

เจเพลเยอร์วาไรตี้

J PLAYER VARIETY



ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน..... 55106
วัน,เดือน,ปี..... 8 เม.ย. 2548

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

.....
.....
.....

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เจเพลเยอร์วารีตี้

J PLAYER VARIETY

คณะผู้จัดทำ นายภาณุมาศ วงษาเลิศ

นางสาววิภาวี วิชญชาคร



สุรินทร์ กิตติธรรมกุล อาจารย์ที่ปรึกษา
(ดร. สุรินทร์ กิตติธรรมกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจเพลเยอร์วารีตี้

นายภาณุมาศ วงษาเลิศ 43010325

นางสาววิภาวี วิชญชาคร 43010399

ดร.สุรินทร์ กิตติธรรมกุล อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

เจเพลเยอร์วารีตี้เป็นโปรแกรมที่พัฒนาขึ้นเพื่อ อำนวยความสะดวกให้กับผู้ใช้ในการบันทึกเสียงร้องของตนเองในขณะที่เล่นไฟล์ภาพและเสียงได้ ทำให้ผู้ใช้สามารถนำเสียงที่บันทึกไว้กลับมาฟังใหม่เพื่อปรับปรุงความสามารถด้านการร้องเพลงของตนเองได้ นอกจากนี้ยังให้ความสะดวกในเรื่องการคัดลอกไฟล์จากแผ่นวีซีดีคาราโอเกะลงเก็บไว้ในเครื่องคอมพิวเตอร์ของผู้ใช้ทำให้ผู้ใช้ไม่ต้องเปลี่ยนแผ่นวีซีดีไปมาหากต้องการร้องเพลงของหลายศิลปินระหว่างการร้องเพลง

เจเพลเยอร์วารีตี้เป็นโปรแกรมเล่นไฟล์ภาพและเสียง ที่มีการพัฒนาขึ้นด้วยภาษาจาวาบนแพลตฟอร์มของวินโดวส์ โดยใช้คอมโพเนนต์เจเอ็มเอฟซึ่งเป็นเอพีไอที่รับผิดชอบการนำเสนอไฟล์จำพวกภาพและเสียง ทำงานกับอุปกรณ์อินพุต คือ ไมโครโฟน ผ่านการกระบวนการทำงานภายในของเจเอ็มเอฟที่ควบคุมโดยออบเจกต์เพลเยอร์ และให้เอาต์พุตออกทาง ลำโพงและจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JPlayerVareity

Panumas Wongsalert

Wipawee Wishayashakorn

Dr. Surin Kittitornkun Advisor

Abstract

J Player Variety is a computer program developed to let the users KARAOKE and record their vocal at the same time. As a result, the users can create their own music video CD. In addition, J Player Variety helps the users improve their singing skills. Moreover, it also provides various functions including copying KARAOKE files from a video CD into a user's computer. Therefore, the users can sing as many songs as they want without changing the CDs. It can be very convenient. J Player Variety is much more than a KARAOKE program.

J Player Variety is developed using Java programming language and JMF (Java Media Framework) application program interface. JMF provides function in controlling input/output such as microphone, speakers and monitor and processing media both audio and video streams. Hence, J Player Variety can support any computing platform including Microsoft Windows, Linux, and Unix.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้เป็นอย่างดีด้วยคำแนะนำคำปรึกษาและคอยดูแลจากหลายๆ ฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาที่ให้โอกาสให้ข้าพเจ้าได้ทำปริญญาบัตรฉบับนี้ คอยให้ความเอาใจใส่ดูแล แนะนำและให้ความช่วยเหลือเสมอมา คือ ดร. สุรินทร์ กิตติธรรมกุล

นอกจากนี้ ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้จัดเตรียมสถานที่และสิ่งอำนวยความสะดวก เพื่อให้การวิจัยและพัฒนาโปรแกรมเป็นไปได้ด้วยความสะดวกและรวดเร็วรวมทั้งยังมีอินเทอร์เน็ตความเร็วสูงให้บริการ สำหรับการค้นคว้าหาความรู้ต่างๆ ซึ่งท้ายที่สุดแล้วก็ได้ประกอบกันเป็นส่วนหนึ่งของโครงการนี้

และขอขอบคุณพี่ๆ เพื่อนๆ ในห้องปฏิบัติการมัลติมีเดียที่คอยสร้างความบันเทิงช่วยทำให้ผ่อนคลายยามอยู่ในห้อง เสมอมา และที่ขาดไม่ได้ต้องขอบคุณห้องฮาร์ดแวร์ที่ให้อาศัย ซื่อซนมาช่วยให้ห้องหายหิว มีแรงกายในการทำงาน

และสุดท้ายต้องขอขอบคุณบุคคลที่สำคัญที่สุดในชีวิตที่ทำให้ข้าพเจ้ามีวันนี้ นั่นคือ บิดา มารดา และบุคคลในครอบครัว อันเป็นที่เคารพรักซึ่งได้เลี้ยงดู คอยสั่งสอนข้าพเจ้ามาเป็นอย่างดี พร้อมให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ ความรักเสมอมา ข้าพเจ้าขอกราบพระคุณมา ณ ที่นี้ด้วย

