่สนับสนุนพอยเตอร์ (point math), การไหลคเกินของตัวดำเนินการ (operator overloading), การสืบทอดจากคลาสหลายคลาส (multiple inheritance)
2. มีชนิดเป็นสแตติก คือ ออบเจกต์ทั้งหมดที่ถูกใช้ในโปรแกรมต้องถูกกำหนดก่อนที่มันจะถูกใช้งาน ทำให้ตัวคอมไพเลอร์ของจาวาสามารถรายงานชนิดที่ขัดแย้งกันได้
3. เมื่อคอมไพล์โปรแกรมแล้วผลลัพธ์จะเป็นไฟล์ข้อมูลแบบไบนารีโค้ด (ซึ่งเหมือนกับ machine code) ซึ่งมันสามารถถูกเอ็กซีคิวต์ภายใต้ระบบปฏิบัติการด้วย Java Interpreter ซึ่งตัว Interpreter จะอ่านไฟล์ที่เป็นไบนารีโค้ด แล้วจะแปลงคำสั่งที่เป็นไบนารีโค้ดไปเป็นคำสั่งที่เป็นภาษาเครื่อง (machine-language commands) ซึ่งจะถูกเอ็กซีคิวต์ได้โดยตรงโดยเครื่องที่กำลังรันโปรแกรมจาวาอยู่ เพราะฉะนั้นเราจะกล่าวได้ว่าจาวาเป็นทั้งตัว compile และ interpret
4. สนับสนุนการทำงานได้หลาย ๆ งานในเวลาเดียวกัน
5. โปรแกรมจาวาจะจัดการกับขยะที่เกิดจากโปรแกรมด้วยตัวมันเอง ซึ่งหมายความว่าโปรแกรมจะไม่ต้องการที่จะลบออบเจกต์ซึ่งอยู่ในหน่วยความจำ
6. ความมั่นคง (Robust) เพราะว่า Java Interpreter จะตรวจสอบระบบทั้งหมดที่จะถูกเข้าถึงโดยโปรแกรม ดังนั้นจึงทำให้มั่นใจได้ว่าโปรแกรมจาวาจะไม่ทำให้ระบบพัง และถ้าเกิดความผิดพลาดขึ้นจะมี throw exception ออกมาซึ่งตัว exception นี้จะถูกจับและจัดการโดยโปรแกรมเพื่อทำให้ไม่เกิดความเสียหายกับระบบ
7. ความปลอดภัย ระบบจาวาไม่ใช่แค่รับรองการทำงานในการเข้าถึงหน่วยความจำเท่านั้น แต่ยังทำให้แน่ใจได้ว่าจะไม่มีการรันโปรแกรม เพราะพอยเตอร์ไม่ถูกใช้ในภาษาจาวา
8. มีความยืดหยุ่น โปรแกรมจาวาสนับสนุน native method (ซึ่งเป็นฟังก์ชันที่ถูกเขียนขึ้นโดยภาษาอื่น ๆ เช่น C++) โดยจะทำให้โปรแกรมเมอร์นั้นเขียนฟังก์ชันซึ่งอาจจะถูกเอ็กซีคิวต์ได้เร็วกว่าฟังก์ชันเดียวกันในการเขียนด้วยภาษาจาวา โดย native method เชื่อมต่อแบบไดนามิกกับ (dynamically link) กับโปรแกรมจาวาแต่เมื่อใดที่ภาษาจาวาแก้ไขในเรื่องความเร็วไปก็ไม่มีผลจำเป็นต้องใช้ native method อีกต่อไป
9. เข้าใจง่าย

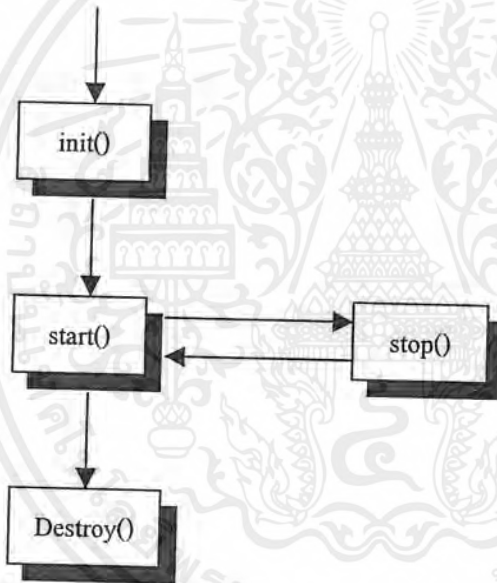
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 จาวาแอปเพล็ต

ความสำเร็จของจาวาส่วนหนึ่งมาจากความสามารถของแอปเพล็ต การเขียนแอปเพล็ตก็มีการเขียนเช่นเดียวกับการเขียนจาวา ซึ่งอาศัยหลักการพื้นฐานจากจาวา แอปเพล็ตสามารถรันได้จากเว็บเบราว์เซอร์ที่สนับสนุนการทำงานกับจาวา (Java - Enabled) ทั้งสามารถโต้ตอบและแสดงอนิเมชันบน HTML ซึ่งกล่าวได้ว่าแอปเพล็ตเป็นตัวทำให้จาวามีเสน่ห์ ดึงดูดและเป็นที่น่าสนใจอย่างแพร่หลาย

คลาส Applet

จาวาแอปเพล็ตต้องทำการ extends java.applet.Applet โดยที่แอปเพล็ตจะจัดหาโครงสร้างการทำงานที่สำคัญให้กับเราในการรันบนเบราว์เซอร์จาวาแอปเพล็ตจะมีเมธอด main() ที่จะทำการรันเมื่อแอปเพล็ตเริ่มทำงาน ต่างจากแอปเพล็ตซึ่งไม่จำเป็นต้องมีเมธอด main() โดยที่เมธอดในแอปเพล็ตจะขึ้นอยู่กับการทำงานของเบราว์เซอร์ ซึ่งมีโครงสร้างการทำงานดังรูป



รูปที่ 2-20 การทำงานของเว็บเบราว์เซอร์ที่ควบคุมการทำงานของเมธอดต่างๆ ในแอปเพล็ต

เบราว์เซอร์จะเป็นตัวควบคุมการเรียกเมธอด `init()`, `start()`, `stop()` และ `destroy()` โดยปกติแล้วเมธอดเหล่านี้จะไม่เกิดการ ทำงานใดๆ เว้นแต่ต้องการทำงานเฉพาะบางอย่าง ซึ่งทำได้โดยการเขียนโค้ดลงให้เหมาะสมกับเมธอดที่ต้องการจะทำงานด้วย

วงจรชีวิตของแอปเพล็ต (Applet Life Cycle)

วงจรชีวิตในที่นี้จะหมายถึง วงจรที่แอปเพล็ตต้องทำงานกับเมธอดต่างๆ ที่ตัวแอปเพล็ตอ้างอิงด้วย ซึ่งปกติจะมีการทำงานวนเวียนดังนี้

1. ตั้งค่าเริ่มต้นให้ตัวเอง (initialize)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เริ่มรันแอปพลิเคชัน (start)
3. หยุดรันแอปพลิเคชัน (stop)
4. ทำลายตัวเองเพื่อเตรียมจะหยุดการทำงานจนกว่าจะมีการเรียกใช้ใหม่ (destroy)

ซึ่งอธิบายได้ว่าแอปพลิเคชันจะเริ่มต้นจากการตั้งค่าเริ่มต้นให้ตัวเอง แล้วจบลงด้วยการทำลายตัวเอง แต่ในขณะที่ทำงานบางครั้งเราทิ้งหน้าจอไว้ชั้กพัก แล้วกลับมาดูอีกครั้ง เราจะพบว่าแอปพลิเคชันไม่ทำงานตัวเอง เพียงแต่จะหยุดการทำงานเท่านั้น และเมื่อเรากลับมาดูใหม่ ก็ไม่จำเป็นต้องทำการตั้งค่าเริ่มต้นใหม่อีก ซึ่งจะทำให้การทำงานเร็วขึ้น แต่อย่างไรก็ตาม ก็จะมีการใช้ทรัพยากรสิ้นเปลืองเช่นกัน

วงจรชีวิตที่กล่าวมาข้างต้นอาศัยเมธอดพื้นฐานดังนี้

1. init()

เมธอด `init()` จะถูกเรียกเมื่อแอปพลิเคชันเริ่มทำงานครั้งแรกเท่านั้นส่วนมากแล้วเราจะใช้เมธอดนี้เพื่อจัดการเตรียมค่าองค์ประกอบเริ่มต้นต่างๆ ให้พร้อมต่อการใช้งาน เช่น การกำหนดค่าเริ่มต้นให้กับตัวแปร การโหลดไฟล์ภาพหรือไฟล์เสียง ซึ่งกิจกรรมเหล่านี้บางครั้งถ้าไปทำขณะที่กำลังทำงานอยู่อาจจะทำให้การทำงานช้าลง ดังนั้นกิจกรรมใดที่เราคิดว่าต้องทำเพียงครั้งเดียวหรือควรจัดการก่อนเข้าสู่การทำงานจริงก็ควรจัดการในเมธอดนี้ ซึ่งมีรูปแบบ

```
public void init()
```

2 start()

เมธอด `start()` จะถูกเรียกหลังจากเมธอด `init()` และจะทำงานใหม่ทุกครั้งเมื่อแอปพลิเคชันถูกเรียกกลับมาให้ทำงาน เช่น เมื่อเราสั่งให้ Internet Explorer ย้อนกลับไปโหลดโฮมเพจก่อนหน้านี้ (Back) แล้วเราสั่งให้กลับมาที่หน้าเดิม (Forward) ที่มีแอปพลิเคชันของเราทำงานอยู่ แอปพลิเคชันจะได้รับเหตุการณ์ `start` อีกครั้งหนึ่ง มีรูปแบบ

```
public void start()
```

3 stop()

เมธอด `stop()` จะทำงานตรงข้ามกับเมธอด `start()` กล่าวคือจะถูกเรียกเมื่อออกจากโฮมเพจหน้าที่มีแอปพลิเคชันนั้นๆ ทำงานอยู่เพื่อไปทำงานที่โฮมเพจหน้าอื่น และเมื่อเรากลับมาทำงานที่แอปพลิเคชันหน้าเดิม เมธอด `start()` ก็จะถูกเรียกใช้งานอีกครั้งหนึ่ง และสุดท้ายเมธอด `stop()` ก็จะถูกเรียกใช้งานก่อนเมธอด `destroy()` เสมอ ซึ่งมีรูปแบบ

```
public void stop()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 destroy()

เมธอด destroy() จะถูกเรียกใช้งานหลังจากเมธอด stop() ทุกครั้งเพื่อทำการระงับทรัพยากรต่างๆ ที่ได้จองไว้ใช้งานก่อนหน้านี้ เช่น ยกเลิกการติดต่อกับระบบเครือข่ายใดๆ ที่เรากำลังติดต่อกันอยู่ หากเรามีกิจกรรมใดๆ ที่จะต้องทำก่อนจบการทำงานของแอปพลิเคชันก็ควรจัดการในเมธอดนี้ มีรูปแบบ

```
public void destroy()
```

เมธอดที่ใช้ในการวาดภาพและเมธอดที่ตอบสนองต่อเหตุการณ์

สำหรับเมธอดที่ใช้ในการแสดงผลหน้าจอประกอบด้วย

```
paint(Graphics g)
```

เมธอด paint(Graphics g) ใช้ในการวาดภาพภายในพื้นที่แสดงผลของแอปพลิเคชัน ซึ่งเมธอดนี้เป็นเมธอดเดียวที่เราสามารถใช้เมธอดที่เกี่ยวข้องกับกราฟฟิคในการวาดภาพหรือวาดตัวอักษรบนพื้นที่แสดงผลของแอปพลิเคชัน มีรูปแบบ

```
public void paint(Graphics g)
```

```
update()
```

เมธอด update() ทำหน้าที่คล้ายเมธอด paint(Graphics g) คือใช้จัดการกับการแสดงผลในแอปพลิเคชัน แต่ใช้ในลักษณะปรับปรุงการแสดงผลของภาพ กรณีการทำ Double Buffered และขณะที่มีการวาดภาพใหม่ เมื่อเรียกเมธอด update() จะไม่ทำการลบพื้นที่แสดงผลภาพใหม่ โดยเมื่อเมธอด update() ถูกเรียกใช้งาน ก็จะไปเรียกให้เมธอด paint(Graphics g) ให้ทำงานอีกครั้งต่อหนึ่ง ดังนั้นเมื่อเราต้องการที่จะปรับปรุงส่วนใดๆ บนพื้นที่การแสดงผล เราก็จะมาจัดการกับเมธอดนี้ มีรูปแบบ

```
public void update()
```

สำหรับเมธอดที่ใช้ตอบสนองต่อเหตุการณ์จะประกอบไปด้วยหลักๆ คือ เมธอดที่ตอบสนองเหตุการณ์จากเมาส์และเมธอดที่ตอบสนองเหตุการณ์จากคีย์บอร์ด

สำหรับเมธอดที่ตอบสนองเหตุการณ์จากเมาส์ประกอบด้วย

1. mouseEntered(MouseEvent e)

เมธอดนี้จะทำงานเมื่อเราเลื่อนเมาส์จากภายนอกพื้นที่แสดงผลของแอปพลิเคชันเข้าสู่พื้นที่การแสดงผลของแอปพลิเคชัน มีรูปแบบ

```
public void mouseEntered(MouseEvent e)
```

2. mouseExited(MouseEvent e)

เมธอดนี้จะทำงานตรงข้ามกับเมธอด mouseEntered(MouseEvent e) คือ จะทำงานเมื่อตำแหน่งของเมาส์หลุดออกนอกพื้นที่แสดงผลของแอปพลิเคชัน มีรูปแบบ

```
public void mouseExited(MouseEvent e)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. `mousePressed(MouseEvent e)`

เมธอดนี้จะทำงานเมื่อมีการกดปุ่มบนเมาส์ภายในพื้นที่แสดงผลของแอปพลิเคชัน มีรูปแบบ

```
public void mousePressed(MouseEvent e)
```

4. `mouseReleased(MouseEvent e)`

เมธอดนี้จะทำงานตรงข้ามกับเมธอด `mousePressed(MouseEvent e)` กล่าวคือ จะทำงานเมื่อปุ่มของเมาส์ถูกปล่อยหลังจากการกด มีรูปแบบ

```
public void mouseReleased(MouseEvent e)
```

5. `mouseMoved(MouseEvent e)`

เมธอดนี้จะทำงานเมื่อมีการเลื่อนตำแหน่งของเมาส์ภายในพื้นที่แสดงผลของแอปพลิเคชัน โดยที่ไม่ได้มีการกดปุ่มเมาส์ มีรูปแบบ

```
public void mouseMoved(MouseEvent e)
```

6. `mouseDragged(MouseEvent e)`

เมธอดนี้ทำงานคล้ายกับเมธอด `mouseMoved(MouseEvent e)` แต่ต่างที่เมธอดนี้จะตอบสนองต่อเหตุการณ์ที่มีการกดปุ่มเมาส์ควบคู่กันไปด้วย มีรูปแบบ

```
public void mouseDragged(MouseEvent e)
```

7. `mouseClicked(MouseEvent e)`

เมธอดนี้จะทำการนับจำนวนการกดปุ่มเมาส์ภายในพื้นที่แสดงผลของแอปพลิเคชัน มีรูปแบบ

```
public void mouseClicked(MouseEvent e)
```

สำหรับเมธอดที่ตอบสนองเหตุการณ์จากคีย์บอร์ดประกอบด้วย

1. `keyPressed(KeyEvent e)`

เมธอดนี้จะทำงานเมื่อมีการกดปุ่มใดๆ บนคีย์บอร์ดในขณะที่แอปพลิเคชันยังคงทำงานอยู่ มีรูปแบบ

```
public void keyPressed(KeyEvent e)
```

2. `keyReleased(KeyEvent e)`

เมธอดนี้จะทำงานตรงข้ามกับเมธอด `keyPressed(KeyEvent e)` กล่าวคือ จะทำงานเมื่อมีการปล่อยคีย์ ซึ่งเกิดหลังจากที่มีการกดคีย์ และอยู่ในระหว่างที่แอปพลิเคชันยังทำงานอยู่ มีรูปแบบ

```
public void keyReleased(KeyEvent e)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. keyTyped(KeyEvent e)

เมธอดนี้ทำงานคล้ายกับ 2 เมธอดข้างต้น กล่าวคือ เมธอดนี้จะทำงานเมื่อมีการกดปุ่มบนคีย์บอร์ด พร้อมทั้งต้องปล่อยปุ่มจากคีย์บอร์ดด้วยจึงจะเริ่มทำงาน มีรูปแบบ

```
public void keyTyped(KeyEvent e)
```

ระบบติดต่อผู้ใช้ (User Interface)

ระบบติดต่อผู้ใช้ คือ ส่วนประกอบหนึ่งของโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่เป็นสื่อกลางสำหรับโต้ตอบกับผู้ใช้ ในความหมายของการโต้ตอบก็คือ การรับคำสั่งและการแสดงผลลัพธ์ ระบบติดต่อผู้ใช้มีอยู่ด้วยกันหลายรูปแบบตั้งแต่อดีตจนถึงปัจจุบัน แต่สามารถแบ่งได้ใหญ่ๆ อยู่ 2 รูปแบบ คือ แบบรับคำสั่งทางคีย์บอร์ดแล้วแสดงผลเป็นตัวอักษร (Command - line interface) และอีกแบบคือ แบบชี้แล้วกด (Point - and - Click) โดยการใช้เมาส์สั่งงานร่วมกับการแสดงผลแบบกราฟฟิก (Graphic User Interface หรือ GUI) ซึ่งระบบติดต่อผู้ใช้แบบนี้เป็นที่นิยมกันในโปรแกรมสมัยใหม่ เพราะค่อนข้างเป็นมาตรฐานใช้งานได้ง่ายกว่าแบบแรก และไม่เสียเวลาในการเรียนรู้มากนัก

การทำงานในระดับล่างของระบบติดต่อผู้ใช้ ระบบปฏิบัติการ (Operating System) จะส่งข้อมูลอินพุทจากคีย์บอร์ดและเมาส์ไปให้แก่โปรแกรมเพื่อประมวลผลและแสดงผลข้อมูลที่เอาท์พุทในรูปของจุดสีหรือบิตแมป (Bitmap) ซึ่งประกอบกันขึ้นเป็นตัวอักษรหรือรูปภาพบนจอภาพ ในจาวา AWT ได้ถูกออกแบบให้โปรแกรมเมอร์หรือผู้พัฒนาโปรแกรม ไม่ต้องเป็นกังวลกับงานในระดับล่าง เช่น การจัดการกับอินเตอร์เฟซของเมาส์หรือคีย์บอร์ด แม้กระทั่งรายละเอียดการเขียนจุดสี (pixel) ลงบนจอภาพ

ด้วยเหตุที่ว่าจาวาเป็นภาษาคอมพิวเตอร์ที่ไม่ขึ้นอยู่กับระบบใดๆ ดังนั้นการทำงานของ AWT ได้ถูกออกแบบให้มีเครื่องมือไม่เครื่องมือที่ใช้สำหรับติดต่อผู้ใช้ ซึ่งสามารถปฏิบัติงานกับระบบปฏิบัติการหลายๆ ระบบได้ โดยการใช้เครื่องมือดั้งเดิมของระบบปฏิบัติการนั้นๆ เพื่อรักษาหน้าตาของเครื่องมือ ซึ่งจะแตกต่างกันบ้างเมื่อทำงานอยู่บน OS ที่ต่างระบบกัน แต่ก็ไม่ใช่เป็นอุปสรรคสำคัญในการพัฒนาโปรแกรม

คอมโพเนนท์และคอนเทนเนอร์

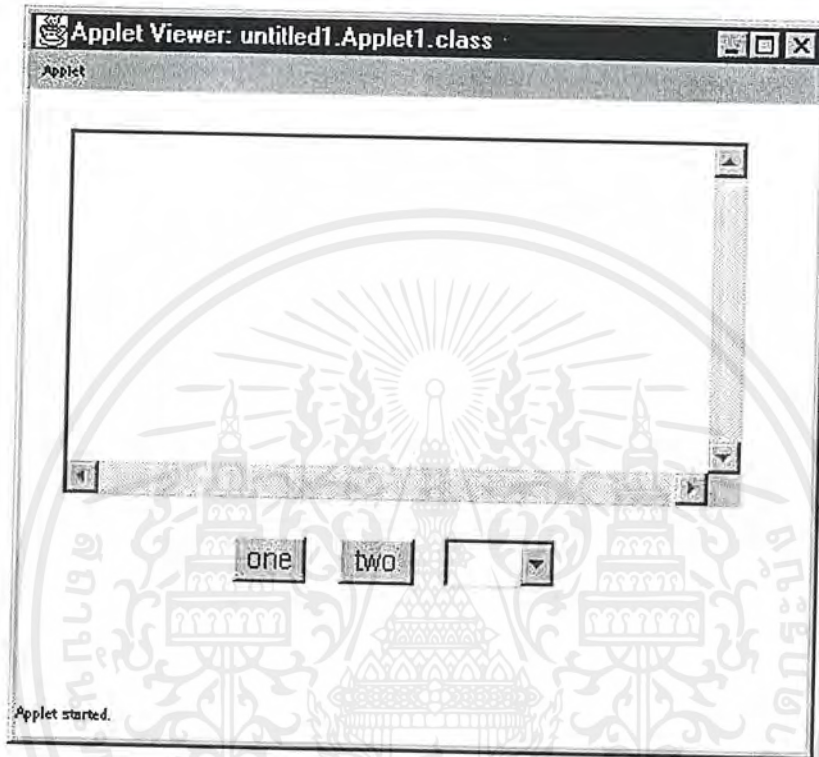
ระบบติดต่อผู้ใช้ถูกสร้างขึ้นจากระบบย่อยๆ ที่เรียกว่า คอมโพเนนท์ (Component) ตัวอย่างของคอมโพเนนท์ที่พบเห็นทั่วไป เช่น ปุ่มกด แถบเลื่อน ช่องเติมอักษร กล่องรายการ ฯลฯ กล่าวคือ คอมโพเนนท์เป็นส่วนที่ทำให้ผู้ใช้สามารถทำการโต้ตอบกับโปรแกรม (สั่งงาน) และคือ ส่วนที่ทำให้ผู้ใช้ทราบสถานะการปฏิบัติงานของโปรแกรม (การแสดงผล) ภายใน AWT คอมโพเนนท์ทุกคอมโพเนนท์ก็คือ อินสแตนซ์ (instance) หรือตัวแทนตัวหนึ่งของคลาส Component หรือเป็นสับไทป์ (sub type) ของคลาส Component

คอมโพเนนท์ใช้เครื่องมือที่ทำงานอยู่โดดเดี่ยวอย่างอิสระ แต่จะอยู่ภายใต้การควบคุมจากคอนเทนเนอร์ (Container) ซึ่งคือที่สำหรับบรรจุคอมโพเนนท์หรือแม้กระทั่งบรรจุคอนเทนเนอร์ด้วยกัน อัน

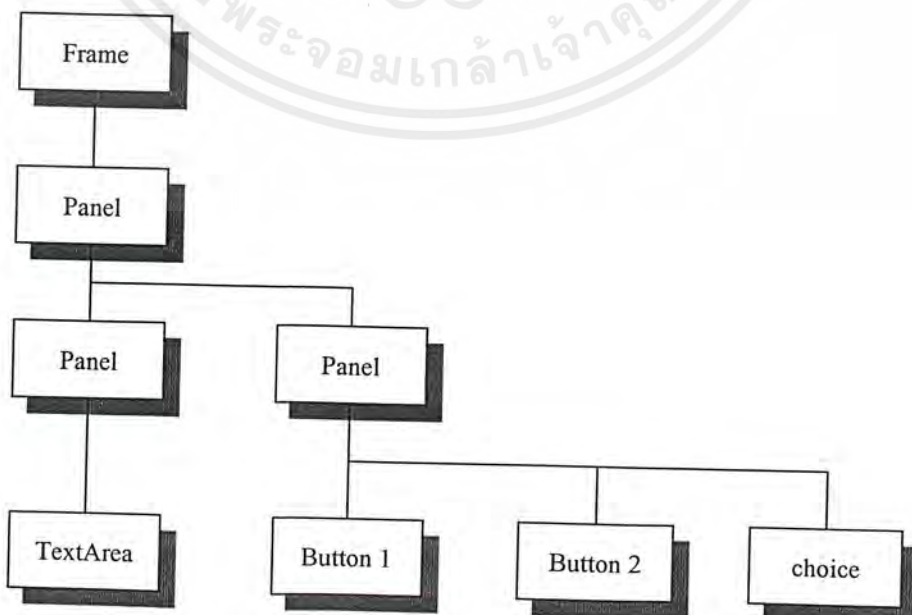
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องมาจากตัวคอนเทนเนอร์เองบางครั้งก็ทำหน้าที่เป็นคอมโพเนนต์เช่นกัน และคอนเทนเนอร์ทุกๆ คอนเทนเนอร์ก็เป็นอินสแตนซ์ของคลาส Container หรือเป็นสับไทป์ของ Container

คอมโพเนนต์จำเป็นต้องอยู่ในคอนเทนเนอร์ซึ่งกลุ่มของคอนเทนเนอร์ (หรือคอนเทนเนอร์ด้วยกัน) ที่บรรจุภายในคอนเทนเนอร์เหล่านี้ เปรียบเสมือนการสร้างโครงสร้างแบบรากต้นไม้ของคอมโพเนนต์ โดยที่คอมโพเนนต์บนสุดเปรียบเสมือนรากแก้วซึ่งจะแตกแขนงรากย่อยๆ ออกไป ในที่นี้คือ คอมโพเนนต์ ตามตัวอย่างดังรูป



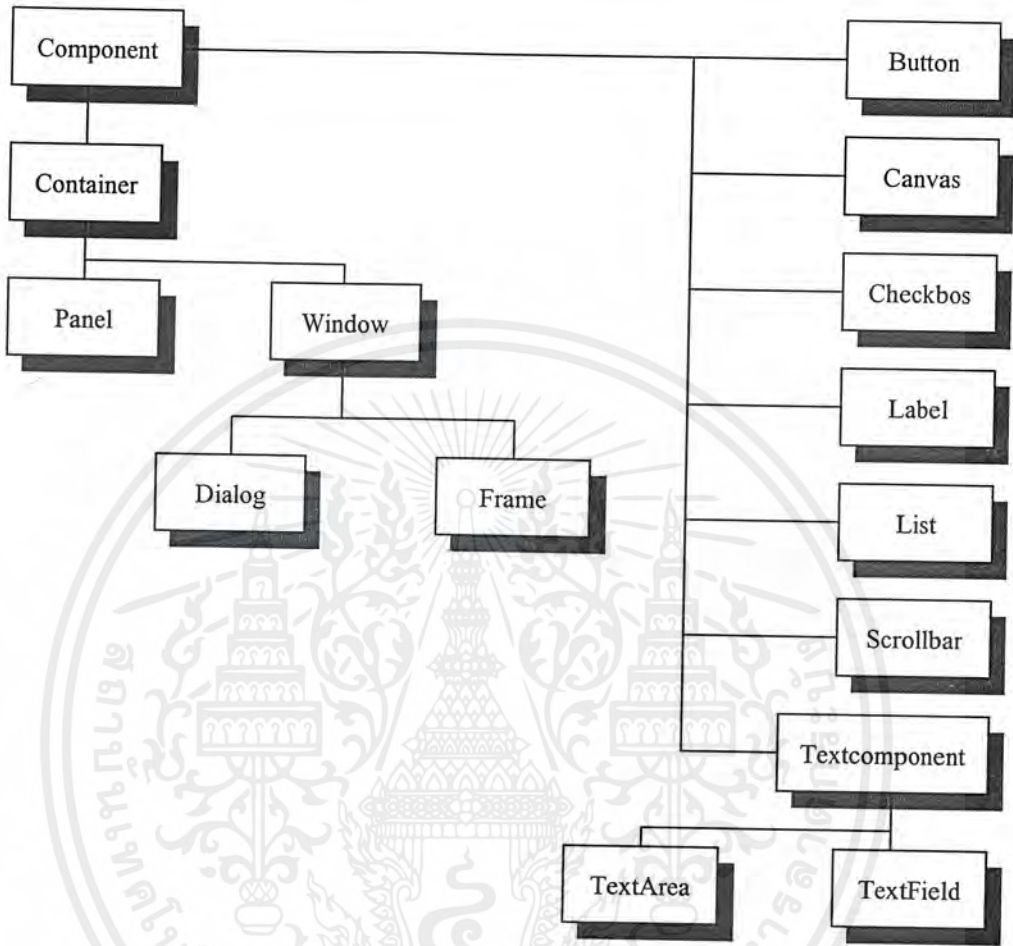
รูปที่ 2-21 ระบบติดต่อผู้ใช้ของ AWT ภายใต้ Window 9x/NT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2-22 แสดงโครงสร้างแบบรากต้นไม้ของระบบเชื่อมต่อตามรูปที่ 2
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อแบบสงวนเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ที่ทำการนำไปใช้

ประเภทของคอมโพเนนท์

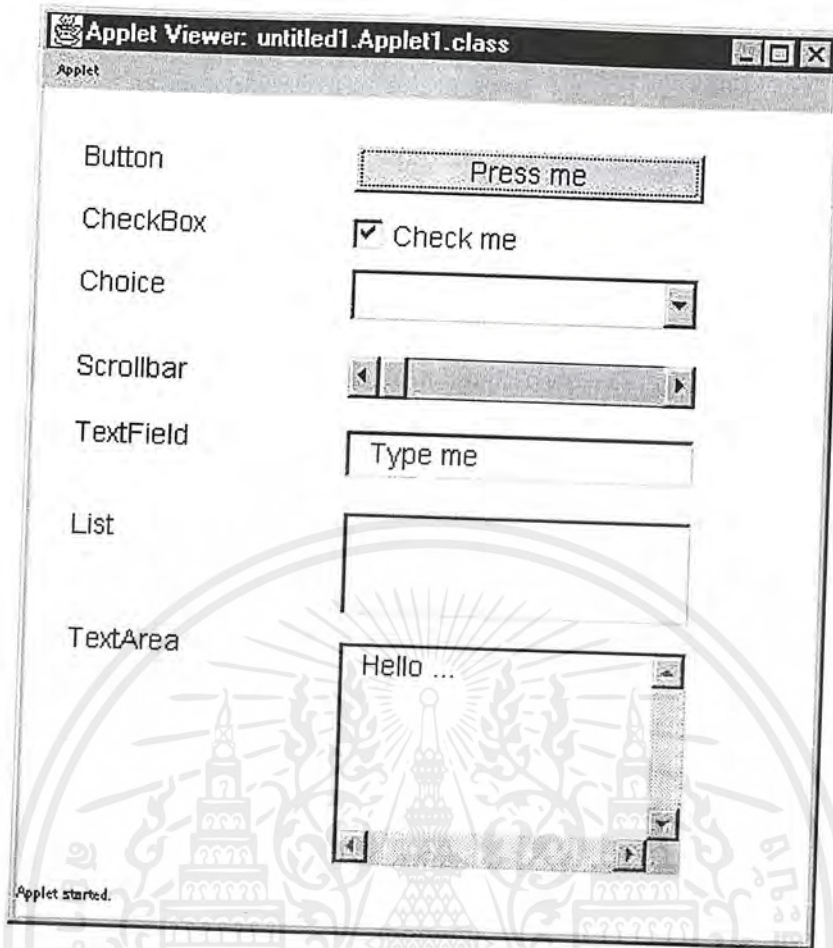
ความสัมพันธ์แบบถ่ายทอดของคลาสต่างๆ ของ AWT แสดงได้ตามรูปต่อไปนี้



รูปที่ 2-23 ความสัมพันธ์แบบถ่ายทอดของคลาส AWT ต่างๆ

นอกจากนี้เรายังสามารถสร้างคอมโพเนนท์ใหม่ขึ้นได้โดยการสืบทอดคุณสมบัติมาจากคอมโพเนนท์พื้นฐานต่างๆ

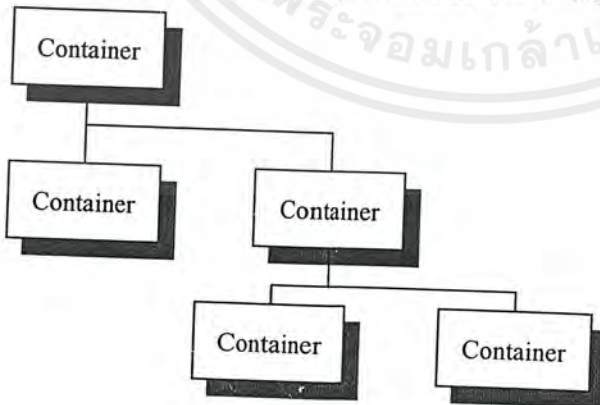
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-24 แสดงอินสแตนซ์ของแต่ละคลาส (หน้าต่างของคอมโพเนนต์พื้นฐาน)

ประเภทของคอนเทนเนอร์

คอนเทนเนอร์มีอยู่ 4 ประเภท คือ Window ซึ่งประกอบไปด้วยซับไทป์อีก 2 คลาส คือ Frame และ Dialog สำหรับ Panel เป็นคลาสที่อยู่ระดับเดียวกับกับ Window ซึ่งแสดงได้ตามรูป



รูปที่ 2-25 แสดงประเภทของคอนเทนเนอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส	คุณสมบัติ
Window	เป็นพื้นที่การแสดงผลระดับบนสุด อินสแตนซ์ของคลาส Window จะไม่สามารถถูกเพิ่มเติมหรือฝังตัวภายในคอนเทนเนอร์ใดๆ ได้ รูปร่างของคลาส Window จะไม่แสดงกรอบและหัวเรื่องของวินโดว์
Frame	เป็นพื้นที่การแสดงผลระดับบนสุด มีการแสดงกรอบและแถบชื่อของวินโดว์ อินสแตนซ์ของคลาส Frame อาจมีเมนูก็ได้ การทำงานโดยทั่วไปจะเหมือนกับอินสแตนซ์ของคลาส Window
Dialog	เป็นพื้นที่การแสดงผลระดับบนสุด มีการแสดงกรอบและแถบชื่อของวินโดว์ อินสแตนซ์ของคลาส Dialog จะเกิดขึ้นไม่ได้ถ้าปราศจากการเชื่อมต่อกับอินสแตนซ์ของคลาส Frame
Panel	คอนเทนเนอร์ทั่วไปสำหรับบรรจุคอมโพเนนต์ โดยการใช้เมธอด add() เพื่อเพิ่มคอมโพเนนต์ให้แก่อินสแตนซ์ของคลาส Panel

ตารางที่ 2-1 แสดงคุณสมบัติคลาสคอนเทนเนอร์ทั้ง 4 ประเภท

การสร้างคอมโพเนนต์

การที่จะเพิ่มเติมคอมโพเนนต์ใดๆ เพื่อสร้างระบบติดต่อกับผู้ใช้นั้น เราควรสร้างคอนเทนเนอร์เพื่อเป็นที่เก็บคอมโพเนนต์เสียก่อน สำหรับจาวาแอปพลิเคชันสิ่งแรกที่เราต้องทำคือ การสร้างอินสแตนซ์ของคลาส Window หรือคลาส Frame แต่สำหรับจาวาแอปพลิเคชัน คลาส Frame จะมีอยู่แล้ว ซึ่งเป็นวินโดว์ของเว็บเบราว์เซอร์นั่นเอง และเนื่องจากคลาส Applet เป็นสับไปป์ของคลาส Panel ดังนั้นเราจึงสามารถที่จะเพิ่มเติมคอมโพเนนต์ต่างๆ ให้กับแอปพลิเคชันได้เลย

การเพิ่มคอมโพเนนต์ให้กับคอนเทนเนอร์

คอมโพเนนต์สามารถนำมาเพิ่มให้กับคอนเทนเนอร์ได้โดยการใช้เมธอด add() การใช้งานเมธอด add() นั้นสามารถทำได้ 3 วิธีขึ้นกับโครงร่างของการวางตำแหน่งคอมโพเนนต์ (Component Layout)

การวางตำแหน่งคอมโพเนนต์

โครงร่างหรือการวางตำแหน่งคอมโพเนนต์ ภายในคอมโพเนนต์นั้นจะไม่ได้ถูกควบคุมจากตัวคอนเทนเนอร์เองแต่จะถูกควบคุมจากตัวจัดการโครงร่าง (Layout Manager) ซึ่งทำหน้าที่โดยตรงกับตัว

คอนเทนเนอร์นั้น ตัวจัดการ โครงร่างเป็นตัวที่จะตัดสินใจให้กับตัวคอมโพเนนท์วางอยู่ในตำแหน่งไหนในคอนเทนเนอร์

การจัดการโครงร่างของตัวจัดการโครงร่างมี 5 รูปแบบ ตั้งแต่รูปแบบง่ายๆ ธรรมดาไปจนถึงรูปแบบที่ซับซ้อนมากๆ

1. FlowLayout

โครงร่างนี้จะจัดวางคอมโพเนนท์จากซ้ายไปขวาในแนวนอนหรือแนวคอลัมน์ก่อน แล้วจึงเริ่มลงมาแถวใหม่เมื่อคอมโพเนนท์เกินขอบเขตในแถวนั้นๆ สำหรับการเพิ่มคอมโพเนนท์ใหม่เข้าไป จะใช้เมธอด `add()` ซึ่งมีอาร์กิวเมนต์เป็นชื่อของคอมโพเนนท์ที่จะทำการเพิ่มเข้าไป

2. BorderLayout

โครงร่างนี้จะจัดแบ่งสัดส่วนพื้นที่ของแอปพลิเคชัน คือ North (ด้านบน), South (ด้านล่าง), West (ด้านซ้าย), East (ด้านขวา) และ Center (กึ่งกลาง) แต่ละคอมโพเนนท์จะถูกจัดวางตำแหน่งภายในพื้นที่เหล่านี้ จุดเด่นของโครงร่างแบบนี้คือ แต่ละพื้นที่จะคงสัดส่วนของตัวเองไว้เสมอแม้จะมีการเปลี่ยนแปลงขนาดของคอมโพเนนท์ไปในรูปแบบใดก็ตาม ซึ่งพื้นที่ที่จะมีขนาดกว้างที่สุดคือ บริเวณ Center และอาร์กิวเมนต์ที่ป้อนเข้าไปในเมธอด `add()` เพื่อที่จะทำการเพิ่มคอมโพเนนท์จะมี 2 ตัว คือ ชื่อของออบเจกต์ (คอมโพเนนท์) และชื่อของพื้นที่ที่จะมีการวางตำแหน่งคอมโพเนนท์

ข้อสังเกตคลาสที่เป็นคอนเทนเนอร์แต่ละแบบจะมีโครงร่างหรือเลย์เอาต์แบบเดิม (default) อยู่แล้ว เช่น คลาส *Frame* และคลาส *Dialog* จะใช้ตัวจัดการโครงร่างแบบ *BorderLayout* ส่วนตัวจัดการโครงร่างดั้งเดิมของคลาส *Panel* และ *Applet* เป็นแบบ *FlowLayout*

3. GridLayout

เป็นส่วนแสดงผลของแอปพลิเคชันภายในกริดของคอลัมน์และแถว แต่ละส่วนที่ควบคุม *User Interface* จะถูกแทนที่ในเซลล์ภายในกริดจากซ้ายไปขวา และจากบนลงล่าง ตัวจัดการ โครงร่างจะจัดส่วนควบคุมที่ถูกต้องไว้ โดยส่วนประกอบของ *User Interface* จะปรากฏภายในแอปพลิเคชัน

4. CardLayout

โครงร่างนี้ใช้จัดที่ว่างบนจอได้ดีกว่าตัวจัดการ โครงร่างอื่น ใช้เพื่อเลือกลำดับของส่วนควบคุมที่จะแสดงในเวลาต่างๆ กัน เมื่อใส่ส่วนควบคุมเพิ่ม ไปยังโครงร่างแบบการ์ด ตัวจัดการ โครงร่างจะไม่แสดงส่วนควบคุมทั้งหมดออกมาพร้อมกันแต่จะเลือกแสดงส่วนควบคุมได้โดยใช้ตัวจัดการ โครงร่างเป็นตัวจัดการ ซึ่งนำมาใช้ในกรณีที่ต้องการ *User Interface* แบบซับซ้อน

5. GridBagLayout

เป็นตัวจัดการ โครงร่างแบบพิเศษ ที่อนุญาตให้แอปพลิเคชันใส่ระดับความสำคัญให้แต่ละส่วนควบคุมภายในแอปพลิเคชัน กล่าวคือ ถ้าแอปพลิเคชันมีการเปลี่ยนขนาดใหม่อีกครั้ง ส่วนควบคุมที่มีระดับความสำคัญสูงที่สุดจะครอบครองพื้นที่หน้าจอบางส่วน

เมธอดพื้นฐานที่ใช้ในการเรียกใช้งานระบบติดต่อผู้ใช้

- `add()` ใช้เพิ่ม UI ต่างๆ เข้าในคอนเทนเนอร์
- `remove()` เป็นการนำ UI ออกจากคอนเทนเนอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- `setLayout()` เป็นการเรียกใช้งานตัวจัดการ โครงร่าง (Layout Manager)

การใส่แอปเพล็ตลงในเอกสาร HTML

สามารถทำได้โดยใส่แท็ก `<APPLET>` ลงในโค้ดของ HTML ซึ่งมีรูปแบบของการเขียนโค้ดดังนี้

```
<APPLET CODE = Applet.class WIDTH = widthInt HEIGHT = heightInt> </APPLET>
```

ทุกรูปแบบข้างต้นแสดงถึงองค์ประกอบพื้นฐานต่อไปนี้

1. เริ่มต้นด้วยการประกาศเป็น `APPLET-tag` คือ `<APPLET>`
2. บอกถึงไฟล์ที่จะใช้ในการรัน ซึ่งต้องเป็นไฟล์ที่คอมไพล์มาแล้ว นั่นคือเป็นไฟล์ `*.class` ตามด้วยการบอกถึงขนาดความกว้างและความยาวของพื้นที่ที่ต้องการแสดงแอปเพล็ตนั้น โดยใส่กับ `WIDTH` และ `HEIGHT` ตามลำดับ

3. ปิดท้ายด้วย `</APPLET>`

นอกจากองค์ประกอบข้างต้นแล้ว ยังมีฟังก์ชันที่บอกถึงที่อยู่ของไฟล์ไบท์โค้ดนั้นด้วย ซึ่งมีรูปแบบ

```
CODEBASE = URL
```

ในส่วน URL ของ `CODEBASE` นี้ นอกจากจะใช้เป็น URL แล้วยังสามารถแทนที่ด้วยที่อยู่ใดเรียกทอริได้ด้วย และกรณีที่มีการรับข้อมูลเข้าสู่แอปเพล็ตก็สามารถทำได้โดยผ่านแท็ก `<PARAM>` ตามรูปแบบ

```
<APPLET CODE = Applet.class WIDTH = 600 HEIGHT = 400>
```

```
<PARAM NAME = param1 VALUE = value1>
```

```
<PARAM NAME = param2 VALUE = value2>
```

```
</APPLET>
```

แต่สำหรับการเขียนโค้ด HTML นั้น ไม่มีความจำเป็นต้องเป็นตัวใหญ่ทั้งหมด และกรณีที่เบราว์เซอร์นั้นไม่สามารถรันแอปเพล็ตได้ เราจะใส่แท็ก `<blockquote>` เพื่อให้การทำงานทำข้ามบล็อกนี้ไป

การติดต่อสื่อสารกับโปรแกรมอื่น

แอปเพล็ตสามารถติดต่อสื่อสารกับโปรแกรมอื่นได้ 3 ทางคือ

1. โดยการเรียกเมธอดของแอปเพล็ตอื่นบนเว็บเดียวกัน
2. โดยการใช้ API ซึ่งจะมีเมธอดที่ช่วยให้แอปเพล็ตสามารถติดต่อกับเบราว์เซอร์หรือแอปเพล็ตวิวเวอร์ที่เก็บแอปเพล็ตนั้น
3. โดยการใช้ API ช่วยให้สามารถติดต่อกับโปรแกรมอื่นๆ ผ่านเครือข่าย โดยจุดสำคัญก็คือจะต้องอยู่ในเครือข่ายที่ใช้โฮสต์เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการติดต่อสื่อสารจะมีหัวข้อย่อยๆ ที่จะต้องพิจารณาดังที่กล่าวข้างต้นคือการส่งเมสเสจไปยังแอปเพล็ตอื่นในเพจเดียวกันซึ่งปกติแอปเพล็ตจะพยายามค้นหาแอปเพล็ตอื่นๆ และสามารถส่งเมสเสจไปยังแอปเพล็ตนั้นได้โดยอยู่บนพื้นฐานของความปลอดภัยดังนี้

1. แอปเพล็ตจะต้องรันอยู่บนเพจเดียวกัน และบนเบราว์เซอร์ตัวเดียวกัน
2. มีแอปเพล็ตวิวเวอร์ (Appletviewer) หลายตัวที่ต้องการให้แอปเพล็ตเหล่านั้นเกิดจากเซิร์ฟเวอร์เดียวกันด้วยการค้นหาแอปเพล็ตอื่น สามารถทำได้โดยใช้เมธอด `getApplet()` โดยการหาอาจหาตามรายชื่อของแอปเพล็ต หรือจะค้นหาทั้งหมดเลยก็ได้ ซึ่งปกติแล้วแอปเพล็ตจะไม่มีชื่อ การที่จะทำให้แอปเพล็ตมีชื่อได้จะต้องมีการกำหนดไว้ในไฟล์ HTML โดยการกำหนดทำได้ 2 แบบคือ

- โดยการใส่ชื่อไว้ใน `APPLET` - tag

```
<applet codebase = example/ code = Sender.class width = 450 height = 250
name = "share">
```

...

```
</applet>
```

- โดยการใส่ `PARAM` - tag

```
<applet codebase = example/ code = Receiver.class width = 450 height = 350>
```

```
<param name = "name value = "value">
```

...