ภาณุมาศ วงษาเลิศ

วิภาวี วิษุชาคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ที่มาและหลักการ	1
1.2 วัตถุประสงค์ของการพัฒนา	1
1.3 ขอบเขตของในการดำเนินงาน	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 เสียงและวิดีโอ	3
2.1 เสียง	3
2.2 วิดีโอ(Video)	4
2.2.1 วิดีโอแอนาล็อก(Analog Video)	4
2.2.2 วิดีโอดิจิทัล(Digital Video)	4
2.3 การบีบอัดวิดีโอ	4
บทที่ 3 จาวามีเดียเฟรมเวิร์ก-เจเอ็มเอฟ(Java Media Framework -JMF)	7
3.1 ไทม์เบสมีเดีย	7
3.2 องค์ประกอบภายใน โปรเซส	8
3.2.1 ดีมัลติเพล็กซ์เซอร์และมัลติเพล็กซ์เซอร์ (Multiplexer and Demultiplexer)	8
3.2.2 โคเด็ค(Codec)	8
3.2.3 เอฟเฟกต์ฟิลเตอร์(Effect Filters)	8
3.2.4 เร็นเดอร์เรอร์(Renderer)	8
3.3 เจเอ็มเอฟ คืออะไร	8
3.4 องค์ประกอบภายในเจเอ็มเอฟ	9
3.4.1 เพลเยอร์(Player)	9
3.4.2 ดาต้าซอส(DataSource)	9
3.4.3 ดาต้าซิงก์(DataSink)	9
3.4.4 มีเดียโลเคเตอร์(MediaLocator)	9
3.4.5 แมเนเจอร์(Manager)	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.6	อีเวนต์โมเดล (Event Model)	10
3.4.7	คอนโทรล (Controls)	10
3.4.8	กระบวนการนำเสนอ(Presentation Process)	10
3.4.9	คอนโทรลเลอร์(Controller)	10
3.4.10	เพลเยอร์ (Player)	10
3.4.10.1	สถานะของเพลเยอร์	11
3.4.11	โพรเซสเซอร์ (Processor)	11
3.4.11.1	สถานะของโพรเซสเซอร์	12
บทที่ 4	โครงสร้างและการออกแบบโปรแกรม	14
4.1	เทคนิคหรือเทคโนโลยีที่ใช้	14
4.2	เครื่องมือที่ใช้ในการพัฒนา	14
4.3	รายละเอียดของอินพุตและเอาต์พุต	14
บทที่ 5	การทดสอบการทำงานของโปรแกรม	23
5.1	อุปกรณ์ที่รองรับการทำงานของโปรแกรม	23
5.2	การทำงานและความสามารถของโปรแกรม	23
บทที่ 6	บทสรุปและวิจารณ์	26
6.1	บทสรุปและวิจารณ์	26
6.2	ปัญหาที่เกิดขึ้น	26
6.3	แนวทางการแก้ปัญหา	26
6.4	แนวทางการพัฒนา	27
ภาคผนวก		
ภาคผนวก ก.	ภาษาจาวา	28
ก.1	ลักษณะของภาษา	28
ก.2	ลักษณะการเขียนโปรแกรมด้วยภาษาจาวา	29
ก.3	ประเภทของโปรแกรมจาวา	30
ก.4	ข้อดีของการพัฒนาโปรแกรมด้วยภาษาจาวา	31
ภาคผนวก ข.	ความรู้พื้นฐานภาษาจาวา	32
ข.1	เอ็กเซปชัน แฮนดลิง (Exception handling)	32
ข.1.1	การตรวจสอบข้อผิดพลาดโดยใช้คำสั่ง try , catch และ finally	33
ข.2	อีเวนต์ แฮนดลิง (Event Handling)	34
ข.2.1	เมาส์ลิสเทนเนอร์ (MouseListener) , เมาส์โมชันลิสเทนเนอร์ (MouseEvent) และ เมาส์อีเวนต์ (MouseEvent)	36
ข.3	ออบเจ็ค	37
ข.3.1	การสร้างออบเจ็ค	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.3.2 การเรียกใช้แอตทริบิวต์(Attribute) และเมทอดของออบเจ็กต์	38
ข.4 คอนสตรัคเตอร์เมทอด	39
ข.5 โอเวอร์โหลดดิงเมทอด	39
ข.6 การใช้ “instanceof”	41
ข.7 การใช้คีชีเวิร์ด “super”	42
ข.7.1 การใช้คีชีเวิร์ด “super” เพื่อเรียกใช้สมาชิกของ ซูเปอร์คลาส	43
ข.7.2 การใช้คีชีเวิร์ด “super” เพื่อเรียกใช้คอนสตรัคเตอร์เมทอด ของซูเปอร์คลาส	44
ข.8 การส่งผ่านตัวแปร	45
ข.8.1 การส่งผ่านตัวแปรด้วยค่า (Pass by Value)	45
ข.8.2 การส่งผ่านด้วยการอ้างอิง (Pass by Reference)	47
ข.9 สวิงค์ คอมโพเนนต์(Swing Component)	48
ข.10 การเรียกใช้โปรแกรมอื่นด้วยจาวา	51
ภาคผนวก ก. รายละเอียดของสไลด์หลักของ โปรแกรม	52
ภาคผนวก ง. คู่มือการติดตั้ง โปรแกรม	60
ง.1 การติดตั้ง Jmf-2_1_le-windows-i586	60
ง.2 การตั้งค่า JMF Customizer	62
ง.3 การตั้งค่า Library ภายใน Jbuilder 7	67
ง.4 การใช้โปรแกรม dat2mpg	68
บรรณานุกรม	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้าที่
รูปที่ 3-1 แสดงองค์ประกอบหลักและการทำงานกับข้อมูลชนิดไทม์เบสมีเดีย	7
รูปที่ 3-2 แสดงภาพรวมของเพลเยอร์	10
รูปที่ 3-3 แสดงสถานะภายในเพลเยอร์	11
รูปที่ 3-4 แสดงภาพรวมของโปรเซสเซอร์	11
รูปที่ 3-5 แสดงสถานะของโปรเซสเซอร์	12
รูปที่ 4-1 แสดงหน้าที่หลักภายในโปรแกรม	14
รูปที่ 4-2 แสดงคลาสภายในโปรแกรมเจเพลเยอร์ว่าไรต์	15
รูปที่ 4-3 แสดงขั้นตอนการทำงานของการเล่นไฟล์มีเดียสตรีม	20
รูปที่ 4-4 แสดงสถานะภายในของเพลเยอร์ภายในคลาส PlayerEventHandler	20
รูปที่ 4-5 แสดงขั้นตอนการทำงานของการเล่นไฟล์มีเดียสตรีม	21
รูปที่ 4-6 แสดงขั้นตอนการทำงานในการบันทึกเสียงของผู้ร้อง	21
รูปที่ 4-7 แสดงขั้นตอนการคัดลอกไฟล์จากแผ่นวีซีดีเก็บลงในเครื่องของผู้ใช้	22
รูปที่ 4-8 แสดงขั้นตอนการเปิดไฟล์มาใส่ไว้ในรายการ	22
รูปที่ 5-1 แสดงรูปลักษณะของโปรแกรม	23
รูปที่ 5-2 แสดงการเลือกใช้ความสามารถในการบันทึกเสียง	24
รูปที่ 5-3 แสดงหน้าต่างเพื่อให้ผู้ใช้ระบุตำแหน่งและชื่อไฟล์ที่จะบันทึก	25
รูปที่ 5-4 แสดงขั้นตอนการเปลี่ยนชื่อเพลงโดยการใส่ชื่อที่ต้องการเปลี่ยนลงไปแทน	25
รูปที่ 5-5 แสดงข้อความที่แจ้งให้ทราบว่า การคัดลอกไฟล์เสร็จสมบูรณ์	25
รูปที่ ก-1 จาวาเอพีไอและจาวาเวอร์ซวลแมชชีน ไม่ขึ้นอยู่กับองค์ประกอบทางด้านฮาร์ดแวร์	28
รูปที่ ก-2 แสดงการทำการแปลภาษาให้เป็น โปรแกรมของจาวาเวอร์ซวลแมชชีน	28
รูปที่ ก-3 แสดงวัตถุของรูปทรงเรขาคณิต	29
รูปที่ ก-4 แสดงขั้นตอนการสร้างและทำงานของจาวาแอปพลิเคชัน	30
รูปที่ ข-1 แสดงการอินเทอร์ริทจาก คลาส “Throwable”	32
รูปที่ ข-2 แสดงความสัมพันธ์ระหว่างอีเวนทชออสซ์ , อีเวนทลิสเทนเนอร์ และ ลิสเทนเนอร์อินเทอร์เฟส	34
รูปที่ ข-3 แสดงความสัมพันธ์ของคลาส ต่าง ๆ ที่ใช้ในอีเวนทแฮนด์ลิ่งของเอดับบิวทีอีเวนท 35	
รูปที่ ข-4 แสดงถึงโอเวอร์โหลดคั้งเมทอด และการใช้งาน	40
รูปที่ ข-5 แสดงความเป็นอินสแทนซ์ของคลาสต่าง ๆ	41
รูปที่ ข-6 แสดงความสัมพันธ์ของคลาสต่าง ๆ	43
รูปที่ ข-7 แสดงการเรียกใช้ คอนสตรัคเตอร์ของซูเปอร์คลาส	44
รูปที่ ข-8 แสดงการส่งผ่านค่าให้กับเมทอดแบบการส่งผ่านด้วยค่า (Pass by Value)	46
รูปที่ ข-9 แสดงการส่งผ่านตัวแปร โดยใช้ออบเจ็กอ้างอิง (Reference)	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข-10 แสดงการสืบทอด Component ต่าง ๆ ของ Swing	49
รูปที่ ง -1 แสดงตำแหน่งของการเลือกไฟล์	60
รูปที่ ง -2 แสดงการทำงานของ การติดตั้ง	60
รูปที่ ง -3 แสดงข้อตกลงของโปรแกรม	61
รูปที่ ง -4 แสดงออบชั่นต่าง ๆ ของโปรแกรมการติดตั้ง	61
รูปที่ ง -5 แสดงการทำงานในการติดตั้งโปรแกรม	62
รูปที่ ง.2 -1 แสดงตำแหน่งของโปรแกรมการตั้งค่า	62
รูปที่ ง.2 -2 แสดงการเลือกออบชั่นต่าง ๆ	63
รูปที่ ง.2 -3 แสดงการเลือกออบชั่นต่าง ๆ	63
รูปที่ ง.2 -4 แสดงการเลือกออบชั่นต่าง ๆ	64
รูปที่ ง.2 -5 แสดงการเลือกออบชั่นต่าง ๆ	64
รูปที่ ง.2 -6แสดงการเลือกออบชั่นต่าง ๆ	65
รูปที่ ง.2 -7แสดงการเลือกออบชั่นต่าง ๆ	65
รูปที่ ง.2 -8 แสดงการเลือกออบชั่นต่าง ๆ	66
รูปที่ ง.2 -9 แสดงการเลือกออบชั่นต่าง ๆ	66
รูปที่ ง.3 -1 แสดงการ add Libraries	67
รูปที่ ง.3 -2 แสดงการ add Libraries	67
รูปที่ ง.3 -3 แสดงการ add Libraries	68
รูปที่ ง.4 -1 แสดงตัวอย่างการเรียกใช้โปรแกรม	69
รูปที่ ง.4 -2 แสดงตัวอย่างการเมื่อโปรแกรมได้ทำงานเสร็จสมบูรณ์แล้ว	69

สารบัญตาราง

ตารางที่ 2-1 แสดงคุณสมบัติของสัญญาณเสียงดิจิทัลในวินาที
 ตารางที่ ข-1 แสดงคอมพิวเตอร์ชนิดต่าง ๆ ในสวีตช์

หน้าที่

3

50



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและหลักการ

ปัจจุบันคอมพิวเตอร์เป็นอุปกรณ์ที่สามารถพบได้มากขึ้น คอมพิวเตอร์จึงไม่ใช่เรื่องใหม่สำหรับคนทั่วไปอีกต่อไป ดังนั้นการนำคอมพิวเตอร์มาใช้เพื่อความบันเทิงจึงนับว่าเป็นทางเลือกที่ช่วยประหยัดได้อีกทางหนึ่ง เพราะคอมพิวเตอร์มีความสามารถในการทำงานที่หลากหลาย สามารถทำงานแทนอุปกรณ์ไฟฟ้าต่างๆ ได้หลายชนิด เช่น วิทยุ โทรทัศน์ เครื่องเล่นวีซีดี เป็นต้น

โปรแกรมที่ทำงานกับไฟล์จำพวกภาพและเสียงในปัจจุบันนี้ โดยส่วนใหญ่จะมีความสามารถในการเล่นไฟล์เพียงอย่างเดียวเท่านั้น ยังไม่มีความสามารถในการบันทึกเสียงของผู้ใช้รวมอยู่ด้วย หรือโปรแกรมที่สามารถบันทึกเสียงของผู้ใช้ก็จะไม่สามารถเล่นไฟล์ภาพและเสียงไปพร้อมๆ กันได้ นอกจากนี้หากต้องการที่จะเล่นไฟล์จากวีซีดีหลายๆ แผ่น ในกรณีที่เป็นไฟล์คาราโอเกะ การเปลี่ยนแผ่นไปมา ทำให้ผู้ใช้ไม่ได้รับความสะดวกและเพลิดเพลินเท่าที่ควรขาดความต่อเนื่องในการใช้งาน