```
</applet>
```

การใช้เมธอด `getApplet()` เพื่อหาแอปเพล็ตบนเพจนั้น เมื่อมีการเรียกใช้เมธอดนี้แล้วก็รีเทิร์นค่าของแอปเพล็ตทั้งหมดที่มีอยู่บนเว็บนั้นออกมาให้

2.4.2 Java Application

จาวาแอปพลิเคชันสามารถรันได้โดยตัวมันเองซึ่งต่างจากจาวาแอปเพล็ตตรงที่แอปเพล็ตต้องการบราวเซอร์ที่สนับสนุน และจาวาแอปพลิเคชันจะรันโดยเรียกคำสั่ง `java` ตามด้วยชื่อโปรแกรมที่มีชนิดเป็นคลาส

จาวาแอปพลิเคชันจะเริ่มรันที่เมธอด `main()` ซึ่งแสดงดังนี้

```
public static void main(String args[])
```

```
{
```

```
....
```

```
}
```

`public` หมายถึง เมธอดนี้เป็นประโยชน์ต่อคลาสและออบเจกต์ต่าง ๆ ดังนั้นเมธอด `main()` จึงสามารถประกาศเป็น `public`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

static หมายถึง main() เป็นเมธอดของคลาส
 void หมายถึง เมธอด main() ไม่ส่งค่าใด ๆ กลับ
 main() ใช้พารามิเตอร์เดียวเป็นอรรถของสตริง พารามิเตอร์นี้จะรับอาร์กิวเมนต์ที่ส่งผ่านมาจากบรรทัดคำสั่งของ DOS (DOS command line)

ส่วนตัวลำดับของเมธอด main() สามารถบรรจุโค้ดใด ๆ ก็ได้ตามที่ต้องการเช่น

- การกำหนดค่าเริ่มแรกให้ตัวแปร
- การสร้าง instance ต่าง ๆ ของคลาสใด ๆ ที่ประกาศ

เมื่อจาวาประมวลผลเมธอด main() คลาสจะยังไม่ถูกสร้าง instance โดยอัตโนมัติ ดังนั้นเมื่อดำเนินการคลาสในลักษณะออบเจกต์แล้วจะต้องสร้าง instance ในเมธอดของ main() ด้วยตัวเอง

เพราะว่าแอปพลิเคชันของจาวาเป็นโปรแกรมเดี่ยว มันใช้ประโยชน์ในการผ่านอาร์กิวเมนต์หรือตัวเลือกให้โปรแกรม เพื่อพิจารณาวิธีการที่โปรแกรมจะรันหรือช่วยโปรแกรมดำเนินการอินพุตที่แตกต่างกันมากมาย อาร์กิวเมนต์ของ command-line ซึ่งเราสามารถทำตามวัตถุประสงค์ที่เราต้องการได้ เช่น ใช้อินพุตของ debug เพื่อระบุชื่อไฟล์ที่จะอ่านหรือบันทึก หรือสารสนเทศอื่น ๆ ที่เราต้องการให้โปรแกรมของจาวารับทราบ

การผ่านอาร์กิวเมนต์ต่าง ๆ ให้โปรแกรมของจาวา

การผ่านอาร์กิวเมนต์ให้โปรแกรมของจาวาเราจะต้องใส่ค่าของอาร์กิวเมนต์ที่ command-line ของ MS-DOS ในโปรแกรมของจาวาดังนี้

```
java myProgram arg1 arg2 arg3
```

บนบรรทัดคำสั่ง (command-line) นี้ได้ใช้อาร์กิวเมนต์ 3 ค่าคือ arg1, arg2 และ arg3 แต่ละอาร์กิวเมนต์จะแยกจากกันโดยช่องว่าง เช่น

```
java myProgram java is best
```

ซึ่งในตัวอย่างจะส่งไป 3 อาร์กิวเมนต์ แต่ถ้ากลุ่มของอาร์กิวเมนต์ล้อมรอบด้วยเครื่องหมาย “ “ แล้วจะถือว่าเป็น 1 อาร์กิวเมนต์ เช่น

```
java myProgram "java is best"
```

การเชื่อมโยงอาร์กิวเมนต์ต่าง ๆ ในโปรแกรม

จาวาสามารถเชื่อมโยงอาร์กิวเมนต์โดยบรรจุไว้ในอรรถของสตริง ซึ่งจะถูกส่งผ่านไปให้เมธอด main() ซึ่งก็คือ args ภายในเมธอด main() เราสามารถเชื่อมโยงอาร์กิวเมนต์ args[] เป็นตัวรับอาร์กิวเมนต์จาก command-line ได้

2.4.3 JAVA SERVLET

Servlet เป็นโมดูลซึ่งรันภายใต้การเรียกร้องและการตอบสนองของเซิร์ฟเวอร์ เช่น Java Web Server โดย servlet ถูกนำมาใช้เพื่อเพิ่มเติมความสามารถของเซิร์ฟเวอร์ในวิธีการหลาย ๆ อย่างทั้งทำให้มี overhead น้อยลง การบำรุงรักษาและการสนับสนุนต่างๆ เพราะว่า servlet เป็นจาวาไบท์โค้ดซึ่งสามารถถูกดาวน์โหลดข้ามเครือข่ายได้ และเป็นการเขียนครั้งเดียวแต่รันที่ใดก็ได้ (write once, run anywhere) และไม่เหมือนกับ CGI script ตรงที่ servlet จะก่อให้เกิดรูปแบบที่ไม่เฉพาะเจาะจง และมีวิธีการที่ง่ายในการสร้างเอกสารที่จะตอบสนองกับผู้ใช้และรันอย่างรวดเร็ว โดยที่ Servlet นั้นเข้ากับเว็บเซิร์ฟเวอร์ทั้งหลายได้ด้วย

ในการติดต่อกับฐานข้อมูล เนื่องจาก Servlet เขียนขึ้นมาจากภาษาจาวาเราจึงสามารถใช้ servlet นี้ทำงานไปด้วยกันกับ JDBC ซึ่ง JDBC เป็นตัวที่ใช้ในการติดต่อกับฐานข้อมูลอีกที

เนื่องจากว่าจาวานั้นสนับสนุนการทำงานหลาย ๆ อย่างได้ในเวลาเดียวกัน ดังนั้นการสร้าง servlet ด้วยจาวาจึงเป็นการทำให้มีประสิทธิภาพที่ดีด้วย ซึ่งถ้าเทียบกับ CGI แล้วการสนับสนุนการทำงานหลาย ๆ อย่างในเวลาเดียวกันนั้นจะขึ้นอยู่กับระบบปฏิบัติการ (Operating System) ซึ่งผู้ที่พัฒนาจำเป็นต้องเรียนรู้ระบบที่จะทำการเรียกรูปแบบแอฟพลิเคชันนั้นขึ้นมารัน แต่ในจาวาเราสามารถใช่คลาส Thread เพื่อเริ่มต้นและหยุดโปรแกรมที่เราต้องการและทำให้เรามั่นใจได้ว่าแอฟพลิเคชันเหล่านั้นจะอยู่ภายใต้สภาพแวดล้อมของเว็บเซิร์ฟเวอร์ใด ๆ ก็ได้

2.4.3.1 Servlet Architecture

สิ่งที่น่าสนใจใน servlet API คือ servlet interface ซึ่ง servlet จะทำการสร้างการอินเทอร์เฟซนี้ขึ้นมาโดยทำการเพิ่มคลาสเช่น HttpServlet และ servlet interface จะจัดหาวิธีการมาจัดการ servlet และการติดต่อกับไคลเอนต์

เมื่อ servlet ได้รับข้อมูลมาจาก client มันจะได้รับ 2 ออบเจกต์คือ ServletRequest และ ServletResponse คลาส ServletRequest จะเกี่ยวข้องกับการติดต่อจากไคลเอนต์ไปยังเซิร์ฟเวอร์ ขณะที่คลาส ServletResponse จะเกี่ยวข้องกับการติดต่อจาก servlet กลับไปยังไคลเอนต์ โดยที่ผู้พัฒนา servlet สามารถใช้ API ต่าง ๆ นี้ดูแลสถานะระหว่าง servlet และไคลเอนต์เพื่อให้มีการเชื่อมต่อพร้อม ๆ กันหลายเครื่องเรื่อย ๆ ระหว่างบางช่วงเวลาได้

2.4.3.2 Servlet lifecycle

1. เมื่อเซิร์ฟเวอร์โหลด servlet ขึ้นมา มันจะเริ่มรัน servlet โดยเริ่มที่เมธอด (ถึงแม้ว่า servlet จะรันใน multithread ได้แต่จะไม่มีการเกิดขึ้นพร้อมๆกันในระหว่างช่วง servlet initialization) ซึ่ง server จะเรียกเมธอด init ครั้งเดียวเมื่อมันโหลด servlet ขึ้นมาและจะไม่เรียกมันอีกจนกว่าเซิร์ฟเวอร์จะทำการรีโหลด servlet ใหม่ โดยที่ server จะไม่สามารถ reload servlet ได้จนกว่ามันจะได้นำ servlet ออกไป (remove) โดยการเรียกเมธอด destroy และที่เมธอด init นี้จะถูกทำให้เสร็จสมบูรณ์ก่อนที่ไคลเอนต์จะทำการเรียกใช้ (คือก่อนที่เมธอด service จะถูกเรียก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หลังจากที่เซิร์ฟเวอร์โหลดและทำตามเมธอด `init` เสร็จแล้ว `servlet` สามารถที่จะรองรับกับการเรียกใช้งานของไคลเอนต์ได้ โดยจะทำการโปรเซสการเรียกใช้งานเหล่านี้ในเมธอด `service` ซึ่งแต่ละการเรียกใช้งานของไคลเอนต์จะมีการเรียกเมธอด `service` และรันใน `servlet thread` ของมันเอง เพราะว่า `servlet` สามารถรันได้หลายเมธอด `service` ที่เวลาหนึ่ง ๆ
3. `servlet` จะรันไปเรื่อย ๆ จนกระทั่งมันได้ถูกนำออกไปจากการบริการ เช่น การเรียกใช้งานของผู้ดูแลระบบ เมื่อเซิร์ฟเวอร์ได้นำ `servlet` ออกไปแล้ว มันจะรัน `destroy method` ของ `servlet` เมธอดนี้จะรันเพียงครั้งเดียว โดยเซิร์ฟเวอร์จะไม่รันมันอีกจนกระทั่งหลังจากที่มันถูกโหลดขึ้นมาใหม่และทำเมธอด `init` ใหม่แล้ว และขณะที่กำลังมีการรัน `destroy method` อยู่ `Thread` อื่นๆอาจจะกำลังรันการเรียกใช้งานการบริการ(`service request`)ก็ได้

2.4.3.3 การเขียน `servlet`

`Servlet` มีอินเตอร์เฟซ `javax.servlet.Servlet` ซึ่งผู้พัฒนาสามารถสร้างการอินเตอร์เฟซได้อย่างตรง ๆ (แต่ปกติไม่จำเป็น) เพราะว่า `servlet` จะเป็นส่วนหนึ่งของ `web server` ซึ่งใช้โปรโตคอล `HTTP` เพื่อการติดต่อกับเครื่องไคลเอนต์ ดังนั้นวิธีที่ดีที่สุดที่จะพัฒนา `servlet` คือ การเจาะจงพิเศษไปที่คลาส `javax.servlet.http.HttpServlet`

คลาส `HttpServlet` จะทำให้เกิด `Servlet interface` โดยการเพิ่มเติมกับ `Generic Servlet base class` และจัดหาเฟรมเวิร์คสำหรับการปฏิบัติการกับโปรโตคอล `HTTP` โดยที่เมธอด `service` จะสนับสนุน `standard HTTP/1.1` ที่ต้องการ โดยการส่งแต่ละการเรียกใช้งานไปยังเมธอดที่ออกแบบให้รองรับการเรียกใช้งานนั้น

โดยทั่วไป `servlet` ที่ถูกเขียนโดยคลาส `HttpServlet` สามารถมีหลาย ๆ เมธอดที่รัน `service method` ไปพร้อม ๆ กัน แต่ถ้าเราใช้เพียงแค่เมธอดเดียวรัน `service method` ที่เวลาหนึ่ง ๆ แล้วในการเพิ่ม `HttpServlet` นั้นทำให้ `servlet` ของเราจะต้อง implement `SingleThread Model Interface` ด้วย เช่น

```
public class SurveyServlet extends HttpServlet
    implements SingleThreadModel
{
}
```

สำหรับการติดต่อกับไคลเอนต์จะมีเมธอดต่าง ๆ ดังนี้

- `doGet` สำหรับการปฏิบัติกับ `Get` , `conditional Get` และ `HEAD request`
- `doPost` สำหรับการปฏิบัติกับ `Post request`
- `doPut` สำหรับการปฏิบัติกับ `Put request`
- `doDelete` สำหรับการปฏิบัติกับ `Delete request`
- `doOption` ถูกเรียกเมื่อได้รับ `option request`
- `doTrace` ถูกเรียกเมื่อได้รับ `Trace request`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใดก็ตามที่เราเขียนเมธอดเหล่านี้ขึ้นเอง(override)เมธอดเหล่านี้จะรับ 2 อาร์กิวเมนต์ อาร์กิวเมนต์แรกคือ `HttpServletRequest` ซึ่งจะรับข้อมูลมาจากไคลเอนต์และอาร์กิวเมนต์ตัวที่ 2 คือ `HttpServletResponse` ซึ่งจะส่งผลการตอบสนองกลับไปยังไคลเอนต์

เราสามารถเข้าถึงข้อมูลจากไคลเอนต์ได้ซึ่งจะขึ้นอยู่กับเมธอด HTTP ของการเรียกครั้งดังนี้

- สำหรับเมธอด HTTP เราสามารถใช้เมธอด `getParameterValue` เพื่อรับค่าของพารามิเตอร์ (เมธอด `getParameterName` จะจัดหาชื่อของพารามิเตอร์มา)
- สำหรับการเรียกครั้งโดยการใช่เมธอด HTTP Get จะใช้เมธอด `getQuerystring` ซึ่งจะส่งค่าที่เป็น string กลับมา
- สำหรับเมธอด HTTP POST,PUT และ DELETE จะมีทางเลือกอยู่ 2 ทาง คือ ถ้าเราคาดว่าข้อมูลเป็นแบบ text เราจะสามารถอ่านได้โดยใช้ `BufferedReader` ที่ถูกส่งกลับมาโดยเมธอด `getReader` หรือถ้าคาดว่าข้อมูลเป็นชนิดไบนารีแล้วเราจะอ่านมันด้วย `ServletInputStream` ที่ถูกส่งกลับมาโดยเมธอด `getInputStream`

2.4.3.4 lifecycle method

1. Overriding the init method

ระหว่างการกำหนดค่าเริ่มต้นต่างๆ servlet จะเตรียมทรัพยากรที่จะถูกจัดการเพื่อพร้อมที่จะรองรับการเรียกครั้งจากไคลเอนต์ และเมื่อกำหนดค่าเริ่มต้นต่างๆเสร็จแล้วก็สามารถที่จะรองรับการเรียกครั้งจากไคลเอนต์ได้ ถ้าเกิดว่าทรัพยากรที่ต้องการไม่มีอยู่ เช่น ยังไม่ได้มีการเชื่อมต่อผ่านเครือข่าย แล้ว เมธอด `init` จะส่งผลให้เกิด exception `UnavailableException` ขึ้น

เมธอด `init` จะรับออบเจกต์ `ServletConfig` เป็นค่าพารามิเตอร์ เมธอดนี้จะเก็บออบเจกต์นี้ไว้ซึ่งเราจะเข้าถึงได้โดยใช้เมธอด `getServletConfig` แต่วิธีที่ง่ายที่สุดที่จะเข้าถึงคือเรียก `super.init` ในเมธอด `init`

ตามตัวอย่างเมธอด `init` ข้างล่างจะรับอินพุทมาจากฟอร์มและบันทึกลงในไฟล์ตั้งนั้นเพื่อที่จะเก็บข้อมูลที่ได้อาจจะต้องมีการกำหนดไคเรคทอรีที่ใช้เก็บซึ่งไคเรคทอรีนี้ได้มาจากค่าพารามิเตอร์ที่ถูกกำหนดในส่วนของการกำหนดค่าเริ่มต้น

```
public void init(ServletConfig config)
    throws ServletException
{
    super.init(config);

    //Store the directory that will hold the survey-results files
    resultsDir = getInitParameter("resultsDir");

    //If no directory was provided, can't handle clients
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (resultsDir == null) {
    throw new UnavailableException (this,
        "Not given a directory to write survey results!");
}
}

```

จากตัวอย่างเมธอด `init` จะเรียกเมธอด `super.init` เพื่อจัดการกับออบเจกต์ `ServletConfig` นอกจากนี้เมธอด `init` ยังเซตฟิลด์ `resultDir` ด้วยพร้อมกับชื่อไดเรกทอรีซึ่งจะจัดหาค่าพารามิเตอร์ที่ถูกกำหนดเริ่มต้น ซึ่งถ้าไม่มีชื่อไดเรกทอรีที่กำหนดแล้ว `servlet` จะทำให้เกิด `exception` ออกมา และถ้าผ่านขั้นตอนในเมธอด `init` แล้ว `servlet` จะสามารถรองรับการเรียกร็องจากไคลเอนท์ได้

2. Initialization parameter

รายละเอียดของค่าพารามิเตอร์ที่ถูกกำหนดขึ้นนั้นเป็นข้อมูลเฉพาะของเซิร์ฟเวอร์ ซึ่งวิธีที่จะใช้รับค่ามาจากค่าพารามิเตอร์ที่ถูกกำหนดขึ้นนั้นคือใช้คำสั่ง `getInitparameter` โดยจะรับชื่อพารามิเตอร์มาเป็นค่าอากิวเมนต์มา

3. Overriding the Destroy method

เมื่อเซิร์ฟเวอร์ยกเลิกการโหลด `servlet` แล้วมันจะเรียกเมธอด `destroy` โดยเมธอด `destroy` จะยกเลิกงานที่ทำเริ่มต้นและสแตทการทำงานที่ยังคงอยู่ในสแตทของหน่วยความจำ

ถึงแม้ว่าจะมีการใช้เมธอด `init` และ `destroy` แล้วก็ไม่จำเป็นที่จะต้องเรียกเมธอด `destroy` เสมอไป เช่น การตั้งค่าเริ่มต้นที่เป็นการอ่านไฟล์และใช้ข้อมูลที่ใช้ร่วมกันเท่านั้น ดังนั้นมันจึงไม่ต้องยกเลิกการทำงานเมื่อเซิร์ฟเวอร์เสร็จสิ้นการใช้ `servlet` แต่สำหรับ `servlet` ทั่วไปแล้วงานที่ถูกตั้งค่าเริ่มต้นต้องถูกยกเลิกเช่นสมมุติว่ามี `servlet` ซึ่งเปิดการเชื่อมต่อกับฐานข้อมูลในระหว่างการตั้งค่าเริ่มต้น ดังนั้นจะต้องมีเมธอด `destroy` เพื่อปิดการเชื่อมต่อกับฐานข้อมูลดังตัวอย่างข้างล่าง

```

/**
Cleans up database connection

```

```

*/ public void destroy()

```

```

{

```

```

try

```

```

{

```

```

    con.close();

```

```

}

```

```

catch (SQLException e)

```

```

{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(e != null)
{
    log("SQLException: " + e.getSQLState() + "\t" + e.getMessage() + "\t" + e.getErrorCode() + "\t");
    e = e.getNextException();
}
}
}

catch (Exception e) { e.printStackTrace(); } }

```

4. คัดลอกด้วย Service threads ที่ Servlet Termination

เมื่อเซิร์ฟเวอร์เคลื่อนย้าย servlet ออกไปแล้วโดยทุกๆ ไปมันจะเรียกเมธอด destroy หลังจากที่มีการบริการทั้งหมดได้ถูกกระทำเสร็จสิ้นสมบูรณ์แล้ว ถ้า servlet มีการทำงานที่ใช้เวลานานแล้ว thread จะยังคงรันอยู่จนกว่าเมธอด destroy จะถูกเรียก ดังนั้นผู้พัฒนาโปรแกรม servlet ควรที่จะทำให้อย่างมั่นใจได้ว่า thread จะยังคงรองรับการเรียกใช้จากไคลเอนท์ที่อยู่จนกว่าจะสมบูรณ์

Servlet จะคอยติดตามเมธอดที่ให้บริการต่างๆ ที่กำลังรันอยู่ โดยจะมีการตรวจสอบเป็นครั้งคราวเพื่อให้มั่นใจได้ว่ามันยังคงรันอยู่ ซึ่งถ้า servlet ถูกทำลายแล้วเมธอดที่กำลังรันอยู่จะถูกหยุด

2.4.4 Java Database Connectivity (JDBC)

ปัจจุบันจาวาเป็นภาษาที่มีการใช้งานกันอย่างแพร่หลายทั้งทางด้านกราฟฟิก หรือใช้ในการทำโฮมเพจ เป็นต้น ซึ่งเป็นเพราะคุณสมบัติของตัวจาวาที่สนับสนุนการทำงานต่าง ๆ มากมาย เช่น เป็นการทำงานที่ไม่ขึ้นอยู่กับแพลตฟอร์ม การได้คอบเบตเรียลไทม์ ความสามารถในการเล่นไฟล์เสียงไฟล์วิดีโอไฟล์กราฟฟิกเคลื่อนไหว เป็นระบบ Multithreaded คือสามารถทำงานได้หลายงานพร้อมกันในเวลาเดียวกัน มีความปลอดภัยสูง เป็นต้น นอกจากนั้นยังมีการพัฒนาองค์ประกอบใหม่ หลาย ๆ อย่างในการปฏิบัติงานในด้านต่าง ๆ ด้วยภาษาจาวา ซึ่งหนึ่งในจำนวนนั้นก็คือ Java Database Connectivity (JDBC) ซึ่งทำหน้าที่อำนวยความสะดวกในการเชื่อมต่อและสื่อสารกับฐานข้อมูล

ความหมายของ JDBC

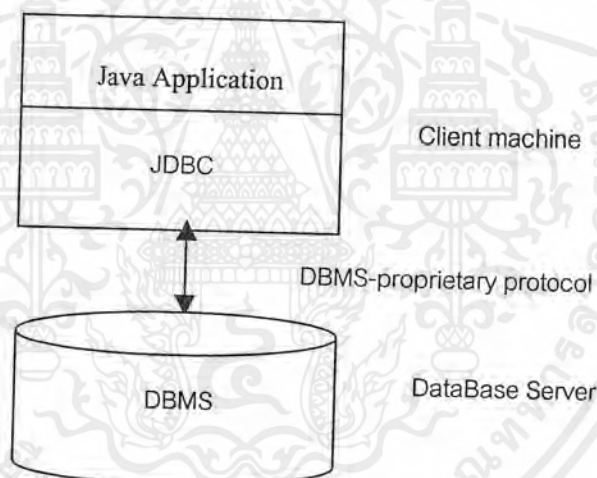
JDBC (Java DataBase Connectivity) เป็นฟังก์ชันมาตรฐานหรือ Java Application Programming Interfaces (API) สำหรับการเชื่อมต่อกับฐานข้อมูล โดย JDBC จะสร้างระดับการเชื่อมต่อเพื่อสื่อสารกับฐานข้อมูลในรูปแบบที่คล้ายคลึงกับ ODBC (Open DataBase Connectivity) ซึ่งในปัจจุบันถือได้ว่าเป็นมาตรฐานของการเชื่อมต่อกับฐานข้อมูล แต่หลักการทำงานของทั้ง JDBC และ ODBC ตั้งอยู่บนมาตรฐานเดียวกันคือ X/Open SQL Call-Level Interface ของระบบ X-Windows และ JDBC driver ต้องเข้ากันได้กับระดับมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standard) ซึ่งหน้าที่หลักของ JDBC คือการส่งคำสั่ง SQL และรับผลลัพธ์ที่ได้มาจากฐานข้อมูล ซึ่งมันจะถูกกำหนดด้วยคลาสของจาวาเพื่อแสดงการติดต่อกับฐานข้อมูล, SQL statement ,result sets, metadata และอื่น ๆ

รูปแบบในการเชื่อมต่อฐานข้อมูลของ JDBC

JDBC API สนับสนุนรูปแบบการเชื่อมต่อฐานข้อมูลทั้งแบบ Two-Tier model และ Three-Tier model

Two-Tier model

จาวาแอปพลิเคชันหรือจาวาแอปพลิเคชันจะติดต่อกับฐานข้อมูลโดยตรงจึงมีความจำเป็นที่โปรแกรมจาวาต้องการ JDBC driver ที่สามารถสื่อสารกับระบบจัดการฐานข้อมูลได้ รูปแบบการเชื่อมต่อแสดงดังรูปที่ 2-26 คำสั่งในการเรียกค้นข้อมูลในรูปของภาษา SQL (Structure Query Language) จะถูกส่งจากผู้ใช้ไปสู่ฐานข้อมูล หลังจากนั้นผลจากการประมวลผลของระบบจัดการฐานข้อมูลก็จะถูกส่งกลับมาสู่ผู้ใช้ฐานข้อมูล รูปแบบ Two-Tier นี้ใช้หลักการทำงานเช่นเดียวกับรูปแบบไคลเอนท์เซิร์ฟเวอร์ที่เรารู้จักกันดี โดยที่เครื่องคอมพิวเตอร์ของผู้ใช้คือ ไคลเอนท์และเครื่องคอมพิวเตอร์ที่ให้บริการฐานข้อมูลคือเซิร์ฟเวอร์ เครื่องข่ายคอมพิวเตอร์ที่นิยมใช้รูปแบบ Two-Tier มักจะเป็นเครือข่ายอินทราเน็ต (Intranet) สำหรับดำเนินธุรกรรมภายในองค์กร

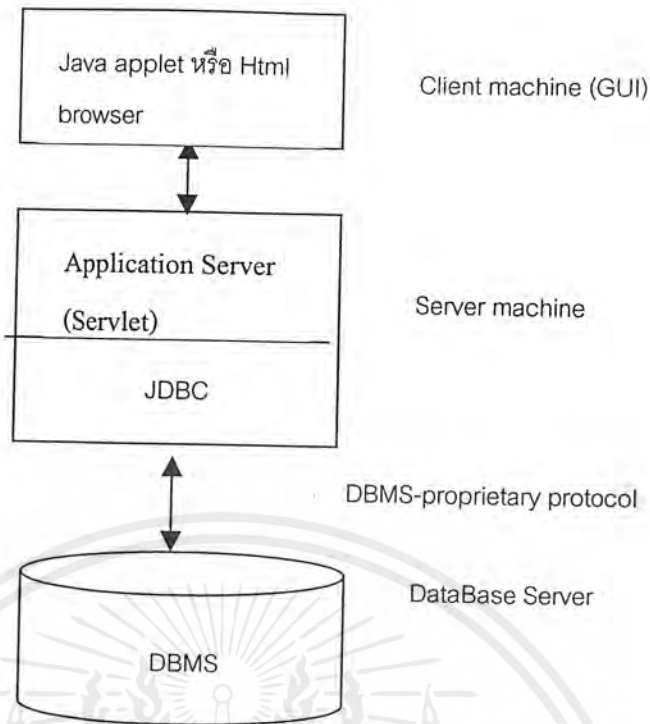


รูปที่ 2-26 แสดงการเชื่อมต่อแบบ Two-Tier model

Three-Tier model

คำสั่งเรียกค้นต่าง ๆ จากผู้ใช้จะถูกส่งไปให้กับ Middle Tier หรือส่วนกลางของการบริการเสียก่อน หลังจากนั้น Middle Tier จะแปลงคำสั่งเหล่านั้นให้เป็นภาษา SQL เพื่อส่งไปให้กับระบบจัดการฐานข้อมูลเพื่อทำการประมวลผล ข้อมูลผลลัพธ์ที่ได้จะถูกส่งกลับคืนไปให้กับ Middle Tier และส่งต่อไปให้กับผู้ใช้ในที่สุด หลักการทำงานเช่นนี้มักพบในเครือข่ายอินเทอร์เน็ตซึ่งประกอบไปด้วยเครื่องคอมพิวเตอร์หลากหลายชนิด และเว็บเซิร์ฟเวอร์ (Web-Server) ซึ่งทำหน้าที่เป็นตัวกลางในการจัดการให้คอมพิวเตอร์ทั้งหลายทั้งไคลเอนท์และเซิร์ฟเวอร์ฐานข้อมูลสามารถสื่อสารกันได้ และนอกจากนี้การปรับเปลี่ยนระบบคอมพิวเตอร์ของไคลเอนท์และเซิร์ฟเวอร์รวมทั้งการเปลี่ยนฐานข้อมูลตัวใหม่ก็จะมีผลกระทบต่อซึ่งกันและกัน รูปที่ 2-27 แสดงโครงสร้างของ Three-Tier model

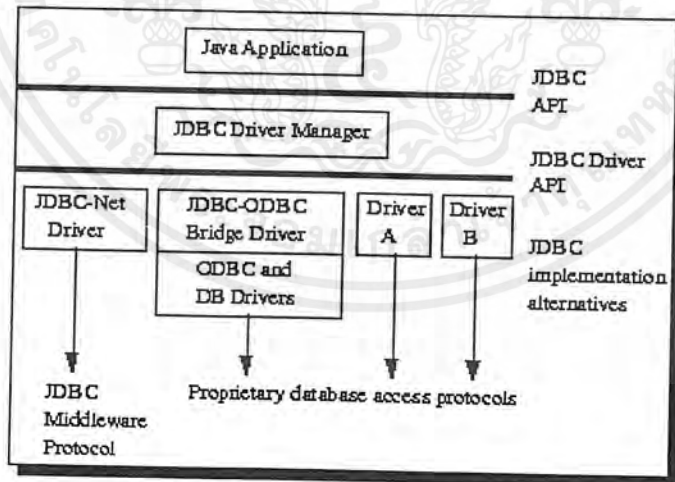
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-27 แสดงโครงสร้างของ Three-Tier model

โครงสร้างของ JDBC

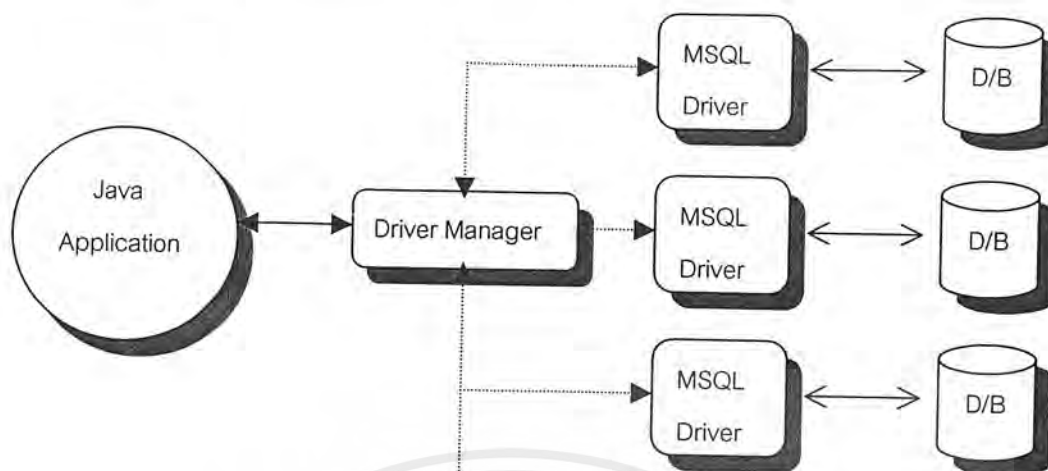
โครงสร้างการเชื่อมต่อภายใน JDBC ประกอบไปด้วย 3 ระดับหลักคือ JDBC API, JDBC Driver API และ JDBC Driver ดังรูปที่ 2-28



รูปที่ 2-28 โครงสร้างการเชื่อมต่อภายใน JDBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนรูปที่ 2-29 แสดงส่วนประกอบหลัก ๆ ของ API



รูปที่ 2-29 แสดงส่วนประกอบหลัก ๆ ของ API

Drive Manager

Drive Manager จะเป็นตัวควบคุมการติดต่อระหว่างแอปพลิเคชันและไดรฟ์เวอร์ที่ใช้ ซึ่งมันสนับสนุนการเชื่อมต่อหลาย ๆ ไดรฟ์เวอร์กับระบบจัดการฐานข้อมูลต่าง ๆ (DBMSs) โดยมันจะดึงเอาไดรฟ์เวอร์ที่ต้องการและจัดการการสนับสนุนสำหรับการจัดการการเชื่อมต่อกับฐานข้อมูล

Drivers

ไดรฟ์เวอร์ให้การอินเตอร์เฟซระหว่างแอปพลิเคชันและระบบจัดการฐานข้อมูล (DBMS) ซึ่งโดยทั่วไปผู้ผลิต DBMS ต่าง ๆ จะมี JDBC drivers มาให้ เนื่องจากไดรฟ์เวอร์สามารถถูกเขียนขึ้นด้วยภาษาจาวาดังนั้นไดรฟ์เวอร์จึงสามารถถูกดาวน์โหลดด้วยแอปพลิเคชันได้ แต่ละไดรฟ์เวอร์จะจัดหาคلاسซึ่งทำให้เกิด `java.sql.Driver` interface ซึ่งอินเตอร์เฟซนี้จะถูกใช้โดยคลาส `java.sql.DriverManager` เพื่อกำหนดไดรฟ์เวอร์สำหรับ database URL และแต่ละไดรฟ์เวอร์ต้องให้การสนับสนุนของการอินเตอร์เฟซดังนี้

- `Java.sql.Connection`
- `Java.sql.Statement`
- `Java.sql.PreparedStatement`
- `Java.sql.CallableStatement`
- `Java.sql.Resultset`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JDBC Driver มีอยู่ด้วยกัน 4 ชนิดคือ

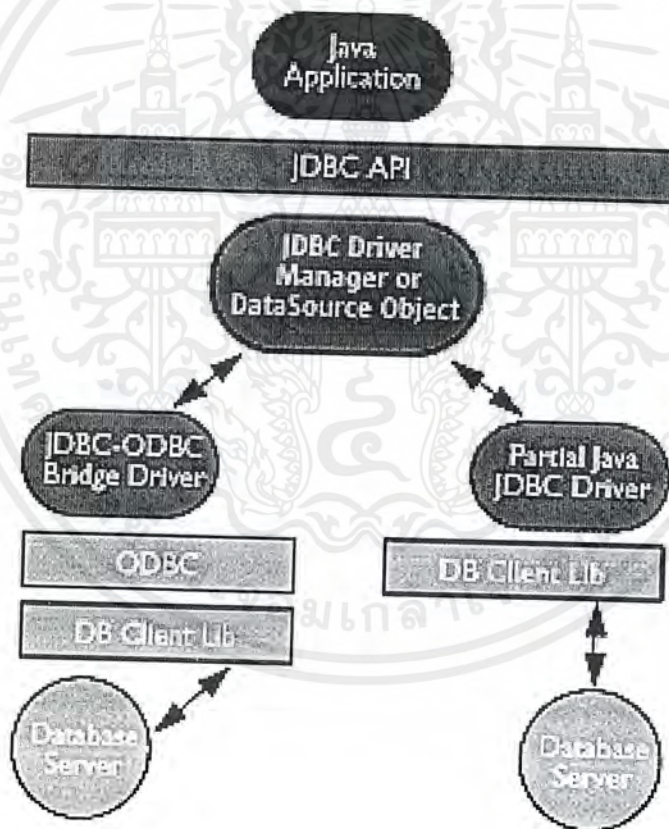
1. JDBC-ODBC Bridge plus ODBC Driver

ไดร์เวอร์ชนิดนี้จะจัดการการเข้าถึงฐานข้อมูลโดยใช้ JDBC ผ่านทางไดร์เวอร์ ODBC ซึ่งไดร์เวอร์ ODBC นี้จะต้องถูกโหลดลงแต่ละเครื่องไคลเอนท์ที่ใช้ JDBC-ODBC Bridge โดยทางบริษัท Sun นั้นให้ไดร์เวอร์ชนิดนี้มาให้แล้วซึ่งเหมาะสำหรับใช้ในการทดลองและจำลองสถานการณ์ต่างๆ ซึ่งไม่สามารถใช้ไดร์เวอร์ชนิดอื่นได้

2. A native-API partly Java Technology-enabled driver

ไดร์เวอร์ชนิดนี้จะแปลง JDBC ไปเป็น Client API สำหรับตัวจัดการฐานข้อมูลแต่ละชนิด (DBMS) ซึ่งจะเหมือนกับไดร์เวอร์ตัวแรกตรงที่ว่า ไดร์เวอร์ตัวนี้ต้องการโค้ดไบนารีบนแต่ละเครื่องไคลเอนท์

ไดร์เวอร์แบบที่ 1 และ 2 นั้นมีลักษณะการเชื่อมต่อดังรูป



รูปที่ 2-30 แสดงลักษณะของการเชื่อมต่อไดร์ฟเวอร์ชนิดที่ 1 และ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

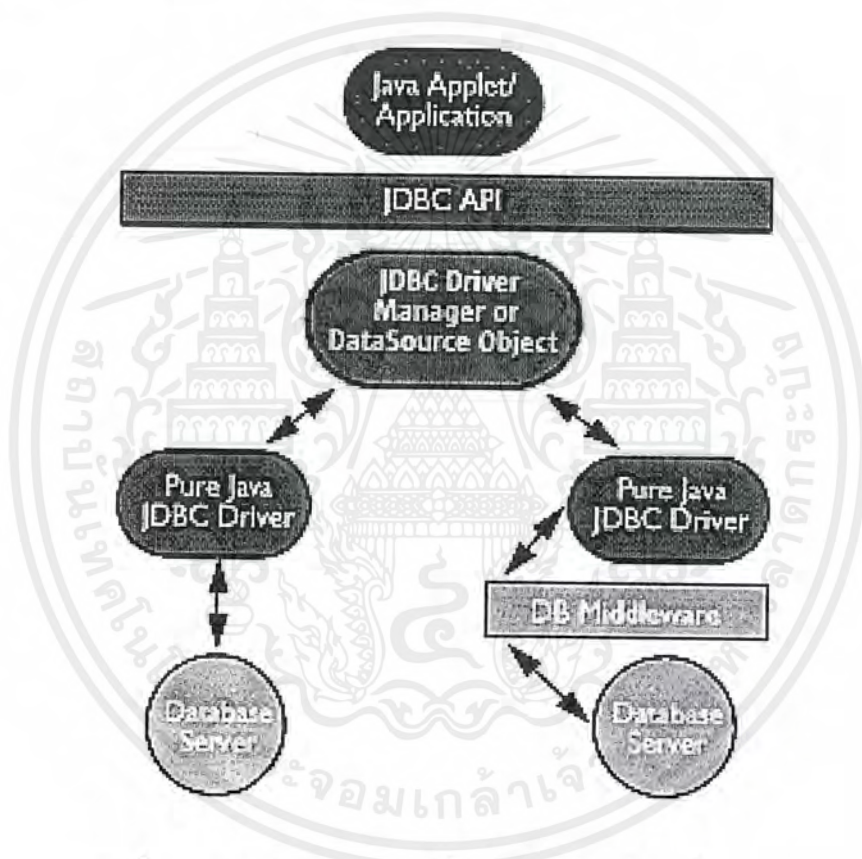
3. A net-protocol fully Java technology-enabled driver

ไดร์เวอร์ชนิดนี้จะแปลง JDBC ไปยังโปรโตคอลของแต่ละฐานข้อมูล โดยมีเครื่องที่เป็นเซิร์ฟเวอร์เป็นตัวกลางซึ่งทำหน้าที่ในการจัดการการเชื่อมต่อกับฐานข้อมูลชนิดต่างๆ

4. A native-protocol fully Java technology-enabled driver

จะแปลง JDBC ไปยังโปรโตคอลของเครือข่ายที่ถูกใช้จริงๆ โดยโปรแกรมจัดการฐานข้อมูล และอนุญาตให้มีการเรียกตรงๆ ได้จากเครื่องไคลเอนท์ไปยังเครื่องเซิร์ฟเวอร์ที่จัดการฐานข้อมูล และยังจัดหาผลลัพธ์สำหรับการเข้าถึงอินทราเน็ตด้วย

ไดร์เวอร์แบบที่ 3 และ 4 นี้มีลักษณะดังรูปที่ 2-31 โดยที่แอปพลิเคชันและแอปเพล็ตสามารถเข้าถึงฐานข้อมูลได้โดยผ่านไดร์เวอร์ JDBC API ทั้ง 2 นี้



รูปที่ 2-31 แสดงลักษณะของการเชื่อมต่อไดร์เวอร์ชนิดที่ 3 และ 4

ขั้นตอนพื้นฐานสำหรับการเชื่อมต่อกับฐานข้อมูล

1. เปิดการเชื่อมต่อกับฐานข้อมูล
2. ส่งคำสั่ง SQL ไปให้แก่ฐานข้อมูล
3. จัดการกับผลลัพธ์ที่ได้
4. ปิดการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเพื่อให้เข้าใจการทำงานของ JDBC ยิ่งขึ้นตัวอย่างโปรแกรมบางส่วนต่อไปนี้นี้จะแสดงให้เห็นถึงขั้นตอนทั้ง 4 ในการติดต่อกับฐานข้อมูลผ่าน ODBC ไดรเวอร์โดยใช้ JDBC-ODBC bridge

```
//(1)เปิดการเชื่อมต่อกับฐานข้อมูล
Class.forName("com.inet.tds.TdsDriver");
Connection con=DriverManager.getConnection("jdbc:inetdae:localhost",login,password);
//(2) ส่งคำสั่ง SQL ไปให้แก่ฐานข้อมูล
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("SELECT a,b,c FROM Table1");
//(3)จัดการกับผลลัพธ์ที่ได้
while(rs.next())
{
int I=rs.getInt("a");
String s=rs.getString("b");
Float f=rs.getFloat("c");
}
//(4)ปิดการเชื่อมต่อ
rs.close();
stmt.close();
con.close();
```

จากตัวอย่างสิ่งแรกที่ทำคือเลือกชนิดของ JDBC ไดรเวอร์(type 3) ระบุชื่อฐานข้อมูล (DSN:Data Source Name) รวมถึงชื่อผู้ใช้ (Login) และรหัสผ่าน>Password) จากนั้นเป็นการสร้างออบเจกต์ stmt สำหรับการเรียกค้นข้อมูล และจัดส่งคำสั่ง SQL ไปในรูปแบบสตริงผ่านเมธอด executeQuery ผลลัพธ์ที่ได้จากเมธอดนี้คือผลลัพธ์จากการประมวลผลของฐานข้อมูลซึ่งจะเก็บอยู่ในออบเจกต์ rs จากนั้นข้อมูลที่ได้รับนี้จะถูกถ่ายทอดให้อยู่ในชนิดข้อมูลที่จำวารู้จัก เช่น int,String,float เพื่อนำไปใช้ประโยชน์อื่น ๆ ต่อไป และสุดท้ายคือการปิดการเชื่อมต่อกับฐานข้อมูล

ความก้าวหน้าของเทคโนโลยี JDBC

- Leverage Existing Enterprise Data

ธุรกิจไม่ได้ผูกติดอยู่กับสถาปัตยกรรมใดๆ และสามารถดำเนินต่อไปโดยใช้ฐานข้อมูลที่ถูกติดตั้ง และสามารถเข้าถึงข้อมูลได้อย่างง่ายดาย

- Simplified Enterprise Development

การรวมกันของ Java API และ JDBC API นั้นทำให้สามารถสร้างแอปพลิเคชันได้ง่ายและประหยัดขึ้น โดย JDBC จะซ่อนความซับซ้อนในส่วนของข้อมูลต่างๆ ที่ถูกเข้าถึงไว้

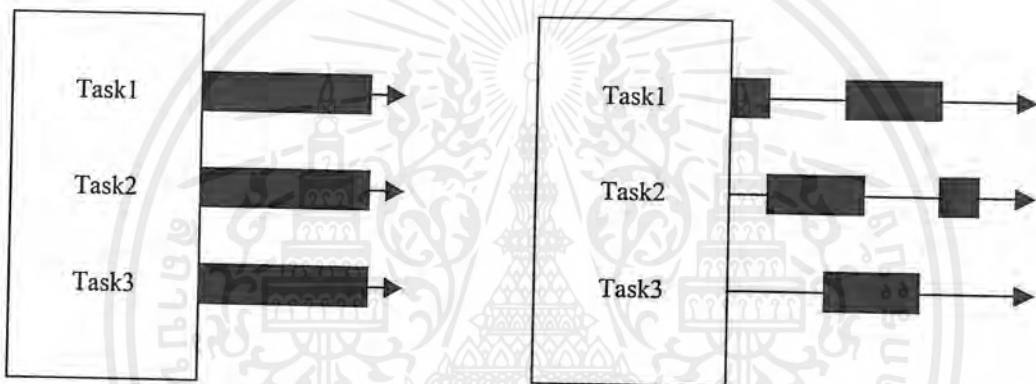
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Zero Configuration for Network Computers

ด้วยการใช้ JDBC API ทำให้ไม่ต้องมีการปรับแต่งบนฝั่งไคลเอนท์ และจากไดรเวอร์ที่ถูกเขียนขึ้นโดยภาษาจาวาทำให้ข้อมูลทั้งหมดที่ถูกส่ง การเพื่อสร้างการเชื่อมต่อนั้นถูกกำหนดขึ้นอย่างสมบูรณ์แบบโดย JDBC URL หรือ ออบเจกต์ DataSource ซึ่งถูก register ด้วย Java Naming and Directory Interface (JNDI)

2.4.5 การทำมัลติเทรด (Multithreading program)

เทรด (thread) มีลักษณะเหมือนโปรแกรมทั่ว ๆ ไปคือมีทั้งจุดเริ่มต้นและจุดสิ้นสุดอย่างชัดเจนในช่วงเวลาใดเวลาหนึ่งและเวลาใด ๆ ขณะที่เทรดกำลังทำงานจะมีจุดประสงค์เพื่อทำการเอ็กซีคิวท์ (execute) โปรแกรมเพียงอย่างเดียว แต่อย่างไรก็ตามโดยตัวมันเองแล้วเทรดไม่ใช่โปรแกรมเพราะไม่สามารถทำงานด้วยตัวเองได้ แต่ต้องทำงานร่วมกับโปรแกรม



รูปที่ 2-32 แสดงถึงการทำงานของเทรดทั้งแบบที่มีโปรเซสเซอร์ตัวเดียวและแบบที่มีโปรเซสเซอร์หลายตัว

การทำงานของเทรดโดยทั่วไปแล้วไม่ได้เป็นการทำงานของเทรดเพียงอย่างเดียว แต่จะเป็นการทำงานของหลาย ๆ เทรดภายในโปรแกรมเดียวกัน โดยให้เทรดทำงานพร้อม ๆ กันแต่จะให้ทำหน้าที่คนละอย่างกัน ในหนังสือบางเล่มจะใช้คำว่า lightweight process แทนคำว่า เทรด และเหตุที่เรียกว่า lightweight ก็เพราะเทรดทำงานภายใต้สภาวะแวดล้อมเดียวกันกับตัว โปรแกรมและใช้ทรัพยากรที่โปรแกรมมีการจองไว้ แต่อย่างไรก็ตามเทรดจะต้องมีทรัพยากรบางอย่างที่เป็นของตัวเอง เช่น สแตก (stack) โปรแกรมเคาน์เตอร์ (program counter) เป็นต้น โค้ดที่อยู่ในเทรดจะทำงานภายใต้สิ่งแวดล้อมที่กล่าวมาเท่านั้น ซึ่งอาจเรียกเทรดได้อีกชื่อหนึ่งว่า executing context

คุณสมบัติของเทรด

ภายในเทรดจะมี Thread body การทำงานทุกอย่างจะเกิดขึ้นใน Thread body ซึ่งอยู่ในเมธอด run() หลังจากเทรดถูกสร้างและทำการตั้งค่าแล้ว รันไทม์ซิสเต็ม (runtime system) จะเรียกเมธอด run() ของเทรดขึ้นมาทำงานโดยโค้ดในเมธอด run() จะเป็นตัวทำงานทุกอย่างของเทรดนั่นเองและบ่อยครั้งการ

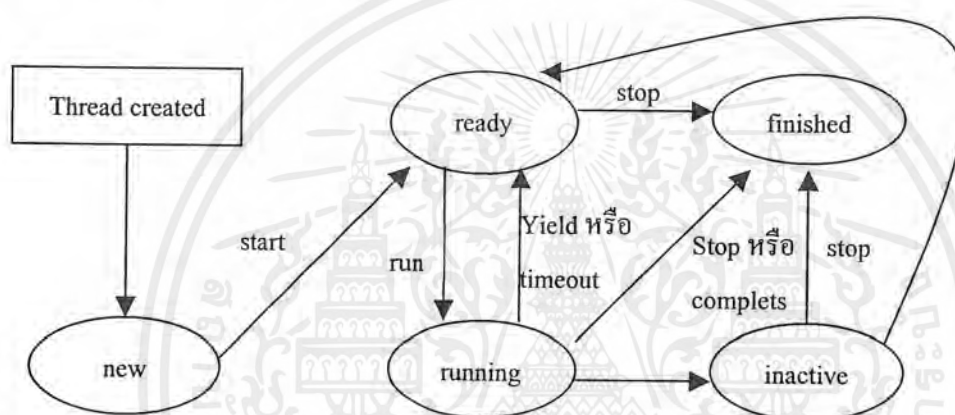
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานของเทรตจะมีการทำงานแบบลูป (loop) ทำให้การทำงานใช้เวลานาน เพื่อลดเวลาและขนาดของเมธอด run() เราสามารถกระทำได้ด้วยวิธีใดวิธีหนึ่งจากวิธีต่อไปนี้

1. สร้างสับคลาส (sub class) ของคลาส Thread ใน java.lang.Thread และทำการโอเวอร์ไรด์เมธอด run()
2. ทำการ implements Runnable interface ในแพ็คเกจ java.lang เช่นกัน แล้วใช้อินสแตนซ์ของเทรตเป็นตัวจัดการงานต่าง ๆ ที่ทำในเมธอด run() เช่นเดียวกัน

สถานะของเทรต

เทรตสามารถอยู่ในสถานะใดสถานะหนึ่งใน 5 สถานะต่อไปนี้คือ new, ready, running, inactive หรือ finished ซึ่งอาจแสดงได้ตามรูปที่ 2



รูปที่ 2-33 เทรตสามารถอยู่ได้ในหนึ่งใน 5 สถานะต่อไปนี้ new, ready, running, inactive หรือ finished

เมื่อเทรตถูกสร้างขึ้นเทรตจะเข้าสู่สถานะ new และหลังจากที่เทรตถูกสั่งให้เริ่มต้นด้วยเมธอด start() เทรตจะเข้าสู่สถานะ ready ซึ่งที่สถานะ ready นี้เทรตพร้อมที่จะทำงานแล้วแต่ยังไม่มีการทำงานใดๆ เกิดขึ้น ขึ้นกับระบบปฏิบัติการจะเป็นผู้จอง CPU time ให้กับเทรตนั้น ๆ

เมื่อเทรตที่พร้อมจะทำงานถูกสั่งให้ทำงาน (execute) เทรตจะเข้าสู่สถานะ running ซึ่งเทรตที่กำลังทำงานอยู่นี้อาจเข้าสู่สถานะ ready ได้อีกครั้งหนึ่ง เมื่อหมดเวลารหรือ CPU time ที่ระบบปฏิบัติการจัดมาให้หรืออาจเกิดจากเมธอด yield() ถูกเรียกให้ทำงานก็ได้

เทรตสามารถเข้าสู่สถานะ inactive ได้จากหลาย ๆ เหตุผล กล่าวคืออาจเรียกใช้เมธอด sleep(), wait() หรือ suspend() ก็ได้ แต่กรณีที่เทรตมีการติดต่อกับระบบ I/O เทรตก็จะเข้าสู่สถานะ inactive ได้เช่นกัน เทรตที่อินแอกทีฟ (inactive thread) อยู่จะกลับมาแอกทีฟได้อีกครั้งหนึ่งเมื่อกิจกรรมที่ทำให้เกิดการอินแอกทีฟเสร็จสิ้น ยกตัวอย่างเช่นเทรตถูกสั่งให้ sleep แล้วเวลาดังกล่าวก็หมดเวลา ในกรณีนี้เองเทรตก็จะกลับมาเป็นแอกทีฟเทรต และเข้าสู่สถานะ ready อีกครั้งนั่นเอง ท้ายสุดเทรตจะเข้าสู่สถานะ finished เมื่อทำการรันจนเสร็จสิ้นในเมธอด run() หรือมีการเรียกใช้เมธอด stop()

ระดับความสำคัญของเทรด (Thread Priority)

เทรดทุกตัวในจาวาจะมีระดับความสำคัญ โดยปกติแล้วเทรดจะได้รับระดับความสำคัญมาจากโปรแกรมที่แตกเทรดออกมา แต่เราก็สามารถเปลี่ยนระดับความสำคัญของเทรดได้โดยใช้เมธอด `setPriority()` และสามารถหาระดับความสำคัญของเทรดได้จากเมธอด `getPriority()` โดยที่ระดับความสำคัญในเทรดนั้นจะเป็นตัวเลขที่มีค่าจาก 1 ถึง 10 แต่ในคลาส `Thread` เองจะมีค่าคงที่ `MIN_PRIORITY`, `NORM_PRIORITY` และ `MAX_PRIORITY` ที่แทนด้วยค่า 1, 5 และ 10 ตามลำดับ โดยระดับความสำคัญของเทรดหลักจะเป็น `Thread.NORM_PRIORITY`

ในจาวารันไทม์ซิสเต็ม (Java runtime system) จะนำเทรดที่มีระดับความสำคัญที่สุดมารันก่อน แต่ถ้าเทรดเหล่านั้นมีลำดับความสำคัญเท่ากัน CPU จะถูกจองให้เทรดเหล่านั้นแบบ round-robin และเทรดที่มีระดับความสำคัญต่ำกว่าจะถูกนำมารันได้ก็ต่อเมื่อเทรดที่มีระดับความสำคัญสูงกว่าถูกรันไปจนหมดแล้วเท่านั้น

เทรดที่ทำงานอยู่เบื้องหลัง (Daemon Thread)

เทรดในจาวาสามารถเป็นเทรดที่ทำงานอยู่เบื้องหลังหรือเดมอนเทรดได้ เดมอนเทรดเป็นเทรดที่ให้บริการกับเทรดที่กำลังทำงานอยู่ในโปรเซส (process) เดียวกัน เมธอด `run()` ของเดมอนเทรดจะทำงานไปเรื่อย ๆ ไม่สิ้นสุดเพื่อรอการขอบริการจากเทรดอื่น และเมื่อมีเทรดเพียงเทรดเดียวเหลืออยู่ในโปรเซสคือเดมอนเทรด ตัวอินเตอร์พรีเตอร์ (interpreter) ก็จะใช้บริการทำงาน (เนื่องจากไม่มีเทรดอื่นที่ต้องบริการ)

ในการกำหนดให้เทรดเป็นเดมอนเทรดทำได้โดยใช้เมธอด `setDaemon()` พร้อมใส่อาร์กิวเมนต์ "true" ส่วนการตรวจสอบว่าเทรดใดเป็นเดมอนเทรดจะใช้เมธอด `isDaemon()`

กลุ่มของเทรด

เทรดทุกเทรดในจาวาเป็นสมาชิกของกลุ่มเทรด (thread group) กลุ่มเทรดเป็นเครื่องมือในการรวมมัลติเทรดเข้าด้วยกันเป็นออบเจกต์เดียวกัน และใช้เทรดเหล่านั้นทั้งหมดพร้อมกัน กลุ่มเทรด implements คลาส `Thread Group` ในแพคเกจ `java.lang`

รันไทม์ซิสเต็มจะทำการรวมเทรดทุกเทรดของเราเข้ากับกลุ่มของเทรดที่มีอยู่แล้ว หรือจะตั้งกลุ่มเทรดขึ้นมาใหม่ก็ได้และเมื่อเทรดเป็นสมาชิกของกลุ่มเทรดใดแล้วก็ไม่สามารถย้ายไปยังเทรดกลุ่มอื่นได้

โปรแกรมแบบมัลติเทรด

การซิงโครไนซ์เทรด (Synchronizing Thread) บ่อยครั้งจะมีการใช้ข้อมูลร่วมกันและจำเป็นต้องพิจารณาถึงสถานะและกิจกรรมที่เทรดอื่นทำอยู่ โดยกลุ่มของโปรแกรมที่อยู่ในสถานะการณ์นี้เรียกว่า producer และ consumer เข้าใจหะกัน (synchronize) ปัญหาที่เกิดจากการทำงานไม่เข้าใจหะกันเรียกว่า race conditions ซึ่งเกิดขึ้นได้เมื่อเทรดทำงานไม่เข้าใจหะกันและทำการเข้าถึงข้อมูลหรือออบเจกต์ตัว

เดียวกัน ทำให้ได้ข้อมูลที่ไม่ถูกต้อง เราสามารถแก้ปัญหานี้ได้ 2 วิธีคือ monitors และใช้ notifyAll คู่กับ เมธอด wait()

1. Monitors

ออบเจกต์ที่ถูกใช้ร่วมกันระหว่างเธรดสองตัวที่จะทำการเข้าถึงจะต้องทำให้เข้าจังหวะกัน ซึ่งเรียกว่า ข้อมูลควบคุม (condition variable) monitor จะเกี่ยวข้องกับการกำหนดข้อมูล (condition variable) และการล็อก (lock) ข้อมูลเหล่านั้น เมื่อเธรดทำการ monitor บนตัวข้อมูลตัวใด แล้วมีเธรดอื่นมาทำการล็อกข้อมูลจะทำให้เธรดที่ทำการ monitor บนข้อมูลนั้น (แต่ไม่ได้เป็นคนล็อกข้อมูล) ไม่สามารถใช้ข้อมูลตัวนั้นได้ ส่วนของโค้ดที่อยู่ในโปรแกรมที่มีการเข้าถึงข้อมูลตัวเดียวกันในเธรดเรียกว่า critical sections ซึ่งในจาวาสามารถกำหนดตัวโปรแกรม (เมธอดหรือตัวแปรให้เป็น critical sections ได้ โดยใช้คีย์เวิร์ด synchronized

2. notifyAll กับเมธอด wait()

เมธอดทั้งคู่นี้เป็นสมาชิกของ java.lang.Object ซึ่ง notifyAll และเมธอด wait() ถูกใช้งานโดย เธรดที่เป็นผู้ถือล็อกอยู่

- notifyAll ใช้ในการบอกให้กับเธรดอื่นที่รอบน monitor โดยให้เธรดปัจจุบันตื่นขึ้นหนึ่งในจำนวนที่รออยู่ ส่วนตัวที่เหลือก็ monitor และทำงานไป
- เมธอด wait() จะทำให้เธรดปัจจุบันหยุดการทำงานและรอจนกว่าเธรดอื่นจะบอกถึงการเปลี่ยนแปลงสถานะ

2.4.6 ซ็อกเกต (socket)

ซ็อกเกตเป็นกลไกระดับล่างที่ใช้ในการแลกเปลี่ยนข้อมูล ซึ่งใช้รับส่งข้อมูลข้ามเครือข่าย TCP/IP โดยข้อมูลที่ใช้ในการรับส่งเรียกว่าดาตาแกรม (Datagram) ซึ่งซ็อกเกตจะมี 2 แบบคืออินเตอร์เน็ตโปรโตคอลซ็อกเกต (IP socket) กับยูสเซอร์ดาตาแกรมซ็อกเกต (UDP socket) โดยไอพีซ็อกเกตใช้รับส่งไอพีดาตาแกรมในขณะที่ยูดีพีซ็อกเกตใช้รับส่งยูดีพีดาตาแกรม โดยส่วนใหญ่แล้วไอพีซ็อกเกตจะนำมาใช้งานแพร่หลายกว่ายูดีพีซ็อกเกต

ในจาวาจะมีแพคเกจ java.net ที่สนับสนุนการใช้อินพุทและเอาท์พุทซ็อกเกต และเนื่องจาก java.net ถือเป็นแพคเกจหลักที่อยู่ในจาวา ดังนั้นจึงนำมาใช้ได้ในทุกมาตรฐานของเครื่องจำลองการทำงานจาวา (java virtual machine)

ข้อดีข้อเสียของซ็อกเกต

ซ็อกเกตเปรียบเหมือนการแลกเปลี่ยนข้อมูลข้ามเครือข่ายกันในระดับล่าง ซึ่งโดยหลักแล้วจะทำงานได้เร็วและใช้หน่วยความจำได้มีประสิทธิภาพมากกว่า ซึ่งพอสรุปได้ว่าซ็อกเกตเหมาะสมสำหรับการรับส่งข้อมูลง่าย ๆ จำนวนไม่มากนัก เช่น ข้อความ Ascii แต่จะไม่เหมาะสมสำหรับการส่งข้อมูลที่มีขนาดใหญ่และมีความซับซ้อนเช่นออบเจกต์

คลาส URL

คลาส URL เป็นคลาสที่ห่อ URL ไว้ใช้บอกที่อยู่ของเอกสารที่เราจะทำการดาวน์โหลดมาจากเซิร์ฟเวอร์ สามารถสร้างคลาสนี้ได้จาก constructor

```
public URL(String url) throws MalformedURLException
```

อาร์กิวเมนต์ตรงเป็นตัวบอกที่ตั้งของเอกสารที่ URL อ้างถึง เช่น ถ้าเราจะอ้างถึงเว็บไซต์ <http://www.kmitl.ac.th> เราสามารถสร้างคลาส URL ได้โดย

```
URL kmitl=new URL("http://www.kmitl.ac.th");
```

ซึ่งจะ throws exception ขึ้นถ้ารูปแบบตัวแปรไม่ถูกต้องตามรูปแบบของ URL ซึ่งเราสามารถจัดการกับ exception ได้โดยใช้ try-catch block หรือให้เมธอดที่มีการเรียกใช้ constructor นี้ throws MalformedURLException ก็ได้ (หรืออาจใช้คลาสบรรพบุรุษ Exception แทนก็ได้)

เมื่อเราทำการสร้าง URL ที่ชี้ไปยังที่ตั้งของเอกสารแล้วเราสามารถที่จะดึงเอกสารออกมาโดยใช้เมธอด OpenStream ซึ่งจะรีเทิร์น InputStream ที่เก็บเนื้อหาของเอกสารไว้ออกมาซึ่งสามารถอ่าน InputStream เพื่อเข้าถึงข้อมูลในเอกสารได้อย่างง่ายดาย เมธอด getContent รีเทิร์นออบเจกต์ที่ห่อเนื้อหาของเอกสารเอาไว้ และทำงานได้สำเร็จเมื่อเครื่องจำลองการทำงานจาวามีการดูแลและจัดการชนิดของเอกสารที่ทำการเข้าถึงอย่างเหมาะสม

คลาส InetAddress

คลาส InetAddress ใช้แสดงถึงแอดเดรสของ Internet host เราใช้คลาสนี้ในการที่จะสร้างชื่อเกตหรือออบเจกต์ดาแกรม ซึ่งคลาส InetAddress ไม่มี constructor แต่ใช้เมธอดสแตติกเพื่อที่จะรีเทิร์นอินสแตนซ์ของ InetAddress ออกมาให้

คลาส Socket

ใน Internet host หรือเครื่องเจ้าบ้านจะมีกลุ่มหมายเลขพอร์ตที่จะใช้สื่อสารกับเครื่องอื่น ซึ่งคลาสของชื่อเกตเป็นตัวห่อหมายเลขพอร์ตทั้งยังจัดหาเมธอดในการรับส่งข้อมูลผ่านพอร์ตดังกล่าวด้วย เราสามารถสร้างชื่อเกตโดยใช้ constructor

```
Socket(String host,int port) หรือ
```

```
Socket(InetAddress address,int port)
```

เช่นต้องการจะเปิดชื่อเกตบนพอร์ตหมายเลข 80 ที่เครื่อง www.kmitl.ac.th สามารถเขียนได้โดย

```
Socket kmitl=new Socket("www.kmitl.ac.th",80)
```

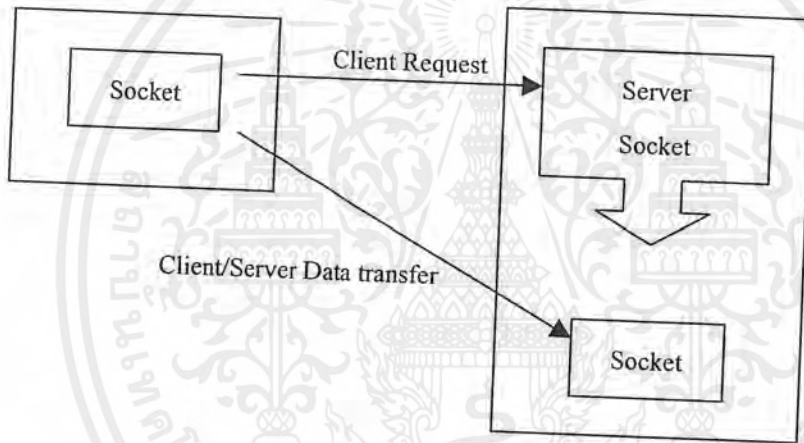
คลาส Socket ยังมี constructor ที่เฉพาะมากกว่าแต่เรานำมาใช้งานน้อยจึงไม่ขอกล่าวถึง ชื่อเกต constructor จะ throws exception ขึ้นเมื่อมีข้อผิดพลาดในการสร้างชื่อเกต ดังนั้นจึงจำเป็นต้องใช้ try-catch block หรือ throws clause เพื่อจัดการกับข้อผิดพลาดดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราได้สร้างซ็อกเก็ตขึ้นมาแล้ว เราสามารถใช้เมธอด `getInputStream` เพื่อใช้ติดต่อกับ `InputStream` และเมธอด `getOutputStream` เพื่อติดต่อกับ `OutputStream` โดยการใช้เมธอดที่ทำการอินพุท/เอาท์พุท ธรรมดาในการที่จะอ่านและเขียนผ่านสายข้อมูล (stream) นี้ เหมือนเหมือนทำการอินพุท/เอาท์พุทลงบนไฟล์ ในกรณีที่ใช้ `InputStream` และ `OutputStream` เราจะพบว่าคลาส `BufferedInputStream` และ `PrintStream` มีความสะดวกในการทำงานมากกว่า

คลาส `ServerSocket`

คลาส `ServerSocket` ถูกใช้โดยโปรแกรมของเซิร์ฟเวอร์ ซึ่งใช้เมธอด `accept` เพื่อรอการร้องขอการติดต่อจากไคลเอนท์และรีเทิร์นซ็อกเก็ตที่ใช้ในการติดต่อระหว่างไคลเอนท์กับเซิร์ฟเวอร์เพื่อแลกเปลี่ยนข้อมูลระหว่างกัน ข้อสำคัญคือ ซ็อกเก็ตออกบเจกต์ที่ส่งไคลเอนท์จะทำการสวิตช์โดยอัตโนมัติโดยใช้ซ็อกเก็ตตัวใหม่เพื่อทำการติดต่อสื่อสาร



รูปที่ 2-34 แสดงถึงการทำงานของ `ServerSocket`

ในการที่จะสร้างเซิร์ฟเวอร์ซ็อกเก็ตเราใช้

```
constructor ServerSocket(int port)
```

เมื่อ constructor เกิดข้อผิดพลาดก็จะ throws `IOException` แต่ถ้าไม่มีข้อผิดพลาดใด ๆ เซิร์ฟเวอร์ซ็อกเก็ตก็พร้อมที่จะใช้งาน เมื่อเราให้เซิร์ฟเวอร์ซ็อกเก็ตทำการรับข้อมูล (`accept`) การทำงานของโปรแกรมจะหยุดรอโดยไม่ทำงานใด ๆ จนกว่าจะมีการร้องขอการติดต่อจากไคลเอนท์ แล้วหลังจากนั้นเซิร์ฟเวอร์ซ็อกเก็ตจะรีเทิร์นซ็อกเก็ตให้ เช่น เพื่อจะสร้างเซิร์ฟเวอร์ซ็อกเก็ตที่รออยู่บนพอร์ตหมายเลข 1234 ในการติดต่อกับไคลเอนท์ทำได้โดย

```
ServerSocket server = new ServerSocket(1234);
```

```
Socket client = server.accept();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดาตาแกรม (Datagram)

ดาตาแกรมใช้ยูติพีซี้อกเกิดในการติดต่อสื่อสาร ซึ่งยูติพีซี้อกเกิดนี้จะไม่ทำการติดต่อให้อัตโนมัตเหมือนในไอพีซี้อกเกิดและยิ่งกว่านั้น TCP/IP ไม่รับประกันว่าเครื่องเจ้าบ้านทำการติดต่อกับยูติพีแพกเกจได้ (บางครั้งเราเรียกดาตาแกรม) ตามลำดับที่ถูกต้องเหมือนตอนที่ทำการส่งมา ดังนั้นยูติพีแพกเกจจึงใช้งานหลัก ๆ อยู่เพียงการส่งข้อมูลสั้น ๆ ซึ่งมีโครงสร้างข้อมูลไม่ซับซ้อน อีกทั้งผู้ดูแลระบบเครือข่ายในบางองค์กรได้ทำการปรับไฟล้วอลเพื่อทำยูติพีแพกเกจจากนอกองค์กร ดังนั้นเมื่อเราต้องการที่จะใช้ยูติพีแพกเกจจึงควรที่จะติดต่อสอบถามกับผู้ดูแลระบบก่อนที่จะทำการลงมือเขียนโปรแกรม เพื่อเป็นการประหยัดค่าใช้จ่ายด้านเวลา



บทที่ 3

การออกแบบและการสร้าง

3.1 การออกแบบและการสร้างในส่วนของเซิร์สเอนจิน

ในส่วนของเซิร์สเอนจิน ได้แบ่งออกเป็น 3 ส่วนด้วยกันคือ

3.1.1 การออกแบบและการสร้างในส่วนของการจัดเก็บข้อมูล

ส่วนนี้จะเป็นการออกแบบในส่วนของฐานข้อมูลซึ่งจะเก็บข้อมูลต่าง ๆ ที่ได้มาจาก spider เพื่อใช้ในการค้นหาทำให้ตรงกับผู้ใช้ต้องการต่อไป โดยข้อมูลที่จะจัดเก็บมีดังนี้
ข้อมูลที่เกี่ยวข้องกับคำที่ใช้ในการค้นหาข้อมูลที่ใช้ต้องการมีดังนี้

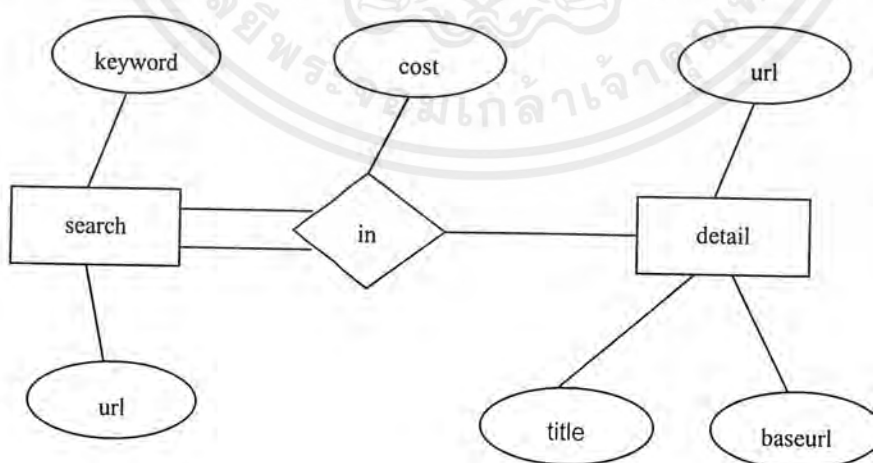
- keyword
- URL คือ เว็บเพจที่พบคีย์เวิร์ดนี้
- Cost คือ ค่าความสำคัญ

ข้อมูลที่เกี่ยวข้องกับ URL ที่ใช้ในการแสดงกลับไปยังผู้ใช้

- URL คือ ชื่อเว็บเพจ
- Title แสดงหัวข้อเรื่องของเว็บเพจ
- Baseurl คือ URL ที่เป็นเว็บไซต์เริ่มต้นของเว็บเพจนี้

และอีกส่วนหนึ่งที่มีการจัดเก็บไว้คือ คำที่เป็น stopword

ต่อไปจะออกแบบฐานข้อมูลโดยใช้หลักการของอีอาร์โมเดล ซึ่งจะได้ดังนี้

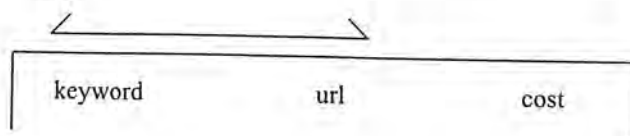


รูปที่ 3-1 แสดงอีอาร์โมเดล

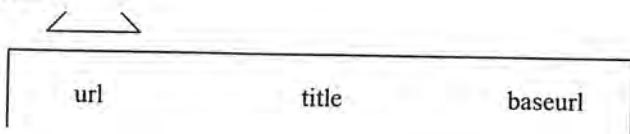
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปนำอีอาร์โมเดลที่ได้มา map เป็นตารางได้ดังนี้

ตาราง search



ตาราง detail



3.1.2 การออกแบบและการสร้างในส่วนของการดึงข้อมูลจากเว็บเพจมาเก็บในฐานข้อมูล

เมื่อสร้างฐานข้อมูลเสร็จแล้วขั้นตอนต่อไปคือการนำข้อมูลไปใส่ในฐานข้อมูลซึ่งข้อมูลต่าง ๆ เหล่านี้ได้มาจากโปรแกรม spider ซึ่งจะคอยวิ่งไปตามเว็บเพจต่าง ๆ เพื่อเก็บคีย์เวิร์ด ซึ่งได้ออกแบบในส่วนของโปรแกรม spider ซึ่งมีส่วนประกอบต่าง ๆ ดังนี้

3.1.2.1 ส่วนที่ทำการดึงข้อมูลมาจากเว็บเพจ

เป็นการดึงข้อมูลมาจากเว็บเพจแล้วนำไปเก็บไว้ในไฟล์ชั่วคราวเพื่อนำไปวิเคราะห์ข้อมูลในเว็บเพจต่อไป

3.1.2.2 ส่วนที่ทำการวิเคราะห์ข้อมูลในเว็บเพจ

ในส่วนนี้จะทำการอ่านข้อมูลที่อยู่ในไฟล์ชั่วคราวแล้วนำไปผ่านการตรวจสอบดังนี้

3.1.2.2.1 ตรวจสอบหาคีย์เวิร์ดที่อยู่ในเว็บเพจ

โดยส่วนต่าง ๆ ของเว็บเพจที่จะถูกนำไปค้นหาคีย์เวิร์ดมีดังนี้

- ใน tag <title>.....</title> ซึ่งคีย์เวิร์ดที่อยู่ในส่วนนี้จะมีค่าความสำคัญมากที่สุด
- ใน tag <meta name="keywords" content="....."> โดยคีย์เวิร์ดที่ได้มีค่าความสำคัญรองลงมาจากส่วนที่อยู่ใน tag <title>
- ใน tag <meta name="description" content="....."> โดยคีย์เวิร์ดที่อยู่ในส่วนนี้จะมีค่าความสำคัญรองมาจาก tag meta keywords

เริ่มแรกนำชื่อเว็บเพจไปเก็บไว้ในตาราง detail โดยนำชื่อเว็บเพจไปเก็บไว้ในคอลัมภ์ url ส่วนรายละเอียดที่ได้มาจาก tag <title>....</title> ให้นำไปใส่ไว้ในคอลัมภ์ title และ baseurl เก็บชื่อเว็บไซต์ที่เป็นเจ้าของเว็บเพจนั้น

เมื่อได้คีย์เวิร์ดมาแล้วก็ต้องนำไปตรวจสอบว่าคำที่ได้มานั้นเป็นคำที่ไม่สื่อความหมายในการค้นหาหรือไม่ถ้าเป็นเช่นนั้นก็ให้ตัดคำนั้นทิ้งไป แต่ถ้าไม่ได้อยู่ในคำที่ไม่สื่อความหมายแล้วก็นำไปวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่าค่า ๆ นั้นมีค่าความสำคัญต่อเว็บเพจนั้นเท่าไร จากนั้นก็นำไปใส่ไว้ในฐานข้อมูลในตาราง search โดยนำคีย์เวิร์ดใส่ไว้ในคอลัมภ์ keyword ส่วนชื่อเว็บเพจนำไปใส่ไว้ในคอลัมภ์ url และค่าความสำคัญที่ได้นำไปใส่ไว้ในคอลัมภ์ cost

ตัวอย่างในการคำนวณค่าความสำคัญ

เช่น พบคำว่า “song” ในเว็บเพจ www.mp3.com/a.html 5 คำ

ตำแหน่งแรกมีค่าความสำคัญเท่ากับ $10 = 10$

ตำแหน่งที่ 2,3 มีค่าความสำคัญเท่ากับ $5 = 10$

ตำแหน่งที่ 4 และ 5 มีค่าความสำคัญเท่ากับ $3 = 6$

เพราะฉะนั้นที่เว็บเพจ www.mp3.com/a.html มีคีย์เวิร์ดคำว่า “song” ซึ่งมีค่าความสำคัญเท่ากับ

26

3.1.2.2.2 ตรวจสอบหาชื่อเว็บเพจและชื่อเว็บไซต์

โดยจะหาชื่อเว็บเพจมาจากส่วนต่าง ๆ ดังนี้

- ใน tag ``
- ใน tag `<frame src=".....">`
- ใน tag `<Area href=".....">`

เมื่อได้ชื่อเว็บเพจมาแล้วก็ต้องนำไปผ่านการตรวจสอบต่อไปว่าใช้ได้หรือไม่ ซึ่งกระบวนการต่าง ๆ มีดังนี้

1. ตรวจสอบว่าชื่อเว็บเพจที่ได้มานั้นเป็นเว็บเพจของเว็บไซต์เดียวกันหรือไม่ ถ้าใช่ก็ผ่านไปยังขั้นตอนต่อไป แต่ถ้าไม่ใช่ก็ออกจากกระบวนการตรวจสอบชื่อเว็บเพจนี้และนำชื่อเว็บไซต์ใหม่ที่ได้ไปตรวจสอบว่ามีในรายการของชื่อเว็บไซต์หรือไม่ ถ้าไม่มีก็เพิ่มเข้าไป (ซึ่งรายการของชื่อเว็บไซต์จะเก็บชื่อเว็บไซต์ที่จะส่งโปรแกรม spider ไปทำการวิเคราะห์ข้อมูล)
2. นำชื่อเว็บไซต์ที่ได้ไปคำนวณลำดับไคเร็คทอรีที่ถูกต้อง และเปลี่ยนแปลงชื่อเว็บเพจให้ถูกต้องตามลำดับไคเร็คทอรี
3. นำชื่อเว็บเพจใหม่ไปตรวจสอบในรายการชื่อเว็บเพจว่ามีชื่อเว็บเพจนี้แล้วหรือไม่ ถ้าไม่มีก็ให้ผ่านไปยังขั้นตอนต่อไป แต่ถ้ามีแล้วก็ออกจากกระบวนการตรวจสอบชื่อเว็บเพจนี้
4. นำชื่อเว็บเพจไปตรวจสอบว่ามีชนิดข้อมูลเป็นแบบใด ถ้าเป็นชนิดข้อมูลที่ใช้ได้แล้วก็ให้ผ่านไปยังขั้นตอนต่อไป แต่ถ้าเป็นชนิดที่ใช้ไม่ได้ก็ให้ออกจากกระบวนการตรวจสอบชื่อเว็บเพจนี้
5. เมื่อชื่อเว็บเพจผ่านการตรวจสอบมาทั้ง 4 ขั้นตอนแล้ว ก็ให้เพิ่มชื่อเว็บเพจนี้เข้าไปในรายการชื่อเว็บเพจเพื่อใช้เป็น url ในการดึงข้อมูลต่อไป

ตัวอย่างเช่นพบคำสั่ง `` ในเว็บเพจ news/list.html ของเว็บไซต์ www.mp3.com

1. ได้ชื่อเว็บเพจเป็น song.html ซึ่งพบว่าเป็นของเว็บไซต์เดียวกันจึงผ่านไปยังขั้นตอนต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เนื่องจากว่าไครเรทอรีปัจจุบันคือ /news/ ดังนั้นชื่อเว็บเพจ song.html จึงอยู่ในไครเรทอรีนี้ ดังนั้นชื่อเว็บเพจที่ถูกต้องคือ news/song.html
3. นำไปตรวจสอบว่ามีชื่อ news/song.html ในรายการของชื่อเว็บเพจหรือไม่ถ้าไม่มีก็ผ่านไปยังขั้นตอนต่อไป แต่ถ้ามีแล้วก็ให้ออกจากการตรวจสอบชื่อเว็บเพจ news/song.html นี้
4. เมื่อนำไปตรวจสอบชนิดข้อมูลแล้วพบว่ามีชนิดเป็นไฟล์ html ซึ่งเป็นชนิดข้อมูลที่ใช้ได้ ดังนั้นจึงผ่านไปยังขั้นตอนต่อไป
5. นำชื่อ news/song.html ไปใส่ไว้ในรายชื่อของเว็บเพจ

3.1.2.3 ส่วนที่ทำหน้าที่ในการจัดเก็บข้อมูลลงฐานข้อมูล

เมื่อผ่านการวิเคราะห์ข้อมูลเสร็จแล้วก็จะนำข้อมูลที่ได้ออกไปเก็บไว้ในฐานข้อมูล โดยจะต้องนำคีย์เวิร์ดที่ได้ไปตรวจสอบก่อนว่าเป็นคำที่ไม่สื่อความหมายหรือไม่ ถ้าไม่เป็นก็ให้จัดเก็บลงฐานข้อมูล แต่ถ้าเป็นคำที่ไม่สื่อความหมายก็ไม่ต้องจัดเก็บลงฐานข้อมูล

การท่องไปตามเว็บเพจของโปรแกรม spider

เริ่มแรกจะต้องกำหนดชื่อเว็บไซต์เริ่มต้นให้กับโปรแกรมก่อน จากนั้นโปรแกรมก็จะเริ่มวิ่งไปตามชื่อเว็บไซต์แรก ถ้าเจอชื่อเว็บไซต์ใด ๆ ในไฟล์ข้อมูลที่ทำกรวิเคราะห์ก็ให้ชื่อเว็บไซต์นั้นเข้าไปในรายชื่อเว็บไซต์และถ้าเจอชื่อเว็บเพจก็ให้เพิ่มในรายชื่อเว็บเพจ

โดยโปรแกรมจะวิ่งไปที่ละเว็บไซต์และวิเคราะห์ข้อมูลจากทุก ๆ เว็บเพจที่เป็นของเว็บไซต์นั้นจนกว่าจะครบทุกเว็บเพจ เมื่อครบทุกเว็บเพจในเว็บไซต์หนึ่งแล้วก็เปลี่ยนไปยังเว็บไซต์ถัดไป และทำเช่นเดิมจนกว่าจะครบทุกเว็บไซต์ที่เจอ

ตัวอย่างกำหนดชื่อเว็บไซต์เริ่มต้นดังนี้ A,B,C

รายการชื่อเว็บไซต์เป็นดังนี้

A	B	C	Null			
---	---	---	------	--	--	--

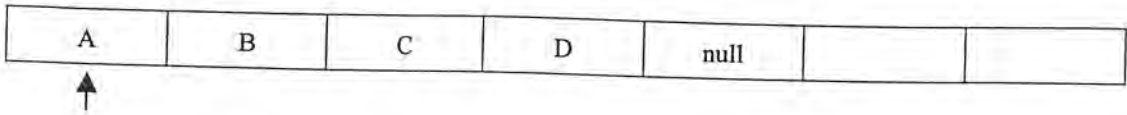


ซึ่งต่อไปจะส่งโปรแกรม spider ไปที่เว็บไซต์แรกคือ A ซึ่งจะสร้างรายชื่อเว็บเพจสำหรับเว็บไซต์ A ขึ้นมาดังนี้

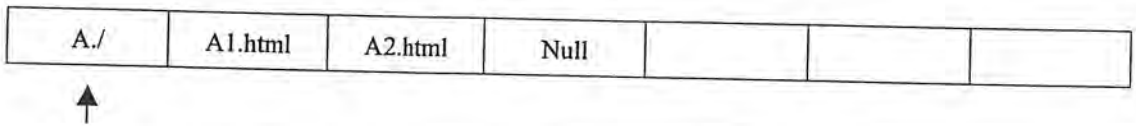
A/	null					
----	------	--	--	--	--	--



โดยชื่อเว็บเพจแรกคือ A/ ซึ่งมันจะไปเรียกเพจที่ถูกกำหนดขึ้นโดยอัตโนมัติของเว็บไซต์นี้มาจากนั้นนำข้อมูลจากเว็บเพจมาวิเคราะห์ ซึ่งถ้าเจอชื่อเว็บไซต์ D ซึ่งเป็นเว็บไซต์ที่ยังไม่มีในรายการชื่อเว็บไซต์ก็ให้เพิ่มเข้าไปในรายการชื่อเว็บไซต์ได้ดังนี้



และถ้าเจอชื่อเว็บเพจ A1.html และ A2.html ใน A/ ก็ให้เพิ่มในรายการชื่อเว็บเพจได้ดังนี้



และเมื่อวิเคราะห์ข้อมูลในเว็บเพจ A/ เสร็จแล้วก็เปลี่ยนเป็นเว็บเพจถัดไปคือ A1.html ดังรูป



และเมื่อวิเคราะห์ข้อมูลครบทุกเว็บเพจในรายการชื่อเว็บเพจของเว็บไซต์ A แล้วก็ให้เปลี่ยนเป็นเว็บไซต์ถัดไปคือเว็บไซต์ B โดยจะสร้างรายการชื่อเว็บเพจของเว็บไซต์ B ขึ้นมาเช่นเดียวกับเว็บไซต์ A และเมื่อไปครบทุกเว็บไซต์ก็จะออกจากโปรแกรมและแสดงผลว่าได้ไปที่เว็บไซต์ใดบ้างและแต่ละเว็บไซต์มีจำนวนเว็บเพจเท่าไร

เนื่องจากว่าเครือข่ายอินเทอร์เน็ตนั้นเป็นเครือข่ายที่มีขนาดใหญ่มากดังนั้นการที่จะสำรวจครบทุกเว็บไซต์จึงต้องใช้เวลาานมาก ดังนั้นจึงได้มีการกำหนดจุดสิ้นสุดของโปรแกรมโดยใช้จำนวนเว็บไซต์เป็นการบอกจำนวนสูงสุด

3.1.3 การออกแบบในส่วนของการติดต่อระหว่างผู้ให้บริการกับฐานข้อมูล

ในส่วนนี้จะเป็นติดต่อระหว่างผู้ให้บริการกับฐานข้อมูล โดยจะใช้ภาษา HTML เป็นตัวส่งข้อมูลต่าง ๆ มายังเครื่องเซิร์ฟเวอร์ ซึ่งการส่งข้อมูลต่าง ๆ มายังเครื่องเซิร์ฟเวอร์นั้นจะเป็นการส่งในรูปแบบของฟอร์ม (form) โดยที่ได้กำหนดจุดหมายปลายทางในการส่งเอาไว้ ซึ่งจุดหมายปลายทางที่เรียกจะเป็นคลาสของโปรแกรม Java Servlet ทั้งหมด ดังตัวอย่าง

```
<form action=http://161.246.6.123/servlet/Basic_search method=POST>
<input type=text name=key >
<input type=submit>
<input type=reset >
</form>
```

จากตัวอย่างเป็นส่วนหนึ่งของภาษา HTML ซึ่งเป็นฟอร์มที่รองรับคีย์เวิร์ดจากผู้ใช้บริการ จากนั้นเมื่อผู้ใช้กรอกปุ่ม submit แล้วจะส่งข้อมูลในฟอร์มนี้ไปยัง <http://161.246.6.123> และจะเรียกคลาส Basic_search

ในส่วนของการติดต่อกับผู้ใช้ได้แบ่งเป็นส่วนต่าง ๆ ดังนี้

3.1.3.1 การค้นหาเว็บเพจให้ตรงกับความต้องการของผู้ใช้

เป็นการรับคีย์เวิร์ดมาจากผู้ใช้และนำคีย์เวิร์ดที่ได้นั้น ไปประมวลผลเพื่อหารายชื่อเว็บเพจให้ตรงกับที่ผู้ใช้ต้องการมากที่สุดโดยในการใช้งานนี้จะแบ่งออกเป็น 2 ประเภทคือ

3.1.3.1.1 การค้นหาแบบธรรมดา (Basic Search)

ไม่ได้สนับสนุนการใช้เครื่องหมายต่าง ๆ ในการค้นหาเพื่อให้ได้เฉพาะคีย์เวิร์ดเท่านั้น โดยคีย์เวิร์ดแต่ละคำจะถูกแยกออกจากกันด้วย spacebar ซึ่งจะมีขั้นตอนการทำงานต่าง ๆ ดังนี้

1. รับอินพุตมาจกผู้ใช้บริการ
2. แยกอินพุตออกเป็นคำ ๆ
3. นำแต่ละคำที่ได้ไปตรวจสอบว่ามีคำใดเป็นคำที่ไม่สื่อความหมายหรือไม่ถ้ามีก็ตัดคำนั้นทิ้งไป
4. นำคำที่เหลือไปค้นหาในฐานข้อมูลที่ละคำและเก็บผลลัพธ์จากแต่ละคำไว้
5. นำชื่อเว็บเพจที่ได้มาจากแต่ละคำมาประมวลผลเพื่อหาว่าเว็บเพจใดที่มีคำครบทุกคำ
6. แสดงผลลัพธ์กลับไปยังผู้ใช้บริการ โดยจะแสดงชื่อเว็บเพจและหัวข้อซึ่งจะเรียงตามลำดับเว็บเพจที่มีความสำคัญมากที่สุดลงมา

3.1.3.1.2 การค้นหาแบบใช้เครื่องหมาย (Advance Search)

เพื่อให้ผู้ใช้ได้ใช้การค้นหาให้มีประสิทธิภาพเพิ่มขึ้นจึงได้มีการเพิ่มเติมส่วนที่เป็นการค้นหาแบบใช้เครื่องหมาย ซึ่งในส่วนนี้จะสนับสนุนให้ใช้เครื่องหมายต่าง ๆ ดังนี้

- เครื่องหมาย "&" มีความหมายว่า "และ"
- เครื่องหมาย "|" มีความหมายว่า "หรือ"
- เครื่องหมาย "(" แทนวงเล็บเปิด
- เครื่องหมาย ")" แทนวงเล็บปิด

เช่น เมื่อผู้ใช้ป้อนคำว่า (a&b)c หมายความว่าต้องการเว็บเพจที่มีคำว่า "a" และ คำว่า "b" ในเว็บเพจเดียวกัน หรือต้องการเว็บเพจที่มีคำว่า "c" ก็ได้

การทำงานในส่วนของการค้นหาแบบใช้เครื่องหมายมีดังนี้

1. รับอินพุตมาจกผู้ใช้
2. ตรวจสอบ syntax ว่าถูกต้องหรือไม่ ถ้าไม่ถูกต้องก็ไม่ต้องทำขั้นตอนต่อไป
3. กระบวนการในการคำนวณหาผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แสดงผลลัพธ์กลับไปยังผู้ใช้บริการ โดยจะแสดงชื่อเว็บเพจและหัวเรื่องซึ่งจะเรียงตามลำดับเว็บเพจที่มีความสำคัญมากที่สุดลงมา

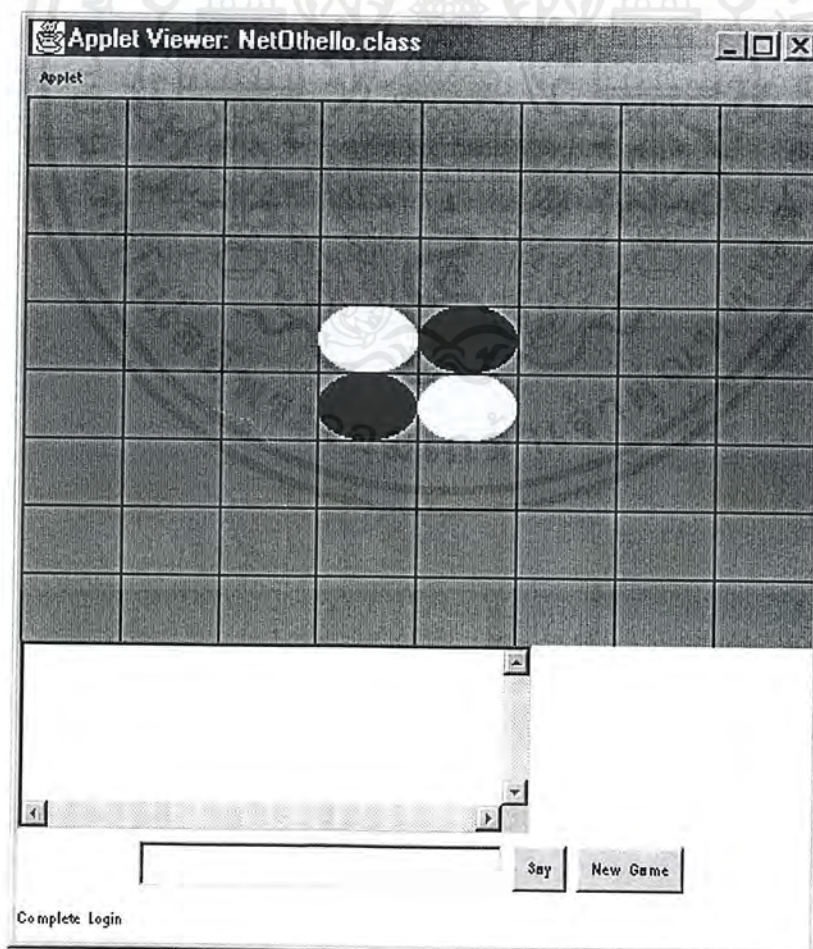
3.1.3.2 ส่วนที่จัดการในการลงทะเบียนรายชื่อเว็บไซต์ (about submit)