ดังนั้นโปรแกรมเจเพลเยอร์ว่าไรดี จึงถูกพัฒนาขึ้นมาเพื่อให้มีความสามารถในการบันทึกเสียงของผู้ใช้ในขณะเปิดไฟล์ภาพและเสียงอยู่ได้ ทำให้ผู้ใช้สามารถนำเสียงที่บันทึกไว้ มาปรับปรุงความสามารถในการร้องเพลงของตนได้ นอกจากนี้โปรแกรมยังสามารถที่จะคัดลอกไฟล์เพลงจากแผ่นวีซีดีเก็บไว้ในเครื่องคอมพิวเตอร์ได้อีกด้วย เพื่ออำนวยความสะดวกให้กับผู้ใช้เพื่อให้ผู้ใช้ไม่ต้องเปลี่ยนแผ่นวีซีดีไปมาหากต้องการเล่นไฟล์จากวีซีดีที่ต่างแผ่นกัน

1.2 วัตถุประสงค์ของการพัฒนา

- เพื่อให้ได้โปรแกรมจำพวกเล่นไฟล์ภาพและเสียงที่พัฒนาอยู่บนภาษาจาวา
- เพื่อให้ได้โปรแกรมที่มีความสามารถในการเล่นไฟล์เพลงคาราโอเกะที่นอกเหนือจากไฟล์มิดี (midi) ได้ ซึ่งเป็นการเพิ่มความบันเทิงอีกทางหนึ่ง เพราะมีรูปประกอบการร้องเพลงที่สวยงาม
- เพื่อให้ได้โปรแกรมที่มีความสามารถในการบันทึกเสียงของผู้ใช้ในขณะร้องเพลงคาราโอเกะเพื่อประโยชน์ในแง่ของการปรับปรุงทักษะการร้องเพลงของตนเอง
- เพื่อให้ผู้ใช้สามารถคัดลอกไฟล์เพลงจากแผ่นวีซีดีคาราโอเกะลงไว้ในเครื่องตนเองได้ ทำให้ไม่ต้องเสียเวลาในการเปลี่ยนแผ่นและสามารถเล่นเพลงได้หลากหลายมากขึ้น

1.3 ขอบเขตในการดำเนินงาน

โครงการนี้เป็นการสร้างและพัฒนาโปรแกรมให้มีความสามารถในการเล่นไฟล์มีเดียต่าง ๆ ที่เจเอ็ม เอฟ สามารถรองรับได้ซึ่งประกอบไปด้วยไฟล์ที่มีรูปแบบต่าง ๆ คือ mpeg, wav, mp3, aiff, au, avi, midi โปรแกรมสามารถทำการบันทึกเสียงของผู้ใช้งานโดยสามารถบันทึกเสียงเพลงพร้อมกับเสียงผู้ใช้ได้ โปรแกรมสามารถคัดลอกไฟล์ลงยังเครื่องของผู้ใช้โดยจะทำการแปลงไฟล์จาก dat เป็น mpeg ได้ซึ่งความสามารถในการแปลงไฟล์นั้น ผู้พัฒนาไม่ได้พัฒนาขึ้นเอง แต่ทำการเรียกใช้โปรแกรมที่ทำหน้าที่ในการแปลงไฟล์อีกทีหนึ่ง

1.4 วิธีการดำเนินงาน

เริ่มต้นโครงการด้วยการศึกษาภาษาจาวา ศึกษาความสามารถของโปรแกรมต่าง ๆ ที่มีอยู่ ศึกษาเจเอ็ม เอฟ (จาวามีเดียเฟรมเวิร์ค) หลังจากได้ทำการศึกษาทฤษฎีต่าง ๆ แล้ว ก็แบ่งงานออกเป็น 4 ส่วนหลัก ๆ คือ

- 1) ส่วนของการพัฒนารูปแบบของอินเทอร์เฟซซึ่งจะต้องพัฒนาให้มีรูปแบบที่ง่ายต่อการใช้งาน
- 2) ส่วนของพัฒนาการเล่นไฟล์ และการหยุดเล่นไฟล์ รวมถึงการทดสอบไฟล์ต่าง ๆ ที่โปรแกรมสามารถรองรับได้
- 3) ส่วนของการพัฒนาการบันทึกเสียงผู้ใช้ โดยให้มีความสามารถตรงตามขอบเขตที่ได้กำหนดเอาไว้ และสุดท้ายคือส่วนของการพัฒนาให้โปรแกรมสามารถทำการแปลงไฟล์จาก dat เป็น mpeg และบันทึกไฟล์ลงยังเครื่องได้

โดยการพัฒนาจะเริ่มที่ส่วนที่ 1 กับส่วนที่ 2 ก่อน เมื่อได้การทำงานที่สามารถทำงานได้แล้วจึงเริ่มพัฒนาส่วนที่ 3 และ 4 ต่อ พร้อมกับปรับแต่งรูปแบบอินเทอร์เฟซให้สามารถใช้งานได้ง่ายเพิ่มมากขึ้น

บทที่ 2

เสียงและวีดีโอ

2.1 เสียง

เสียงสามารถแบ่งออกเป็น 2 ชนิด คือ

1. เสียงแบบมีดี (MIDI:Musical Instrument Digital Interface) คือ ข้อมูลที่แสดงถึงลักษณะเสียงที่แทนเครื่องดนตรีชนิดต่างๆ ในมุมมองของนักดนตรี มีดีจะหมายถึง โน้ตเพลงที่มีรูปแบบเป็นสัญลักษณ์หรือตัวเลข ที่จะบอกให้รู้ว่าต้องเล่นตัวโน้ตตัวใดในเวลานานเท่าไร เพื่อให้เกิดเป็นเสียงดนตรี ไฟล์มีดีจะมีขนาดเล็ก แต่ความสมจริงของเสียงยังด้อยอยู่
2. เสียงแบบดิจิทัล (Digital Audio) คือสัญญาณเสียงที่ส่งมาจากอุปกรณ์ต่างๆ เช่น ไมโครโฟน เครื่องเล่นเทป แล้วนำข้อมูลที่ได้ไปแปลงเป็นสัญญาณดิจิทัล ซึ่งจะถูกรวมให้อยู่ในรูปแบบของบิต และไบต์ โดยเรียกอัตราการสุ่มข้อมูลที่ได้นี้ว่า Sampling Rate และขนาดของข้อมูลที่ได้นี้เรียกว่า Sampling Size ซึ่งทั้งสองตัวนี้จะเป็นตัวกำหนดคุณภาพของเสียงที่ได้จากการเล่นเสียงแบบดิจิทัล

Sampling Rate(kHz)	Sampling Size(bit)	Stereo หรือ Mono	จำนวนByte ที่ใช้ใน 1 วินาที
44.1	16	Stereo	8.5MB
44.1	16	Mono	5.25MB
44.1	8	Stereo	5.25MB
44.1	8	Mono	2.6MB
22.05	16	Stereo	5.25MB
22.05	16	Mono	2.5MB
22.05	8	Stereo	2.6MB
22.05	8	Mono	1.3MB
11.025	8	Stereo	1.3MB
11.025	8	Mono	650KB

ตาราง 2-1 แสดงคุณสมบัติของสัญญาณเสียงดิจิทัลใน 1 วินาที

การบันทึกเสียงแบบสเตอริโอ (Stereo Recording) จะให้คุณภาพเสียงที่ฟังแล้วสมจริงมากขึ้นและเมื่อเปรียบเทียบกับการบันทึกเสียงแบบโมโน (Mono Recording) ในระยะเวลาที่เท่ากัน เพิ่มข้อมูลเสียงแบบสเตอริโอที่บันทึกได้จะใช้พื้นที่มากกว่าเพิ่มข้อมูลเสียงแบบโมโน ซึ่งการคำนวณขนาดของเพิ่มข้อมูลที่ได้จากการบันทึกทั้งสองแบบมีดังนี้

การบันทึกเสียงแบบโมโน : $\text{Sampling Rate} * \text{ระยะเวลาในการบันทึก} * (\text{Sampling Size}/8) * 1$

การบันทึกเสียงแบบสเตอริโอ : $\text{Sampling Rate} * \text{ระยะเวลาในการบันทึก} * (\text{Sampling Size}/8) * 2$

2.2 วีดีโอ (Video)

วีดีโอเป็นองค์ประกอบที่สำคัญสำหรับงานมัลติมีเดียจึงได้มีการพัฒนาในหลายๆส่วนให้มีประสิทธิภาพ โดยสามารถแบ่งวีดีโอที่มีอยู่ในปัจจุบันออกเป็น 2 ชนิด คือ

2.2.1 วีดีโออนาล็อก (Analog Video)

วีดีโออนาล็อก เป็นวีดีโอที่ทำการบันทึกข้อมูลภาพและเสียงให้อยู่ในรูปสัญญาณอนาล็อก ได้แก่ VHS (Video Home System) ซึ่งเป็นม้วนวีดีโอที่ใช้ดูกันตามบ้าน

2.2.2 วีดีโอดิจิทัล (Digital Video)

เป็นวีดีโอที่ทำการบันทึกข้อมูลภาพและเสียงที่ได้มาจากกล้องวีดีโอดิจิทัล ให้อยู่ในรูปสัญญาณดิจิทัล คือ 0 กับ 1 วีดีโอชนิดนี้เมื่อผ่านการตัดต่อแล้ว ก็ยังคงคุณภาพความชัดเท่ากับข้อมูลต้นฉบับ

2.3 การบีบอัดวีดีโอ

การบีบอัดวีดีโอ เป็นการจัดการเก็บข้อมูลภาพและเสียงของไฟล์ต้นฉบับให้มีขนาดลดน้อยลง แต่ยังคงมีปริมาณของข้อมูลเท่าเดิม โดยได้มีการพัฒนามาตรฐานที่ใช้สำหรับการบีบอัดข้อมูลชนิดนี้ไว้ดังนี้

- **เจเพ็ก (JPEG)**

เป็นมาตรฐานการบีบอัดข้อมูลที่คอมพิวเตอร์จะทำการสุ่มตัวอย่างของจุดภาพในส่วนต่างๆ ก่อนที่จะบีบอัดข้อมูล โดยจะตรวจสอบพื้นที่ขนาด 8*8 พิกเซล ว่าน่าจะมีสีอะไรอยู่มากที่สุด จากนั้นจะยุบพื้นที่ที่เหลือเพียงสีที่ต้องการเพียงหนึ่งพิกเซล ซึ่ง JPEG จะถูกนำมาใช้กับภาพนิ่งที่อัตราส่วนการบีบอัดประมาณ 25:1, 40:1 จนถึง 100:1

- **Motion-JPEG หรือ M-JPEG**

เป็นมาตรฐานการบีบอัดข้อมูล ที่สามารถบีบอัดและขยายสัญญาณ ได้ตั้งแต่ 12:1, 5:1 และ 2:1 ทำให้ภาพที่ออกมามีคุณภาพเป็นที่น่าพอใจ สำหรับกลุ่มผู้ผลิตงานที่ไม่ต้องการความละเอียดมากนัก

- **CODEC**

เป็นเทคโนโลยีการบีบอัดและการคลายข้อมูล ซึ่งสามารถนำไปใช้กับซอฟต์แวร์และฮาร์ดแวร์ หรืออย่างใดอย่างหนึ่งก็ได้ โดยส่วนมาก CODEC จะนิยมใช้กับการบีบอัดแบบ

MPEG, Indeo และ Cinepak

- **เอ็มเพ็ก (MPEG : Moving Picture Experts Group)**

เป็นมาตรฐานการบีบอัดสัญญาณภาพและเสียง โดยใช้ระบบ DCT ซึ่งเป็นระบบที่ใช้กับระบบวีดีโอคุณภาพสูงทั่วไป จะมีความคล้ายคลึงกับการบีบอัดข้อมูลแบบ JPEG แต่จะลดจำนวนข้อมูลที่ซ้ำกันของภาพต่อไปด้วย การบีบอัดข้อมูลแบบ MPEG นี้เป็นไปแบบไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมาตร เนื่องจากขั้นตอน ในการเข้ารหัสสัญญาณวีดิโอนานกว่าขั้นตอนการถอดรหัสข้อมูล โดยMPEG ได้ถูกพัฒนาขึ้นอย่างต่อเนื่องดังนี้

MPEG-1

ถูกพัฒนาขึ้นมาเพื่อใช้งานกับวีดิโอที่ดูตามบ้าน หรือเป็นที่รู้จะกันดีในชื่อ VHS ที่ใช้อัตราการส่งผ่านข้อมูลเพียง 1.5 Mbps ซึ่งไฟล์ที่ได้จากการบีบอัดข้อมูลแบบนี้ สามารถใช้เครื่องเล่น CD ทั่วไปอ่านหรือเขียนข้อมูลได้ แต่ยังไม่ให้ภาพที่ค่อนข้างหยาบ สัญญาณสีแต่ละจุดไม่สามารถกำหนดเป็นสีที่ถูกต้องได้

MPEG-2

ถูกพัฒนาขึ้นมาเพื่อใช้ในอุตสาหกรรมภาพยนตร์ โดยเฉพาะการบีบอัดข้อมูลแบบนี้ก่อนที่คอมพิวเตอร์จะคำนวณผลเพื่อแทนค่าจุดสีต่าง ภาพจะถูกแบ่งออกเป็นส่วนๆ และจำคำนวณทีละหลายๆภาพ เรียกว่า GOP (Group of Picture) ซึ่งเป็นการมองภาพครั้งละ 8-24 ภาพ โดยจะดูจากภาพที่หนึ่งของ GOP เป็นหลัก จากนั้นจะทำการเข้ารหัสภาพ แล้วมองภาพถัดไปว่ามีความแตกต่างจากภาพแรกที่จุดใด จากนั้นจะทำการเปรียบเทียบและเก็บเฉพาะข้อมูลที่แตกต่างของภาพในเฟรมนั้น ส่วนภาพต่อไปก็ทำการเปรียบเทียบกับภาพ ติดกัน แล้วเก็บส่วนต่างไว้เช่นกัน ทำให้สามารถลดจำนวนข้อมูลที่ต้องการเก็บ และเก็บบันทึกข้อมูลที่ต้องการถอดรหัสได้

MPEG-3

ถูกพัฒนาขึ้นมาเพื่อใช้งานกับ โทรทัศน์ที่มีความคมชัดสูงหรือที่เรียกว่า HDTV (High-Definition Television) เป็นโทรทัศน์ดิจิทัลชนิดใหม่ แต่มาตรฐานนี้ไม่เป็นที่ยอมรับเท่าที่ควร

MPEG-4

เป็นมาตรฐานที่ใกล้เคียงกับ Quick Time ที่พัฒนาโดย Apple เพื่อใช้งานทางด้านมัลติมีเดียที่มีแบนวิธต่ำ ซึ่งสามารถรวมภาพ เสียง และส่วนประกอบอื่นที่คอมพิวเตอร์สร้างขึ้นได้ ที่สำคัญก็คือ MPEG-4 ถูกออกแบบให้มีความสามารถในการเข้ารหัสได้ด้อยกว่าวีดิโอต่างๆในภาพได้

MPEG-7

เป็นตัวเชื่อมรายละเอียดเนื้อหาของมัลติมีเดียเข้าด้วยกัน โดยมีจุดมุ่งหมายที่จะสร้างมาตรฐานการอธิบายข้อมูลข่าวสารของมัลติมีเดีย เพื่อใช้ในการสนับสนุนความหมายของข้อมูลข่าวสารต่างๆบนสื่อ

- **Microsoft Video**

ทำงานในขั้นตอนการบีบอัดข้อมูลที่อัตราส่วนการบีบอัดต่ำได้อย่างรวดเร็ว เหมาะสำหรับภาพที่มีการเคลื่อนไหวมากๆ แต่ความละเอียดต่ำ (240*180 พิกเซล)

- **Microsoft RLE**

ใช้อัตราส่วนในการบีบอัดต่ำ เหมาะสำหรับภาพเคลื่อนไหวต่างๆที่มีความชัดเจน แต่ไม่เหมาะกับงานวีดิโอ

- **DV Format**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นระบบเข้ารหัสเพื่อการบันทึกเป็นสัญญาณดิจิทัลโดยตรง เพื่อใช้กับกล้องถ่ายวิดีโอแบบดิจิทัล ซึ่งเรียกการเข้ารหัสแบบนี้ว่า DV Format โดยสัญญาณที่ถูกบันทึกจะผ่านการบีบอัดข้อมูลเรียบร้อยแล้ว สามารถส่งผ่านเข้าสู่คอมพิวเตอร์ได้โดยตรงไม่มีปัญหาการสูญเสียความคมชัดของภาพ แต่ข้อมูลของภาพดิจิทัลวีดีโอค่อนข้างใหญ่การส่งผ่านข้อมูลจะใช้เวลานาน จึงได้มีการพัฒนา IEEE 1394 หรือที่เรียกว่า Fire Wire มารองรับการส่งข้อมูลแบบ DV จนกระทั่ง Fire Wire ได้กลายเป็น มาตรฐานการเชื่อมต่อสำหรับกล้องดิจิทัลวีดีโอในที่สุด

- **Div X**

กลุ่มโปรแกรมเมอร์ได้ค้นได้ร่วมกันพัฒนาระบบซอฟต์แวร์ CODEC แบบใหม่ ซึ่งสามารถลดข้อมูลเหลือเพียง 10-20 เปอร์เซ็นต์ของปริมาณข้อมูลเดิม และยังสามารถเปิดชมภาพยนตร์ด้วยโปรแกรมธรรมดา เช่น Media Player ได้อีกด้วย

- **DVI**

เป็นเทคโนโลยี CODEC ที่ถูกพัฒนาโดยIntel ซึ่งมีมาตรฐาน NTST ในการแสดงภาพที่มีอัตรา 30 เฟรมต่อวินาที สามารถบันทึกและแสดงภาพวีดีโอที่มีการเคลื่อนไหวที่สมจริงเหมือนในโทรทัศน์ แต่บางครั้งมักจะเกิดปัญหาเพราะเนื้อที่ของฮาร์ดดิสก์ ไม่เพียงพอ เนื่องจากสามารถบันทึกข้อมูลได้ในปริมาณมาก ดังนั้น DVI ด้วยอุปกรณ์ที่เป็นฮาร์ดแวร์ทั้งหมด