ในส่วนนี้จะทำหน้าที่ให้บริการกับผู้ใช้เกี่ยวกับการลงทะเบียนรายชื่อเว็บไซต์ของผู้ใช้ซึ่งประกอบไปด้วยส่วนต่าง ๆ ดังนี้

- ส่วนที่ตรวจสอบว่ามีชื่อเว็บไซต์นี้ในฐานข้อมูลรึยัง
- ส่วนที่รับชื่อเว็บไซต์ใหม่เข้ามาเก็บไว้เพื่อให้เราส่ง โปรแกรม spider ไปยังเว็บไซต์นั้น
- ส่วนที่รับชื่อเว็บไซต์ซึ่งมีอยู่แล้วในฐานข้อมูลของเราเข้ามาเพื่อให้เราส่ง โปรแกรม spider ไปยังเว็บไซต์นั้นใหม่เนื่องจากเว็บไซต์นั้นมีการเปลี่ยนแปลง

3.2 การออกแบบและการสร้างในส่วนของเกมออนไลน์

เกมแบบออนไลน์ที่จะกล่าวถึงในที่นี้คือเกม Othello ซึ่งเป็นเกมที่สามารถเล่นได้หลายคนพร้อมกันผ่านระบบอินเทอร์เน็ต ในกรณีที่ไม่เคยเล่นเกม Othello มาก่อนก็จะได้อธิบายกติกาในการเล่นไว้ด้วย ซึ่งเกม Othello นี้อาจเรียกได้อีกชื่อหนึ่งว่า Reversi รูปที่ 3-2 แสดงถึงรูปร่างหน้าตาของเกมบอร์ด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนเพื่อการศึกษาเท่านั้น ไม่เอามาเผยแพร่ไปใช้ประโยชน์ด้านการค้า
รูปที่ 3-2 แสดงถึงบอร์ดของเกม Othello โดยทั่วไปที่อยู่ในสถานะเริ่ม
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 ลักษณะของเกม

เกม Othello เป็นเกมแบบบอร์ดเกม (Board Game) ที่มีผู้เล่นหลายคน แต่จะมีจับกลุ่มผู้เล่นเข้ามาในกลุ่ม ในแต่ละกลุ่มจะมีผู้เล่นอยู่ 2 คน โดยคนจะมีคนหนึ่งเป็นผู้เล่นสีดำ และอีกคนจะผู้เล่นสีขาว โดยแต่ละคนสามารถเล่นจากที่ไหนก็ได้และสามารถเล่นอยู่บนเครื่องชนิดใดก็ได้ ไม่ว่าจะเป็นเครื่อง Sun หรือเครื่อง PC ธรรมดาก็ได้เพราะเป็นเกมที่ไม่ขึ้นกับแพลตฟอร์มใดๆ เนื่องจากเขียนขึ้นจากจาวา ซึ่งมีการควบคุมการเล่นจากเมาส์และใช้คีย์บอร์ดในการส่งข้อมูลสื่อสารกัน เกม Othello นี้ถือเป็นเกมแบบเหตุการณ์ (Event - base Game) หรือเกมที่ดำเนินเกมตามอินพุทของเหตุการณ์ที่ป้อนเข้ามา แล้วดำเนินเกมไปเรื่อยๆ จนกว่าจะจบเกม (Game Over)

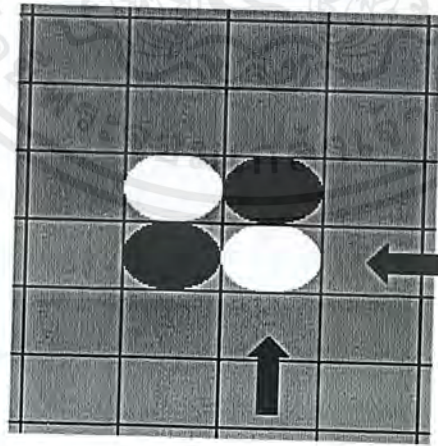
3.2.2 แนวการออกแบบเกม (Game Design)

การออกแบบเกม Othello มีขั้นตอนในการออกแบบดังนี้

3.2.2.1 การออกแบบการดำเนินเกม

เป็นการออกแบบโดยรวมของเกม กล่าวคือ จะมีการเริ่มต้นอย่างไร การดำเนินเกมเป็นอย่างไร จนกระทั่งจะจบเกมอย่างไร ซึ่งพอสรุปได้ดังนี้

เริ่มต้นจากมีหมากกระดานในบอร์ดที่ได้วางหมากเริ่มต้นไว้แล้ว โดยการเล่นผู้เล่นจะเล่นเป็นผู้เล่นสีขาวหรือเป็นผู้เล่นสีดำก็ได้ โดยการเล่นนี้ต้องใช้ผู้เล่น 2 คนในการเล่น และในการเล่นเกมจะผลัดกันเล่นทีละคน (Turn - base Game) โดยในแต่ละครั้งจะต้องวางหมากในตำแหน่งที่สามารถวางได้ ซึ่งตำแหน่งที่จะวางหมากได้นั้นจะต้องมีหมากติดตรงเคียงกันกับผู้เล่นวางอยู่ระหว่างหมากของผู้เล่นที่จะทำการวาง โดยสามารถวางได้ในแนวตั้ง แนวนอน หรือแนวทะแยงก็ได้ รูปที่ 3-3 แสดงตัวอย่างหมากของผู้เล่นที่สามารถวางได้



รูปที่ 3-3 ลูกศรแสดงถึงตำแหน่งที่สามารถจะวางได้เมื่อเราเป็นผู้เล่นสีดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการเล่นจะผลัดเปลี่ยนกันเล่น ไปเรื่อยๆ จนกว่าจะจบเกม ซึ่งการจบเกมจะเกิดขึ้นเมื่อผู้เล่นฝ่ายใดฝ่ายหนึ่งไม่มีหมากของตนเหลืออยู่บนบอร์ดเลข หรือเกิดจากหมากบนบอร์ดถูกวางจนเต็ม (ไม่สามารถวางหมากได้อีก) ผู้ที่ชนะของเกมนี้คือผู้ที่มีจำนวนหมากมากกว่านั่นเอง

3.2.2.2 การออกแบบการรับอินพุตจากผู้เล่น

การรับอินพุตจากผู้เล่นของเกม Othello นี้จะรับอินพุตจากเมาส์และคีย์บอร์ด โดยส่วนที่รับจากเมาส์ จะใช้ในการวางหมากจากผู้เล่น ซึ่งการวางหมากนี้จะวางได้เมื่อดำแหน่งของหมากที่จะวางอยู่ในตำแหน่งตามที่ได้อธิบายไปแล้ว และในส่วนที่รับอินพุตจากคีย์บอร์ดจะใช้ในการติดต่อสื่อสารกันระหว่างผู้เล่น

3.2.2.3 การออกแบบหน้าจอของเกม

หน้าจอของเกมจะประกอบไปด้วย 3 ส่วนหลัก คือ

1. ส่วนที่ใช้แสดงบอร์ด
2. ส่วนที่ใช้แสดงข้อความที่ผู้เล่นสื่อสารกัน
3. ส่วนที่รับข้อความจากผู้เล่น และปุ่มควบคุม

3.2.2.4 หลักในการเขียนโปรแกรม

ตัวเกม Othello นี้แบ่งออกเป็น 2 ส่วนคือ ส่วนที่ทำหน้าที่เป็นเซิร์ฟเวอร์กับส่วนที่เป็นไคลเอนท์ Client - side ทำหน้าที่ในการรับอินพุตมาจากผู้เล่นและแสดงหน้าจอติดต่อกับผู้เล่น สำหรับการออกแบบในส่วนไคลเอนท์ ได้ทำการออกแบบโดยอาศัยหลักของ Object - Oriented Programming ออกแบบไว้ 3 คลาสคือ

1. คลาส GamePiece

คลาสนี้ทำหน้าที่ในการเก็บสถานะของหมากและเก็บตำแหน่งของหมากบนบอร์ด สามารถที่จะเปลี่ยนสถานะของหมาก (flip) และสามารถวาดตัวเองบนหน้าจอได้

2. คลาส GameBoard

คลาสนี้ทำหน้าที่เก็บหมากที่วางอยู่บนบอร์ดโดยเก็บเป็นอาร์เรย์แบบ 2 มิติ สามารถวาดตัวบอร์ดและหมากที่วางอยู่ทั้งหมดลงบนหน้าจอ สามารถนับจำนวนหมากของผู้เล่นบนบอร์ดและสามารถนับจำนวนที่วางที่ยังไม่ได้วางหมาก สามารถที่จะรับการวางหมากจากผู้เล่นทั้งยังสามารถรีเซ็ตสถานะหมากที่วางอยู่ในตำแหน่งของหน้าจอแล้วเทียบกับบอร์ด

3. คลาส NetOthello

คลาสนี้ทำหน้าที่ในการจัดการกับกติกาและฟังก์ชันการทำงานของเกมทั้งหมด ไม่ว่าจะเป็นการรับอินพุตจากผู้เล่นทั้งจากเมาส์และจากคีย์บอร์ด การจัดการแสดงผลหน้าจอกับผู้เล่น การรับชื่อของผู้เล่น และการจัดการเครือข่ายทั้งหมด

Server - side

เกม Othello ที่ได้ออกแบบขึ้นมาขึ้นอยู่กับการทำงานในฝั่งไคลเอนต์ (Client - Driven) หน้าที่หลักๆ ที่เซิร์ฟเวอร์ทำคือทำการรับและกระจายเมสเสจ (message) คืนให้กลับไคลเอนต์อีกทีเพื่อทำการซิงโครไนซ์ (synchronize) ให้กับไคลเอนต์ โดยในส่วนการออกแบบด้วย 4 คลาสคือ

1. คลาส GameServer

เป็นคลาสที่เป็นตัวรันหลักของเซิร์ฟเวอร์ ซึ่งทำการรันจากคอมมานด์ไลน์ (Command - line) ซึ่งรับอาร์กิวเมนต์เป็นหมายเลขพอร์ต แล้วส่งอาร์กิวเมนต์ผ่านให้กับคลาส GameServerThread

2. คลาส GameServerThread

เป็นคลาสที่ทำหน้าที่จัดการทางด้านเครือข่าย ซึ่งเป็นตัวที่จัดการคลาส GameGroup กล่าวคือเป็นคลาสที่ทำหน้าที่รับการติดต่อมาจากไคลเอนต์และจัดการจัดกลุ่มให้กับผู้เล่นคือ ผู้เล่น 2 คนจะจัดเข้ามาเป็นกลุ่มและยังจัดการในการยกเลิกการติดต่อจากไคลเอนต์ไม่ว่าจะโดยตั้งใจหรือไม่ตั้งใจก็ตาม

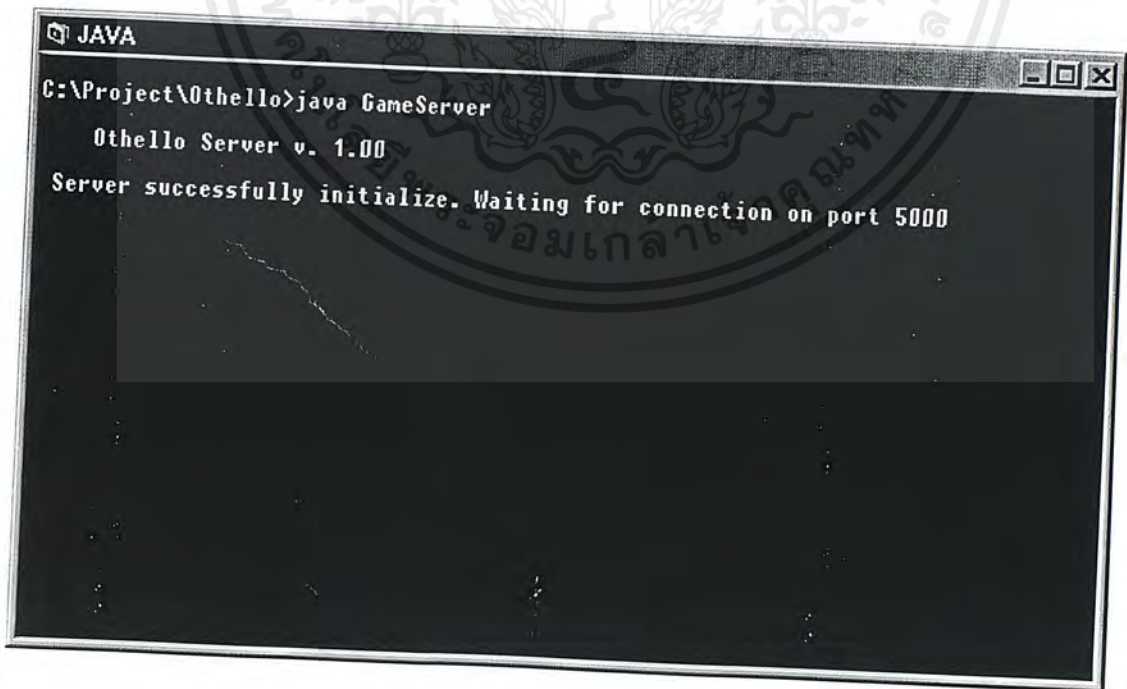
3. คลาส GameGroup

โดยหลักๆ แล้วคลาสนี้รับไคลเอนต์เข้ามาและยังทำหน้าที่รับ/ส่งเมสเสจคืนให้กลับไคลเอนต์ภายในกลุ่มทั้งยังตรวจสอบว่าผู้เล่นเข้ามาครบกลุ่มหรือยัง ถ้าครบกลุ่มแล้วก็ทำการเริ่มเกมขึ้น

4. คลาส GameClientThread

เป็นคลาสที่ทำหน้าที่ในการรับพารามิเตอร์มาจาก GameGroup ทำหน้าที่เปิด Stream รอรับเมสเสจที่ไคลเอนต์ที่จะติดต่อเข้ามา

3.2.2.5 ตัวอย่างหน้าจอของเกม

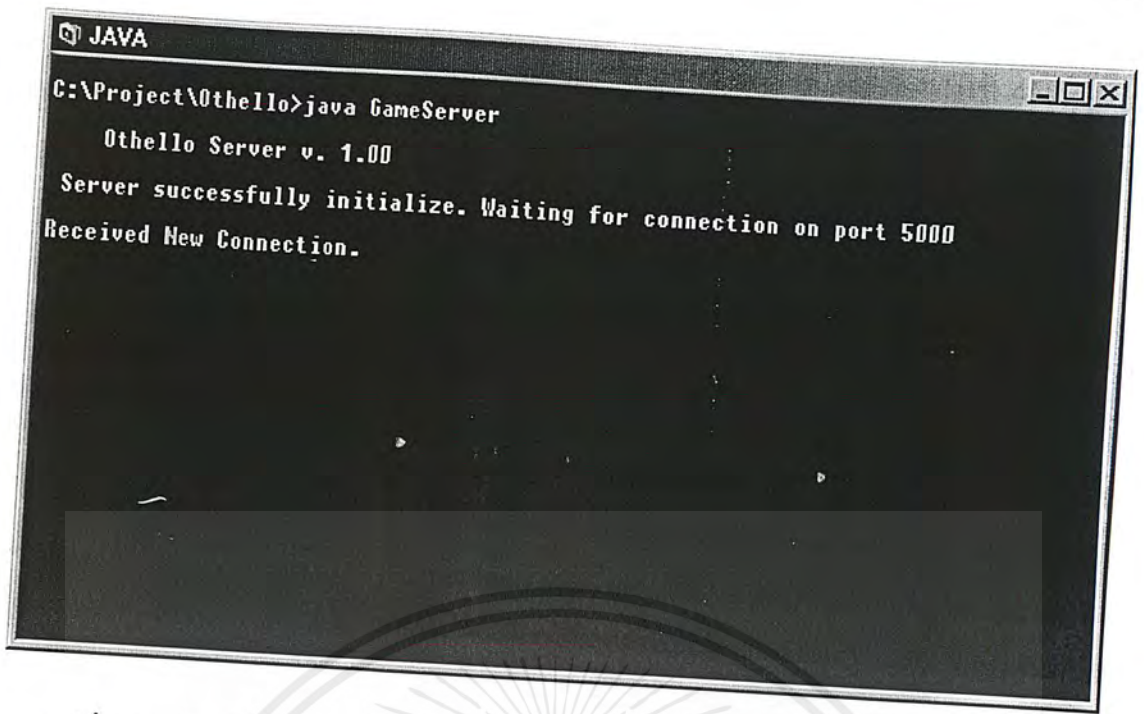


```

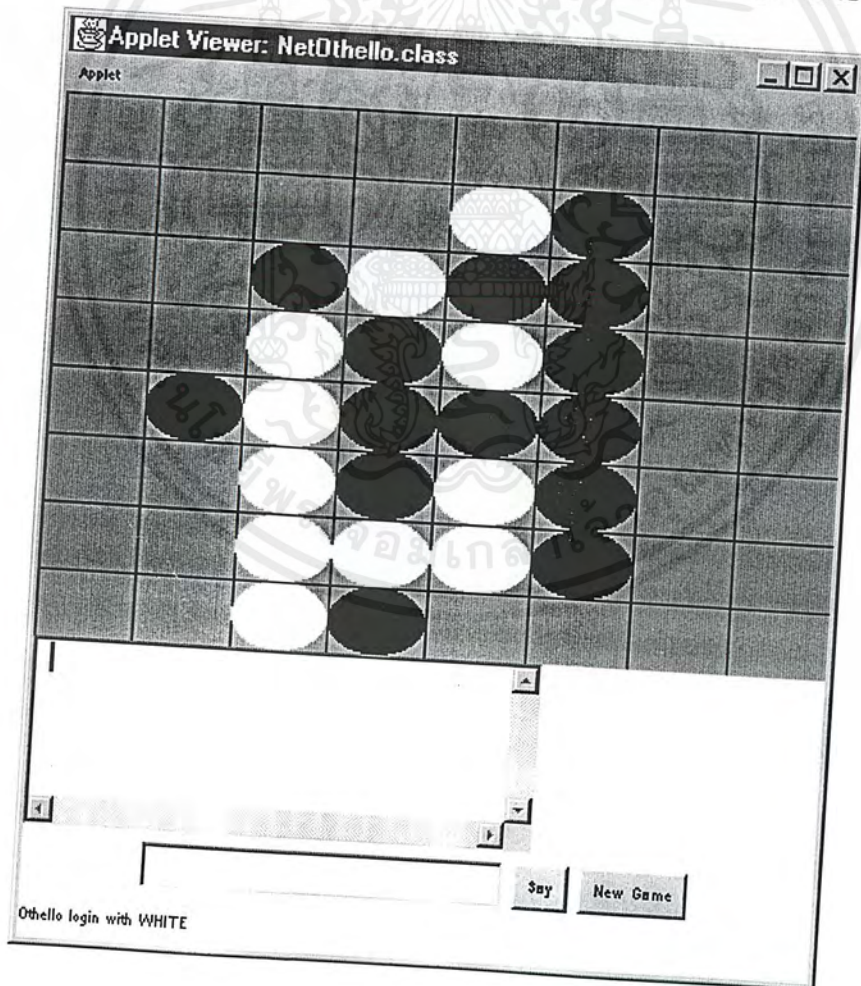
C:\Project\Othello>java GameServer
Othello Server v. 1.00
Server successfully initialize. Waiting for connection on port 5000
  
```

รูปที่ 3-4 แสดงหน้าจอการทำงานของเซิร์ฟเวอร์ที่กำลังรอการติดต่อจากไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-5 แสดงหน้าจอการทำงานของเซิร์ฟเวอร์หลังจากทำการติดต่อจากไคลเอนท์เรียบร้อยแล้ว



รูปที่ 3-6 แสดงผลการทำงานที่ฝั่งไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดสอบและผลการทดสอบ

4.1 การทดสอบและผลการทดสอบในส่วนของเซิร์ฟเวอร์

เนื่องจากในส่วนของเซิร์ฟเวอร์ได้แบ่งออกเป็น 3 ส่วนจึงมีขั้นตอนการทดสอบในการทำงานในแต่ละส่วนดังนี้

4.1.1 ส่วนที่ทำการจัดการด้านฐานข้อมูล

ส่วนนี้ได้มีการทดสอบการติดต่อระหว่าง JDBC กับ DBMS เพื่อทดสอบว่าฐานข้อมูลที่สร้างสามารถถูก query ข้อมูลในฐานข้อมูลจากภายนอกได้หรือไม่ทั้งนี้ก็เพื่อให้สามารถดำเนินการในขั้นตอนต่อไปได้ โดยได้ใช้ไดรเวอร์ SprintaJDBC_2.zip ซึ่งเป็นไดรเวอร์ชนิดที่ 4 ของเว็บไซต์ www.inet-software.com เพื่อใช้ในการติดต่อระหว่าง JDBC กับ DBMS

โปรแกรมที่ทดสอบเป็น Java Application ชื่อ CreateCoffees.java โดยโปรแกรมนี้อาจทำการติดต่อกับฐานข้อมูล project ซึ่งจะทำการสร้างตารางชื่อ COFFEES ประกอบด้วยคอลัมน์ต่าง ๆ ดังนี้

ชื่อคอลัมน์	ชนิดข้อมูล
COF_NAME	VARCHAR
SUB_ID	INT
PRICE	FLOAT
SALES	INT
TOTAL	INT

จากนั้นนำไปทดสอบโดยใช้คำสั่งดังนี้

```
javac CreateCoffees.java
```

และ `java CreateCoffees ทั้ง 2`

```

Microsoft Command Prompt
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\>pre4
C:\>path=c:\jdk1.2.2\bin
C:\>set classpath=.;c:\SprintaJDBC_2.zip;c:\jsdk2.0\lib\jsdk.jar;
C:\>cd class
C:\class>javac CreateCoffees.java
C:\class>java CreateCoffees
C:\class>

```

รูปที่ 4-1 แสดงการรันคำสั่ง

จากนั้นไปดูในฐานข้อมูลว่ามีตาราง COFFEES หรือไม่ซึ่งผลลัพธ์แสดงดังรูปที่ 4-2 จากรูปจะเห็นว่ามีการสร้างตาราง COFFEES ในฐานข้อมูลแล้ว

Column Name	Datatype	Length	Precision	Scale	Allow Nulls
COF_NAME	varchar	32	0	0	<input type="checkbox"/>
SUP_ID	int	4	10	0	<input checked="" type="checkbox"/>
PRICE	float	8	53	0	<input checked="" type="checkbox"/>
SALES	int	4	10	0	<input checked="" type="checkbox"/>
TOTAL	int	4	10	0	<input checked="" type="checkbox"/>

รูปที่ 4-2 แสดงผลจากการรันคำสั่งในฐานข้อมูล

ขั้นต่อไปจะทดสอบการส่งข้อมูลเข้าไปในตาราง COFFEES และ Query ข้อมูลขึ้นมาจากราย โดยต้องการคอลัมน์ COF_NAME และ PRICE ของทุกแถวที่มีในตารางซึ่งผลลัพธ์ควรจะแสดงดังนี้

COF_NAME	PRICE
Colombian	8.99
French_Roas	8.99
Espresso	9.99
Colombian_Decaf	8.99
French_Roast_Decaf	9.99

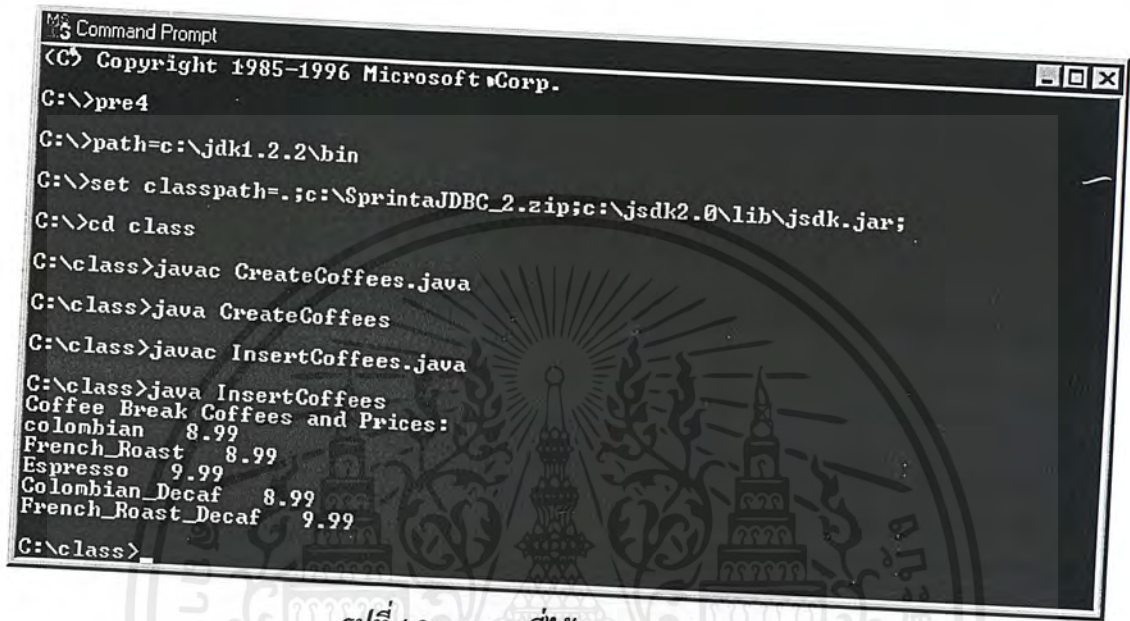
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโปรแกรมนี้ชื่อว่า InsertCoffees.java จากนั้นก็รันโปรแกรมด้วยคำสั่ง

```
javac InsertCoffees.java
```

และ `java InsertCoffees`

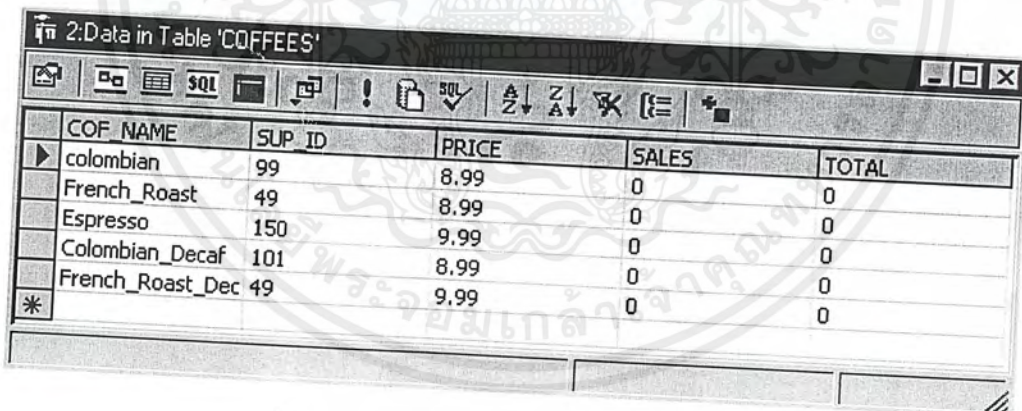
ผลที่ได้จากการรันโปรแกรมแสดงดังรูปที่ 4-3 ซึ่งจะเห็นว่าผลลัพธ์ออกมาถูกต้องตรงตามที่เรากาดไว้ แสดงว่าส่วนของการจัดเก็บข้อมูลนั้นพร้อมที่จะใช้งานได้แล้ว



```

C:\>pre4
C:\>path=c:\jdk1.2.2\bin
C:\>set classpath=.;c:\SprintaJDBC_2.zip;c:\jsdk2.0\lib\jsdk.jar;
C:\>cd class
C:\class>javac CreateCoffees.java
C:\class>java CreateCoffees
C:\class>javac InsertCoffees.java
C:\class>java InsertCoffees
Coffee Break Coffees and Prices:
colombian      8.99
French_Roast   8.99
Espresso       9.99
Colombian_Decaf 8.99
French_Roast_Decaf 9.99
C:\class>
  
```

รูปที่ 4-3 แสดงผลที่ได้จากการรันโปรแกรม



COF_NAME	SUP_ID	PRICE	SALES	TOTAL
colombian	99	8.99	0	0
French_Roast	49	8.99	0	0
Espresso	150	9.99	0	0
Colombian_Decaf	101	8.99	0	0
French_Roast_Dec	49	9.99	0	0

รูปที่ 4-4 แสดงผลลัพธ์จากการรันโปรแกรมในฐานข้อมูล

4.1.2 ส่วนที่ทำการค้นหาข้อมูลจากเว็บเพจมาเก็บในฐานข้อมูล

เนื่องจากส่วนนี้ได้แบ่งออกเป็น 3 ส่วนด้วยกันซึ่งจะทำการทดสอบในแต่ละส่วนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.1 ส่วนที่ทำการดึงข้อมูลมาจากเว็บเพจ

ในส่วนนี้จะเป็นการทดสอบในการติดต่อกับเว็บเพจเพื่อร้องขอข้อมูลจากเว็บเพจ โดยใช้เมธอด `openConnection` ของคลาส `URL` เพื่อขอข้อมูลแล้วนำมาเก็บไว้ในไฟล์ชั่วคราวซึ่งวิธีการต่าง ๆ แสดงดังไฟล์ชาร์ตในรูปที่ 4-4 หน้าที่ 3-3 ภาคผนวก ค

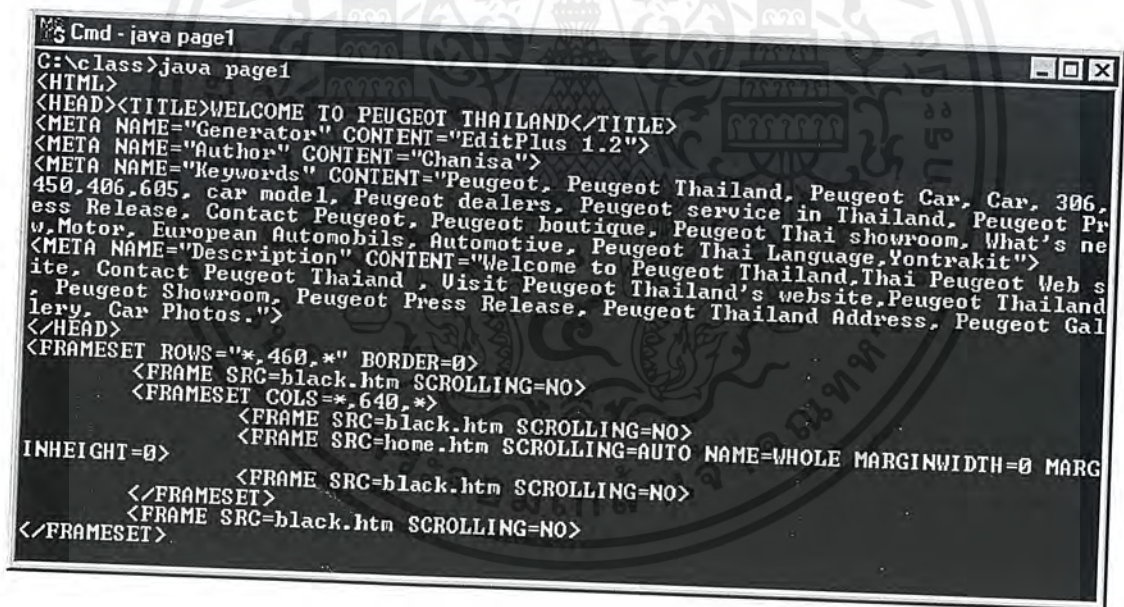
โปรแกรมที่ใช้ทดสอบชื่อ `page1.java` ซึ่งเป็นโปรแกรม Java application ซึ่งจะทำการร้องขอข้อมูลมาจากเว็บเพจที่ต้องการ ซึ่งในที่นี้กำหนดให้เป็น `www.peugeot-th.com` ผลที่คาดว่าจะได้รับนั้นคือแสดงโค้ดข้อมูลของเพจนี้ออกมาทางจอภาพและบันทึกข้อมูลทั้งหมดที่ได้ลงในไฟล์ในที่นี้กำหนดให้เป็น `tempurl`

จากนั้นรันโปรแกรมด้วยคำสั่ง

```
javac page1.java
```

```
และ java page1
```

ผลที่ได้ออกมาที่แสดงดังรูปที่ 4-5 และรูปที่ 4-6 (เนื่องจากว่าแสดงในหน้าจอเดียวไม่พอ) ซึ่งจากรูปจะเห็นว่าได้แสดงโค้ดโปรแกรมในภาษา HTML ออกมา จากนั้นนำไปเปรียบเทียบกับไฟล์ `tempurl` และ source code จริง ๆ ของเว็บเพจ (อยู่ในภาคผนวก) นี้ว่าตรงกันหรือไม่ ซึ่งปรากฏว่าตรงกัน แสดงว่าสามารถใช้เมธอด `openConnection` ของคลาส `URL` ในการร้องขอข้อมูลจากเว็บเพจได้



```

C:\class>java page1
<HTML>
<HEAD><TITLE>WELCOME TO PEUGEOT THAILAND</TITLE>
<META NAME="Generator" CONTENT="EditPlus 1.2">
<META NAME="Author" CONTENT="Chanisa">
<META NAME="Keywords" CONTENT="Peugeot, Peugeot Thailand, Peugeot Car, Car, 306, 450, 406, 605, car model, Peugeot dealers, Peugeot service in Thailand, Peugeot Press Release, Contact Peugeot, Peugeot boutique, Peugeot Thai showroom, What's new, Motor, European Automobils, Automotive, Peugeot Thai Language, Yontrakit">
<META NAME="Description" CONTENT="Welcome to Peugeot Thailand, Thai Peugeot Website, Contact Peugeot Thailand, Visit Peugeot Thailand's website, Peugeot Thailand Peugeot Showroom, Peugeot Press Release, Peugeot Thailand Address, Peugeot Gallery, Car Photos.">
</HEAD>
<FRAMESET ROWS="*, 460, *" BORDER=0>
  <FRAME SRC=black.htm SCROLLING=NO>
  <FRAMESET COLS="*, 640, *>
    <FRAME SRC=black.htm SCROLLING=NO>
    <FRAME SRC=home.htm SCROLLING=AUTO NAME=WHOLE MARGINWIDTH=0 MARGINHEIGHT=0>
  </FRAMESET>
  <FRAME SRC=black.htm SCROLLING=NO>
</FRAMESET>

```

รูปที่ 4-5 ผลจากการรันโปรแกรมการดึงข้อมูลในส่วนบน

```

MS-DOS Command Prompt
<FRAME SRC=black.htm SCROLLING=NO>
<FRAMESET COLS=*,640,*>
  <FRAME SRC=black.htm SCROLLING=NO>
  <FRAME SRC=home.htm SCROLLING=AUTO NAME=WHOLE MARGINWIDTH=0 MARG
INHEIGHT=0>
  <FRAME SRC=black.htm SCROLLING=NO>
</FRAMESET>
<FRAMESET>
  <FRAME SRC=black.htm SCROLLING=NO>
</FRAMESET>
<!-- Start of NedStat Basic code -->
<a href="http://usa.nedstatbasic.net/cgi-bin/viewstat?name=lion">ing
src="http://usa.nedstatbasic.net/cgi-bin/nedstat.gif?name=lion"
border=0 alt="" nosave width=22 height=22</a>
<!-- End of NedStat Basic code -->
</HTML>
file: /
host: www.peugeot-th.com
ref: null
protocol: http
page: http://www.peugeot-th.com/
Retrieve Successful!
bytes read from file: 0
G:\class>

```

รูปที่ 4-6 ผลจากการรันโปรแกรมการดึงข้อมูลในส่วนล่าง

4.1.2.2 ส่วนที่ทำการวิเคราะห์ข้อมูลในไฟล์ชั่วคราว

ในส่วนนี้จะทำการวิเคราะห์ข้อมูลในไฟล์ชั่วคราวเพื่อหาคีย์เวิร์ด ค่าความสำคัญของแต่ละคำ รวมทั้งหาชื่อเว็บเพจและชื่อเว็บไซต์ โดยในที่นี้ใช้โปรแกรมที่ทำเสร็จแล้วมาทำการทดสอบคือโปรแกรม Spider.java โดยเมธอดที่ทำหน้าที่ในการหาคีย์เวิร์ดคือเมธอด findkeyword() และเมธอดที่ทำหน้าที่ในการหาชื่อเว็บเพจและชื่อเว็บไซต์คือเมธอด findlink() ผลที่คาดว่าจะได้รับคือแสดงคีย์เวิร์ดที่พบรวมทั้งค่าความสำคัญของแต่ละคำและชื่อเว็บไซต์และเว็บเพจออกทางหน้าจอ

เริ่มทดสอบโดยป้อนชื่อเว็บไซต์เริ่มต้นเข้าไปในไฟล์ fileurl.txt ซึ่งไฟล์นี้จะเก็บชื่อเว็บไซต์เริ่มต้นทั้งหมดที่โปรแกรม spider จะทำการค้นหา จากนั้นเรียกคำสั่ง

```
java Spider
```

ผลที่ออกมาแสดงดังรูปที่ 4-7 และรูปที่ 4-8 (รูปที่ 4-7 และรูปที่ 4-8 นั้นต่อกัน) จากนั้นนำไปเปรียบเทียบกับ source code ของเว็บเพจ www.peugeot-th.com ซึ่งจะเห็นว่าผลที่ออกมานั้นถูกต้อง ตัวอย่างคีย์เวิร์ดที่ได้มาจากรูปที่ 4-7 มีดังนี้

- คีย์เวิร์ดคำแรกที่พบคือ welcome มีค่าความสำคัญเท่ากับ 3
- คีย์เวิร์ดคำที่ 2 คือ to มีค่าความสำคัญเท่ากับ 3
- คีย์เวิร์ดคำที่ 3 คือ peugeot มีค่าความสำคัญเท่ากับ 3
- คีย์เวิร์ดคำสุดท้ายคือ photos มีค่าความสำคัญเท่ากับ 1
- และคีย์เวิร์ดอื่น ๆ แสดงดังรูป

```

MS-DOS Cmd - java Spider
C:\>cd class
C:\class>java Spider
Set for delete www.peugeot-th.com
SELECT URL FROM DETAIL WHERE BASEURL = 'www.peugeot-th.com'
executing
passexecute
No www.peugeot-th.com in database

Find keyword and link in page number :0
http://www.peugeot-th.com/
0:WELCOME(3) 1:TO(3) 2:PEUGEOT(3) 3:THAILAND(3) 4:Peugeot(2) 5:Peugeot(2) 6:Thailand(2) 7:Peugeot(2) 8:Car(2) 9:Car(2) 10:306(2) 11:450(2) 12:406(2) 13:605(2) 14:car(2) 15:model(2) 16:Peugeot(2) 17:dealers(2) 18:Peugeot(2) 19:service(2) 20:in(2) 21:Thailand(2) 22:Peugeot(2) 23:Press(2) 24:Release(2) 25:Contact(2) 26:Peugeot(2) 27:Peugeot(2) 28:boutique(2) 29:Peugeot(2) 30:Thai(2) 31:showroom(2) 32:What(2) 33:s(2) 34:new(2) 35:Motor(2) 36:European(2) 37:Automobiles(2) 38:Automotive(2) 39:Peugeot(2) 40:Thai(2) 41:Language(2) 42:Yontrakit(2) 43:Welcome(1) 44:to(1) 45:Peugeot(1) 46:Thailand(1) 47:Thai(1) 48:Peugeot(1) 49:Web(1) 50:site(1) 51:Contact(1) 52:Peugeot(1) 53:Thailand(1) 54:Visit(1) 55:Peugeot(1) 56:Thailand(1) 57:s(1) 58:website(1) 59:Peugeot(1) 60:Thailand(1) 61:Peugeot(1) 62:Showroom(1) 63:Peugeot(1) 64:Press(1) 65:Release(1) 66:Peugeot(1) 67:Thailand(1) 68:Address(1) 69:Peugeot(1) 70:Gallery(1) 71:Car(1) 72:Photos(1)

```

รูปที่ 4-7 ผลจากการวิเคราะห์ข้อมูลในส่วนของคีย์เวิร์ดที่ได้

```

MS-DOS Cmd - java Spider
1 : black.htm
2 : home.htm
3 : black.htm
4 : black.htm
5 : http://usa.nedstatbasic.net/cgi-bin/viewstat?name=lion

Find keyword and link in page number :1
http://www.peugeot-th.com/black.htm

Find keyword and link in page number :2
http://www.peugeot-th.com/home.htm
0:Welcome(3) 1:to(3) 2:Peugeot(3) 3:Peugeot(2) 4:Peugeot(2) 5:Thailand(2) 6:Peugeot(2) 7:Car(2) 8:Car(2) 9:306(2) 10:450(2) 11:406(2) 12:605(2) 13:car(2) 14:model(2) 15:Peugeot(2) 16:dealers(2) 17:Peugeot(2) 18:service(2) 19:in(2) 20:Thailand(2) 21:Peugeot(2) 22:Press(2) 23:Release(2) 24:Contact(2) 25:Peugeot(2) 26:Peugeot(2) 27:boutique(2) 28:Peugeot(2) 29:Thai(2) 30:showroom(2) 31:What(2) 32:s(2) 33:new(2) 34:European(2) 35:Automobiles(2) 36:Automotive(2) 37:Peugeot(2) 38:Thai(2) 39:Language(2) 40:Welcome(1) 41:to(1) 42:Automotive(2) 43:Thailand(1) 44:Thai(1) 45:Peugeot(1) 46:Web(1) 47:site(1) 48:Contact(1) 49:Peugeot(1) 50:Thailand(1) 51:Visit(1) 52:Peugeot(1) 53:Thailand(1) 54:s(1) 55:website(1) 56:Peugeot(1) 57:Thailand(1) 58:Peugeot(1) 59:Showroom(1) 60:Peugeot(1) 61:Press(1) 62:Release(1) 63:Peugeot(1) 64:Thailand(1) 65:Address(1) 66:Peugeot(1) 67:Gallery(1) 68:Car(1) 69:Photos(1)

```

รูปที่ 4-8 ผลจากการวิเคราะห์ข้อมูลในส่วนของชื่อลิงค์ที่ได้

ชื่อลิงค์ที่ได้มาจากรูปที่ 4-8 มีดังนี้

1. black.htm
2. home.htm
3. black.htm
4. black.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. <http://usa.nedstatbasic.net/cgi-bin/viewstat?name=lion>

4.1.2.3 ส่วนที่ทำหน้าที่ในการจัดเก็บข้อมูลลงฐานข้อมูล

จะนำคีย์เวิร์ดไปจัดเก็บลงฐานข้อมูลซึ่งจะต้องตรวจสอบก่อนว่าค่านั้นเป็นคำที่ไม่สื่อความหมายหรือไม่ถ้าเป็นก็ไม่ต้องจัดเก็บลงฐานข้อมูล จากตัวอย่างคำที่ 2 นั้นคือ to ซึ่งเป็นคำที่ไม่สื่อความหมายในทางทฤษฎีเซิร์ชเอนจินดังนั้นจึงไม่เก็บคำนี้ลงในฐานข้อมูล

รูปที่ 4-9 แสดงข้อมูลส่วนหนึ่งในตาราง search ที่ได้จากการวิเคราะห์เว็บเพจ www.peugeot-th.com/

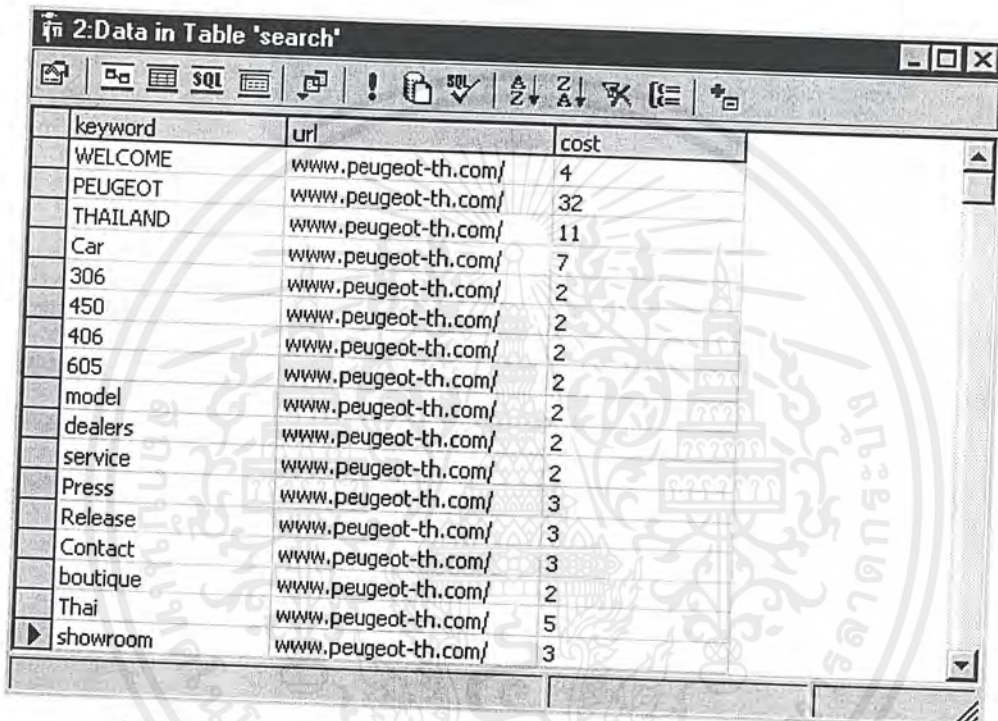


Figure 4-9: Data in Table 'search'

keyword	url	cost
WELCOME	www.peugeot-th.com/	4
PEUGEOT	www.peugeot-th.com/	32
THAILAND	www.peugeot-th.com/	11
Car	www.peugeot-th.com/	7
306	www.peugeot-th.com/	2
450	www.peugeot-th.com/	2
406	www.peugeot-th.com/	2
605	www.peugeot-th.com/	2
model	www.peugeot-th.com/	2
dealers	www.peugeot-th.com/	2
service	www.peugeot-th.com/	2
Press	www.peugeot-th.com/	3
Release	www.peugeot-th.com/	3
Contact	www.peugeot-th.com/	3
boutique	www.peugeot-th.com/	2
Thai	www.peugeot-th.com/	5
showroom	www.peugeot-th.com/	3

รูปที่ 4-9 แสดงข้อมูลส่วนหนึ่งในตาราง search ที่ได้จากการวิเคราะห์เว็บเพจ

นอกจากนี้ยังมีการเก็บข้อมูลลงตาราง detail ดังแสดงในรูปที่ 4-10



Figure 4-10: Data in Table 'detail'

url	title	baseurl
www.peugeot-th.com/	WELCOME TO PEUGEOT THAILAND	www.peugeot-th.com

รูปที่ 4-10 แสดงข้อมูลในตาราง detail ที่ได้จากการวิเคราะห์เว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

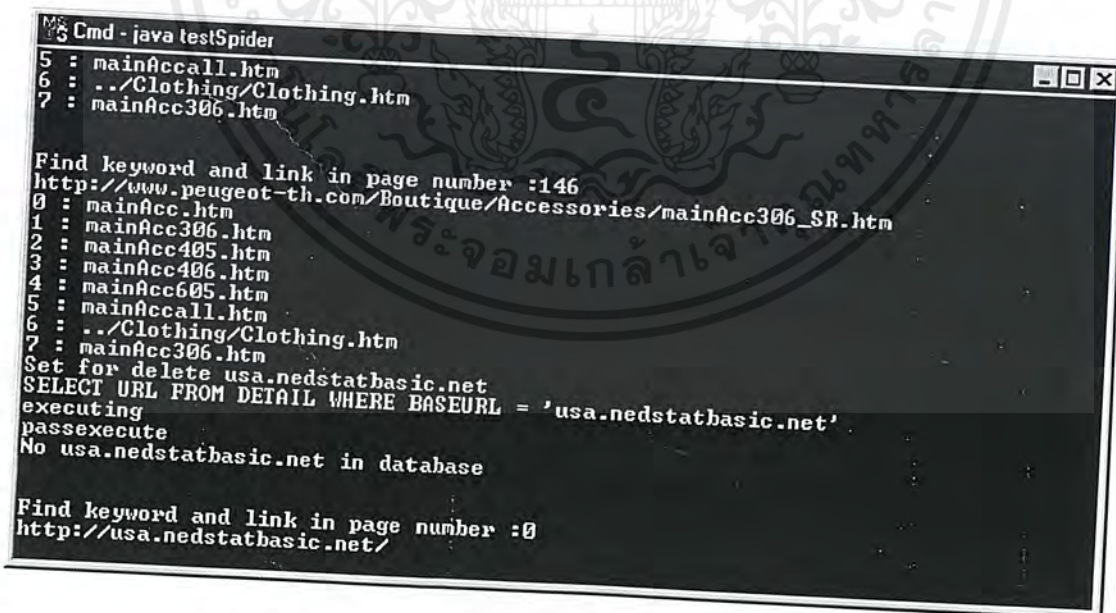
ผลจากการทดสอบจะเห็นว่าได้ผลลัพธ์ออกมาถูกต้อง

ทดสอบการท่องไปตามเว็บเพจของ Spider

จากการทดสอบการวิเคราะห์ข้อมูลในเว็บไซท์ www.peugeot-th.com นั้นปรากฏว่าสามารถวิเคราะห์ข้อมูลในเว็บเพจแรก คือ www.peugeot-th.com/ ได้อย่างถูกต้อง โดยเว็บเพจแรกนั้นได้ชื่อลิงค์มาทั้งหมด 5 ชื่อจากนั้นจะนำแต่ละชื่อไปตรวจสอบซึ่งปรากฏว่าจะเหลือชื่อลิงค์ที่ใช้ได้อยู่ 3 ชื่อคือ black.htm และ home.htm ซึ่งทั้งสองชื่อนี้เป็นเว็บเพจของเว็บไซท์ที่กำลังพิจารณาตั้งนั้น www.peugeot-th.com/black.htm และ www.peugeot-th.com/home.htm จะถูกเพิ่มเข้าไปในรายการชื่อเว็บเพจ ส่วนอีกชื่อที่พบคือ usa.nedstatbasic.net ซึ่งจะเป็นชื่อเว็บไซท์ตั้งนั้นจึงเพิ่มเข้าไปในรายการชื่อเว็บไซท์

เมื่อวิเคราะห์ข้อมูลในเว็บเพจ www.peugeot-th.com/ เสร็จแล้วต่อไปต้องไปดึงข้อมูลจากเว็บเพจถัดไปที่มีรายชื่ออยู่ในรายการชื่อเว็บเพจ ซึ่งก็คือ www.peugeot-th.com/black.htm และทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะครบทุกเว็บเพจ และเมื่อครบทุกเว็บเพจแล้วก็จะไปตั้งค้นใหม่ในเว็บไซท์ดังไปซึ่งก็คือ usa.nedstatbasic.net

จากรูปที่ 4-7 และ รูปที่ 4-8 จะแสดงการท่องเว็บไซท์ของ Spider ในตอนเริ่มต้นของเว็บไซท์ www.peugeot-th.com จากนั้นโปรแกรมก็จะดำเนินต่อไปเรื่อย ๆ ซึ่งในรูปที่ 4-11 แสดงเว็บเพจสุดท้ายของเว็บไซท์นี้คือ www.peugeot-th.com/Boutique/Accessories/mainAcc306_SR.htm ซึ่งเป็นเว็บเพจลำดับที่ 146 ของเว็บไซท์นี้ และในรูปนี้ได้เปลี่ยนเป็นเว็บไซท์ใหม่ด้วยซึ่งก็คือ usa.nedstatbasic.net ซึ่งเริ่มที่เว็บเพจแรกคือ usa.nedstatbasic.net/ และโปรแกรมจะดำเนินต่อไปเรื่อย ๆ จนกว่าจะถึงขอบเขตที่กำหนดไว้



รูปที่ 4-11 แสดงผลการวิเคราะห์ข้อมูลให้เว็บเพจสุดท้ายของเว็บไซท์แรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากการทดสอบจะพบว่าโปรแกรมสามารถทำงานได้ตามถูกต้องตามที่ต้องการทุกประการดังนั้นจึงนำโปรแกรมที่ได้ทิ้งไปตามเว็บเพจต่าง ๆ โดยกำหนด URL เริ่มต้นและเริ่มต้นโปรแกรมโดยได้จำกัดจำนวนเว็บไซต์ไว้เนื่องจากไม่มีพื้นที่เพียงพอในการเก็บข้อมูล

4.1.3 ส่วนของการติดต่อระหว่างผู้ให้บริการกับฐานข้อมูล

ได้มีการทดสอบในส่วนต่าง ๆ ดังต่อไปนี้

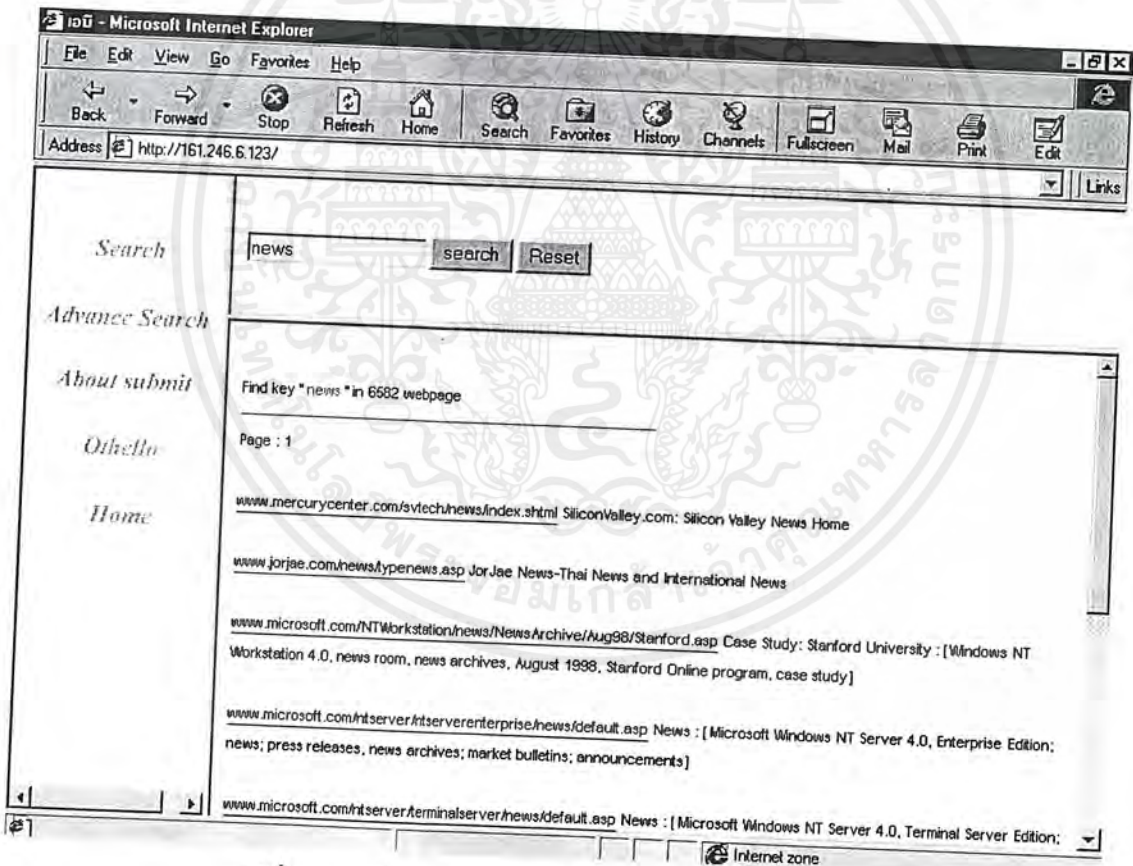
4.1.3.1 การค้นหาเว็บเพจให้ตรงกับความต้องการของผู้ใช้

ในการทดสอบนั้นคอนแรกนั้นได้ทำการทดสอบตั้งแต่ช่วงที่ยังไม่ได้ทำการดึงข้อมูล โดยจะทำการทำการป้อนข้อมูลเข้าไปในฐานข้อมูลเอง ซึ่งได้ทำการทดสอบมาเรื่อยๆจนกระทั่งโปรแกรมสามารถรองรับการใช้งานของผู้ใช้ได้แล้วทั้งการค้นหาแบบธรรมดาและการค้นหาโดยใช้เครื่องหมาย

ต่อไปจะเป็นการทดสอบหลังจากที่ดึงข้อมูลมาใส่ในฐานข้อมูลแล้ว

4.1.3.1.1 การค้นหาแบบธรรมดา (basic search)

สามารถใส่คำที่คำก็ได้ซึ่งในการทดสอบจะป้อนคำว่า news เข้าไปผลที่ได้แสดงดังรูปที่ 4-12

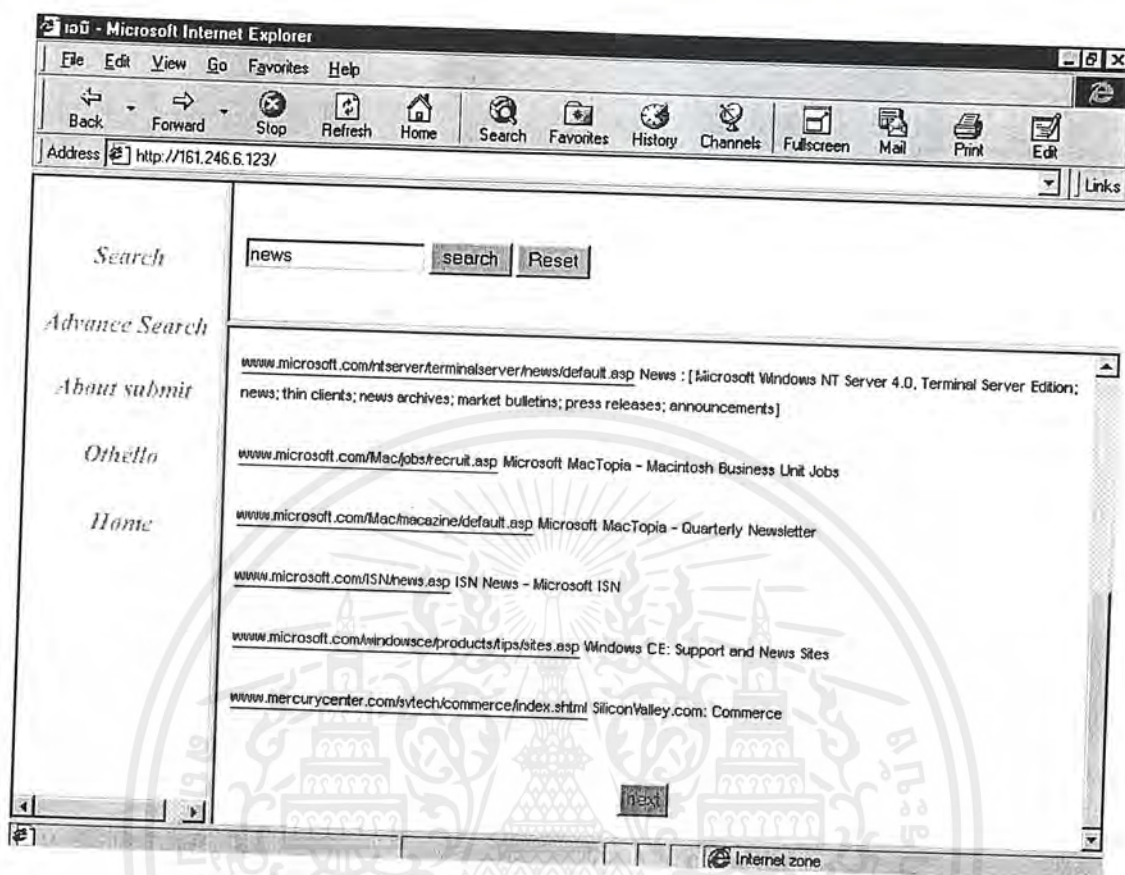


รูปที่ 4-12 ผลที่ได้จากการค้นหาในการค้นหาแบบธรรมดา (ส่วนบน)

จากรูปที่ 4-12 เป็นส่วนบนของเว็บเพจซึ่งจะเห็นว่าเว็บเพจที่พบทั้งหมด 6582 เว็บเพจ โดยในที่นี่จะเป็นหน้าที่ 1 ซึ่งแสดง 10 เว็บเพจแรกที่มีค่าสูงสุดเรียงตามลำดับลงมา โดยจะแสดงชื่อเว็บเพจ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

title ส่วนรูปที่ 4-13 เป็นส่วนท้ายของหน้านี้โดยจะมีปุ่ม Next ให้กดเมื่อผู้ใช้ต้องการดูชื่อเว็บเพจ 10 ชื่อถัดไป



รูปที่ 4-13 ผลที่ได้จากการค้นหาในการค้นหาแบบธรรมดา (ส่วนล่าง)

4.1.3.1.2 การค้นหาโดยใช้เครื่องหมาย (advance search)

คือเวิร์ดที่ป้อนเข้าไปคือ news&thailand ซึ่งผลลัพธ์ที่ได้แสดงดังรูปที่ 4-14 และ รูปที่ 4-15 ซึ่งรูปที่ 4-14 แสดงส่วนบนของเว็บเพจ และรูปที่ 4-15 แสดงส่วนล่างของเว็บเพจ

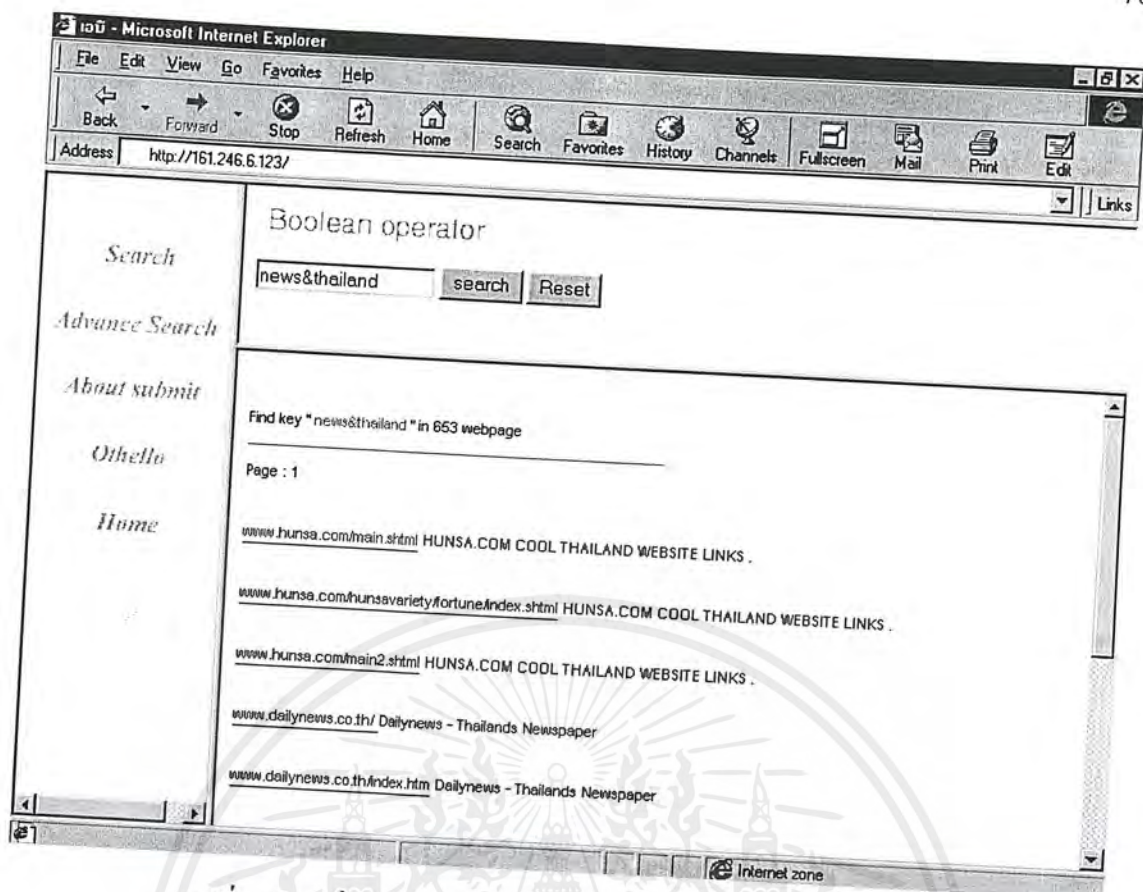
4.1.3.2 การทดสอบเกี่ยวกับการลงทะเบียนรายชื่อเว็บไซต์จากผู้ใช้

จะทำการทดสอบโดยเริ่มแรกจะตรวจสอบว่ามีชื่อเว็บไซต์ที่ต้องการจะลงทะเบียนในฐานะข้อมูลหรือไม่ในที่นี้กำหนดให้เป็น www.apache.org ผลการทดสอบเป็นดังรูปที่ 4-16

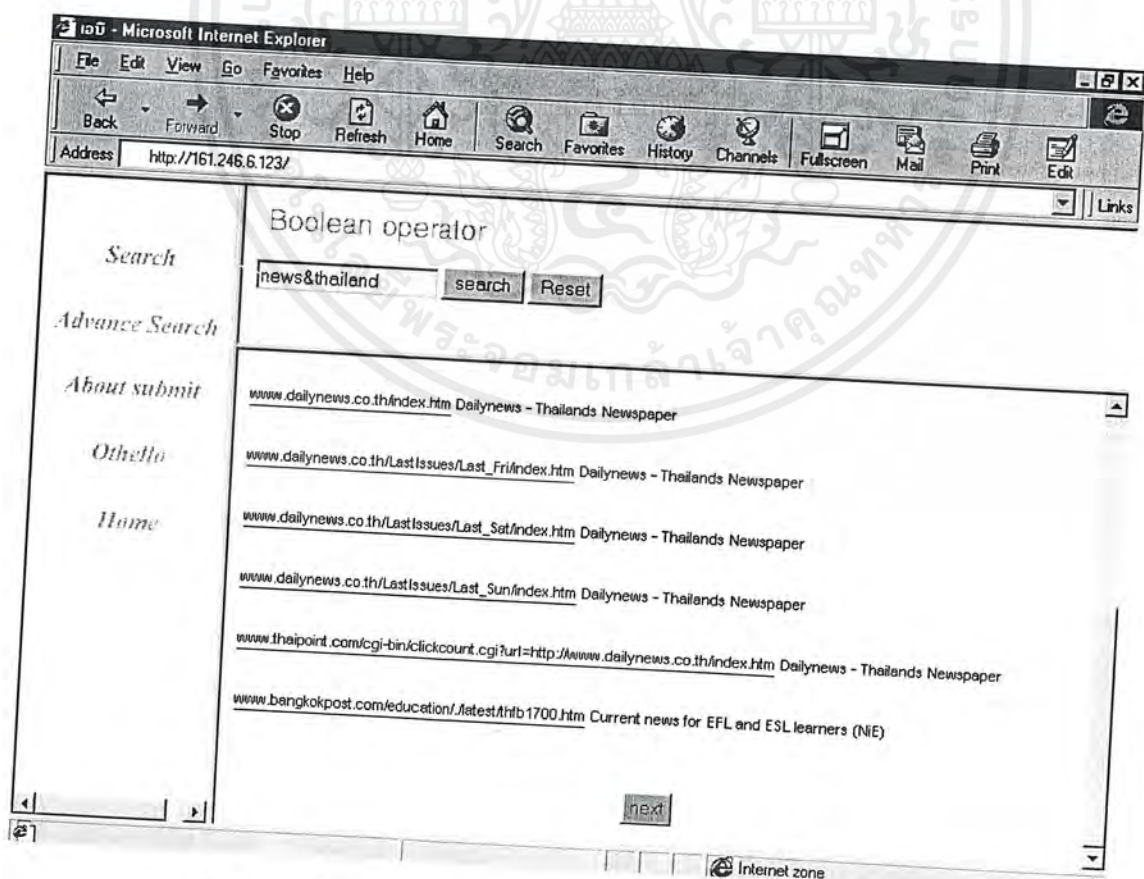
ซึ่งจะแสดงผลออกมาว่ามีอยู่ในฐานข้อมูลแล้วซึ่งถ้าต้องการ submit ใหม่ก็ให้กดปุ่ม Reindex จากนั้นจะขึ้นหน้าจอว่าจะทำการส่งโปรแกรม spider ไปยังเว็บไซต์นี้ในการ update ครั้งต่อไป โดยจะเก็บชื่อ www.apache.org ไว้ในไฟล์ fileurl.txt ซึ่งเป็นไฟล์ที่เก็บรายชื่อเว็บไซต์

แต่ถ้ายังไม่มีในฐานข้อมูลจะขึ้นว่าต้องการที่จะ submit หรือไม่ เช่น กำหนดชื่อเว็บไซต์เป็น www.linux.com ผลที่ออกมาแสดงดังรูปที่ 4-17 ซึ่งถ้าต้องการ submit ก็ให้กดปุ่ม submit เมื่อกดปุ่มแล้วผลที่ได้แสดงดังรูปที่ 4-18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

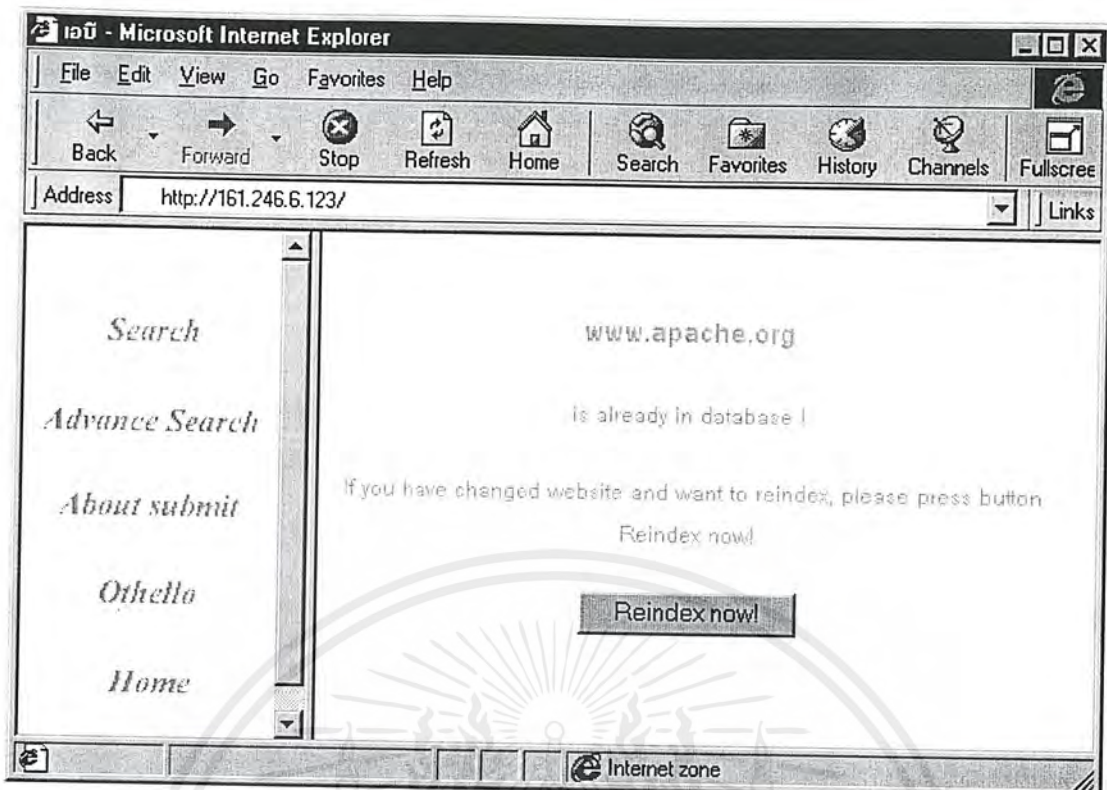


รูปที่ 4-14 ผลที่ได้จากการค้นหาในการค้นหาแบบใช้เครื่องหมาย (ส่วนบน)

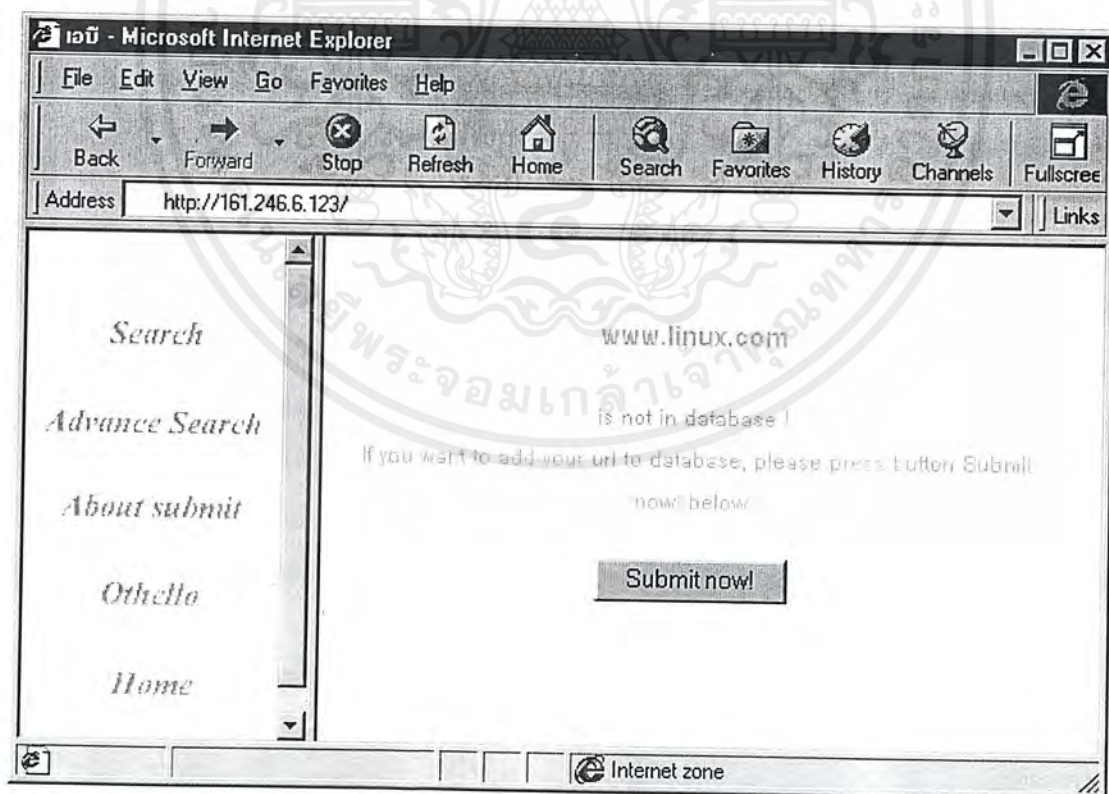


รูปที่ 4-15 ผลที่ได้จากการค้นหาในการค้นหาแบบใช้เครื่องหมาย (ส่วนล่าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นการใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

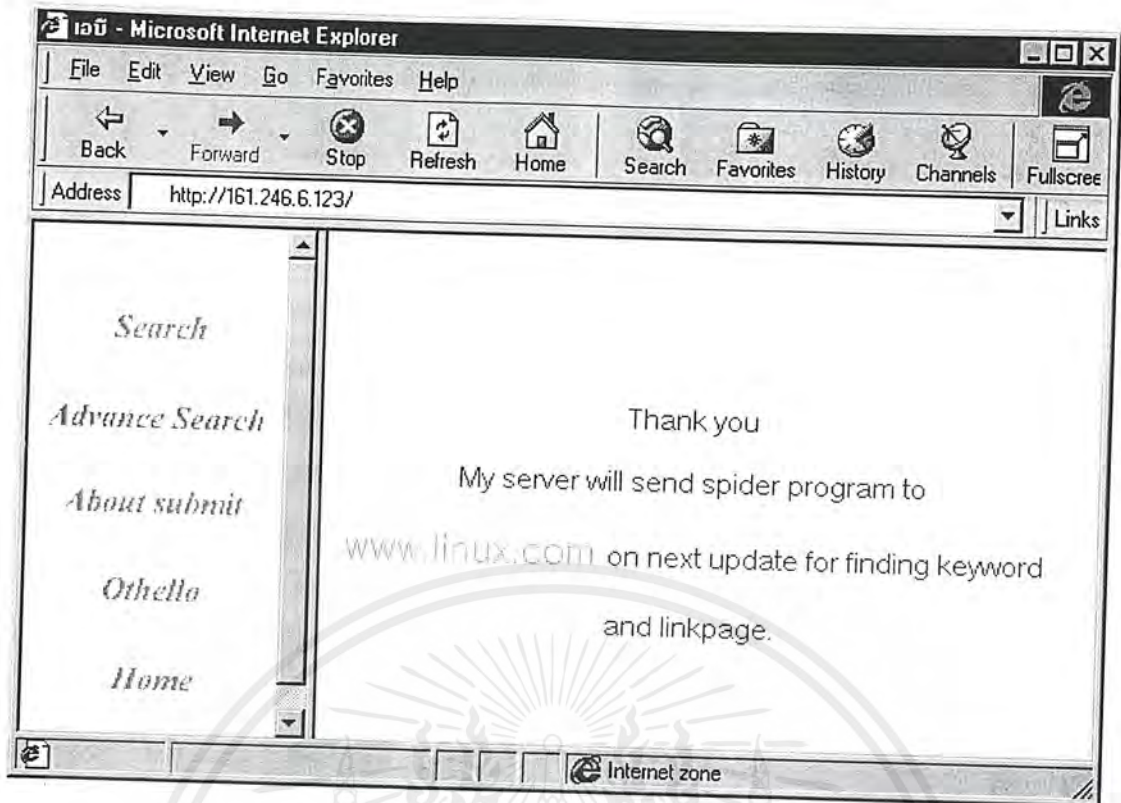


รูปที่ 4-16 ผลจากการตรวจสอบรายชื่อเว็บไซต์



รูปที่ 4-17 ผลจากการตรวจสอบรายชื่อเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-18 ผลจากการ submit ชื่อเว็บไซต์

4.2 การทดสอบและผลการทดสอบในส่วนของเกมออนไลน์

ประกอบด้วย 2 ส่วนหลักที่จะต้องทดสอบคือส่วนไคลเอนท์และส่วนเซิร์ฟเวอร์

4.2.1 Client - side

ส่วนนี้จะเน้นทดสอบในส่วนการเชื่อมต่อเข้ากับระบบเครือข่ายและการดำเนินเกม ซึ่งตัวเกมต้องเป็นไปตามการดำเนินเกมที่ได้ออกแบบที่ได้วางไว้ โดยสามารถทำการตรวจสอบเป็นส่วนๆ ไป ซึ่งส่วนที่จะทำการทดสอบสามารถสรุปได้ดังนี้

1. การตรวจสอบการเชื่อมต่อเข้ากับเครือข่ายในเครื่องเดียวกัน
2. การตรวจสอบการเชื่อมต่อเข้ากับเครือข่ายต่างเครื่องกัน
3. การตรวจสอบการเข้าจังหวะ (synchronize) ที่เซิร์ฟเวอร์ส่งมา
4. การตรวจสอบการเล่นแบบ Multi-user
5. การตรวจสอบการส่งเมสเสจของผู้เล่นแต่ละคน
6. การตรวจสอบการรับอินพุตจากผู้เล่น
7. การตรวจสอบการวางหมากเป็นไปตามกฎที่ได้วางเอาไว้หรือไม่
8. การตรวจสอบการผลัดเปลี่ยนเทิร์นของผู้เล่น
9. การตรวจสอบการเปลี่ยนสถานะของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

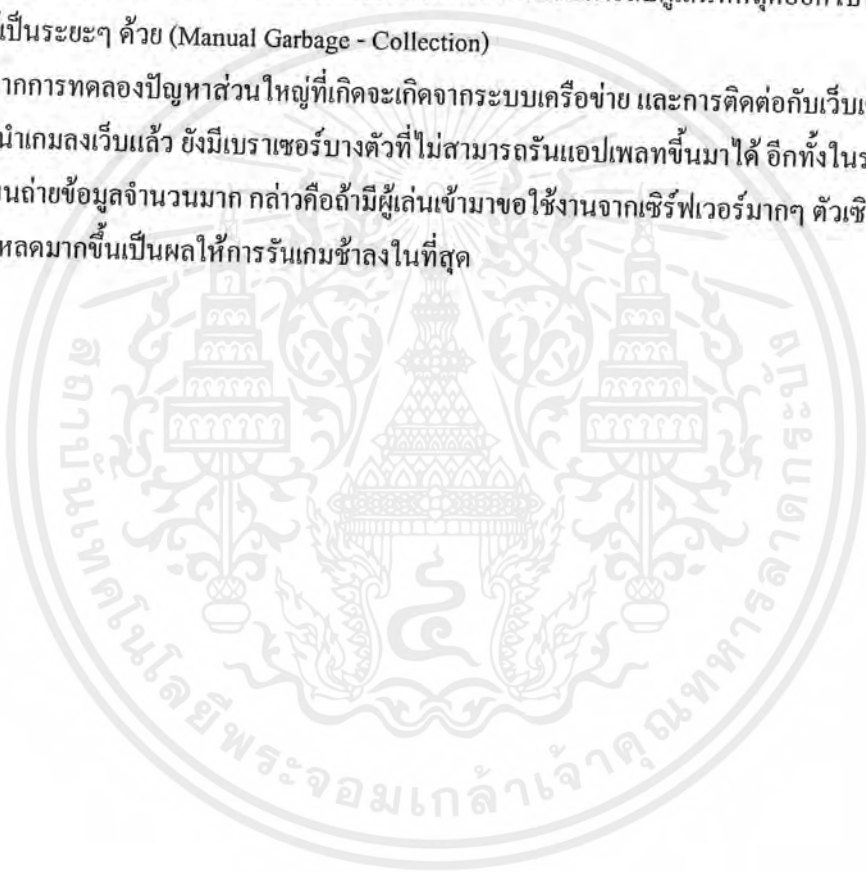
10. การตรวจสอบการสิ้นสุดเกมทั้งที่ตั้งใจและไม่ตั้งใจ

เมื่อทำการตรวจสอบการทำงานจนถูกต้องแล้วก็มาถึงขั้นตอนในการนำเกมลงเว็บ โดยผ่านเว็บเซิร์ฟเวอร์เป็นตัวลิงค์ไปยังเกมแอปพลิเคชัน

4.2.2 Server - side

สำหรับทางฝั่งเซิร์ฟเวอร์จะต้องตรวจสอบการทำงานให้สามารถรอการเชื่อมต่อจากไคลเอนท์สามารถทำการเข้าจังหวะให้กับผู้เล่นแต่ละคนได้อย่างถูกต้อง นอกจากนี้ยังต้องตรวจสอบในส่วนที่รับส่งข้อมูล (message) ของผู้เล่นทั้งที่ผู้เล่นส่งถึงกันและในส่วนที่ผู้เล่นส่งคำสั่งให้กับเซิร์ฟเวอร์ซึ่งต้องส่งข้อมูลถึงกันได้อย่างถูกต้อง และในขณะเดียวกันจะต้องคอยตรวจสอบการลบผู้เล่นที่หลุดออกไปจากเซิร์ฟเวอร์เป็นระยะๆ ด้วย (Manual Garbage - Collection)

จากการทดลองปัญหาส่วนใหญ่ที่เกิดจะเกิดจากระบบเครือข่าย และการติดต่อกับเว็บเซิร์ฟเวอร์คือ ตอนที่นำเกมลงเว็บแล้ว ยังมีเบรเซอร์บางตัวที่ไม่สามารถรันแอปพลิเคชันขึ้นมาได้ อีกทั้งในระบบเครือข่ายมีการขนถ่ายข้อมูลจำนวนมาก กล่าวคือถ้ามีผู้เล่นเข้ามาขอใช้งานจากเซิร์ฟเวอร์มากๆ ตัวเซิร์ฟเวอร์เองจะรับโหลดมากขึ้นเป็นผลให้การรันเกมช้าลงในที่สุด



บทที่ 5

บทสรุปและวิจารณ์

5.1 สรุปผลการดำเนินงาน

สามารถสรุปผลงานแต่ละส่วนในโครงการได้ดังนี้

5.1.1 ส่วนของการท่องไปตามเว็บไซต์ของโปรแกรม Spider

สรุปความสามารถได้ดังนี้

- สามารถวิเคราะห์ข้อมูลเพื่อค้นหาคีย์เวิร์ดและค่าความสำคัญแต่ละค่าที่ได้จากเว็บเพจได้อย่างถูกต้อง
- สามารถวิเคราะห์ข้อมูลเพื่อค้นหาชื่อเว็บเพจและชื่อเว็บไซต์ที่มีอยู่ในเว็บเพจได้อย่างถูกต้อง
- สามารถท่องไปตามเว็บไซต์ต่าง ๆ ที่พบได้อย่างมีประสิทธิภาพ
- สามารถนำข้อมูลที่ได้จากการวิเคราะห์มาจัดเก็บลงฐานข้อมูลได้อย่างถูกต้อง

ซึ่งการท่องไปตามเว็บไซต์ของโปรแกรม Spider นั้น ได้ถูกจำกัดจำนวนเว็บไซต์ไว้เนื่องจากว่าเราไม่สามารถเก็บข้อมูลได้ทั้งหมดจากเว็บไซต์ที่มีอยู่ทั่วโลกเพราะมีข้อจำกัดในพื้นที่ที่เซิร์ฟเวอร์

5.1.2 ส่วนของการค้นหาข้อมูลจากผู้ใช้

สรุปความสามารถได้ดังนี้

- สามารถรองรับการใช้งานจากผู้ใช้ได้หลาย ๆ คนในเวลาเดียวกันอย่างมีประสิทธิภาพ
- สามารถรองรับการค้นหาคีย์เวิร์ดได้พร้อมกันหลาย ๆ คำจากผู้ใช้ได้
- สามารถรองรับการค้นหาโดยใช้วิธีการทางคณิตศาสตร์มาช่วยซึ่งจะช่วยเพิ่มประสิทธิภาพในการค้นหาจากผู้ใช้มากยิ่งขึ้น
- สามารถติดต่อกับฐานข้อมูลได้อย่างมีประสิทธิภาพ โดยมีเครื่องเซิร์ฟเวอร์เป็นตัวกลาง

5.1.3 ส่วนของเกมออนไลน์

ซึ่งในที่นี้ได้พัฒนาเกม Othello ขึ้น โดยสรุปความสามารถได้ดังนี้

- สามารถจัดการการเล่นเกมระหว่างผู้เล่นได้อย่างมีประสิทธิภาพ
- สามารถเล่นได้พร้อมกันหลาย ๆ คู่ โดยมีเครื่องเซิร์ฟเวอร์คอยจัดการอย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 แนวทางในการพัฒนาต่อ

โครงการนี้สามารถที่จะพัฒนาขีดความสามารถได้เพิ่มเติมต่อไปอีกดังนี้

5.2.1 ส่วนของการท่องเที่ยวตามเว็บไซต์ของโปรแกรม Spider

- พัฒนาหลักการในการเทียบค่าและคำนวณค่าความสำคัญให้เหมาะสมยิ่งขึ้น
- พัฒนาในส่วนของการ update ข้อมูลในฐานข้อมูลให้มีประสิทธิภาพเพิ่มขึ้น
- พัฒนารูปแบบการจัดเก็บข้อมูลในฐานข้อมูลให้มีประสิทธิภาพเพิ่มขึ้น

5.2.2 ส่วนของการค้นหาข้อมูลจากผู้ใช้

- พัฒนาการค้นหาข้อมูลให้มีประสิทธิภาพเพิ่มขึ้น
- พัฒนารูปแบบในการค้นหาข้อมูลเพื่อให้ตรงกับความต้องการของผู้ใช้มากขึ้น

5.2.3 ส่วนของเกมออนไลน์

- พัฒนาประสิทธิภาพในการรองรับการเล่นพร้อมกันหลายๆ คู่ ของผู้ใช้งานมากขึ้น
- พัฒนารูปแบบของการนำเสนอเกมบนเว็บไซต์ให้ตรงกับความต้องการของผู้ใช้มากขึ้น
- พัฒนาให้ผู้เล่นสามารถเล่นคนเดียวได้