- **Cinepak**

เป็นเทคโนโลยีการบีบอัดและการคลายข้อมูลที่ถูกพัฒนาโดย SuperMac Inc. สามารถส่งข้อมูลวีดีโอขนาด 24 บิต บนพื้นที่ขนาด 1 ต่อ 4 ของจอภาพวินโดว (320*240พิกเซล) ซึ่งนิยมใช้ในรูปแบบของไฟล์วีดีโอ ที่เป็น *.avi โดยสามารถบีบอัดข้อมูลได้ดีแม้มีข้อเสียตรงที่ใช้เวลานานในการบีบอัดข้อมูล

- **Indeo**

ถูกพัฒนาโดย Intel Corporation ซึ่งมีพื้นฐานมาจาก DVI ที่เป็นฮาร์ดแวร์ล้วนๆ ส่วนการเข้าและถอดรหัสของ Indeo จะเป็นซอฟต์แวร์ทั้งหมด โดยนิยมนำมาประยุกต์ใช้ในการประชุมด้วยภาพ (Video Conferencing)

บทที่ 3

จาวามีเดียเฟรมเวิร์ก-เจเอ็มเอฟ (Java Media Framework-JMF)

3.1 ไทม์เบสมีเดีย

ไทม์เบสมีเดีย คือ ข้อมูลชนิดใดๆที่มีการทำงานสัมพันธ์กับเวลา เช่น ไฟล์เสียง ไฟล์ภาพยนตร์ไฟล์ภาพเคลื่อนไหวต่างๆ โดยเมื่อทำงานกับข้อมูลเหล่านี้ ระบบจำเป็นต้องคำนึงถึงเวลาในการส่งและรับข้อมูลเข้าไปในโปรเซส ไม่ให้เกิดการติดขัดหรือล่าช้า ซึ่งจะส่งผลให้ข้อมูลที่แสดงออกมามีข้อผิดพลาดได้ ซึ่งอัตราการส่งข้อมูลจะต้องคงที่ เราจึงเรียกรูปแบบการส่งข้อมูลนี้ว่า สตรีมมีเดีย (Streaming Media) และเรียกข้อมูลที่ส่งไปว่า มีเดียสตรีม (Media Stream) ซึ่งประกอบไปด้วยไฟล์หลายรูปแบบ (Format) มีทั้งรูปแบบภาพ และรูปแบบเสียงโดยเราจะเรียกรูปแบบเหล่านี้ว่า คอนเทนต์ไทป์ (Content type) หรือ ไฟล์ไทป์ (File Type) ซึ่งไฟล์นั้นมักประกอบไปด้วยหลายแชนเนล (Multiple channels) เราเรียกแต่ละแชนเนลว่า แทร็ก (Track) และเรียกไฟล์นั้นว่า มัลติเพล็กซ์ไฟล์ ซึ่งประกอบด้วยหลายแทร็ก ที่มีรูปแบบของข้อมูลที่แตกต่างกันออกไป

องค์ประกอบหลักภายในไทม์เบสมีเดีย (Time-based Media) ได้แก่

1. อินพุต (Input) มีได้หลายทางคือ ทางอุปกรณ์ต่างๆเช่น ไมโครโฟน หรือ กล้องวิดีโอ การอ่านจากไฟล์ การรับข้อมูลจากเครือข่าย
2. โปรเซส (Process) ทำหน้าที่ต่างๆได้แก่ เพิ่มเอฟเฟกต์ (effect) บีบอัดข้อมูลเข้าหรือขยายข้อมูลออก เปลี่ยนรูปแบบของไฟล์
3. เอาท์พุท (Output) ไปได้หลายทางคือ ให้เสียงและรูปภาพ บันทึกลงสู่ไฟล์ และส่งข้อมูลไปในระบบเครือข่าย



รูปที่ 3-1 แสดงองค์ประกอบหลักและการทำงานกับข้อมูลชนิดไทม์เบสมีเดีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีเดียสตรีม สามารถแบ่งออกได้เป็น 2 ประเภทตามทิศทางการส่งข้อมูล ได้แก่

- 1.Pull ข้อมูลจะถูกส่งและควบคุมจากฝั่งไคลเอนต์
- 2.Push ข้อมูลจะถูกส่งและควบคุมจากฝั่งเซิร์ฟเวอร์

คุณภาพของมีเดียสตรีมนั้น ขึ้นอยู่กับปัจจัยต่างๆ ดังนี้

- 2.1 วิธีการบีบอัดข้อมูล
- 2.2 สมรรถภาพของโปรเซสในการเล่นไฟล์
- 2.3 แบนด์วิธที่มีอยู่

ดังนั้นจึงจำเป็นที่จะต้องคำนึงถึงทรัพยากรภายในระบบที่จะทำงานกับข้อมูลจำพวกนี้อย่างระมัดระวังด้วย เนื่องจากในบางลักษณะงานถือความถูกต้องของตัวข้อมูลเป็นเงื่อนไขอันดับแรก

การโปรเซสมีเดียสตรีม มีการกระทำต่างๆ ดังนี้

1. ถ้าสตรีมข้อมูลเป็นแบบมัลติเพล็กซ์ (Multiplex) จะต้องมีการแยกออกมาเป็นแต่ละแทร็ก
2. ถ้าแต่ละแทร็กถูกบีบอัดอยู่ จะต้องนำมาถอดรหัสเสียก่อน
3. การเปลี่ยนรูปแบบอาจเกิดขึ้น ถ้างานนั้นๆจำเป็นต้องใช้
4. การเพิ่มเอฟเฟกต์ ใช้โดยคำนึงถึงคุณภาพงานที่ต้องการ

3.2 องค์ประกอบภายในโปรเซส ได้แก่

3.2.1 ดิมัลติเพล็กซ์เซอร์และมัลติเพล็กซ์เซอร์ (Multiplexer and Demultiplexer)

ดิมัลติเพล็กซ์เซอร์ ทำหน้าที่ในการนำแต่ละแทร็กออกมาจากมีเดียสตรีม และมัลติเพล็กซ์เซอร์ทำหน้าที่ในการนำแต่ละแทร็กรวมกลับไปเป็นมีเดียสตรีม

3.2.2 โคเด็ค (Codec)

โคเด็ค ทำหน้าที่ในการบีบอัดและขยายข้อมูล เมื่อแต่ละแทร็กถูกเข้ารหัส (Encode) มันจะทำการเปลี่ยนรูปแบบข้อมูลให้อยู่ในรูปแบบที่เหมาะสมสำหรับงานนั้นๆ และเมื่อแต่ละแทร็กถูกถอดรหัส (Decode) มันจะทำการเปลี่ยนรูปแบบข้อมูลไปให้เหมาะสมกับการนำเสนอ

3.2.3 เอฟเฟกต์ฟิลเตอร์ (Effect Filters)

เอฟเฟกต์ฟิลเตอร์จะทำการแก้ไขคัดแปลงแต่ละแทร็กของข้อมูล เพื่อให้เกิดเป็นลักษณะพิเศษที่ต้องการ เช่น การทำภาพเบลอ การทำเอคโโล เป็นต้น

3.2.4 เร็นเดอร์เรอร์ (Renderer)

เร็นเดอร์เรอร์เป็นเหมือนตัวแทนของการนำเสนอข้อมูล โดยจะทำหน้าที่แทนการคำนวณเสียงและลำโพงในการนำเสนอข้อมูลเสียง และจะทำหน้าที่เสมือนจอภาพสำหรับการนำเสนอภาพ

3.3 เจเอ็มเอฟ คืออะไร

เจเอ็มเอฟ คือ เอพีไอ* ที่ทำหน้าที่ในการจัดการกับข้อมูลจำพวกไทม์เบสมีเดีย (time-based media) ในภาษาจาวา (Java) สำหรับนำเสนอผ่านทางอุปกรณ์ต่างๆ ขึ้นกับชนิดและรูปแบบของข้อมูล

เจเอ็มเอฟ ได้จัดการในเรื่องของโครงสร้างและโปรโตคอลในการทำงานสำหรับจัดการกับการโปรเซส และการส่งข้อมูลจำพวกไทม์เบสมีเดีย และถูกออกแบบมาให้รองรับกับไฟล์ในรูปแบบต่างๆ เช่น AIFF,

AU, AVI, GSM, MIDI, MPEG, QuickTime, RMF และ WAV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 องค์ประกอบภายในเจเอ็มเอฟ

3.4.1 เพลเยอร์ (Player)

เพลเยอร์ คือ ออฟเจ็ทที่มีหน้าที่หลักสำหรับรองรับการทำงานภายในกระบวนการต่างๆของเจเอ็มเอฟ โดยจะมีส่วนที่ย่อยลงไปทีเรียกว่า โพรเซสเซอร์ (Processor) รองรับการทำงานในระดับล่างกว่า

3.4.2 ดาต้าซอร์ส (DataSource)

DataSource เป็นชนิดข้อมูลชนิดหนึ่งภายในเจเอ็มเอฟ ที่ทำหน้าที่รองรับการนำมีเดียสตรีมต่างๆที่จะมาทำงานภายในเจเอ็มเอฟ เข้ามา เช่น การที่เจเอ็มเอฟใช้ DataSource ในการจัดการเรื่องการส่งเนื้อหาข้อมูล โดยจะเอนแคพซูลตำแหน่งของข้อมูล โปรโตคอลที่ใช้และซอฟต์แวร์ที่ใช้ในการส่งไว้ด้านใน ซึ่งทำให้ไม่สามารถถูกนำไปใช้ใหม่ได้ ซึ่งถูกแทนด้วย มีเดียโลเคเตอร์ (MediaLocator) เป็นต้น

3.4.3 ดาต้าซิงก์ (DataSink)

DataSink เป็นข้อมูลเป็นชนิดข้อมูลชนิดหนึ่งภายในเจเอ็มเอฟ ที่ทำหน้าที่รองรับการนำมีเดียสตรีมต่างๆ ออกไปสู่ภายนอก เช่น การบันทึกไฟล์ เป็นต้น

3.4.4 มีเดียโลเคเตอร์ (MediaLocator)

มีเดียโลเคเตอร์ คือ ตำแหน่งของข้อมูลที่ทำงานกับเจเอ็มเอฟผ่านทางDataSource และ DataSink

3.4.5 เมเนเจอร์ (Manager)

Manager เป็นอินเตอร์เฟสที่กำหนดพฤติกรรมและรูปแบบการทำงานของโปรเซสในแต่ละชนิดไว้ โดยMangerที่ถูกใช้ภายในเจเอ็มเอฟ ได้แก่

- Manager รองรับการสร้างเพลเยอร์, โพรเซสเซอร์, ดาต้าซอร์สและดาต้าซิงก์
 - PackageManager สำหรับการจัดการกับบริจิสทรีภายในแพ็คเกจที่มีคลาสของเจเอ็มเอฟ เช่น เพลเยอร์, โพรเซสเซอร์, ดาต้าซอร์ส และดาต้าซิงก์
 - CaptureDeviceManager สำหรับจัดการกับบริจิสทรีสำหรับอุปกรณ์การบันทึก (Capture Device)
 - PlugInManger สำหรับจัดการกับบริจิสทรีในคอมโพเนนท์ปลั๊กอิน (Plug-in)ของเจเอ็มเอฟ เช่น มัลติเพล็กซ์เซอร์, ดีมัลติเพล็กซ์เซอร์, โคเด็ก, เอฟเพ็ก และเรนเดอร์เรอร์
- ในการพัฒนาโครงการนี้ ได้ใช้ออฟเจ็ทManager สำหรับการกระทำต่าง ดังตัวอย่าง

```
player = Manager.createPlayer(mediaLocator);
```

ใช้สำหรับสร้างเพลเยอร์ที่สำหรับทำงานกับเจเอ็มเอฟ

```
processor = Manager.createRealizedProcessor(new ProcessorModel(
inSource, outFormats, outFileType ) );
```

ใช้สำหรับนำให้โปรเซสเข้าสู่สถานะที่พร้อมจะทำงาน

```
// create data source from MediaLocator of device
inSource = Manager.createDataSource(deviceInfo.getLocator() );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้สำหรับสร้างค่าค่าขอสมารองรับอินพุท

```
// put available devices' information into vector
deviceList=CaptureDeviceManager.getDeviceList( null );
```

ใช้สำหรับจัดการเรื่องการเลือกรายการอุปกรณ์ที่จะนำมาใช้ในการนำอินพุทมาจากอุปกรณ์
จำพวกไมโครโฟน

3.4.6 อีเวนต์โมเดล (Event Model)

ใช้สำหรับการรายงานสถานะของข้อมูลที่กำลังกระทำด้วย และตอบสนองต่อข้อผิดพลาดต่างๆที่เกิดขึ้น โดยการส่งค่าMediaEvent ออกมา ซึ่งจะมีหลายชนิดขึ้นกับลักษณะของเหตุการณ์

3.4.7 คอนโทรล (Controls)

เจเอ็มเอฟ มีการจัดการในเรื่องการกำหนดค่าให้แต่ละแอททริบิวต์ (Attributes)ของออปเจ็ก นอก
จากนี้ยังทำหน้าที่ในการควบคุมการเข้าถึงส่วนของอินเตอร์เฟส ที่ติดต่อกับผู้ใช้ที่ทำให้ผู้ใช้สามารถควบคุม
แอททริบิวต์ของออปเจ็กได้ ออปเจ็กของเจเอ็มเอฟหลายตัวใช้ Controls เช่น ออปเจ็กController, ออป
เจ็กDataSource, ออปเจ็กDdatasink เป็นต้น Control มีรูปแบบการกำหนดค่าและค่าเฉพาะของแต่ละรูปแบบ
ของข้อมูลอยู่ เพื่อรองรับการทำงานในรูปแบบต่างๆที่จำเป็น โดยในโครงงานนี้ได้ใช้ในส่วนของยูส
เซอร์อินเตอร์เฟส และการควบคุมไฟล์ให้เหมาะสมกับรูปแบบของไฟล์ ดังตัวอย่าง

```
formatControls = ( ( CaptureDevice ) inSource ).getFormatControls();
```

ใช้สำหรับการเลือกรูปแบบไฟล์ที่ต้องการทำงานด้วย