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ตัวอย่างโค้ดของเว็บเพจ www.peugeot-th.com ที่ถูกนำไปใช้ในการทดสอบในบทที่ 4 มีดังนี้

```
<HTML>
<HEAD><TITLE>WELCOME TO PEUGEOT THAILAND</TITLE>
<META NAME="Generator" CONTENT="EditPlus 1.2">
<META NAME="Author" CONTENT="Chanisa">
<META NAME="Keywords" CONTENT="Peugeot, Peugeot Thailand, Peugeot Car, Car,
306,450,406,605, car model,
Peugeot dealers, Peugeot service in Thailand, Peugeot Press Release, Contact Peugeot, Peugeot
boutique, Peugeot
Thai showroom, What's new, Motor, European Automobils, Automotive, Peugeot Thai
Language, Yontrakit">
<META NAME="Description" CONTENT="Welcome to Peugeot Thailand, Thai Peugeot Web site,
Contact Peugeot
Thailand , Visit Peugeot Thailand's website, Peugeot Thailand, Peugeot Showroom, Peugeot Press
Release, Peugeot
Thailand Address, Peugeot Gallery, Car Photos.">
</HEAD>
<FRAMESET ROWS="*,460,*" BORDER=0>
  <FRAME SRC=black.htm SCROLLING=NO>
  <FRAMESET COLS="*,640,*">
    <FRAME SRC=black.htm SCROLLING=NO>
    <FRAME SRC=home.htm SCROLLING=AUTO NAME=WHOLE
MARGINWIDTH=0
MARGINHEIGHT=0>
  <FRAME SRC=black.htm SCROLLING=NO>
</FRAMESET>
<FRAME SRC=black.htm SCROLLING=NO>
</FRAMESET>
<!-- Start of NedStat Basic code -->
<a href="http://usa.nedstatbasic.net/cgi-bin/viewstat?name=lion"></a>

<!-- End of NedStat Basic co



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข

## Stop Words

รายการของคำที่ไม่สื่อความหมายมีดังนี้

---

A | B | C | D | E | F | H | I | L | M | N | O | P | R | S | T | U | V | W | X | Y |

---

## A

a, about, above, across, adj, after, afterwards, again, against, albeit, all, almost, alone, along, already, also, although, always, among, amongst, an, and, another, any, anyhow, anyone, anything, anywhere, are, around, as, at

---

## B

be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, below, beside, besides, between, beyond, both, but, by

---

## C

can, cannot, co, could

---

## D

down, during

---

## E

each, eg, either, else, elsewhere, enough, etc, even, ever, every, everyone, everything, everywhere, except

---

## F

few, first, for, former, formerly, from, further

---

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

---



---

H

had, has, have, he, hence, her, here, hereafter, hereby, herein, hereupon, hers, herself, him, himself, his, how, however

---

I

ie, if, in, inc, indeed, into, is, it, its, itself

---

L

last, latter, latterly, least, less, ltd

---

M

many, may, me, meanwhile, might, more, moreover, most, mostly, much, must, my, myself

---

N

namely, neither, never, nevertheless, next, no, nobody, none, noone, nor, not, nothing, now, nowhere

---

O

of, off, often, on, once one, only, onto, or, other, others, otherwise, our, ours, ourselves, out, over, own

---

P

per, perhaps

---

R

rather

---

S

same, seem, seemed, seeming, seems, several, she, should, since, so, some, somehow, someone, something, sometime, sometimes, somewhere, still, such

---

---



---

T

than, that, the, their, them, themselves, then, thence, there, thereafter, thereby, therefor, therein, thereupon, these, they, this, those, though, through, throughout, thru, thus, to, together, too, toward, towards

---



---

U

under, until, up, upon, us

---



---

V

very, via

---



---

W

was, we, well, were, what, whatever, whatsoever, when, whence, whenever, whensoever, where, whereafter, whereas, whereat, whereby, wherefrom, wherein, whereinto, whereof, whereon, whereto, whereunto, whereupon, wherever, wherewith, whether, which, whichever, whichsoever, while, whilst, whither, who, whoever, whole, whom, whomever, whomsoever, whose, whosoever, why, will, with, within, without, would

---



---

X

xsubj, xcal, xauthor, xother, xnote

---



---

Y

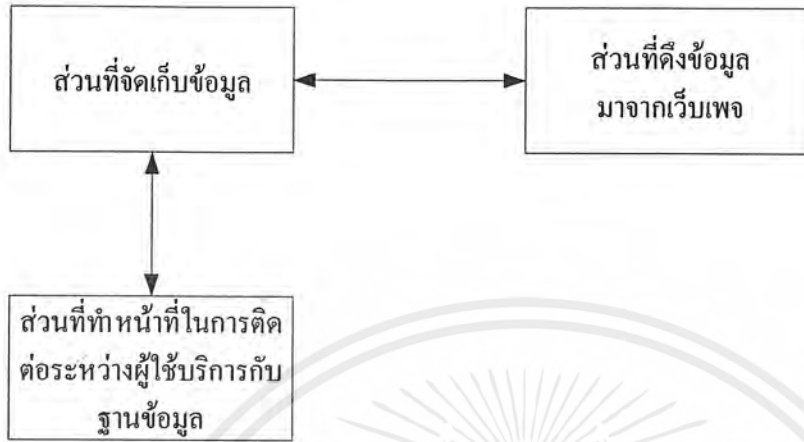
yet, you, your, yours, yourself, yourselves

---

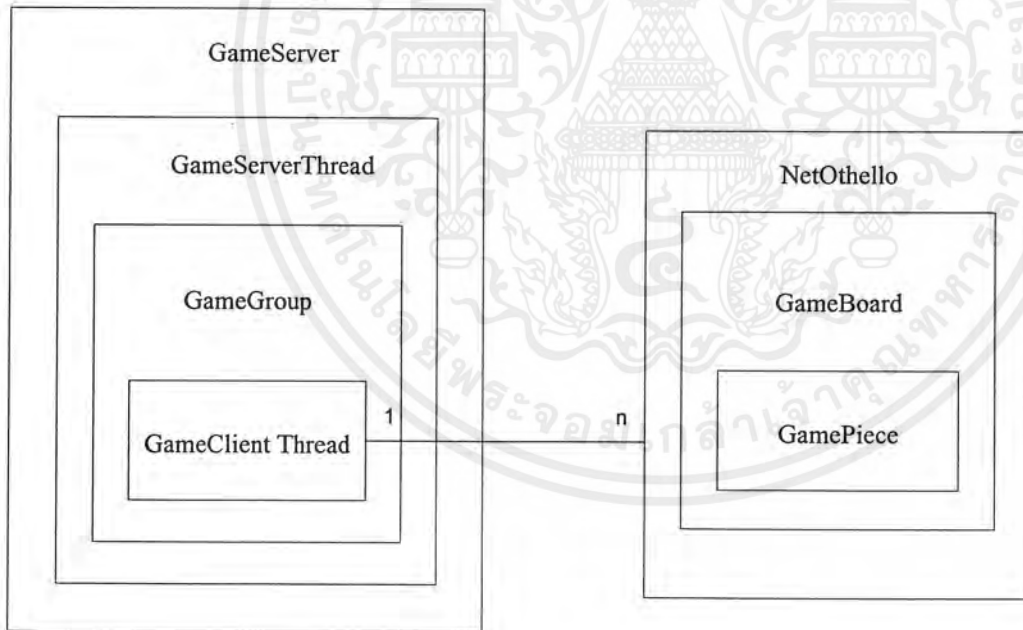


---

ภาคผนวก ค  
แผนภาพการทำงานต่าง ๆ

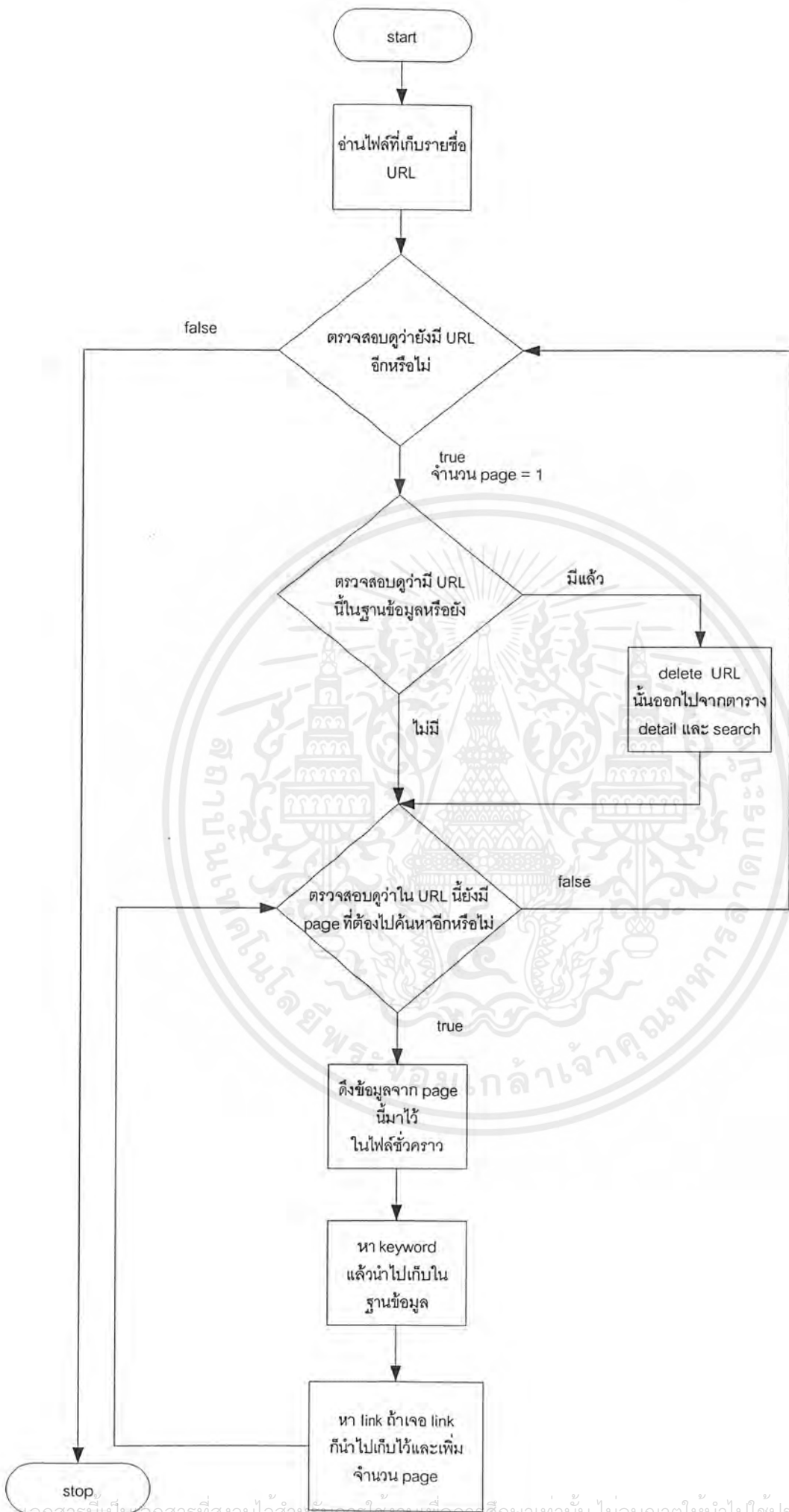


รูปที่ ค-1 แสดงการทำงานในส่วนของเซิร์ฟเวอร์เอ็นจิน

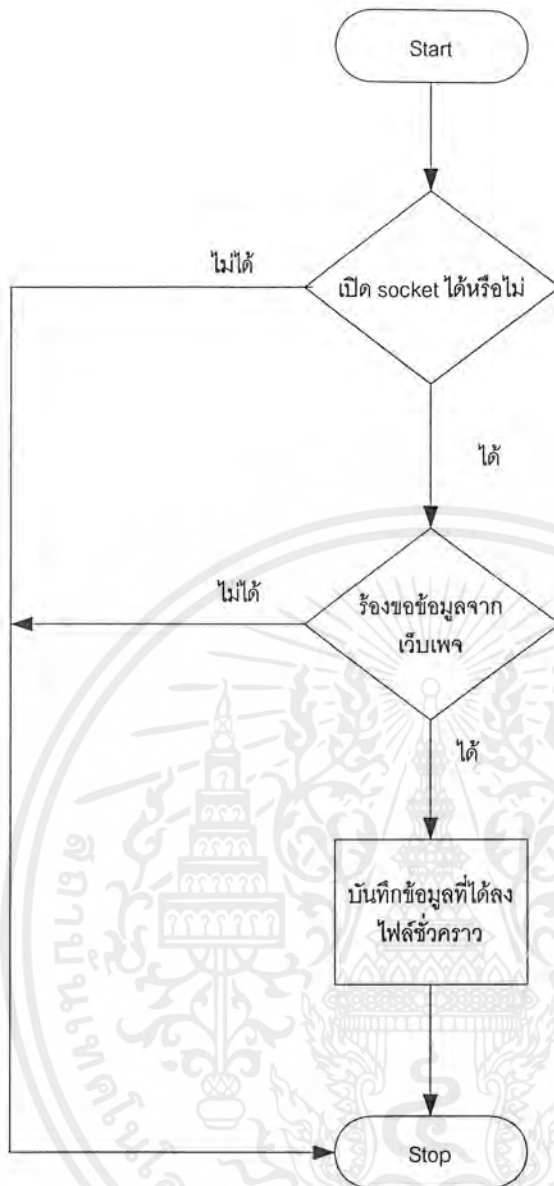


รูปที่ ค-2 แสดงคลาสโคดของเกม Othello

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

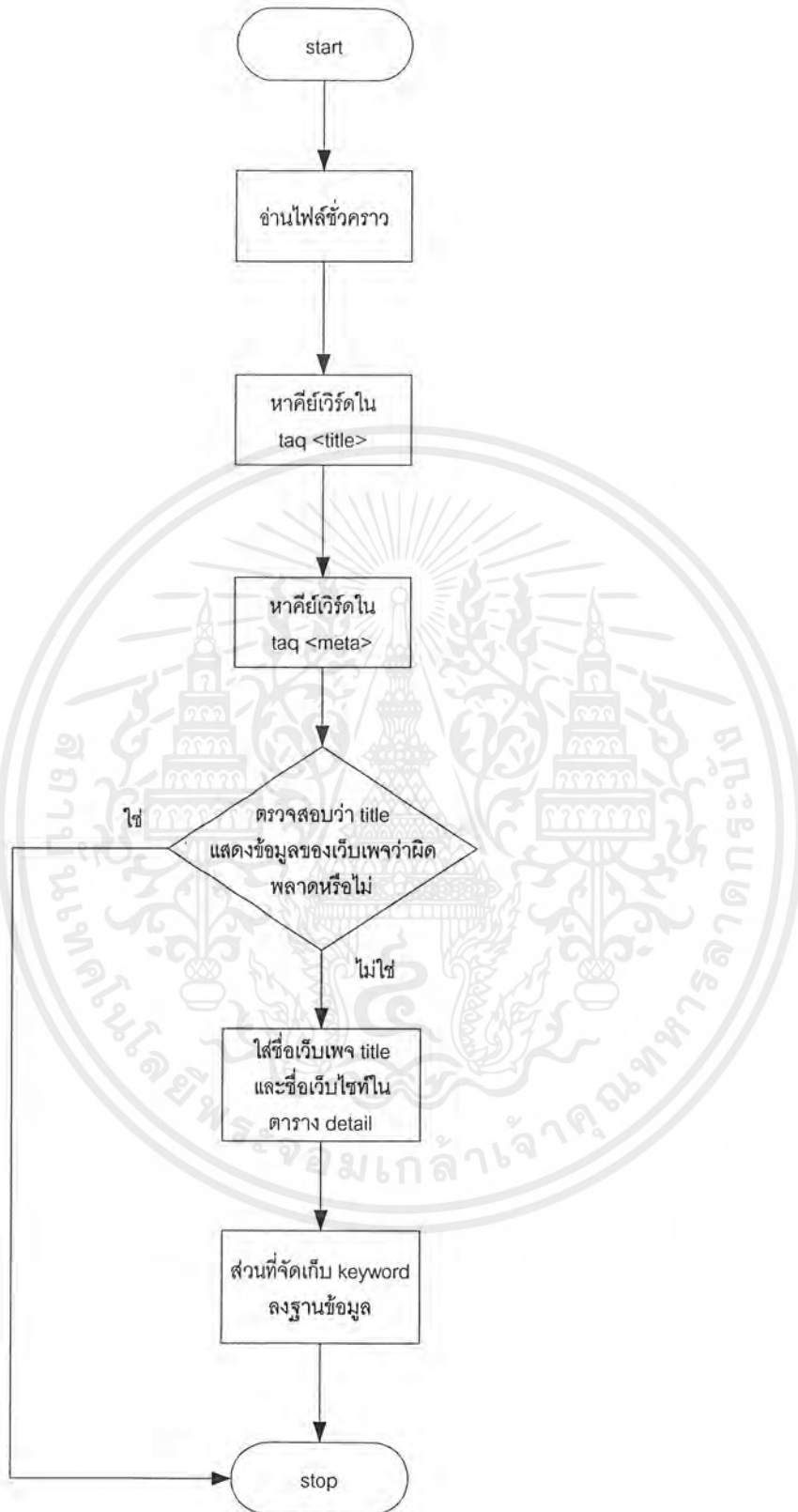


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีก **รูปที่ ก-3 แสดงโฟลว์ชาร์ตในส่วนของ Spider** ไปถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



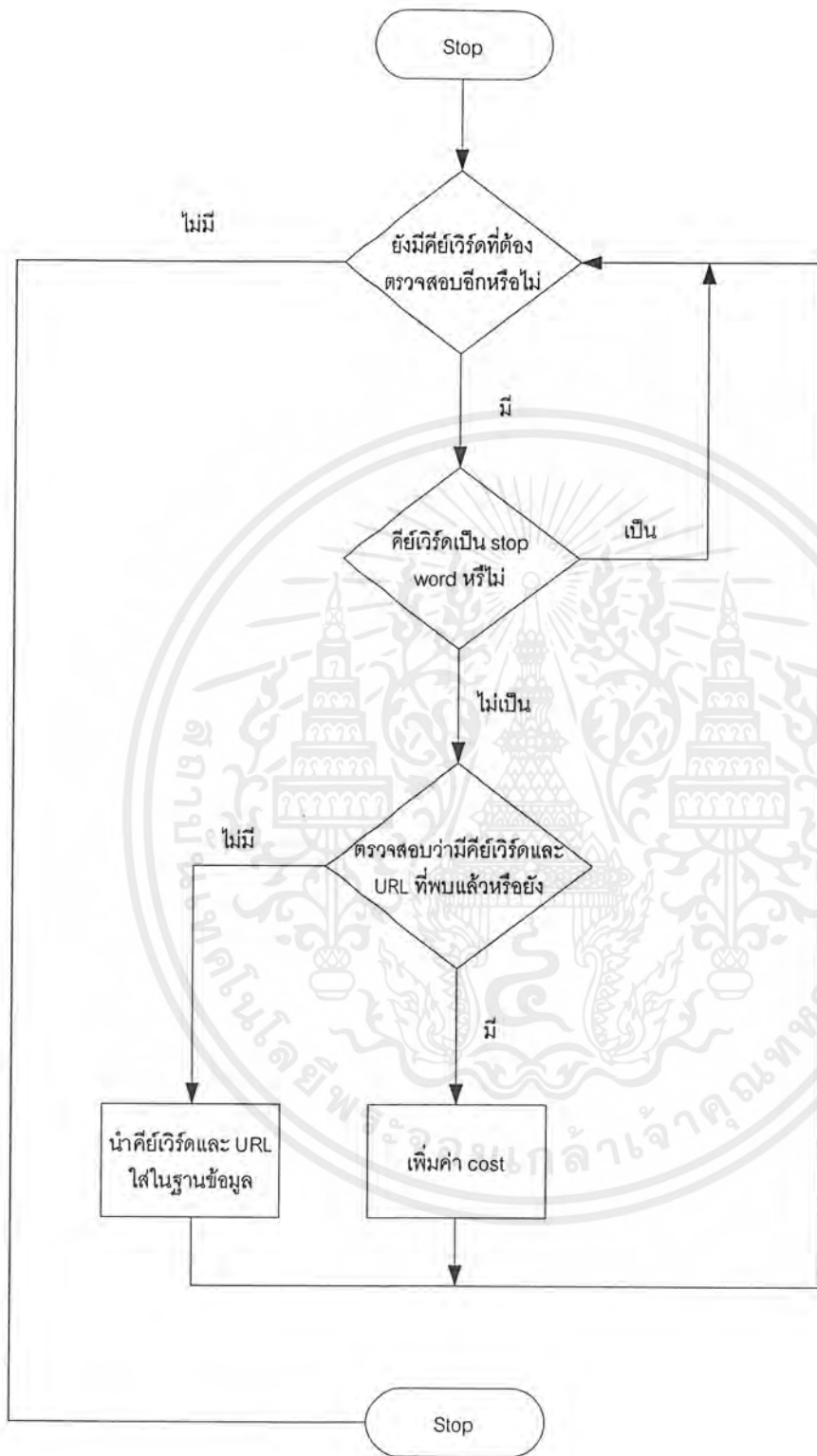
รูปที่ ก-4 แสดงโฟลว์ชาร์ตในส่วนของการร้องขอข้อมูลจากเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



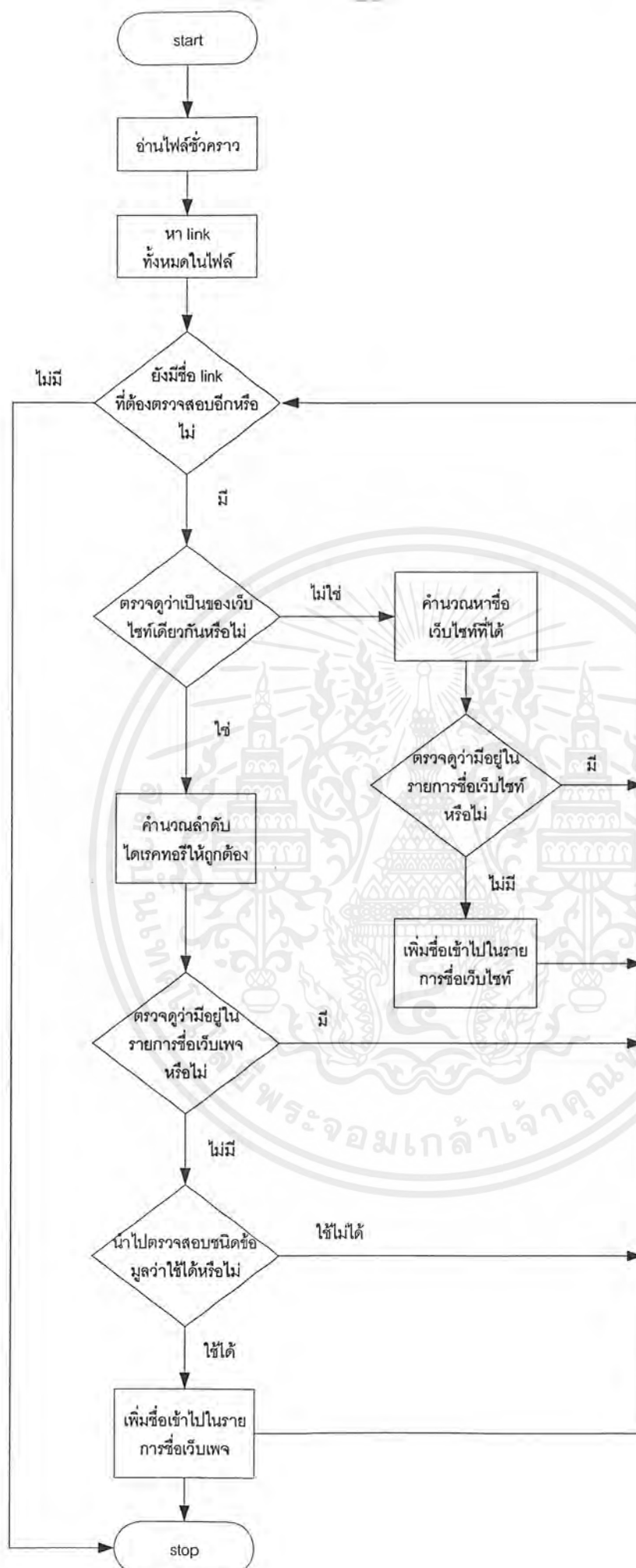
รูปที่ ก-5 แสดงโฟลว์ชาร์ตในส่วนของการวิเคราะห์ข้อมูลเพื่อหาคีย์เวิร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

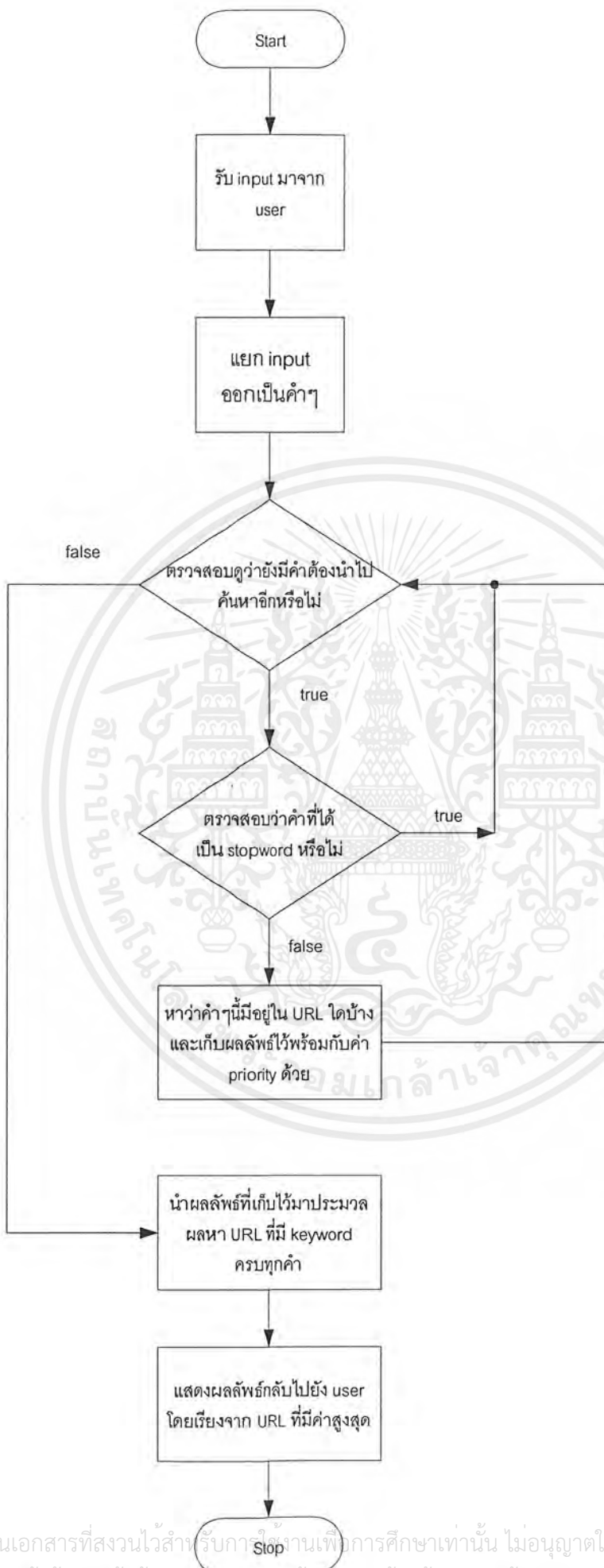


รูปที่ ค-6 แสดงโฟลว์ชาร์ตในส่วนของการตรวจสอบคีย์เวิร์ดเพื่อจัดเก็บลงฐานข้อมูล

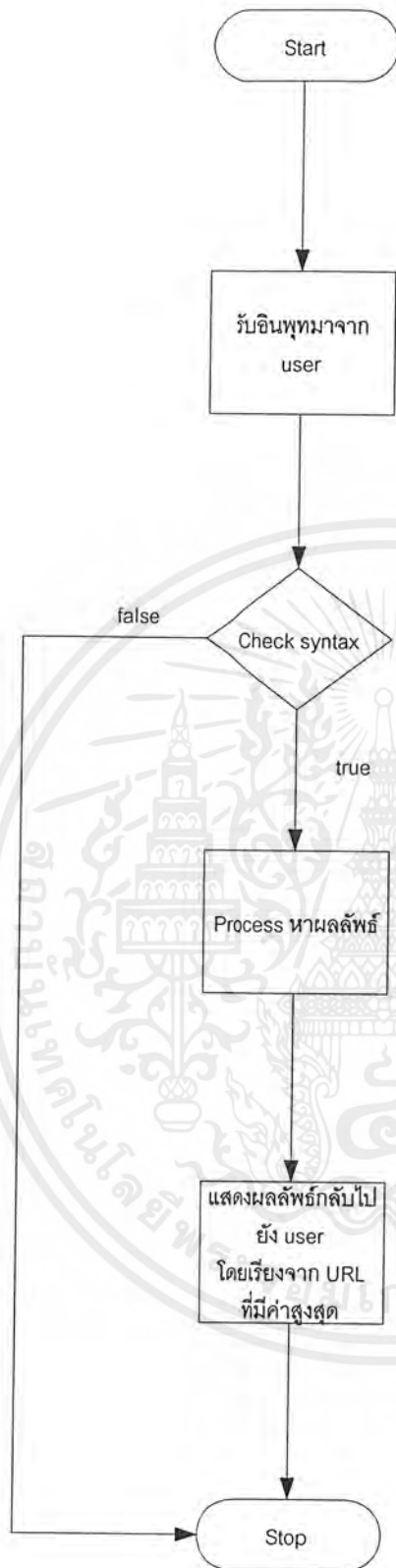
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ หากมีข้อผิดพลาดในส่วนนี้ของกรวิเคราะห์ข้อมูลเพื่อหาชื่อเว็บเพจและชื่อเว็บไซต์  
**รูปที่ ก-7 แสดงไฟล์ชาร์ตในส่วนของการวิเคราะห์ข้อมูลเพื่อหาชื่อเว็บเพจและชื่อเว็บไซต์**

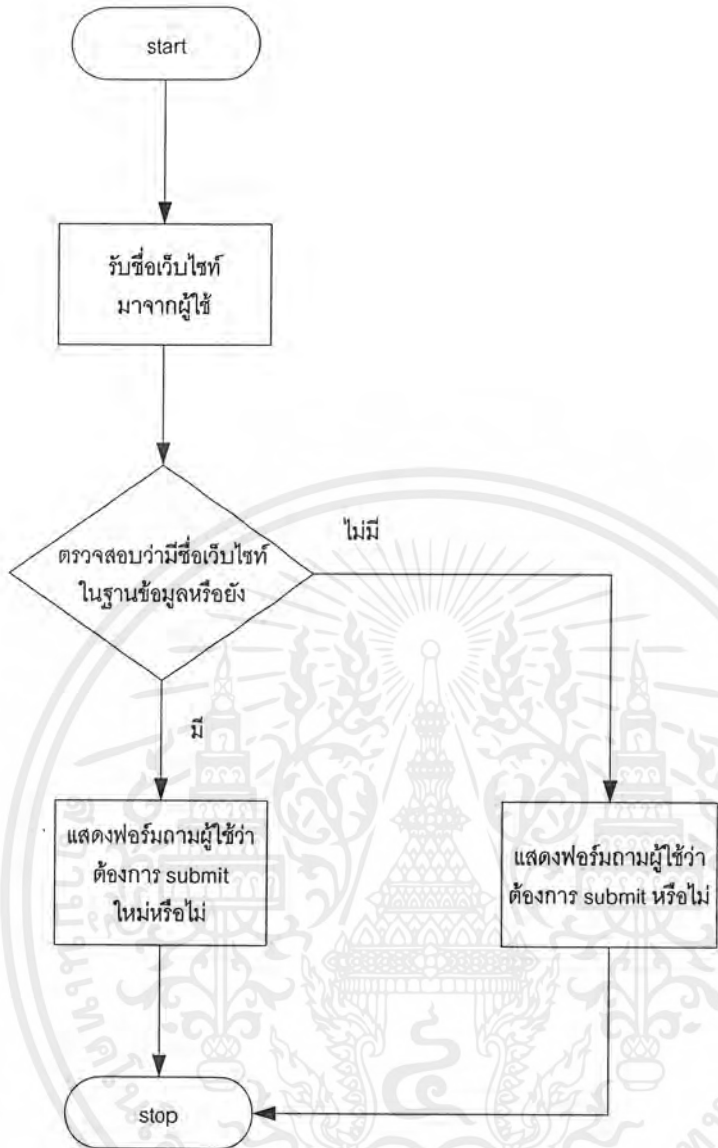


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
**รูปที่ ก-8 แสดงโฟลว์ชาร์ตในส่วนของการค้นหาข้อมูลจากผู้ใช้แบบธรรมชาติ**



รูปที่ ค-9 แสดงโฟลว์ชาร์ตในส่วนของการค้นหาข้อมูลจากผู้ใช้แบบใช้เครื่องหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-10 แสดงโฟลว์ชาร์ตในส่วนของการตรวจสอบชื่อเว็บไซต์จากผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-11 แสดงโฟลว์ชาร์ตในส่วนของการรับชื่อเว็บไซต์จากการ submit ของผู้ใช้

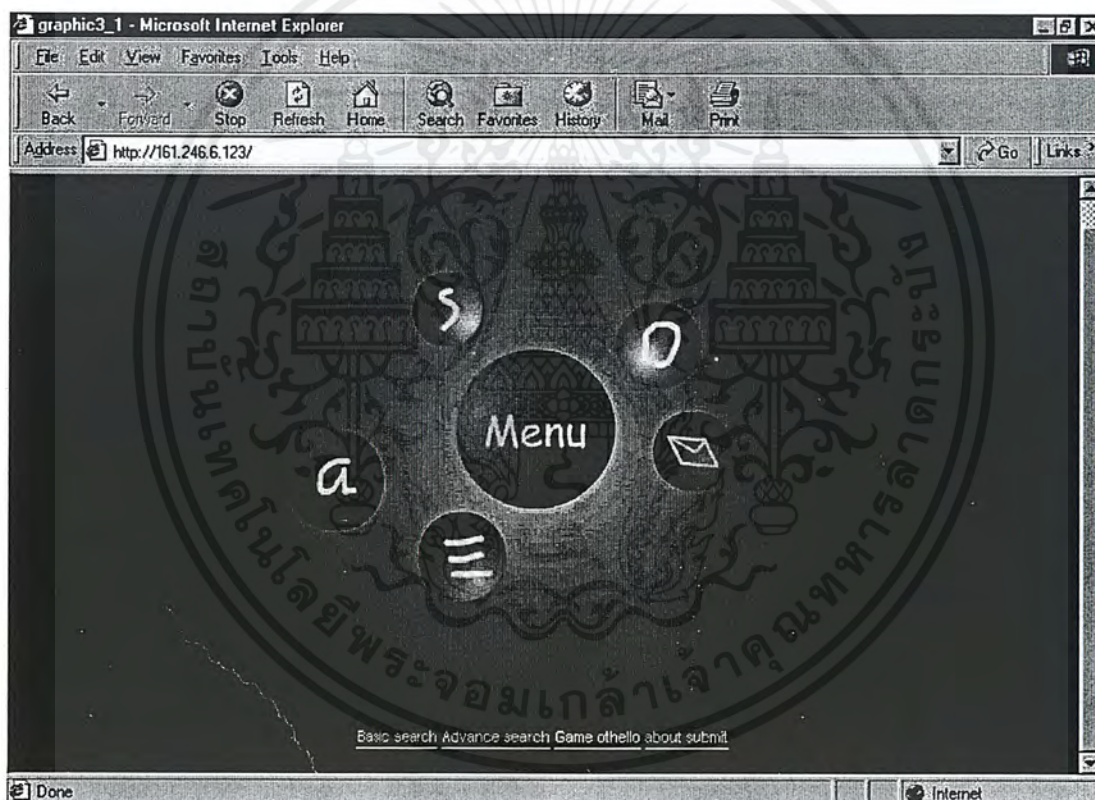
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ง

### คู่มือการใช้งานสำหรับผู้ใช้

หน้าจอแรกจะมีเมนูให้เลือก 5 เมื่อดังรูปที่ ง-1 ดังนี้

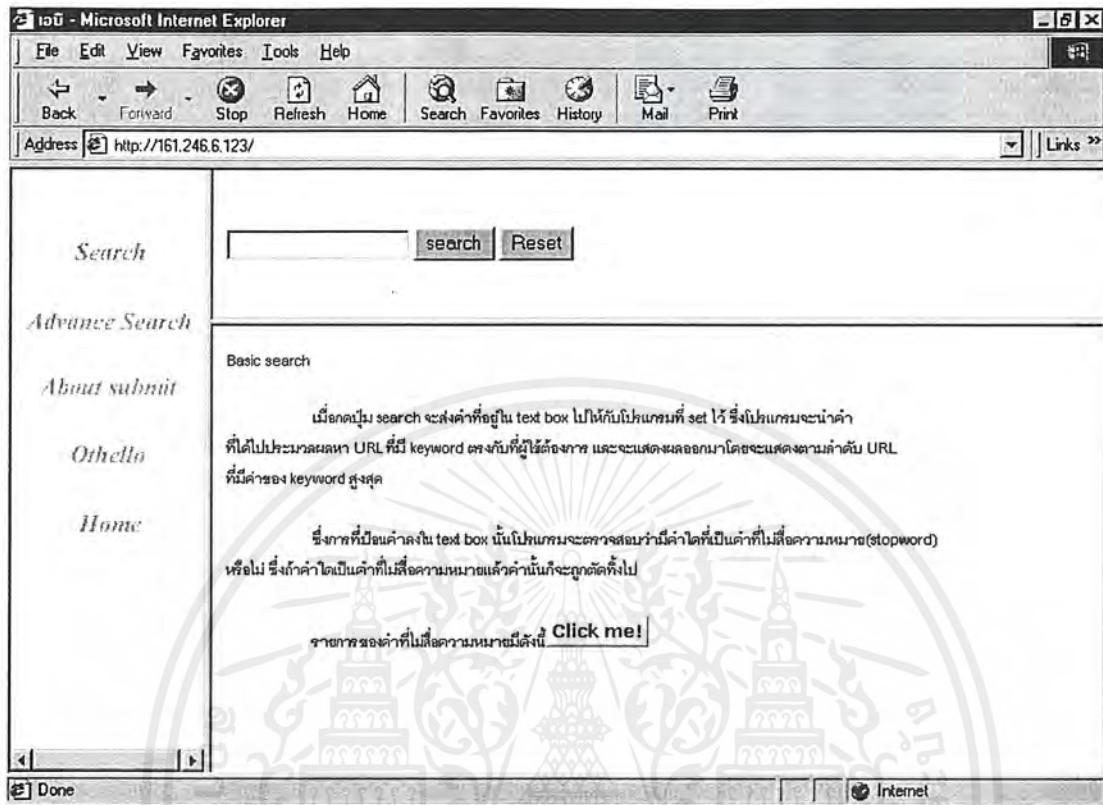
1. Search( ในหน้าจอเป็นรูปตัว S) เป็นเมนูสำหรับการค้นหาข้อมูลแบบธรรมดา
2. Advance Search(ในหน้าจอเป็นรูปตัว a)เป็นเมนูสำหรับการค้นหาข้อมูลแบบใช้เครื่องหมาย
3. About submit (ในหน้าจอเป็นรูปขีด 3 เส้น) เป็นเมนูเกี่ยวกับการ submit ชื่อเว็บไซต์
4. Othello (ในหน้าจอเป็นรูปตัว O) เป็นเมนูสำหรับการเล่นเกม Othello แบบออนไลน์
5. Email (ในหน้าจอเป็นรูปจดหมาย)



รูปที่ ง-1 แสดงหน้าแรกของโฮมเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

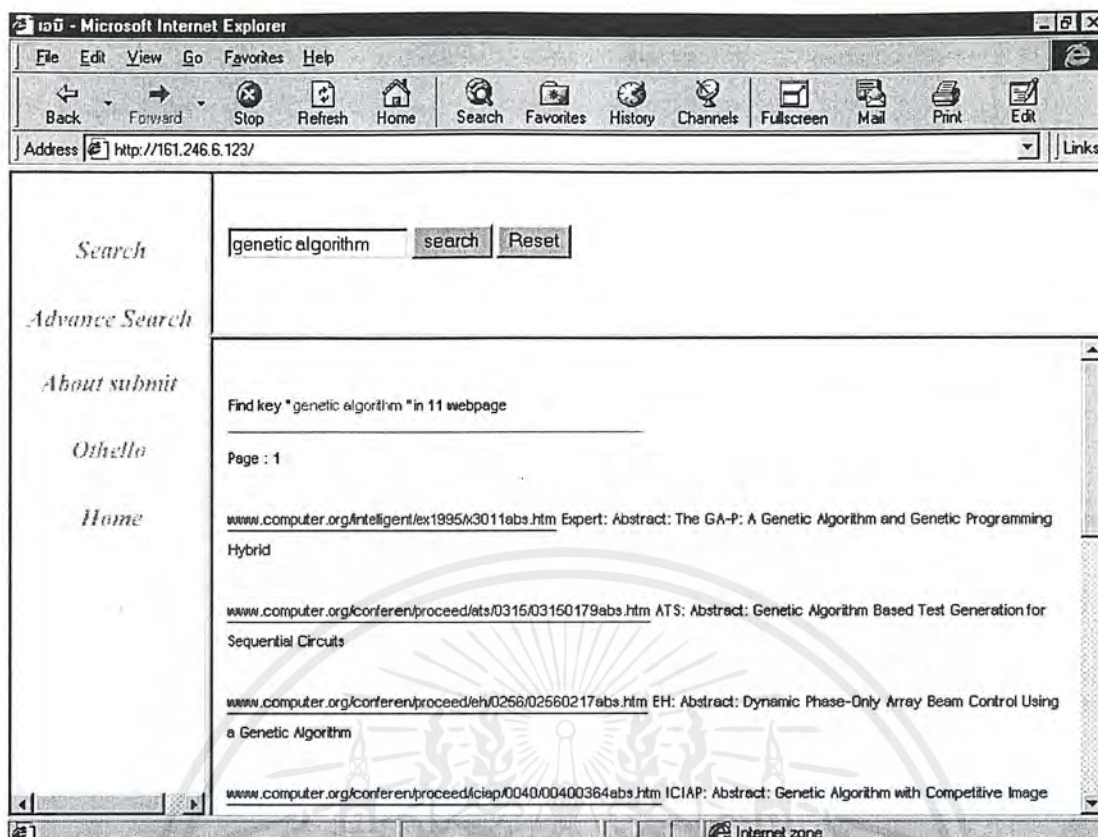
## ในเมนูแรกเมื่อกด Search จะแสดงหน้าจอตั้งรูปที่ จ-2



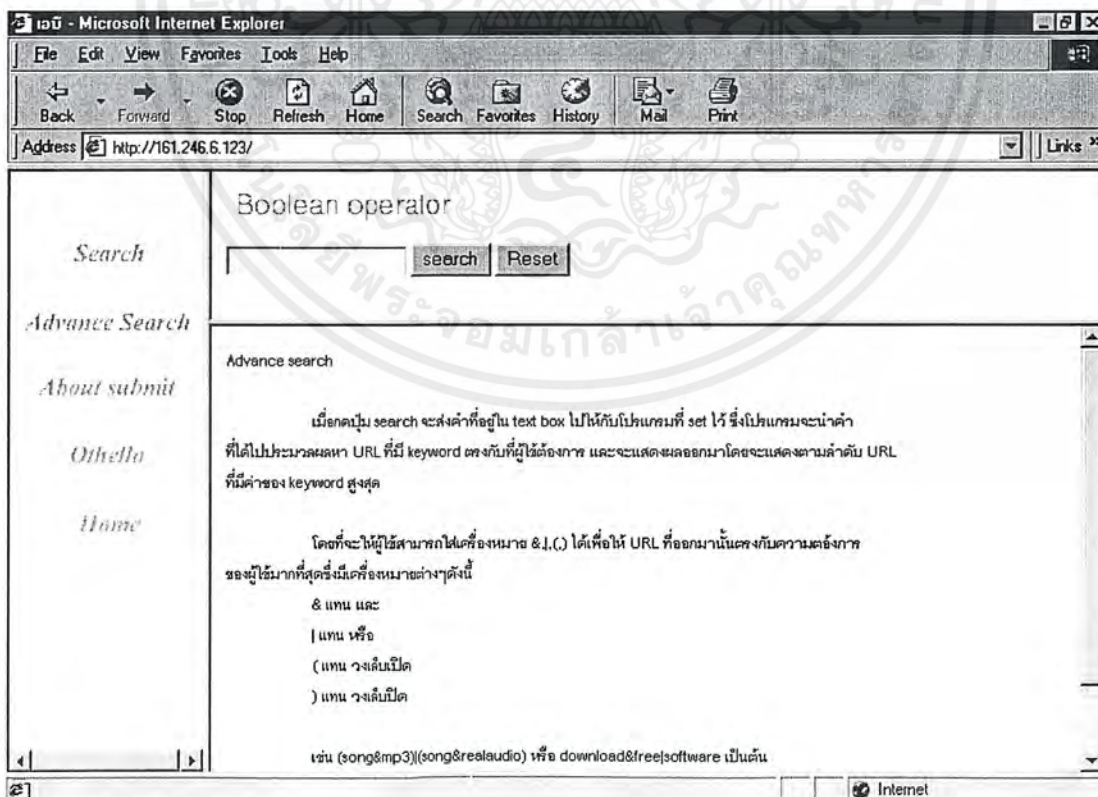
รูปที่ จ-2 แสดงหน้าจอเมื่อกดเมนู Search

ซึ่งในหน้าจอนี้ได้อธิบายวิธีการในการค้นหาข้อมูลเพื่อให้ผู้ใช้ได้ใช้อย่างถูกต้องตรงตามความต้องการและเมื่อป้อนคีย์เวิร์ดลงใน textbox แล้วกดปุ่ม Search จะแสดงผลลัพธ์ออกทางจอภาพเช่นป้อนคำว่า genetic algorithm จะแสดงผลลัพธ์ดังรูปที่ จ-3

ต่อไปเมื่อกดเมนู Advance Search แล้วจะปรากฏหน้าจอตั้งรูปที่ จ-4 โดยได้อธิบายวิธีการในการทำงานไว้เพื่อให้ผู้ใช้ได้ใช้อย่างถูกต้องตรงตามความต้องการและเมื่อป้อนคีย์เวิร์ดลงใน textbox แล้วกดปุ่ม Search จะแสดงผลลัพธ์ออกทางจอภาพเช่นป้อนเข้าไปว่า genetic algorithm จะแสดงผลลัพธ์ดังรูปที่ จ-5

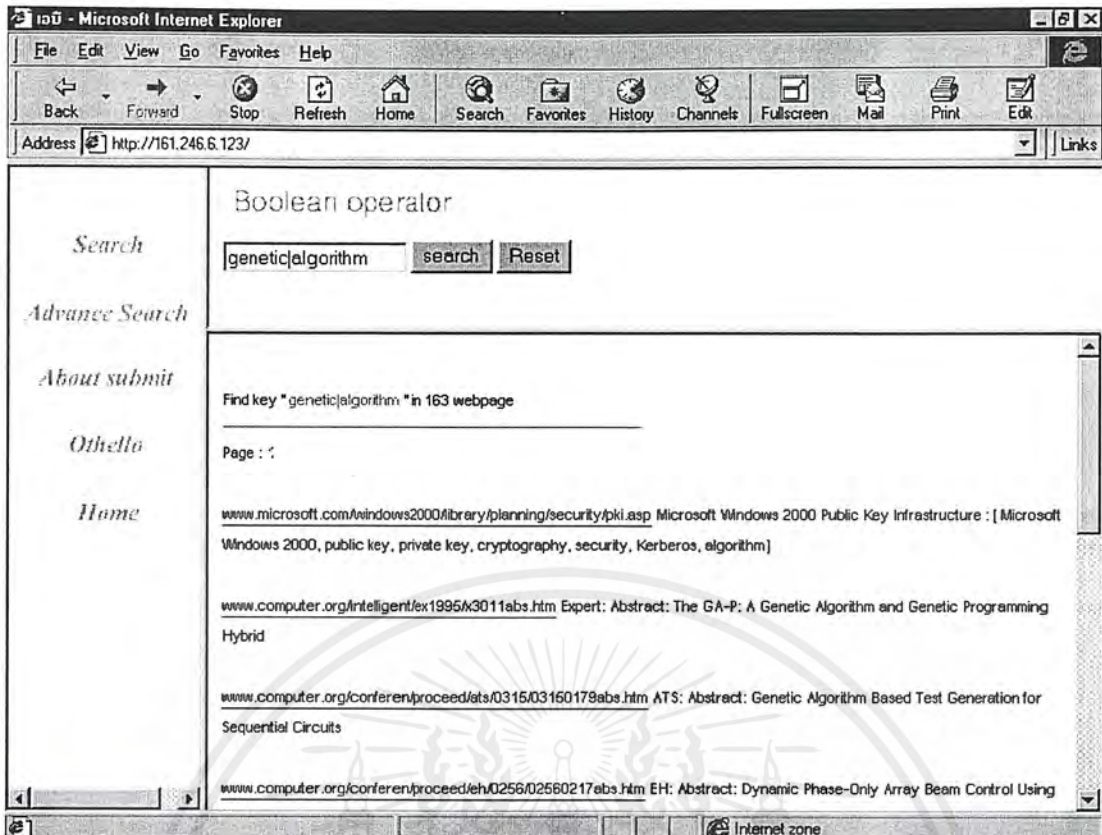


รูปที่ จ-3 แสดงรายชื่อเว็บเพจที่ได้จากการค้นหาแบบธรรมดา

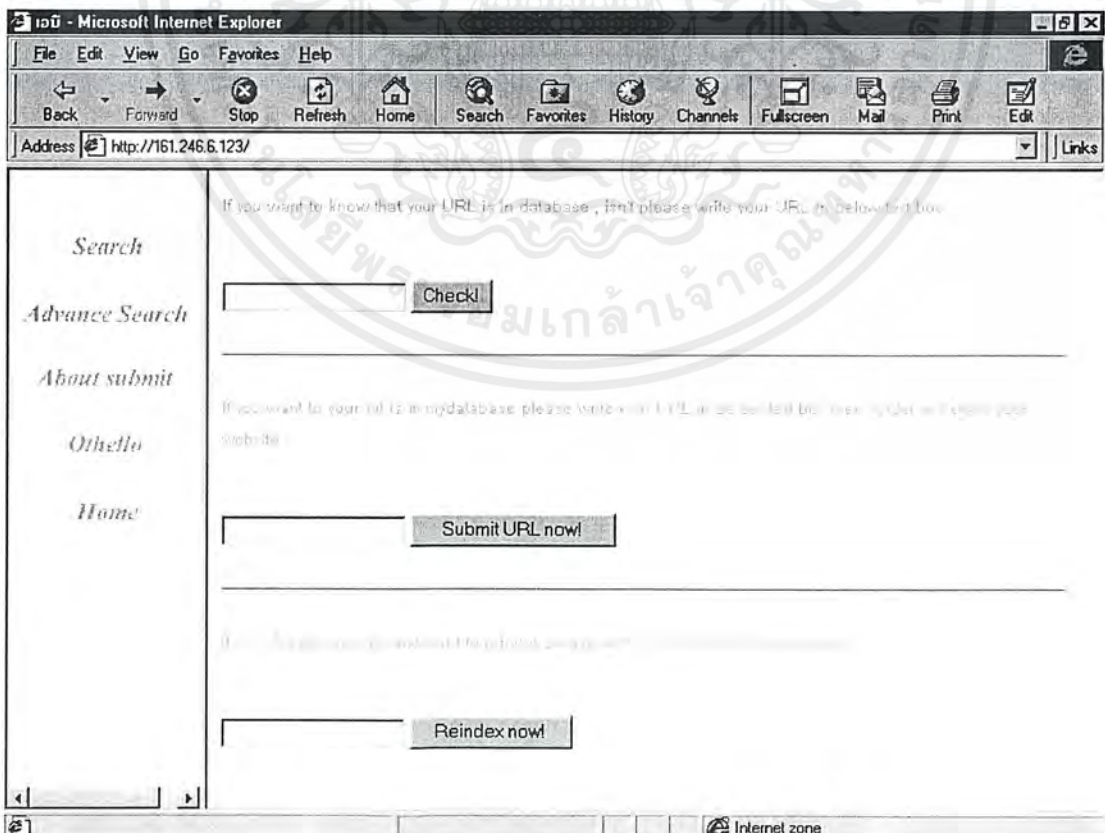


รูปที่ จ-4 แสดงหน้าจอเมื่อคลิกเมนู Advance Search

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ-5 แสดงรายชื่อเว็บเพจที่ได้จากการค้นหาแบบใช้เครื่องหมาย  
 ต่อไปเมื่อกดเมนู About submit แล้วจะปรากฏหน้าจอดังรูปที่ จ-6

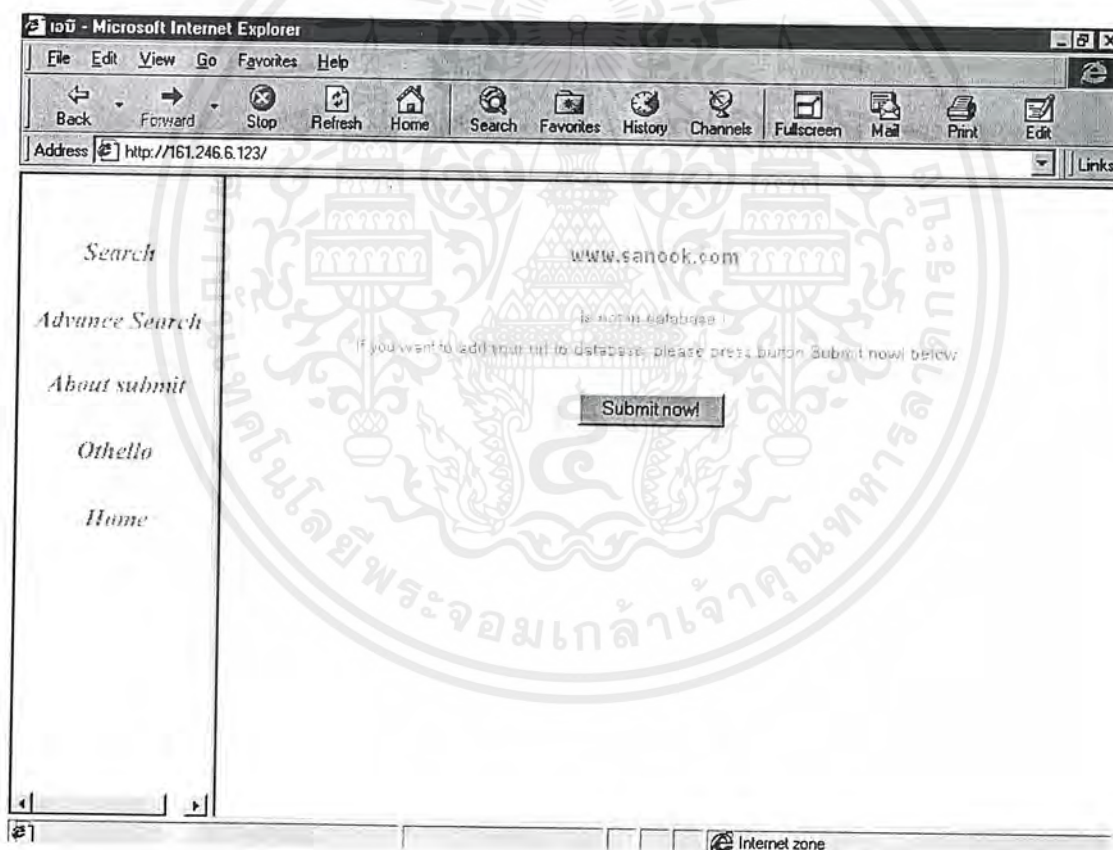


เอกสารนี้เป็นเอกสารที่สงวนรูปที่ จ-6 แสดงหน้าจอกดปุ่ม About submit อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ ง-6 จะเห็นได้ว่าหน้าจอได้แบ่งออกเป็น 3 ส่วนดังนี้

1. ส่วนที่ทำหน้าที่ในการตรวจสอบรายชื่อเว็บไซต์ จะให้ผู้ใส่ชื่อเว็บไซต์ที่ต้องการตรวจสอบว่ามีในฐานข้อมูลหรือยัง
2. ส่วนที่ทำหน้าที่ในการรองรับการ submit ชื่อเว็บไซต์จากผู้ใส่ จะให้ผู้ใส่ชื่อเว็บไซต์ที่ต้องการ submit เมื่อเว็บไซต์นั้นไม่มีอยู่ในฐานข้อมูล
3. ส่วนที่ทำหน้าที่ในการรองรับการ submit ชื่อเว็บไซต์จากผู้ใส่เมื่อเว็บไซต์นั้นมีการเปลี่ยนแปลง จะให้ผู้ใส่ชื่อเว็บไซต์ที่ต้องการ submit เมื่อเว็บไซต์นั้นมีอยู่ในฐานข้อมูลในฐานข้อมูลแล้วและเว็บไซต์นั้นมีการเปลี่ยนแปลง

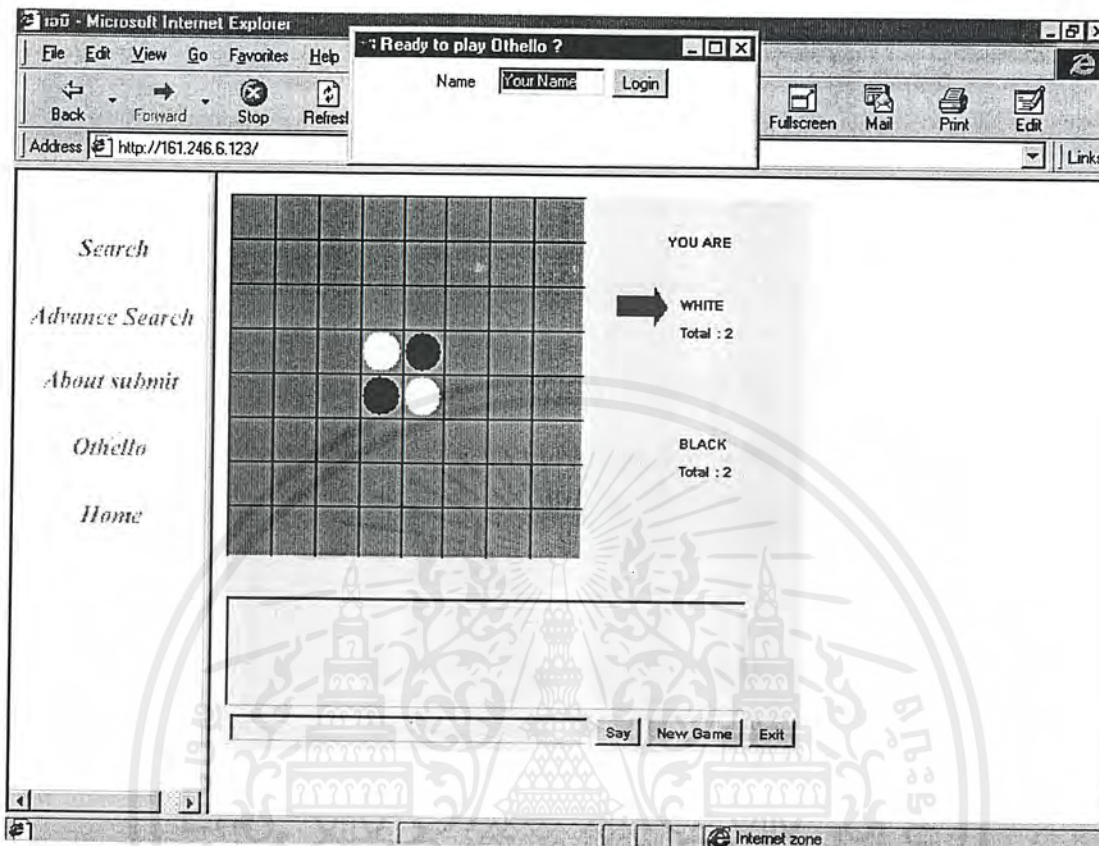
ตัวอย่างเมื่อผู้ใส่ชื่อเว็บไซต์ [www.sanook.com](http://www.sanook.com) ในการตรวจสอบรายชื่อเว็บไซต์แล้วจะปรากฏหน้าจอดังรูปที่ ง-7 ซึ่งจากรูปจะเห็นว่าไม่มีชื่อเว็บไซต์นี้ในฐานข้อมูลดังนั้นถ้าต้องการ submit ก็ให้คลิกปุ่ม submit now!



รูปที่ ง-7 ผลจากการตรวจสอบรายชื่อเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนูต่อไปคือเมนู Othello ซึ่งเมื่อกดจะปรากฏหน้าจอดังรูปที่ ง-8



รูปที่ ง-8 แสดงหน้าจอเมื่อกดปุ่ม Othello

จากรูปจะเห็นว่า มีช่องให้ใส่ชื่อผู้เล่นและเมื่อกด login แล้วก็เข้าสู่โหมดการเล่นเกมที่ ระหว่างการเล่นจะสามารถส่งข้อความไปติดต่อกับผู้เล่นอีกฝ่ายได้โดยกรอกข้อความลงใน textbox ข้างล่าง แล้วกดปุ่ม say หรือกด enter แล้วข้อความจะไปปรากฏในช่องแสดงข้อความที่อยู่ด้านล่าง กระดาน และเกมจะดำเนินไปเรื่อย ๆ จนกว่าผู้เล่นฝ่ายใดฝ่ายหนึ่งจะออกไปจากเกมหรือวางหมากจนเต็ม กระดานแล้ว

ขณะที่ทำการเล่นอยู่นั้นจะมีสถานะที่แสดงว่าผู้เล่นว่าฝ่ายใดเป็นสีอะไร และจะมีลูกศรสีแดงชี้ เพื่อบอกว่าขณะนี้ เป็นเทิร์นของใคร และเมื่อเล่นจนจบเกมแล้วจะมีข้อความบอกว่าฝ่ายใดเป็นผู้ชนะ

สุดท้ายเมื่อผู้เล่นต้องการจะออกหรือเล่นเกมใหม่ให้คลิกที่ปุ่ม Exit และ New Game ตามลำดับ

## ภาคผนวก จ

## Source Code

ในส่วนของเซิร์ฟเวอร์เอนจินมีคลาสต่าง ๆ ดังนี้

- คลาส Basic\_search เป็น Java Servlet ซึ่งจะรองรับการค้นหาชื่อเว็บเพจให้ตรงกับผู้ใช้  
ต้องการ โดยจะเป็นการค้นหาแบบธรรมดา
- คลาส Advance\_search เป็น Java Servlet ซึ่งจะรองรับการค้นหาชื่อเว็บเพจให้ตรงกับผู้ใช้  
ต้องการ โดยจะเป็นการค้นหาโดยใช้เครื่องหมาย
- คลาส checkwebsite เป็น Java Servlet ซึ่งจะรองรับการตรวจสอบชื่อเว็บไซต์พื้นฐานข้อมูล
- คลาส addwebsite เป็น Java Servlet ซึ่งจะรองรับการขั้บมิทชื่อเว็บไซต์จากผู้ใ้
- คลาส Control เป็น Java Application จะเรียกคลาส Frame1 ขึ้นมา
- คลาส Frame1 เป็น JFrame ซึ่งจะเป็นการกำหนดค่าต่าง ๆ ให้กับคลาส Spider
- คลาส Spider เป็นการท่องไปตามเว็บเพจต่าง ๆ เพื่อเก็บคีย์เวิร์ดและค่าความสำคัญ

ในส่วนของเกมออนไลน์ประกอบไปด้วยคลาสต่าง ๆ ดังนี้

- คลาส GameBoard เป็นคลาสที่ใช้เก็บ array ของหมากกระดาน
- คลาส GameClientThread เป็นคลาสที่รับการเชื่อมต่อมาจากผู้เล่นแต่ละคน
- คลาส GameGroup เป็นกลุ่มของผู้เล่นที่แข่งขันกันขณะนั้น
- คลาส GamePiece เป็นคลาสที่เก็บคุณสมบัติของหมากต่าง ๆ
- คลาส GameServer เป็นคลาสที่ใช้รันตัวเกมขึ้นมา
- คลาส GameServerThread เป็นเธรดที่ใช้ควบคุมการทำงานของเซิร์ฟเวอร์
- คลาส NetOthello เป็นคลาสที่ควบคุมกฎกติกาต่าง ๆ ของเกมในฝั่งไคลเอนต์
- คลาส PromptFrame เป็นเฟรมที่รับชื่อของผู้เล่น

### รหัส Basic\_search

```

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.lang.*;

public class Basic_search extends HttpServlet {
 String url;
 String query;

 public void init(ServletConfig config)
 throws ServletException {
 super.init(config);
 url="jdbc:inetdae:161.246.6.123";
 query = "SELECT URL,COST FROM SEARCH "+
 " WHERE KEYWORD = ?";
 try { Class.forName("com.inet.tds.TdsDriver");
 }
 catch(java.lang.ClassNotFoundException e) {
 System.err.print("ClassNotFoundException: ");
 System.err.println(e.getMessage());
 }
 }

 public void doPost(HttpServletRequest req, HttpServletResponse res)
 throws ServletException,IOException {
 Connection con;
 PreparedStatement pre;
 ResultSet rs[];
 res.setContentType("text/html");
 PrintWriter toClient = res.getWriter();

 Enumeration values = req.getParameterNames();
 toClient.println("<!DOCTYPE HTML PUBLIC \"-//IETF//DTD HTML//EN\">");

 toClient.println("<html>");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

toClient.println("<head><title>Thank you!</title></head><body>");
String value="";
String name="";
int page=1;
int e=0;
String tvalue="";
String nvalue="";
while(values.hasMoreElements()){
 name=(String)values.nextElement();
 if(name.compareTo("key")==0)
 value = req.getParameterValues(name)[0];
 if(name.compareTo("current")==0){
 nvalue=req.getParameterValues(name)[0];
 page=Integer.parseInt(nvalue);
 }
 if(name.compareTo("action")==0){
 tvalue=req.getParameterValues(name)[0];
 }
}
if(tvalue.compareTo("next")==0){
 e=page*10;
 ++page;
}
if(tvalue.compareTo("prev")==0){
 e=(page-2)*10;
 --page;
}

StringTokenizer st=new StringTokenizer(value);
while(values.hasMoreElements())
 name=(String)values.nextElement();

String t[];
int newnum=st.countTokens();
t = new String[newnum];
newnum=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int num=1;
try {
con=DriverManager.getConnection(url,"general","search");
pre=con.prepareStatement("SELECT COUNT(*) FROM STOPWORD "+
" WHERE KEYWORD = ?");
ResultSet test;
while(st.hasMoreTokens())
{
t[newnum]=st.nextToken();
pre.setString(1,t[newnum]);
test=pre.executeQuery();
test.next();
if(test.getInt(1)!=0){
--newnum;
}
++newnum;
++num;
test.close();
}
pre.close();
String word[];
word = new String[newnum];
for(int aa=0;aa<newnum;++aa)
word[aa]=t[aa];
if(newnum>0){
pre= con.prepareStatement(query);
rs = new ResultSet[newnum];
int maxrs0=0;
for(int a=0;a<newnum;++a){
pre.setString(1,word[a]);
rs[a]=pre.executeQuery();
while (rs[a].next()) {
if(a==0)
++maxrs0;
}
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
}
String temps="";
int tempi=0;
String storeurl[];
int storecost[];
String tempurl="";
int tempcost=0;
int findurl=0;
int tempfindurl;
storeurl=new String[maxrs0]; //number max of rs[0]
storecost=new int[maxrs0]; //number max of rs[0]
rs[0].beforeFirst();
while(rs[0].next()){
temps=rs[0].getString(1);
tempi=rs[0].getInt(2);
boolean nextto=true;
int a=1;
while(a<newnum && nextto){
rs[a].beforeFirst();
nextto=false;
boolean outfind=true;
while(rs[a].next() && outfind){
iff(temps.compareTo(rs[a].getString(1))==0){
tempcost=tempcost+rs[a].getInt(2);
nextto=true;
outfind=false;
}
}
++a;
}
if(a==newnum && nextto){
storeurl[findurl]=temps;
storecost[findurl]=tempcost;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

++findurl;
}
}
for(int j=0;j<newnum;++j)
 rs[j].close();
pre.close();
tempfindurl=findurl;
if(findurl>0){
while(findurl != 1){
 for(int c=0;c<(findurl-1);++c){
 if(storecost[c+1]>storecost[c]){
 tempcost=storecost[c];
 storecost[c]=storecost[c+1];
 storecost[c+1]=tempcost;
 tempurl=storeurl[c];
 storeurl[c]=storeurl[c+1];
 storeurl[c+1]=tempurl;
 }
 }
 --findurl;
}
findurl=tempfindurl;
}
toClient.println("
Find key \" "+
value+" \" in "+ findurl+" webpage");
toClient.println("<hr align=\"left\" width=\"50%\" ");
if(findurl>10)
toClient.println("
Page : "+page+"
");
PreparedStatement pre1;
pre1 = con.prepareStatement("SELECT * FROM DETAIL WHERE URL = ?");
ResultSet rs1;
findurl=tempfindurl;
boolean first=true;
while(e<findurl && (e%10!=0 || first)){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 pre1.setString(1,storeurl[e]);
 rs1=pre1.executeQuery();
 while (rs1.next()) {
 toClient.println("
"
+rs1.getString(1)+"");
 toClient.println(" "+rs1.getString(2)+"
");
 }
 rs1.close();
 ++e;

 if(e<findurl && e%10==0 && first==false){
toClient.println("<form action=http://161.246.6.123/servlet/Basic_search method=post>");
toClient.println("<center><input type=hidden name=key value="+""+value+"\""+"" >");
toClient.println("<input type=hidden name=current value="+page+">
");
 if(page>1){
 toClient.println("<input type=submit name=action "+" value=\"prev\" style=\"{background-
color:#539ae2;color:#546270}\">");
 }
 toClient.println("<input type=submit name=action "+" value=\"next\" style=\"{background-
color:#539ae2; "+" color:#546270}\"></center></form>");
 }
 if(e==findurl && page>1){
toClient.println("<form action=http://161.246.6.123/servlet/Basic_search "+" method=post>");
toClient.println("<center><input type=hidden name=key "+" value="+""+value+"\""+"" >");
toClient.println("<input type=hidden name=current value="+page+">");
toClient.println("
<input type=submit name=action "+" value=\"prev\" style=\"{background-
color:#539ae2; color:#546270}\">"+ "</center></form>");
 }
 first=false;
 }
 pre1.close();
}

con.close();

}

catch(SQLException ex)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 { System.err.println("SQLException: " + ex.getMessage());
 }
 toClient.println("</body></html>");
 toClient.close();
}
}

```

### คลาส Advance\_search

```

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.lang.*;

public class Advance_search extends HttpServlet {
 String url;
 String query;
 public void init(ServletConfig config)
 throws ServletException{
 super.init(config);
 url="jdbc:inetdae:161.246.6.123";
 query = "SELECT URL,COST FROM SEARCH WHERE KEYWORD = ?";
 try{
 Class.forName("com.inet.tds.TdsDriver");
 }
 catch(java.lang.ClassNotFoundException e) {
 System.err.print("ClassNotFoundException: ");
 System.err.println(e.getMessage());
 }
 }

 public void doPost(HttpServletRequest req, HttpServletResponse res)
 throws ServletException,IOException {
 Connection con;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PreparedStatement pre;

ResultSet rs[];

res.setContentType("text/html");

PrintWriter toClient = res.getWriter();

Enumeration values = req.getParameterNames();

toClient.println("<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">");
toClient.println("<html>");
toClient.println("<head><title>Thank you!</title></head><body>");

 String value="";
 String name="";
 int page=1;
 int e=0;
 String tvalue="";
 String nvalue="";
 while(values.hasMoreElements()){
 name=(String)values.nextElement();
 if(name.compareTo("key")==0)
 value = req.getParameterValues(name)[0];
 if(name.compareTo("current")==0){
 nvalue=req.getParameterValues(name)[0];
 page=Integer.parseInt(nvalue);
 }
 if(name.compareTo("action")==0){
 tvalue=req.getParameterValues(name)[0];
 }
 }

 if(tvalue.compareTo("next")==0){
 e=page*10;
 ++page;
 }

 if(tvalue.compareTo("prev")==0){
 e=(page-2)*10;
 --page;
 }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

nvalue="";
int newlength=0;
for(int n=0;n<value.length();++n){
 if(value.charAt(n)!=' '){
 nvalue=nvalue+value.charAt(n);
 ++newlength;
 }
}
nvalue=nvalue+' ';
++newlength;
String tempa[];
tempa = new String[newlength];
int numa=-1;
for(int zzz=0;zzz<nvalue.length();++zzz)
tempa[zzz]="";
int numstate=0;
int numstack=0;
boolean syntax=true;
int s=0;
int maxbrace=0;
while(s<nvalue.length() && syntax){
switch(numstate){
 case 0 : switch(nvalue.charAt(s)){
 case '(' : numstate=1; ++numstack; ++maxbrace; break;
 case '|' : syntax=false; break;
 case '&' : syntax=false; break;
 case ')' : syntax=false; break;
 case ' ' : syntax=false; break;
 default : numstate=2; ++numa;
tempa[numa]=tempa[numa]+nvalue.charAt(s);
 }
 break;
 case 1 : switch(nvalue.charAt(s)){
 case '(' : numstate=1; ++numstack; ++maxbrace; break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 case '|': syntax=false;break;
 case '&': syntax=false;break;
 case ')': syntax=false;break;
 case ' ': syntax=false;break;
 default : numstate=2;++numa;
 tempa[numa]=tempa[numa]+nvalue.charAt(s);
 }
 break;
case 2 : switch(nvalue.charAt(s)){
 case '(' : syntax=false;break;
 case '|' : numstate=5;break;
 case '&': numstate=5;break;
 case ')': numstate=4;
 if(numstack>0)- numstack;else syntax=false;break;
 case ' ': if(numstack==0)syntax=true;else syntax=false;break;
 default : numstate=2;
 tempa[numa]=tempa[numa]+nvalue.charAt(s);
 }
 break;
case 4 : switch(nvalue.charAt(s)){
 case '(' : syntax=false;break;
 case '|' : numstate=5;break;
 case '&': numstate=5;break;
 case ')': if(numstack>0)- numstack;else syntax=false;break;
 case ' ': if(numstack==0)syntax=true;else syntax=false;break;
 default : syntax=false;
 }
 break;
case 5 : switch(nvalue.charAt(s)){
 case '(' : numstate=1;++numstack;++maxbrace;break;
 case '|' : syntax=false;break;
 case '&': syntax=false;break;
 case ')': syntax=false;break;
 case ' ': syntax=false;break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

}
String t1[][];
t1=new String[maxbrace+1][maxsite];
int t2[][];
t2=new int[maxbrace+1][maxsite];
boolean t3[][];
t3=new boolean[maxbrace+1][maxsite];
for(int abc1=0;abc1<=maxbrace;++abc1)
 for(int abc=0;abc<maxsite;++abc){
 t1[abc1][abc]="";
 t2[abc1][abc]=0;
 t3[abc1][abc]=false;
 }
int n=0;
int tempbrace;
for(int i=maxbrace;i>=0;--i){
int c=0;
int b=0;
int st=0;
while(c!=i){
if(nvalue2.charAt(b)=='(')++c;
if(nvalue2.charAt(b)=='s')++st;
++b;
}
tempbrace=i;
boolean first1=true;
boolean outdo=false;
do{
if(first1){
if(nvalue2.charAt(b)=='(' || first1==false){
 first1=false;
 ++tempbrace;
 int braceinbrace=0;
 int tb=b+1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int tempbrace2=0;
while(nvalue2.charAt(tb)!='' || braceinbrace!=0){
 if(nvalue2.charAt(tb)=='s')++st;
 if(nvalue2.charAt(tb)=='('){
 ++braceinbrace;
 ++tempbrace2;
 }
 if(nvalue2.charAt(tb)=='')
 --braceinbrace;
 ++tb;
}
b=tb;
n=0;
for(int ab1=0;ab1<maxsite;++ab1){
 if(t3[tempbrace][ab1]){
 t1[i][n]=t1[tempbrace][ab1];
 t2[i][n]=t2[tempbrace][ab1];
 t3[i][n]=true;
 ++n;
 }
}
tempbrace=tempbrace+tempbrace2;
}
else if(nvalue2.charAt(b)=='s' && first1){
 first1=false;
 rs[st].beforeFirst();
 n=0;
 while(rs[st].next()){
 t1[i][n]=rs[st].getString(1);
 t2[i][n]=rs[st].getInt(2);
 t3[i][n]=true;
 ++n;
 }
 ++st;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
} //close of if(first1)
if((b+1)!=nvalue2.length() && nvalue2.charAt(b+1)!='')
switch(nvalue2.charAt(b+2)){
 case '(': ++tempbrace;
 if(nvalue2.charAt(b+1)=='&'){
 for(int m=0;m<n;++m){
 if(t3[i][m]){
 int p=0;
 boolean outfind=true;
 while(p<maxsite && outfind){
 if(t3[tempbrace][p])
 if(t1[i][m].compareTo(t1[tempbrace][p])==0){
 t2[i][m]=t2[i][m]+t2[tempbrace][p];
 outfind=false;
 }
 ++p;
 }
 if(outfind){
 t3[i][m]=false;
 t2[i][m]=0;
 }
 } //close if
 } //close for
 } //close of &
 if(nvalue2.charAt(b+1)=='|'){
 for(int p=0;p<maxsite;++p){
 boolean outfind=true;
 int m=0;
 while(m<n && outfind){
 if(t1[i][m].compareTo(t1[tempbrace][p])==0){
 if(t3[i][m] || t3[tempbrace][p]){
 t2[i][m]=t2[i][m]+t2[tempbrace][p];
 t3[i][m]=true;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 outfind=false;
 }

 ++m;
 }
 if(outfind)
 if(t3[tempbrace][p]){
 t1[i][n]=t1[tempbrace][p];
 t2[i][n]=t2[tempbrace][p];
 t3[i][n]=true;
 ++n;
 }
 }
} //close of |
b+=3;
int braceinbrace=0;
int tempbrace2=0;
while(nvalue2.charAt(b)!='' || braceinbrace!=0){
 if(nvalue2.charAt(b)=='s')++st;
 if(nvalue2.charAt(b)=='('){
 ++braceinbrace;
 ++tempbrace2;
 }
 if(nvalue2.charAt(b)=='')- braceinbrace;
 ++b;
}
tempbrace=tempbrace+tempbrace2;
break;
case 's' : if(nvalue2.charAt(b+1)=='&'){
 for(int m=0;m<n;++m){
 if(t3[i][m]){
 rs[st].beforeFirst();
 boolean outfind=true;
 while(rs[st].next() && outfind){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 if(t1[i][m].compareTo(rs[st].getString(1))==0){
 t2[i][m]=t2[i][m]+rs[st].getInt(2);
 outfind=false;
 }
 }
 if(outfind){
 t3[i][m]=false;
 t2[i][m]=0;
 }
} //close if t3 is true
} //close for
} //close of &
if(nvalue2.charAt(b+1)=='|'){
 rs[st].beforeFirst();
 while(rs[st].next()){
 boolean outfind=true;
 int m=0;
 while(m<n && outfind){
 if(t1[i][m].compareTo(rs[st].getString(1))==0){
 t2[i][m]=t2[i][m]+rs[st].getInt(2);
 t3[i][m]=true;
 outfind=false;
 }
 ++m;
 }
 if(outfind){
 t1[i][n]=rs[st].getString(1);
 t2[i][n]=rs[st].getInt(2);
 t3[i][n]=true;
 ++n;
 }
 }
} //close of '|'
b+=2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 ++st;
 break;
 }
 if((b+1)==nvalue2.length())outdo=true;
 else if(nvalue2.charAt(b+1)=='')outdo=true;
 }
 while(outdo==false);
}
String storeurl[];
int storecost[];
int findurl=0;
storeurl = new String[maxsite];
storecost = new int[maxsite];
for(int bb=maxbrace;bb>=0;--bb)
for(int cc=0;cc<maxsite;++cc)
if(t3[bb][cc]){
if(bb==0){
 storeurl[findurl]=t1[0][cc];
 storecost[findurl]=t2[0][cc];
 ++findurl;
}
}
for(int j=0;j<numa;++j)rs[j].close();
pre.close();
int tempfindurl=findurl;
String tempurl="";
int tempcost=0;
if(findurl>0){
while(findurl != 1){
 for(int c=0;c<(findurl-1);++c){
 if(storecost[c+1]>storecost[c]){
 tempcost=storecost[c];
 storecost[c]=storecost[c+1];
 storecost[c+1]=tempcost;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 tempurl=storeurl[c];
 storeurl[c]=storeurl[c+1];
 storeurl[c+1]=tempurl;
 }
}
--findurl;
}
findurl=tempfindurl;
}
toClient.println("
Find key \" "+
 value+" \" in "+ findurl+" webpage");
toClient.println("<hr align=\"left\" width=\"50%\" ");
if(findurl>10)toClient.println("
Page : "+page+"
");
PreparedStatement pre1;
pre1 = con.prepareStatement("SELECT * FROM DETAIL WHERE URL = ?");
ResultSet rs1;
findurl=tempfindurl;
boolean first=true;
while(e<findurl && (e%10!=0 || first)){
 pre1.setString(1,storeurl[e]);
 rs1=pre1.executeQuery();
 while (rs1.next()) {
 toClient.println("
"+rs1.getString(1)+"");
 toClient.println(" "+rs1.getString(2)+"
");
 }
 rs1.close();
 ++e;
}
if(e<findurl && e%10==0 && first==false){
 toClient.println("<form action=http://161.246.6.123/servlet/Advance_search "+
 " method=post>");
 toClient.println("<center><input type=hidden name=key "+
 " value=\""+value+"\"> ");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 toClient.println("<input type=hidden name=current value="+
 page+">
");
 if(page>1){
 toClient.println("<input type=submit name=action "+
 " value=\"prev\" style=\"{background-color:#539ae2;color:#546270}\">");
 }
 toClient.println("<input type=submit name=action "+
 value=\"next\" style=\"{background-color:#539ae2; "+ " color:#546270}\"></center></form>");
 }
 if(e= =findurl && page>1){
 toClient.println("<form action=http://161.246.6.123/servlet/Advance_search "+" method=post>");
 toClient.println("<center><input type=hidden name=key "+" value=\"+\"\""+value+\"\""+ " >");
 toClient.println("<input type=hidden name=current value="+page+">");
 toClient.println("
<input type=submit name=action "+
 " value=\"prev\" style=\"{background-color:#539ae2; color:#546270}\">"+</center></form>");
 }
 first=false;
 }
 pre1.close();
 con.close();
 }
 catch(SQLException ex) {
 System.err.println("SQLException: " + ex.getMessage());
 }
 }
 toClient.println("</body></html>");
 toClient.close();
}
}

```

#### คลาส checkwebsite

```

import java.io.*;
import java.util.*;
import javax.servlet.*;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import javax.servlet.http.*;
import java.sql.*;
import java.lang.*;

public class checkwebsite extends HttpServlet
 implements SingleThreadModel{
 String url;

public void init(ServletConfig config)
 throws ServletException {
 super.init(config);
 url="jdbc:inetdae:161.246.6.123";
 try {
 Class.forName("com.inet.tds.TdsDriver");
 }
catch(java.lang.ClassNotFoundException e) {
 System.err.print("ClassNotFoundException: ");
 System.err.println(e.getMessage());
 }
}

public void doPost(HttpServletRequest req, HttpServletResponse res)
 throws ServletException,IOException {
 Connection con;
 PreparedStatement pre;
 ResultSet rs;
 String query = "SELECT COUNT(*) FROM DETAIL WHERE BASEURL = ?";
 res.setContentType("text/html");
 PrintWriter toClient = res.getWriter();
 Enumeration values = req.getParameterNames();
 toClient.println("<html>");
 toClient.println("<title>Thank you!</title>");
 toClient.println("<body text=#79BCFF bgcolor=#000000>");
 String value="";
 String name = (String)values.nextElement();
 if(name.compareTo("key")==0)value = req.getParameterValues(name)[0];
 toClient.println("
");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

try {
con=DriverManager.getConnection(url,"general","search");
pre= con.prepareStatement(query);
pre.setString(1,value);
rs=pre.executeQuery();
boolean have=false;
if(rs.next()){
if(rs.getInt(1)>=1)
 have=true;
}
if(have){
toClient.println("<center><h2>"+value+"</h2> <h3> is already in database !</h3>");
toClient.println("<h3>If you have changed website and want to "+
 "reindex, please press button Reindex now!</h3>");
toClient.println("<form action=http://161.246.6.123/servlet/addwebsite method=POST>");
toClient.println("<input type=hidden name=key value='"+value+"' >");
toClient.println("<input type=submit value='"+"Reindex now!"
 +" style='"+" {background-color:79bcff;"+ "color:#000000}"+">");
toClient.println("</form></center>");
}
else {
toClient.println("<center><h2>"+value+"</h2> <h3> is not in database !
");
toClient.println("If you want to add your url to "+
 "database, please press button Submit now! below</h3>");
toClient.println("<form action=http://161.246.6.123/servlet/addwebsite method=POST>");
toClient.println("<input type=hidden name=key value='"+value+"' >");
toClient.println("<input type=submit value='"+"Submit now!"
 +" style='"+" {background-color:79bcff;"+ "color:#000000}"+">");
toClient.println("</form></center>");
}
rs.close();
pre.close();
con.close();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

catch(SQLException ex) {
 System.err.println("SQLException: " + ex.getMessage());
}

toClient.println("</body></html>");
toClient.close();
}
}

```

### กติกาส์ addwebsite

```

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.lang.*;

public class addwebsite extends HttpServlet
 implements SingleThreadModel{
public void init(ServletConfig config)
 throws ServletException{
 super.init(config);
}

public void doPost(HttpServletRequest req, HttpServletResponse res)
 throws ServletException,IOException {
 res.setContentType("text/html");
 PrintWriter toClient = res.getWriter();
 Enumeration values = req.getParameterNames();
 toClient.println("<html>");
 toClient.println("<title>Thank you!</title><body text=#79BCFF bgcolor=#000000>");
 String value="none";
 while(values.hasMoreElements()){
 String name = (String)values.nextElement();
 if(name.compareTo("key")==0){
 value = req.getParameterValues(name)[0];
 }
 }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
String arrayurl[];
FileReader file1;
File myfile;
FileWriter fw;
BufferedWriter bw;
try{
file1=new FileReader("c:/class/fileurl.txt");
boolean mark=false;
int i=0,tempi=10;
int numurl=0;
while(i!=-1){
i=file1.read();
if(i==13){
if(tempi!=10)mark=true;
}
else if(i==10 && mark){
++numurl;
mark=false;
}
if(i==10) if(tempi!=10)++numurl;
tempi=i;
}
arrayurl=new String[numurl];
for(int b=0;b<numurl;++b)arrayurl[b]="";
file1=new FileReader("c:/class/fileurl.txt");
mark=false;
i=0;
tempi=10;
int a=0;
char in;
while(i!=-1){
i=file1.read();
if(i!=32){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(i==13){
if(tempi!=10)mark=true;
}
else if(i==10 && mark){
 ++a;
 mark=false;
}
else if(i==1){
 if(tempi!=10)++a;}
 else if(i!=13 && i!=10 && i!=1){
 in=(char)i;
 arrayurl[a]=arrayurl[a]+in;
 }
} //close if(i!=32)
temp=i;
}
boolean addurl=true;
int b=0;
myfile=new File("c:/class/fileurl.txt");
fw=new FileWriter(myfile);
bw=new BufferedWriter(fw);
while(b<numurl){
 if(arrayurl[b].compareTo(value)==0)addurl=false;
 bw.write(arrayurl[b]);
 bw.write(13);
 bw.write(10);//write 13 and 10 for enter
 bw.flush();
 ++b;
}
if(addurl){
 bw.write(value);
 bw.write(13);
 bw.write(10);
 bw.flush();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 bw.close();
}
catch(IOException ioe){
 System.out.println("IO error:"+ioe);
}

toClient.println("

");
toClient.println("<center>");
toClient.println("Thank you");
toClient.println("
My server will send spider program to ");
toClient.println("" +value);
toClient.println("");
toClient.println(" on next update");
toClient.println("for finding keyword and linkpage.");
toClient.println("</center>");
toClient.println("</body></html>");
toClient.close();
}
}

```

### คลาส Control

```

import javax.swing.UIManager;
import JFrame1;
public class control {
 boolean packFrame = false;
 public control() {
 JFrame1 frame = new JFrame1();
 if (packFrame)frame.pack();
 else frame.validate();
 frame.setVisible(true);
 }
 public static void main(String[] args) {
 try {
 UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 catch(Exception e) {
 }
 new control();
}
}

```

### คลาส Frame1

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.lang.*;
import Spider;

public class Frame1 extends JFrame {
 JLabel jLabel1 = new JLabel();
 JLabel jLabel2 = new JLabel();
 JFileChooser chooser=new JFileChooser();
 JButton button=new JButton("Start URL");
 JTextField jTextField2 = new JTextField();
 JButton jButton1 = new JButton();
 JButton jButton2 = new JButton();
 private int limiturl;
 File file;
 public Frame1(){
 enableEvents(AWTEvent.WINDOW_EVENT_MASK);
 try{
 jbInit();
 }
 catch(Exception e) {
 e.printStackTrace();
 }
 }
 private void jbInit() throws Exception {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

jLabel1.setFont(new java.awt.Font("Dialog", 0, 20));
jLabel1.setForeground(Color.cyan);
jLabel1.setHorizontalAlignment(SwingConstants.CENTER);
jLabel1.setText("File start URL");
jLabel1.setBounds(new Rectangle(7, 39, 145, 41));
this.getContentPane().setLayout(null);
this.getContentPane().setBackground(new java.awt.Color(183, 143, 207));
this.setSize(new Dimension(384, 268));
this.setTitle("Frame Title");
jLabel2.setFont(new java.awt.Font("Dialog", 0, 20));
jLabel2.setForeground(Color.cyan);
jLabel2.setHorizontalAlignment(SwingConstants.CENTER);
jLabel2.setText("Limit");
jLabel2.setBounds(new Rectangle(12, 106, 88, 41));
jTextField2.setFont(new java.awt.Font("SansSerif", 0, 20));
jTextField2.setText("0");
jTextField2.setHorizontalAlignment(SwingConstants.RIGHT);
jTextField2.setBounds(new Rectangle(163, 109, 182, 40));
button.setBackground(Color.cyan);
button.setFont(new java.awt.Font("Dialog",0,16));
button.setText("Choose File");
button.setBounds(new Rectangle(163, 39, 182, 40));
button.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent e) {
int state=chooser.showOpenDialog(null);
file=chooser.getSelectedFile();
if(file!=null && state==JFileChooser.APPROVE_OPTION){
JOptionPane.showMessageDialog(null,file.getPath());
}
else if(state==JFileChooser.CANCEL_OPTION){
JOptionPane.showMessageDialog(null,"Canceled");
}
}
});

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

jButton1.setBackground(Color.cyan);
jButton1.setFont(new java.awt.Font("Dialog", 0, 16));
jButton1.setText("RUN");
jButton1.setBounds(new Rectangle(105, 183, 87, 34));
jButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(ActionEvent e) {
 jButton1_actionPerformed(e);
}
});
jButton2.setBackground(Color.cyan);
jButton2.setFont(new java.awt.Font("Dialog", 0, 16));
jButton2.setText("Cancel");
jButton2.setBounds(new Rectangle(213, 183, 83, 35));
jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(ActionEvent e) {
 jButton2_actionPerformed(e);
}
});
this.getContentPane().add(button, null);
this.getContentPane().add(jTextField2, null);
this.getContentPane().add(jButton2, null);
this.getContentPane().add(jButton1, null);
this.getContentPane().add(jLabel2, null);
this.getContentPane().add(jLabel1, null);
}
protected void processWindowEvent(WindowEvent e) {
 super.processWindowEvent(e);
 if(e.getID() == WindowEvent.WINDOW_CLOSING) {
 System.exit(0);
 }
}
void jButton1_actionPerformed(ActionEvent e) {
 String temp=jTextField2.getText();
 limitur1=Integer.parseInt(temp);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 Spider spider=new Spider();
 spider.setfileurl(file);
 spider.setlimiturl(limiturl);
 try{
 spider.spiderstart();
 }
 catch(IOException ioe)
 {}
 }
 void jButton2_actionPerformed(ActionEvent e) {
 dispose();
 }
}