```
mediaControl = player.getControlPanelComponent();
```

ใช้สำหรับเรียกคอมโพเนนต์ที่ทำหน้าที่ในการควบคุมของเจเอ็มเอฟมาใช้

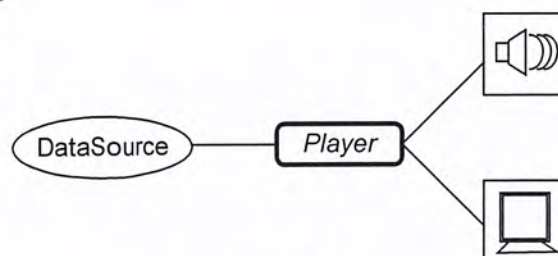
3.4.8 กระบวนการนำเสนอ (Presentation Process)

ในกระบวนการของการนำเสนอ นั้นถูกกำหนดโดยอินเตอร์เฟสคอนโทรลเลอร์ (Controller) ซึ่ง
ประกอบไปด้วยการควบคุม การนำเสนอ การบันทึกข้อมูลใหม่เบสมีเดีย

3.4.9 คอนโทรลเลอร์ (Controller)

เป็นอินเตอร์เฟสที่ถูกใช้สำหรับกระบวนการนำเสนอของเจเอ็มเอฟ ซึ่งได้กำหนดสถานะพื้นฐาน
พร้อมทั้งการควบคุมไว้ให้ สำหรับแต่ละออปเจ็ก ในการทำงาน MediaEventจะเป็นตัวบอกการเปลี่ยน
สถานะของใหม่เบสมีเดีย ซึ่ง ControllerListener เป็นอินเตอร์เฟสที่มารองรับ

3.4.10 เพลเยอร์ (Player)



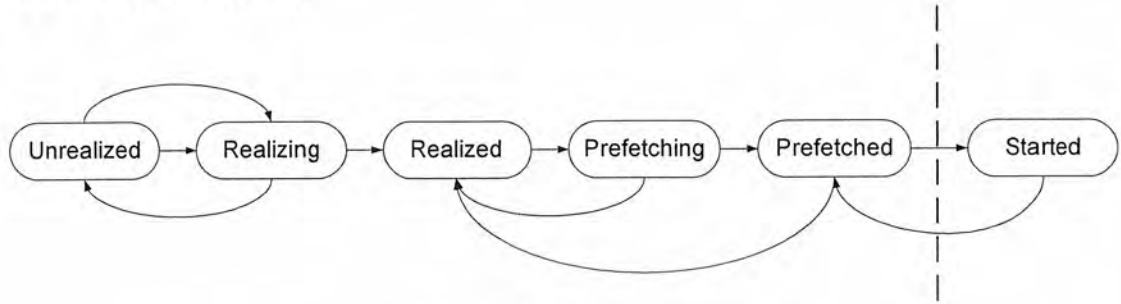
Player Model

รูปที่ 3-2 แสดงภาพรวมของเพลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพลเยอร์ จะรับโทรมีเดียที่เข้ามาเป็นอินพุตจากคาล์ซอส และแสดงออกมาในเวลาที่ต้องการด้วยอุปกรณ์ต่างๆกันไปขึ้นกับชนิดของข้อมูลนั้นๆ ซึ่งการทำงานของเพลเยอร์นั้นจะมีทั้งหมด 5 สถานะดังนี้

3.4.10.1 สถานะของเพลเยอร์

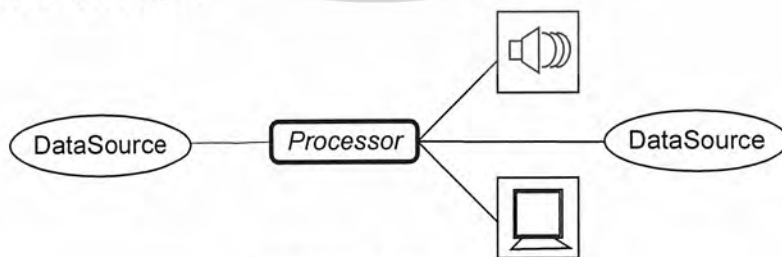


Player States

รูปที่ 3-3 แสดงสถานะภายในเพลเยอร์

1. **Unrealized** ในสถานะนี้เพลเยอร์จะถูกสร้างขึ้น แต่จะยังไม่รู้รายละเอียดเกี่ยวกับโทรมีเดียที่เข้ามา
2. **Realizing** เมื่อเรียกคำสั่ง realize เพลเยอร์จะเปลี่ยนสถานะจากUnrealized มาเป็น Realizing ซึ่งเป็นสถานะที่ให้โปรเซสจัดการกับความต้องการทรัพยากรของระบบสำหรับมีเดียต่างๆ และสำหรับในบางทรัพยากรที่ให้ใช้ได้เฉพาะช่วงเวลาที่ทำงานเท่านั้น จะถูกจัดการเมื่ออยู่ในสถานะ Prefetching แทน
3. **Realized** ในสถานะนี้ เพลเยอร์จะรู้วิธีการเรนเดอร์ข้อมูลแล้ว ทำให้สามารถหาลงค์ประกอบที่ช่วยในการควบคุมเตรียมไว้ก่อนได้
4. **Prefetching** เมื่อมีการเรียกคำสั่ง prefetch เพลเยอร์จะเตรียมตัวนำเสนอมีเดียที่เข้ามาโดยการเคลื่อนย้ายข้อมูลเข้ามาเตรียมไว้ก่อน และเริ่มจับจองทรัพยากรที่ต้องการใช้
5. **Prefetched** ในสถานะนี้ เพลเยอร์พร้อมที่จะเริ่มทำงานแล้ว
6. **Started** เมื่อเรียกคำสั่ง start เพลเยอร์พร้อมที่จะนำเสนอมีเดียแล้ว โดยรอเพียงเวลาที่เหมาะสมเท่านั้น

3.4.11 โปรเซสเซอร์ (Processor)



Processor Model

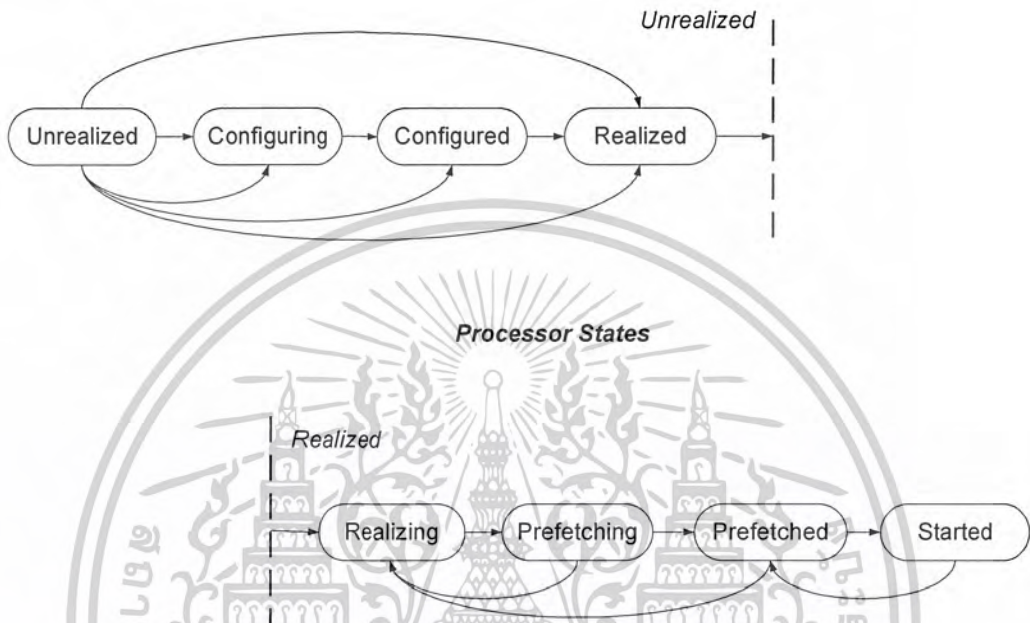
รูปที่ 3-4 แสดงภาพรวมของโปรเซสเซอร์

โปรเซสเซอร์ เป็นเพลเยอร์ชนิดหนึ่งที่มีความสามารถในการควบคุมกระบวนการที่เกิดขึ้นกับข้อมูลอินพุต ซึ่งนอกจากจะสามารถส่งเอาท์พุตออกไปเป็นภาพและเสียงตามชนิดของข้อมูลได้แล้ว เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถส่งเอาต์พุตออกไปเป็น DataSource ที่สามารถนำไปทำงานต่อกับโปรเซสเซอร์หรือ เฟลเยอร์ตัวอื่นๆ ได้อีกด้วย

3.4.11.1 สถานะของโปรเซสเซอร์

จะมีสถานะที่ถูกเพิ่มเติมขึ้นมาอีก 2 สถานะ คือ



รูปที่ 3-5 แสดงสถานะของโปรเซสเซอร์

1. Configuring สถานะนี้จะเริ่มขึ้นเมื่อเรียกคำสั่ง configure ซึ่งทำให้โปรเซสเซอร์ติดต่อยังดาต้าซอสและทำการคิดเฟล็กซ์มีเดียที่เป็นอินพุท และเตรียมรายละเอียดของรูปแบบของมีเดียต่างๆ ไว้

2. Configured โดยทั้งสองสถานะนี้ถูกเพิ่มก่อนสถานะ Realizing

ในการนำเสนอโคมมีเดีย จะใช้เฟลเยอร์ ในการควบคุมกระบวนการต่างๆ หากต้องการที่จะทำงานกับสตรีมหลายสตรีม จะต้องสร้างเฟลเยอร์แยกกัน โดยจะมีโปรเซสเซอร์ที่มาทำการควบคุมการนำเสนอมีเดียในรายละเอียดอีกที

เมื่อต้องการเล่นมีเดียสตรีม จะต้องมีการสร้างเฟลเยอร์สำหรับสตรีมข้อมูลนั้นๆ โดยเฟลเยอร์ที่พร้อมจะทำงานได้นั้นจะต้องผ่านสถานะ Realized มาก่อน โดยการใช้คำสั่งดังนี้

```
Manager.createRealizedProcessor();
```

เจเอ็มเอฟมีคอมโพเนนต์สำหรับจัดการเกี่ยวกับส่วนที่ติดต่อกับผู้ใช้ ซึ่งสามารถแบ่งได้เป็น 2 ส่วนหลักๆคือ

1. Visual Component เป็นส่วนที่รองรับเกี่ยวกับการนำเสนอมีเดียสตรีมออกมา เช่น การแสดงภาพ หรือ การส่งเสียงออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Control Panel Component เป็นส่วนที่ควบคุมมีเดียสตรีมที่กำลังนำเสนอ โดยทำให้ผู้ใช้สามารถ ดูรายละเอียด หรือ ควบคุมได้

ตัวอย่างแสดงการเรียกใช้งานส่วนอินเทอร์เฟซที่ติดต่อกับผู้ใช้

```
visualMedia = player.getVisualComponent();
if (visualMedia != null)
    container.add(visualMedia, BorderLayout.CENTER);

mediaControl = player.getControlPanelComponent();
if (mediaControl != null)
    container.add(mediaControl, BorderLayout.NORTH);
```

เมื่อจะให้เพลเยอร์ที่สร้างขึ้นเริ่มทำงาน เราจะใช้คำสั่ง