```

#### คลาส Spider

```

import java.net.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.lang.*;
class Spider{
 FileReader file1;
 File myfile;
 File resulturlfile;
 String linkurl[];
 int numlinkurl;
 File tempfile;
 String arrayurl[];
 int currenturl;
 int numurl;
 int limiturl;
 Connection con;
 String url;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

File namelisturl;
public Spider(){
myfile=new File("temppage");
tempfile=new File("tempfile");
resulturlfile=new File("resulturlfile");
currenturl=0;
url="jdbc:inetdae:161.246.6.123";
try {
 Class.forName("com.inet.tds.TdsDriver");
}
catch(java.lang.ClassNotFoundException e){
 System.err.print("ClassNotFoundException: ");
 System.err.println(e.getMessage());
}
}
public void spiderstart()
throws IOException {
try{
con=DriverManager.getConnection(url,"general","search");
}
catch(SQLException ex) {
 System.err.println("SQLException: " + ex.getMessage());
}
try
{
 file1=new FileReader(namelisturl);
 boolean mark=false;
 int i=0,tempi=10;
 numurl=0;
 while(i!=1){
i=file1.read();
if(i==13){
 if(tempi!=10)mark=true;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 else if(i==10 && mark){
 ++numurl;
 mark=false;
 }

 if(i==1)if(tempi!=10)++numurl;
 tempi=i;
}

limiturl=limiturl+numurl;
System.out.println("All Limit = "+limiturl);
arrayurl=new String[limiturl];
for(int b=0;b<limiturl;++b)arrayurl[b]="";
file1=new FileReader(namelisturl);
mark=false;
i=0;
tempi=10;
int a=0;
char in;
while(i!=1){
 i=file1.read();
 if(i!=32){
 if(i==13){
 if(tempi!=10)mark=true;
 }
 else if(i==10 && mark){
 ++a;
 mark=false;
 }
 else if(i==1){
 if(tempi!=10)++a;
 }
 else if(i!=13 && i!=10 && i!=1){
 in=(char)i;
 arrayurl[a]=arrayurl[a]+in;
 }
 }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 } //close if(i!=32)
 tempi=i;
 }
 FileWriter urlwriter;
 BufferedWriter urlbufferwriter;
 urlwriter=new FileWriter(resulturlfile);
 urlbufferwriter=new BufferedWriter(urlwriter);
 while(currenturl<numurl){
 boolean findcon=true;
 linkurl=new String[10000];
 for(int ii=0;ii<10000;++ii)linkurl[ii]="";
 deleteurl(arrayurl[currenturl]);
 numlinkurl=1;//start for each url
 linkurl[0]=" ";
 finddataurl(arrayurl[currenturl]);
 urlbufferwriter.write("Number "+currenturl+" : "+arrayurl[currenturl]+" : have link = "+
 numlinkurl);
 urlbufferwriter.newLine();
 urlbufferwriter.flush();
 ++currenturl;
 }
 urlbufferwriter.close();
}
catch(IOException ioe){
 System.out.println("IO error:"+ioe);
}

try{
 con.close();
}
catch(SQLException ex) {
 System.err.println("SQLException: " + ex.getMessage());
}

System.out.println();
System.out.println("*****Update finish*****");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 public boolean urlindbms(String tempnameurl){
 boolean findurlcon=true;
 String quer="SELECT COUNT(*) FROM DETAIL WHERE BASEURL=" +
 tempnameurl+"";
 Statement st;
 ResultSet rst;
 try{
 st=con.createStatement();
 rst=st.executeQuery(quer);
 if(rst.next()){
 if(rst.getInt(1)!=0)
 findurlcon=false;
 }
 rst.close();
 st.close();
 }
 catch(SQLException ex) {
 System.err.println("SQLException: " + ex.getMessage());
 }
 return findurlcon;
 }
 public void deleteurl(String deletename){
 String querydel="";
 querydel="SELECT URL FROM DETAIL WHERE BASEURL = '"+deletename+"";
 System.out.println("Set for delete "+deletename);
 System.out.println(querydel);
 Statement stmt,stmt1;
 ResultSet result;
 try{
 stmt=con.createStatement();
 System.out.println("executing");
 result=stmt.executeQuery(querydel);
 System.out.println("passexecute");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(result.next())
{
System.out.println("have result");
String newquery="DELETE FROM SEARCH WHERE URL='"+result.getString(1)+"'";
while(result.next()){
newquery=newquery+" OR URL= '"+result.getString(1)+"'";
}
stmt1=con.createStatement();
System.out.println(newquery);
System.out.println("deleting");
stmt1.executeUpdate(newquery);
System.out.println("delete finish in search table");
stmt1.close();
stmt.close();
querydel="DELETE FROM DETAIL WHERE BASEURL = '"+deletename+"'";
System.out.println(querydel);
System.out.println("deleting");
stmt = con.createStatement();
stmt.executeUpdate(querydel);
System.out.println("deleting "+deletename+" finish");
}
else System.out.println("No "+arrayurl[currenturl]+" in database");
result.close();
stmt.close();
}
catch(SQLException ex) {
System.err.println("SQLException: " + ex.getMessage());
}
}
public void finddataurl(String baseurl){
int count=0;
while(count<numlinkurl){
DataInputStream dis=null;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

URL u=null;
System.out.println();
System.out.println();
boolean doread;
doread=true;
 try{
 u=new URL("http://"+baseurl+"/"+linkurl[count]);
 System.out.println("Find keyword and link in page number :"+count);
 System.out.println(u);
 dis=new DataInputStream(u.openConnection().getInputStream());
 }
 catch(Exception e){
 doread=false;
 System.out.println(e);
 }
if(doread)
{
 InputStreamReader in;
 BufferedReader br;
 FileWriter fw;
 BufferedWriter bw;
 try{
 in=new InputStreamReader(dis);
 br=new BufferedReader(in);
 fw=new FileWriter(myfile);
 bw=new BufferedWriter(fw);
 String temp="";
 while((temp=br.readLine())!=null){
 bw.write(temp);
 bw.newLine();
 bw.flush();
 }
 bw.close();
 }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

catch(IOException ioe){
System.out.println("getToFile error: "+ioe);
}

 findkeyword(linkurl[count]);
 findlink(linkurl[count]);
}

 ++count;
}
}

public void findkeyword(String linkpage){
String col1[]=new String[10000];
int col2[]=new int[10000];
for(int a2=0;a2<10000;++a2){
 col1[a2]="";
 col2[a2]=0;
}
int countcol=0;
String temptitle="";
try{
FileReader fr=new FileReader(myfile);
BufferedReader br=new BufferedReader(fr);
String title[];
title=new String[3000];
int numtitle=0;
for(int a1=0;a1<3000;++a1)title[a1]="";
int i=0;
char in;
i=br.read();
int numstate=0;
boolean out=false;
while(i!=-1 && out==false){
in=(char)i;
if(i<129){
switch(numstate){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 0 : switch(i){
 case 60 : numstate=1;break;
 default : numstate=0;
 }
 break;
case 1 : switch(i){
 case 84 :
 case 116 : numstate=2;break;
 default : numstate=9;
 }
 break;
case 2 : switch(i){
 case 73 :
 case 105 : numstate=3;break;
 default : numstate=9;
 }
 break;
case 3 : switch(i){
 case 84 :
 case 116 : numstate=4;break;
 default : numstate=9;
 }
 break;
case 4 : switch(i){
 case 76 :
 case 108 : numstate=5;break;
 default : numstate=9;
 }
 break;
case 5 : switch(i){
 case 69 :
 case 101 : numstate=6;break;
 default : numstate=9;
 }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 break;
case 6 : switch(i){
 case 62 : numstate=7;break;
 case 32 : numstate=6;break;
 default : numstate=9;
 }
 break;
case 7 : switch(i){
 case 39 : numstate=8;break;
 case 10 : case 13 : case 40 : case 41 : case 46 : case 33 : case 63 : case 35 : case 36 :
 case 37 : case 38 : case 43 : case 45 : case 47 : case 59 : case 91 : case 92 : case 93 :
 case 94 : case 95 : case 61 : case 64 : case 34 :
 case 32 : numstate=8;
 break;
 case 44 : numstate=8;
 temptitle=temptitle+in;
 break;
 case 58 : numstate=8;
 temptitle=temptitle+in;
 break;
 case 60 : numstate=10;break;
 default : numstate=7;
 title[numtitle]=title[numtitle]+in;
 temptitle=temptitle+in;
 }
 break;
case 8 : switch(i){
 case 39 : numstate=8;break;
 case 10 : case 13 : case 40 : case 41 : case 46 : case 33 : case 63 : case 35 : case 36 :
 case 37 : case 38 : case 43 : case 45 : case 47 : case 59 : case 91 : case 92 : case 93 :
 case 94 : case 95 : case 61 : case 64 : case 34 : case 44 : case 58
 case 32 : numstate=8;
 temptitle=temptitle+in;
 break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 case 60 : numstate=10;break;
 default : numstate=7;
 if(title[numtitle].compareTo("")!=0)++numtitle;
 title[numtitle]=title[numtitle]+in;
 temptitle=temptitle+in;
 }
 break;
 case 9 : switch(i){
 case 62 : numstate=0;break;
 default : numstate=9;
 }
 break;
 case 10 : out=true;
 }
}
i=br.read();
}
if(out)++numtitle;
for(int a3=0;a3<numtitle;++a3){
 col1[countcol]=title[a3];
 col2[countcol]=3;
 ++countcol;
}
}

catch(IOException ioe){
 System.out.println("IO error:"+ioe);
}

try{
 FileReader fr1=new FileReader(myfile);
 BufferedReader br1=new BufferedReader(fr1);
 int nummeta=0;
 int j=0;
 char charj;
 j=br1.read();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int metastate=0;
boolean metaout=false;
boolean first=true;
boolean dest=true;
while(j!=-1 && metaout==false){
charj=(char)j;
if(j<129){
switch(metastate){
case 0 : switch(j){
case 60 : metastate=1;break;
default : metastate=0;
}
break;
case 1 : switch(j){
case 77 :
case 109 : metastate=2;break;
default : metastate=45;
}
break;
case 2 : switch(j){
case 69 :
case 101 : metastate=3;break;
default : metastate=45;
}
break;
case 3 : switch(j){
case 84 :
case 116 : metastate=4;break;
default : metastate=45;
}
break;
case 4 : switch(j){
case 65 :
case 97 : metastate=5;break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 default : metastate=45;
 }
 break;
case 5 : switch(j){
 case 32 : metastate=6;break;
 default : metastate=45;
 }
 break;
case 6 : switch(j){
 case 78 :
 case 110 : metastate=7;break;
 case 32 : metastate=6;break;
 default : metastate=45;
 }
 break;
case 7 : switch(j){
 case 65 :
 case 97 : metastate=8;break;
 default : metastate=45;
 }
 break;
case 8 : switch(j){
 case 77 :
 case 109 : metastate=9;break;
 default : metastate=45;
 }
 break;
case 9 : switch(j){
 case 69 :
 case 101 : metastate=10;break;
 default : metastate=45;
 }
 break;
case 10 : switch(j){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 case 61 : metastate=11;break;
 case 32 : metastate=10;break;
 default : metastate=45;
 }
 break;

case 11 : switch(j){
 case 34 : metastate=12;break;
 case 32 : metastate=11;break;
 default : metastate=45;
 }
 break;

case 12 : switch(j){
 case 68 :
 case 100 : metastate=13;
 dest=true;
 break;
 case 75 :
 case 107 : metastate=38;
 dest=false;
 break;
 default : metastate=45;
 }
 break;

case 13 : switch(j){
 case 69 :
 case 101 : metastate=14;break;
 default : metastate=45;
 }
 break;

case 14 : switch(j){
 case 83 :
 case 115 : metastate=15;break;
 default : metastate=45;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 break;
case 15 : switch(j){
 case 67 :
 case 99 : metastate=16;break;
 default : metastate=45;
 }
 break;
case 16 : switch(j){
 case 82 :
 case 114 : metastate=17;break;
 default : metastate=45;
 }
 break;
case 17 : switch(j){
 case 73 :
 case 105 : metastate=18;break;
 default : metastate=45;
 }
 break;
case 18 : switch(j){
 case 80 :
 case 112 : metastate=19;break;
 default : metastate=45;
 }
 break;
case 19 : switch(j){
 case 84 :
 case 116 : metastate=20;break;
 default : metastate=45;
 }
 break;
case 20 : switch(j){
 case 73 :

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 case 105 : metastate=21;break;
 default : metastate=45;
 }
 break;
case 21 : switch(j){
 case 79 :
 case 111 : metastate=22;break;
 default : metastate=45;
 }
 break;
case 22 : switch(j){
 case 78 :
 case 110 : metastate=23;break;
 default : metastate=45;
 }
 break;
case 23 : switch(j){
 case 34 : metastate=24;break;
 case 32 : metastate=23;break;
 default : metastate=45;
 }
 break;
case 24 : switch(j){
 case 32 : metastate=25;break;
 default : metastate=45;
 }
 break;
case 25 : switch(j){
 case 67 :
 case 99 : metastate=26;break;
 case 32 : metastate=25;break;
 default : metastate=45;
 }
 break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 26 : switch(j){
 case 79 :
 case 111 : metastate=27;break;
 default : metastate=45;
 }
 break;
case 27 : switch(j){
 case 78 :
 case 110 : metastate=28;break;
 default : metastate=45;
 }
 break;
case 28 : switch(j){
 case 84 :
 case 116 : metastate=29;break;
 default : metastate=45;
 }
 break;
case 29 : switch(j){
 case 69 :
 case 101 : metastate=30;break;
 default : metastate=45;
 }
 break;
case 30 : switch(j){
 case 78 :
 case 110 : metastate=31;break;
 default : metastate=45;
 }
 break;
case 31 : switch(j){
 case 84 :
 case 116 : metastate=32;break;
 default : metastate=45;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 break;
case 32 : switch(j){
 case 61 : metastate=33;break;
 case 32 : metastate=32;break;
 default : metastate=45;
 }
 break;
case 33 : switch(j){
 case 34 : metastate=34;break;
 case 32 : metastate=33;break;
 default : metastate=45;
 }
 break;
case 34 : switch(j){
 case 39 : metastate=35;break;
 case 10 : case 13 : case 40 : case 41 : case 46 : case 33 : case 63 : case 35 : case 36 :
 case 37 : case 38 : case 43 : case 45 : case 47 : case 59 : case 91 : case 92 : case 93 :
 case 94 : case 95 : case 60 : case 61 : case 62 : case 64 :
 case 32 : metastate=35;
 if(dest){
 col2[countcol]=1;
 }
 else{
 col2[countcol]=2;
 }
 }
 break;
case 44 : metastate=35;
 if(dest){
 col2[countcol]=1;
 }
 else{
 col2[countcol]=2;
 }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 break;
 case 58 : metastate=35;
 if(dest){
 col2[countcol]=1;
 }
 else{
 col2[countcol]=2;
 }
 break;
 case 34 : metastate=36;
 if(dest)col2[countcol]=1;else col2[countcol]=2;
 ++countcol;
 break;
 default : metastate=34;
 col1[countcol]=col1[countcol]+charj;
 if(dest){
 col2[countcol]=1;
 }
 else{
 col2[countcol]=2;
 }
 }
 break;
 case 35 : switch(j){
 case 39 : metastate=35;break;
 case 10 : case 13 : case 40 : case 41 : case 46 : case 33 : case 63 : case 35 : case 36 :
 case 37 : case 38 : case 43 : case 45 : case 47 : case 59 : case 91 : case 92 : case 93 :
 case 94 : case 95 : case 60 : case 61 : case 62 : case 64 : case 32 : case 44 :
 case 58 : metastate=35;break;
 case 34 : metastate=36;++countcol;break;
 default : metastate=34;
 if(col1[countcol].compareTo("")!=0)
 ++countcol;
 col1[countcol]=col1[countcol]+charj;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 if(dest){
 col2[countcol]=1;
 }
 else{
 col2[countcol]=2;
 }
 }
 break;
case 36 : switch(j){
 case 62 : if(first){
 metastate=0;
 first=false;
 }
 else metastate=37;
 break;
 case 32 : metastate=36;break;
 default : metastate=45;
 }
 break;
case 37 : metaout=true;
 break;
case 38 : switch(j){
 case 69 :
 case 101 : metastate=39;break;
 default : metastate=45;
 }
 break;
case 39 : switch(j){
 case 89 :
 case 121 : metastate=40;break;
 default : metastate=45;
 }
 break;
case 40 : switch(j){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 case 87 :
 case 119 : metastate=41;break;
 default : metastate=45;
 }
 break;
case 41 : switch(j){
 case 79 :
 case 111 : metastate=42;break;
 default : metastate=45;
 }
 break;
case 42 : switch(j){
 case 82 :
 case 114 : metastate=43;break;
 default : metastate=45;
 }
 break;
case 43 : switch(j){
 case 68 :
 case 100 : metastate=44;break;
 default : metastate=45;
 }
 break;
case 44 : switch(j){
 case 83 :
 case 115 : metastate=23;break;
 default : metastate=45;
 }
 break;
case 45 : switch(j){
 case 62 : metastate=0;break;
 default : metastate=45;
 }
 break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 //close switch(metastate)
 } //close if(j<129)
 j=br1.read();
 }

 for(int a4=0;a4<countcol;++a4)
 System.out.print(a4+":"+col1[a4]+(" "+col2[a4]+" "));
} //close try
catch(IOException ioe){
 System.out.println("IO error:"+ioe);
}

 boolean puttortable=true;
 if(temptitle.compareTo("Error 404")==0 || temptitle.compareTo("404 Not Found")==0)
 puttortable=false;
 if(puttortable){
 Statement stmt;
 ResultSet rs;
 try{
 stmt = con.createStatement();
 String update="INSERT INTO DETAIL "+ "VALUES(" + arrayurl[currenturl]+ "/" + linkpage +
 "" + temptitle + "" + arrayurl[currenturl] + "");
 stmt.executeUpdate(update);
 stmt.close();
 }
 catch(SQLException ex){
 System.err.println("SQLException: " + ex.getMessage());
 }
 for(int a5=0;a5<countcol;++a5)
 if(col1[a5].compareTo("")!=0)
 uttodatabase(col1[a5],col2[a5],linkpage);
 }
}

public void puttodatabase(String keyword,int place,String linkpage){
 Statement stmt;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PreparedStatement pre;
ResultSet rs;
boolean usekey=true;
try{
 pre=con.prepareStatement("SELECT COUNT(*) FROM STOPWORD "+
 " WHERE KEYWORD = ?");
 pre.setString(1,keyword);
 rs=pre.executeQuery();
 if(rs.next())
 if(rs.getInt(1)!=0)
 usekey=false;
 rs.close();
 pre.close();
 if(usekey){
 String query;
 query="SELECT COST FROM SEARCH WHERE KEYWORD = "+
 keyword+" AND URL = "+arrayurl[currenturl]"/"+linkpage+"";
 stmt = con.createStatement();
 rs=stmt.executeQuery(query);
 boolean alreadyinsearch=false;
 int newnum=0;
 while(rs.next()){
 newnum=rs.getInt(1);
 alreadyinsearch=true;
 }
 stmt.close();
 rs.close();
 if(alreadyinsearch){
 newnum=newnum+place;
 query="UPDATE SEARCH SET COST = "+
 newnum+" WHERE KEYWORD = "+keyword+" AND URL = "+
 arrayurl[currenturl]"/"+linkpage+"";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

stmt = con.createStatement();
stmt.executeUpdate(query);
stmt.close();
}
else{
query="INSERT INTO SEARCH VALUES('"+keyword+"','"+
arrayurl[currenturl]+"'"+linkpage+"','"+place+"");
stmt=con.createStatement();
stmt.executeUpdate(query);
stmt.close();
}
}
}
}
catch(SQLException ex){
System.err.println("SQLException: " + ex.getMessage());
}
}
}
public void findlink(String tempage){
String dir="";
boolean outcheckurl=false;
int j=tempage.length();
while(j>0 && outcheckurl==false){
-j;
if(tempage.charAt(j)=='/')outcheckurl=true;
}
if(outcheckurl)dir=tempage.substring(0,j+1);
try{
FileReader fr=new FileReader(myfile);
BufferedReader br=new BufferedReader(fr);
int i=0;
char in;
i=br.read();
int numstate=0;
int s=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

String templink[];
templink = new String[30000];
for(int i1=0;i1<30000;++i1)templink[i1]="";
int numlink=-1;
while(i!=-1){
in=(char)i;
switch(numstate){
case 0 : switch(i){
 case 60 : numstate=1;break;
 default : numstate=0;
 }
 break;
case 1 : switch(i){
 case 60 : numstate=1;break;
 case 70 :
 case 102 : numstate=2;break;
 case 62 : numstate=0;break;
 case 65 :
 case 97 : numstate=15;break;
 default : numstate=14;
 }
 break;
case 2 : switch(i){
 case 62 : numstate=0;break;
 case 82 :
 case 114 : numstate=3;break;
 default : numstate=14;
 }
 break;
case 3 : switch(i){
 case 62 : numstate=0;break;
 case 65 :
 case 97 : numstate=4;break;
 default : numstate=14;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 break;
case 4 : switch(i){
 case 62 : numstate=0;break;
 case 77 :
 case 109 : numstate=5;break;
 default : numstate=14;
 }
 break;
case 5 : switch(i){
 case 62 : numstate=0;break;
 case 69 :
 case 101 : numstate=6;break;
 default : numstate=14;
 }
 break;
case 6 : switch(i){
 case 62 : numstate=0;break;
 case 32 : numstate=7;break;
 default : numstate=14;
 }
 break;
case 7 : switch(i){
 case 32 : numstate=7;break;
 case 60 : numstate=1;break;
 case 83 :
 case 115 : numstate=8;break;
 case 62 : numstate=0;break;
 default : numstate=7;
 }
 break;
case 8 : switch(i){
 case 62 : numstate=0;break;
 case 82 :

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 case 114 : numstate=9;break;
 case 60 : numstate=1;break;
 default : numstate=7;
 }
 break;
case 9 : switch(i){
 case 62 : numstate=0;break;
 case 67 :
 case 99 : numstate=10;break;
 case 60 : numstate=1;break;
 default : numstate=7;
 }
 break;
case 10 : switch(i){
 case 62 : numstate=0;break;
 case 32 : numstate=11;break;
 case 61 : numstate=12;break;
 case 60 : numstate=1;break;
 default : numstate=7;
 }
 break;
case 11 : switch(i){
 case 62 : numstate=0;break;
 case 32 : numstate=11;break;
 case 61 : numstate=12;break;
 default : numstate=14;
 }
 break;
case 12 : switch(i){
 case 62 : numstate=0;break;
 case 32 : numstate=12;break;
 case 34 : numstate=13;++numlink;break;
 default : numstate=13;++numlink;
 templink[numlink]=templink[numlink]+in;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 }
 break;
case 13 : switch(i){
 case 62 : case 32 :
 case 34 : numstate=0;break;
 default : templink[numlink]=templink[numlink]+in;
 }
 break;
case 14 : switch(i){
 case 32 : numstate=16;break;
 case 62 : numstate=0;break;
 default : numstate=14;
 }
 break;
case 15 : switch(i){
 case 62 : numstate=0;break;
 case 32 : numstate=16;break;
 default : numstate=14;
 }
 break;
case 16 : switch(i){
 case 62 : numstate=0;break;
 case 32 : numstate=16;break;
 case 72 :
 case 104 : numstate=17;break;
 default : numstate=1;
 }
 break;
case 17 : switch(i){
 case 62 : numstate=0;break;
 case 82 :
 case 114 : numstate=18;break;
 default : numstate=1;
 }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 break;
 case 18 : switch(i){
 case 62 : numstate=0;break;
 case 69 :
 case 101 : numstate=19;break;
 default : numstate=1;
 }
 break;
 case 19 : switch(i){
 case 62 : numstate=0;break;
 case 70 :
 case 102 : numstate=20;break;
 default : numstate=1;
 }
 break;
 case 20 : switch(i){
 case 62 : numstate=0;break;
 case 32 : numstate=11;break;
 case 61 : numstate=12;break;
 default : numstate=1;
 }
 break;
 }
 i=br.read();
 }
 ++numlink;
 boolean add=false;
 boolean ishtml=false;
 String sameurl="";
 for(int i2=0;i2<numlink;++i2){
 System.out.println(i2+" : "+templink[i2]);
 sameurl=checksameurl(templink[i2]);
 if(sameurl.compareTo("notuseforlinkaaazz")!=0){
 if(sameurl.compareTo(templink[i2])==0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 {
 if(templink[i2].length(>1){
 if(templink[i2].charAt(0)=='/ ')
 templink[i2]=templink[i2].substring(1,templink[i2].length());
 else templink[i2]=checklevel(dir,templink[i2]);
 }
 }

 else templink[i2]=sameurl;
 add=checkhavelink(templink[i2]);
 if(add){
 ishtml=checkurl(templink[i2]);
 if(ishtml)
 addtolinkurl(templink[i2]);
 }
}
}
}

catch(IOException ioe){
System.out.println("IO error:"+ioe);
}
} //close findlink

public String checklevel(String direc,String linkpage){
int numstate=0;
int i=0;
boolean out=false;
int uplevel=0;
int place=0;
while(i<linkpage.length() && out==false){
switch(numstate){
 case 0 : if(linkpage.charAt(i)=='.')numstate=1;
 else numstate=4;
 break;
 case 1 : if(linkpage.charAt(i)=='.')numstate=2;
 else numstate=4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 break;
 case 2 : if(linkpage.charAt(i)=='/'){
 numstate=3;
 ++uplevel;
 place=i;
 }
 else numstate=4;
 break;
 case 3 : if(linkpage.charAt(i)=='.')numstate=1;
 else numstate=4;
 break;
 case 4 : out=true;
 break;
 }//close switch
 ++i;
} //close while
i=0;
int len=direc.length();
if(uplevel!=0){
linkpage=linkpage.substring(place+1,linkpage.length());
while(i!=uplevel+1 && len>0){
 --len;
 if(direc.charAt(len)=='/')++i;
}
if(len==0)direc=linkpage;//because direc=""
 else direc=direc.substring(0,len+1)+linkpage;
} //close if(uplevel!=0)
else direc=direc+linkpage;
return direc;
} //close checklevel

public String checksameurl(String linkpage){
String temp="";
String newvalue="notuseforlinkaaazzz";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int templen=0;
templen=linkpage.length();
if(templen>6)temp=linkpage.substring(0,7);
temp=temp.toUpperCase();
if(temp.compareTo("HTTP://")==0){
 boolean notsame=true;
 int len=arrayurl[currenturl].length();
 if(linkpage.length() > len+7+1){
 if((linkpage.substring(7,len+7)).compareTo(arrayurl[currenturl])==0){
 newvalue=linkpage.substring(len+7+1,linkpage.length());
 notsame=false;
 if(newvalue.length(>1)
 if(newvalue.charAt(0)=='/'){
 newvalue=newvalue.substring(1,newvalue.length());
 }
 }
 if(numurl<limiturl && notsame)addlisturl(linkpage);
 }
 else newvalue=linkpage;
 }
 return newvalue;
}
//close checksameurl
public void addlisturl(String linkpage){
 boolean docheck=true;
 int i=7;
 while(i<linkpage.length() && docheck){
 if(linkpage.charAt(i)=='/')docheck=false;
 ++i;
 }
 if(docheck==false)- i;
 String newurl;
 newurl=linkpage.substring(7,i);
 boolean add=true;
 i=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(i<numurl && add){
if(arrayurl[i].compareTo(newurl)==0)add=false;
++i;
}
if(add){
 arrayurl[numurl]=newurl;
 ++numurl;
}
} //close addlisturl

public void addtolinkurl(String linkpage){
linkurl[numlinkurl]=linkpage;
++numlinkurl;
} //close addtolinkurl

public boolean checkhavelink(String linkpage){
boolean add=true;
int i=0;
while(i<numlinkurl && add){
if(linkurl[i].compareTo(linkpage)==0)add=false;
++i;
}
return add;
} //close checkhavelink

public boolean checkurl(String linkpage){
int i=linkpage.length();
boolean boo=true;
int plot=0;
--i;
while(i>=0 && boo) {
if(linkpage.charAt(i)=='.'){
 plot=i;
 boo=false;
}
--i;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

i=0;
int numstate=0;
boolean eject=false;
if(plot!=0){
++plot;
String checktype="";
checktype=linkpage.substring(plot,linkpage.length());
checktype=checktype.toLowerCase();
if(checktype.compareTo("htm")==0 || checktype.compareTo("html")==0
|| checktype.compareTo("jhtm")==0 || checktype.compareTo("jhtml")==0
|| checktype.compareTo("shtm")==0 || checktype.compareTo("shtml")==0
|| checktype.compareTo("phtm")==0 || checktype.compareTo("phtml")==0
|| checktype.compareTo("asp")==0 || checktype.compareTo("xml")==0
|| checktype.compareTo("mif")==0 || checktype.compareTo("rtf")==0
|| checktype.compareTo("php")==0 || checktype.compareTo("lbi")==0
|| checktype.compareTo("dwt")==0 || checktype.compareTo("css")==0
|| checktype.compareTo("cfm")==0 || checktype.compareTo("cfml")==0
|| checktype.compareTo("lasso")==0 || checktype.compareTo("htx")==0
|| checktype.compareTo("js")==0 || checktype.compareTo("nsf")==0)
 eject=true;
} //close if(plot!=0)
return eject;
} //close checkurl
public void setfileurl(File turl){
 namelisturl=turl;
}
public void setlimiturl(int lim){
 limiturl=lim;
}
} //close class Spider

```

### คลาส GameBoard

```

import java.awt.*;
import GamePiece;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public class GameBoard {
 final boolean BLACK = false;
 final boolean WHITE = true;
 public int pieceWidth,pieceHeight;
 public int rows,cols;
 GamePiece board[][];
 public int empty;
 public GameBoard(int w,int h){
 board = new GamePiece[w][h];
 cols = w;
 rows = h;
 empty = w*h;
 }
 public void paintBoard(Graphics g,Rectangle r){
 int x,y;
 pieceWidth = (int)(r.width/cols);
 pieceHeight = (int)(r.height/rows);
 g.setColor(Color.green);
 g.fillRect(r.x,r.y,r.width,r.height);
 for (x=0;x < cols;x++){
 g.setColor(Color.black);
 g.drawLine(x*pieceWidth,0,x*pieceWidth,r.height);
 for (y=0;y < rows;y++){
 g.setColor(Color.black);
 g.drawLine(0,y*pieceHeight,r.width,y*pieceHeight);
 if (board[x][y] != null)
 board[x][y].paintPiece(g,new Rectangle(r.x +
 pieceWidth*x,r.y + pieceHeight*y,pieceWidth,pieceHeight),(board[x][y].color));
 }
 }
 }
 //end paintBoard
 public GamePiece pieceAt(int x,int y){
 if (x >= cols || x < 0 || y >= rows || y < 0) return null;
 }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 return board[x][y];
 }

 public int count(boolean color){
 int i,j,num = 0;
 for (i=0;i < rows;i++)
 for (j=0;j < cols;j++)
 if (board[i][j] != null && board[i][j].color == color)
 { num++; }

 return num;
 }

 public void addPiece(int x,int y,boolean c){
 board[x][y] = new GamePiece(c);
 }
}

```

#### คลาส GameClientThread

```

import java.net.*;
import java.io.*;
import java.util.*;
import GameGroup;

public class GameClientThread extends Thread {
 GameGroup parent;
 Socket theSock;
 DataInputStream dis;
 PrintStream ps;
 String alias;

 public GameClientThread(Socket s,GameGroup p){
 theSock = s;
 parent = p;
 }

 public void run(){
 try{
 dis = new DataInputStream(new BufferedInputStream(theSock.getInputStream()));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 ps = new PrintStream(theSock.getOutputStream());
}
catch (Exception e){ stop(); }
while (theSock != null){
 String input = null;
 try{
 input = dis.readLine();
 if (input != null){
 parent.output(input);
 if (input.equals("bye")){
 System.out.println("Client Remove");
 stop();
 }
 }
 }
 catch (Exception e){
 stop();
 theSock = null;
 }
 try { sleep(100); } catch (Exception e) {}
}
}

public boolean message(String str){
 boolean flag = false;
 while (!flag){
 try{
 ps.println(str);
 flag = true;
 }
 catch (Exception e){ flag = false; }
 }
 return true;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void finalize(){
 try{
 ps.close();
 dis.close();
 theSock.close();
 }
 catch (Exception e) {}
 theSock = null;
}
}

```

### คลาส GameGroup

```

import java.net.*;
import java.io.*;
import java.util.*;
import GameClientThread;
public class GameGroup extends Thread {
 GameClientThread arr[];
 final int SIZE = 2;
 public GameGroup(Socket s){
 arr = new GameClientThread[SIZE];
 addClient(s);
 }
 public void addClient(Socket s){
 int x;
 for (x=0;x < SIZE;x++)
 if (arr[x] == null || !arr[x].isAlive()){
 arr[x] = new GameClientThread(s,this);
 arr[x].start();
 return;
 }
 }
 public void run(){
 int x;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

arr[0].message("start|white");
arr[1].message("start|black");
while (true){
 for (x=0;x < SIZE;x++){
 if (arr[x] == null || !arr[x].isAlive()){
 output("bye");
 stop();
 }
 try { sleep(5000); } catch (Exception e) {}
 }
}

public void output(String str){
 int x;
 for (x=0;x < SIZE;x++){
 if (arr[x] != null){ arr[x].message(str); }
 }
}

public boolean full(){
 int x;
 for (x=0;x < SIZE;x++){
 if (arr[x] == null)
 return false;
 }
 return true;
}

public void finalize(){
 int x;
 output("bye");
 try{
 for (x=0;x < SIZE;x++){
 if (arr[x] != null)
 { arr[x].stop(); }
 }
 }
 catch (Exception e) {}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

```

### คลาส GamePiece

```
import java.awt.*;

public class GamePiece{

 final boolean BLACK = false;

 final boolean WHITE = true;

 public boolean color;

 public Rectangle rect;

 public GamePiece(boolean c){

 color = c;

 }

 public void flip(){ color = !color; }

 public void paintPiece(Graphics g,Rectangle r,boolean c){

 rect = r;

 color = c;

 if (!r.intersects(g.getClipRect())) return;

 if (color == WHITE)

 g.setColor(Color.white);

 else

 g.setColor(Color.black);

 g.fillOval(rect.x+2,rect.y+2,rect.width-4,rect.height-4);

 }

}

```

### คลาส GameServer

```
import GameServerThread;

public class GameServer{

 public static void main(String[] arg){

 int thePort;

 try{

 thePort = Integer.valueOf(arg[0]).intValue();

 }

 catch (Exception e)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 { thePort = 5000; }
 new GameServerThread(thePort).start();
}
}

```

### คลาส GameServerThread

```

import java.net.*;
import java.util.*;
import GameGroup;

public class GameServerThread extends Thread {
 ServerSocket servSock = null;
 Vector v; //use to keep game group
 public GameServerThread(int port){
 try{
 servSock = new ServerSocket(port);
 }
 catch (Exception e){
 System.out.println("Could not initialize. Exiting.");
 System.exit(1);
 }
 System.out.println("\n Othello Server v. 1.00\n");
 System.out.println(" Server successfully initialize. Waiting for connection on port "+port+"\n");
 v = new Vector();
 }
 public void run(){
 GameGroup tempGroup = null;
 while (servSock != null){
 Socket tempSock;
 try{
 tempSock = servSock.accept();
 System.out.println("Received New Connection.");
 if (!v.isEmpty()) //if have client already login{
 tempGroup = (GameGroup)v.lastElement();
 if (tempGroup.full()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 { v.addElement(new GameGroup(tempSock)); }
 else{
 tempGroup.addClient(tempSock);

 if (tempGroup.full())
 tempGroup.start();
 }
 } else v.addElement(new GameGroup(tempSock));
 for (int x=0;x < v.size()-1;x++)
 if (!((GameGroup)v.elementAt(x)).isAlive()){
 System.out.println("Remove the Group "+x);
 v.removeElementAt(x);
 }
 }
 catch (Exception e){
 System.out.println("New Connection Failure. Exiting.");
 System.exit(1);
 }
 try { sleep(100); } catch (Exception ex) {}
}
}
public void finalize(){
 try
 { servSock.close(); }
 catch (Exception e) {}
 servSock = null;
}
}
}

```

### คลาส NetOthello

```

import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import java.net.*;
import java.io.*;
import GameBoard;
import PromptFrame;
public class NetOthello extends Applet implements Runnable{
 final boolean BLACK = false;
 final boolean WHITE = true;
 GameBoard theBoard;
 boolean turn;
 boolean local;
 Cursor plusCursor = new Cursor(Cursor.CROSSHAIR_CURSOR);
 Cursor originalCursor = new Cursor(Cursor.DEFAULT_CURSOR);
 Font overFont = new Font("Helvetica",Font.BOLD,30);
 int squareSize = 260;
 boolean click = false;
 Image arrow;
 String gameState;
 String playerColor = null;
 TextArea dispA = new TextArea(5,60);
 TextField sayTextField = new TextField(40);
 Button newGame = new Button("New Game");
 Button sayButton = new Button("Say");
 Panel inputPanel, turnPanel;
 PromptFrame pf;
 Thread kicker;
 PrintStream ps;
 Socket s = null;
 DataInputStream dis = null;
 String name, theHost;
 int thePort = 5000;
 public void run(){
 name = null;
 while (name == null){
 name = pf.gotName;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. Bill McCarty : " Java Distributed Objects “, SAMS, First edition, 1989
2. Y. Daniel Liang : " An Introduction to Java Programming “, Que E&T , First edition, 1998
3. Joel Fan, Eric Ries, Calin Tenitchi : " Black Art of Java Game Programming “, Waite Group Press, First edition, 1996
4. Gray Cornell, Cay s.Horstmann,Sun Microsystems : " Core Java “,Sun Microsystems, Second editon, 1997
5. Mertin Hughes, Michale Shoffner, Derek Hamner,Derek : “ Java Network Programming “ , Manning Publications, Second edition, 1997
6. Walnum, Clayton : “Java by Example”, Que Corparation, First edition, 1996
7. Janes Gosling, Frank Yellin : “Java application programming interface “, Addison-wesley, First edition, 1996
8. [www.java.sun.com](http://www.java.sun.com)
9. [www.searchenginewatch.com](http://www.searchenginewatch.com)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

 catch (Exception e)
 { input = null; }
}
if (input.equals("bye")){
 display("Your partner has left the game... Restarting");
 repaint();
 return;
} else
if (input.equals("end")){
 gameState = "OVER";
 repaint();
 stop();
 kicker.stop();
 return;
}
st = new StringTokenizer(input,"|");
String cmd = st.nextToken();
String val = st.nextToken();
if (cmd.equals("start")){
 display("Another player come");
 if (val.equals("black")) local = BLACK;
 else local = WHITE;
 if (local == WHITE)
 playerColor = "WHITE";
 else
 playerColor = "BLACK";
 display("You will play "+val);
 repaint();
} else
if (cmd.equals("move")){
 doMove(Integer.valueOf(val).intValue(),Integer.valueOf(st.nextToken()).intValue(),turn);
 click = false;
 repaint();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 } else
 if (cmd.equals("say")){
 dispA.append(val+"\n");
 }
} //end kicker is not null
} //end run method
public void init(){
 newGame.addActionListener(new java.awt.event.ActionListener(){
 public void actionPerformed(ActionEvent e){
 newGame_actionPerformed(e);
 }
 });
 sayButton.addActionListener(new java.awt.event.ActionListener(){
 public void actionPerformed(ActionEvent e){
 sayButton_actionPerformed(e);
 }
 });
 sayTextField.addActionListener(new java.awt.event.ActionListener(){
 public void actionPerformed(ActionEvent e){
 sayTextField_actionPerformed(e);
 }
 });
 this.addMouseListener(new java.awt.event.MouseAdapter(){
 public void mouseClicked(MouseEvent e){
 this_mouseClicked(e);
 }
 });
 this.addMouseMotionListener(new java.awt.event.MouseMotionAdapter(){
 public void mouseMoved(MouseEvent e){
 this_mouseMoved(e);
 }
 });
 arrow = getImage(getCodeBase(),"Arrow.gif");
 newGame();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

this.setSize(420,390);
setBackground(Color.yellow);
setLayout(new BorderLayout());
inputPanel = new Panel();
inputPanel.setLayout(new BorderLayout());
inputPanel.add("West",dispA);
inputPanel.add("South",turnPanel = new Panel());
turnPanel.add(sayTextField);
turnPanel.add(sayButton);
turnPanel.add(newGame);
dispA.setEditable(false);
add("South",inputPanel);
try{
 theHost = this.getParameter("HOST");
 thePort = Integer.parseInt(getParameter("PORT"));
}
catch (Exception e){
 theHost = "127.0.0.1";
 thePort = 5000;
}
}
public void newGame(){
 setFont(new Font("Helvetica",Font.BOLD,10));
 gameState = null;
 initBoard();
 promptUser();
 dispA.setText(null);
 sayTextField.setText(null);
 kicker = new Thread(this);
 kicker.setPriority(Thread.NORM_PRIORITY);
 kicker.start();
}
public void initBoard(){
 turn = WHITE;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

theBoard = new GameBoard(8,8);
theBoard.addPiece(3,3,WHITE);
theBoard.addPiece(3,4,BLACK);
theBoard.addPiece(4,3,BLACK);
theBoard.addPiece(4,4,WHITE);
}

public void promptUser(){
 pf= new PromptFrame();
 pf.setSize(300,100);
 pf.show();
}

public void paint(Graphics g){
 if (gameState == "OVER"){
 if (this.getFont() != overFont)
 setFont(overFont);
 g.drawString("GAME OVER",100,130);
 int bl = theBoard.count(BLACK);
 int wh = theBoard.count(WHITE);
 if (wh > bl)
 g.drawString("White Win!!",110,180);
 else if (wh < bl)
 g.drawString("Black Win!!",110,180);
 else if (wh == bl)
 g.drawString("It's tied",110,180);
 return;
 }

 theBoard.paintBoard(g,new Rectangle(0,0,squareSize,squareSize));
 this.showStatus(name+" login with "+playerColor);
 g.drawString("YOU ARE",320,35);
 if (playerColor != null){
 if (playerColor == "WHITE")
 g.setColor(Color.white);
 else
 g.setColor(Color.black);
 }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

g.fillOval(380,20,25,25);
g.setColor(Color.black);
}
g.drawString("WHITE",330,80);
g.drawString("Total : "+new Integer(theBoard.count(WHITE)).toString(),330,100);
g.drawString("BLACK",330,180);
g.drawString("Total : "+new Integer(theBoard.count(BLACK)).toString(),330,200);
if (turn == WHITE)
 g.drawImage(arrow,280,60,this);
else
 g.drawImage(arrow,280,160,this);
}
public void stop(){
try{
 ps.println("bye");
 dis.close();
 ps.close();
 s.close();
}
catch (Exception e) {}
}
public boolean validMoveXY(int x,int y,boolean color) {
if ((theBoard.pieceWidth == 0) || (theBoard.pieceHeight == 0))
 return false;
else
 return validMove((int)(x/theBoard.pieceWidth), (int)(y/theBoard.pieceHeight),color);
}
public boolean validMove(int x,int y,boolean color) {
if (theBoard.pieceAt(x,y) != null) return false;
int a,b,i,j,num;
GamePiece temp = null;
for (i = -1;i <= 1;i++)
 for (j = -1;j <= 1;j++) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (!(j == 0 && i == 0)) {
 a = x;
 b = y;
 num = 0;
 do {
 a += i;
 b += j;
 temp = theBoard.pieceAt(a,b);
 if (temp != null && temp.color == !color)
 num++;
 if (temp != null && temp.color == color)
 if (num > 0) return true;
 else temp = null;
 }
 while (temp != null);
}
return false;
}

public int validMoves(boolean color) {
 int i,j,num = 0;
 for (i = 0;i < 8;i++)
 for (j = 0;j < 8;j++) {
 if (validMove(i,j,color))
 num++;
 }
 return num;
}

public void doMove(int x,int y,boolean color) {
 int a = x,b = y;
 int i,j,num;
 GamePiece temp = null;
 theBoard.addPiece(x,y,color);
 repaint(a*theBoard.pieceWidth,b*theBoard.pieceHeight,theBoard.pieceWidth,theBoard.pieceHeight);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i = -1; i <= 1; i++)
 for (j = -1; j <= 1; j++) {
 if (!(j == 0 && i == 0)) {
 a = x;
 b = y;
 num = 0;
 do {
 a += i;
 b += j;
 temp = theBoard.pieceAt(a,b);
 if (temp != null) {
 if (temp.color == !color)
 num++; else
 if (temp.color == color){
 if (num > 0){
 a = x + i;
 b = y + j;
 temp = theBoard.pieceAt(a,b);
 while (temp.color == !color){
 theBoard.pieceAt(a,b).flip();
 }
 }
 }
 repaint(a*theBoard.pieceWidth,b*theBoard.pieceHeight,theBoard.pieceWidth,theBoard.pieceHeight);
 a += i;
 b += j;
 temp = theBoard.pieceAt(a,b);
 }
 }
 temp = null;
 }
 }
 while (temp != null);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (validMoves(!turn) > 0)
 turn = !turn;
if (endGame()) {
 boolean flag = false;
 while (!flag) {
 try{
 ps.println("end");
 ps.flush();
 flag = true;
 }
 catch (Exception ex)
 { flag = false; }
 }
}
}

public boolean endGame(){
 if (theBoard.empty == 0)
 return true;
 else if ((theBoard.count(BLACK) == 0) || (theBoard.count(WHITE) == 0))
 return true;
 else if ((validMoves(BLACK) == 0) && (validMoves(WHITE) == 0))
 return true;
 return false;
}

public void display(String str){
 System.out.println(str);
}

public void newGame_actionPerformed(ActionEvent e){
 kicker.stop();
 newGame();
 repaint();
}

public void sayButton_actionPerformed(ActionEvent e){
 if (sayTextField.getText().trim() != ""){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 ps.println("say|" + name + " says : " + sayTextField.getText());
 sayTextField.setText(null);
}
}
public void exitButton_actionPerformed(ActionEvent e){
 if (gameState == "OVER") return;
 boolean flag = false;
 while (!flag){
 try{
 ps.println("end");
 ps.flush();
 flag = true;
 }
 catch (Exception ex)
 { flag = false; }
 }
 repaint();
}
public void sayTextField_actionPerformed(ActionEvent e){
 sayButton_actionPerformed(e);
}
public void this_mouseMoved(MouseEvent e){
 if ((validMoveXY(e.getX(),e.getY(),local) && (turn == local)){
 if (this.getCursor().getType() != Cursor.CROSSHAIR_CURSOR){
 this.setCursor(plusCursor);
 repaint(e.getX(),e.getY(),theBoard.pieceWidth,theBoard.pieceHeight);
 }
 }
}
else{
 if (this.getCursor().getType() != Cursor.DEFAULT_CURSOR){
 this.setCursor(originalCursor);
 repaint(e.getX(),e.getY(),theBoard.pieceWidth,theBoard.pieceHeight);
 }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
public void this_mouseClicked(MouseEvent e){
 int x = e.getX();
 int y = e.getY();
 if (turn == local){
 if (validMoveXY(x,y,turn)){
 if (click) return;
 int xx,yy;
 xx = (int)(x/theBoard.pieceWidth);
 yy = (int)(y/theBoard.pieceHeight);
 boolean flag = false;
 while (!flag){
 try{
 ps.println("move|" +xx+"|" +yy);
 ps.flush();
 click = true;
 flag = true;
 }
 catch (Exception ex)
 { flag = false; }
 }
 } //end if validmove
 } //end if my turn
}
}

```

### คลาส PromptFrame

```

import java.awt.event.*;
import java.awt.*;

public class PromptFrame extends Frame{
 FlowLayout flowLayout1 = new FlowLayout();
 Label label1 = new Label();
 TextField textField1 = new TextField();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Button button1 = new Button();
String gotName = null;
public PromptFrame() {
 super("Ready to play Othello ?");
 try {
 jbInit();
 }
 catch (Exception e) {
 e.printStackTrace();
 }
}
private void jbInit() throws Exception{
 this.addWindowListener(new java.awt.event.WindowAdapter() {
 public void windowClosing(WindowEvent e) {
 this.windowClosing(e);
 }
 });
 button1.addActionListener(new java.awt.event.ActionListener() {
 public void actionPerformed(ActionEvent e) {
 button1_actionPerformed(e);
 }
 });
 textField1.addActionListener(new java.awt.event.ActionListener() {
 public void actionPerformed(ActionEvent e){
 textField1_actionPerformed(e);
 }
 });
 label1.setText("Name");
 textField1.setText("Your Name");
 button1.setLabel("Login");
 this.setLayout(flowLayout1);
 this.add(label1, null);
 this.add(textField1, null);
 this.add(button1, null);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 textField1.selectAll();
}
void this_windowClosing(WindowEvent e){
 this.setVisible(false);
}
void textField1_actionPerformed(ActionEvent e){
 if (textField1.getText() != "")
 gotName = textField1.getText();
 this.setVisible(false);
}
void button1_actionPerformed(ActionEvent e) {
 if (textField1.getText() != "")
 gotName = textField1.getText();
 this.setVisible(false);
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. Bill McCarty : " Java Distributed Objects ", SAMS, First edition, 1989
2. Y. Daniel Liang : " An Introduction to Java Programming ", Que E&T , First edition, 1998
3. Joel Fan, Eric Ries, Calin Tenitchi : " Black Art of Java Game Programming ", Waite Group Press, First edition, 1996
4. Gray Cornell, Cay s.Horstmann,Sun Microsystems : " Core Java ",Sun Microsystems, Second editon, 1997
5. Mertin Hughes, Michale Shoffner, Derek Hamner,Derek : " Java Network Programming " , Manning Publications, Second edition, 1997
6. Walnum, Clayton : "Java by Example", Que Corparation, First edition, 1996
7. Janes Gosling, Frank Yellin : "Java application programming interface ", Addison-wesley, First edition, 1996
8. [www.java.sun.com](http://www.java.sun.com)
9. [www.searchenginewatch.com](http://www.searchenginewatch.com)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้