```
Player.start();
```

และเมื่อต้องการให้เพลเยอร์หยุดทำงาน เราจะใช้คำสั่ง

```
Player.stop();
```

และในการคืนทรัพยากรให้กับระบบ เราจะใช้ คำสั่ง

```
Player.deallocate();
```

ซึ่งจะต้องเรียกหลังจากที่มีการหยุดการทำงานของเพลเยอร์แล้วเท่านั้น และหากมีการเรียกใช้คำสั่งนี้ในขณะที่อยู่ในสถานะ Prefetching หรือ Prefetched เพลเยอร์จะกลับไปอยู่ในสถานะ Realized แต่ถ้าหากมีการเรียกใช้คำสั่งนี้ในขณะที่เพลเยอร์อยู่ในสถานะ Realizing เพลเยอร์จะกลับไปอยู่ในสถานะ Unrealized

บทที่ 4

โครงสร้างและการออกแบบโปรแกรม

4.1 เทคนิคหรือเทคโนโลยีที่ใช้

- ภาษาจาวา
- สวิงก์ สำหรับการมีส่วนเชื่อมประสานทางกราฟฟิคติดต่อกับผู้ใช้
- จาวามีเดียเฟรมเวิร์ค

4.2 เครื่องมือที่ใช้ในการพัฒนา

- เจบิวเดอร์เวอร์ชัน 7.0 (JBuilder 7.0) ใช้สำหรับการเขียนภาษาจาวา
- เจเอ็มสตูดิโอ (JMStudio)

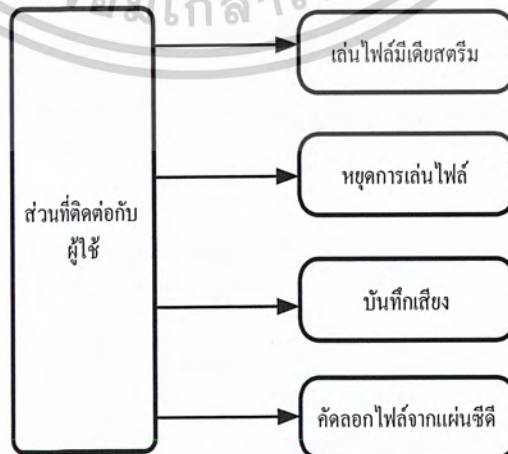
4.3 รายละเอียดของอินพุตและเอาต์พุต

อินพุต

1. คำสั่งในการเล่นไฟล์มีเดียสตรีม
2. คำสั่งในการหยุดเล่นไฟล์มีเดียสตรีม
3. คำสั่งในการบันทึกเสียงร้องของผู้ร้อง
4. คำสั่งในการคัดลอกไฟล์จากแผ่นวีซีดีลงไว้ในเครื่องคอมพิวเตอร์
5. เสียงร้องของผู้ร้องที่เข้ามาทางไมโครโฟน

เอาต์พุต

1. การแสดงผลการเล่นไฟล์มีเดียสตรีม ที่อยู่ในรูปแบบต่างๆ
2. การหยุดเล่นไฟล์มีเดียสตรีม
3. ไฟล์ที่ได้จากการบันทึกเสียงของผู้ร้อง
4. ไฟล์ที่ได้จากการคัดลอกไฟล์ต้นฉบับในแผ่นวีซีดีลงสู่เครื่องคอมพิวเตอร์
5. เสียงร้องของผู้ร้องที่ออกทางลำโพง



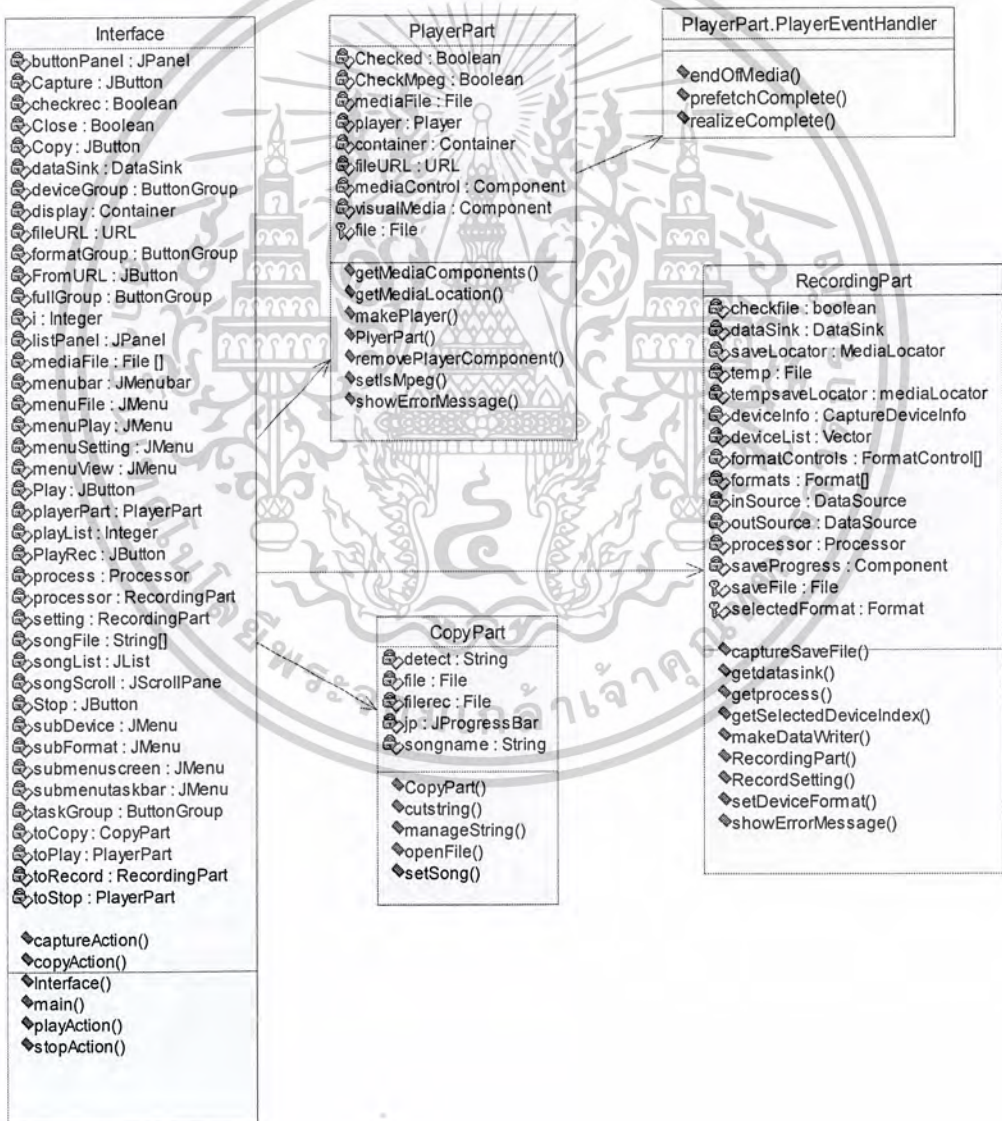
รูปที่ 4-1 แสดงหน้าที่หลักภายในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแจเพลเยอร์มีความสามารถ ดังนี้

1. เล่นไฟล์มีเดียสตรีม โดยอาศัยเพลเยอร์ของเจเอ็มเอฟ ในการควบคุมการแสดงผลทั้งภาพและเสียง
2. หยุดการเล่นไฟล์ เป็นโปรเซสในการหยุดการเล่นของเพลเยอร์ของเจเอ็มเอฟที่กำลังทำงานอยู่ในขณะนั้น
3. บันทึกเสียงผู้ร้อง เป็นโปรเซสที่ทำหน้าที่ในการบันทึกเสียงของผู้ร้อง
4. คัดลอกไฟล์จากแผ่นซีดี เป็นโปรเซสการทำงานในการคัดลอกไฟล์จากแผ่นซีดีลงในเครื่องคอมพิวเตอร์ของตนเอง

โปรแกรมแจเพลเยอร์ว่าไรตี้ ได้มีการแบ่งการจัดการกับการกระทำต่างๆภายในโปรแกรม ออกเป็นคลาสต่างๆตามความสามารถของโปรแกรม ดังรูป



รูปที่ 4-2 แสดงคลาสภายในโปรแกรมแจเพลเยอร์ว่าไรตี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสต่างๆในโปรแกรม มีรายละเอียดดังนี้

1.Interface

การทำงานในส่วนอินเทอร์เฟซและเมนโปรแกรม การทำงานในส่วนนี้จะเป็นการสร้างและกำหนดรูปแบบการทำงานของอินเทอร์เฟซซึ่งประกอบไปด้วย ปุ่มต่าง ๆ เมนูต่าง ๆ และส่วนส่วนแสดงภาพ ซึ่งอธิบายรายละเอียดต่าง ๆ ได้ดังนี้ เริ่มแรกจะเป็นการสร้างเมนูต่าง ๆ ของโปรแกรมโดยประกอบไปด้วย

เมนู File สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + f ภายในเมนูประกอบด้วย

- เมนูไอเท็ม Open File ทำหน้าที่ในการเลือกไฟล์ที่ต้องการเล่นซึ่งสามารถเลือกได้ที่ละหลาย ๆ ไฟล์ด้วยกันโดยเมื่อทำการเลือกเสร็จแล้วไฟล์ที่เลือกก็จะไปแสดงยังลิสเพลงของโปรแกรม สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + O

- เมนูไอเท็ม Exit ทำหน้าที่ในการออกจากโปรแกรม สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + x

เมนู Play สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + p ภายในเมนูประกอบด้วย

- เมนูไอเท็ม Play ทำหน้าที่ในการเล่นไฟล์ที่ถูกเลือกจากลิสเพลง โดยการทำงานจะไปเรียกเมทอด playAction สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + l

- เมนูไอเท็ม Stop ทำหน้าที่หยุดการเล่นไฟล์ โดยการทำงานจะไปเรียกเมทอด stopAction สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + t

- เมนูไอเท็ม Record ทำหน้าที่ในการบันทึกเสียงของผู้ใช้โดยก่อนที่จะใช้งานในส่วนนี้จะต้องไปกำหนดรายละเอียดของการบันทึกในเมนู Setting/Record ก่อน การทำงานในส่วนนี้จะไปเรียกเมทอด captureAction สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + e

- เมนูไอเท็ม Copy ทำหน้าที่ในการแปลงไฟล์จากรูปแบบตัดเป็นเอ็มเปคแล้วบันทึกไฟล์ลงยังเครื่อง การทำงานในส่วนนี้จะไปเรียกใช้เมทอด copyAction สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + o

เมนู Setting สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + s ภายในเมนูประกอบด้วย

- เมนูไอเท็ม Record ทำหน้าที่ในการกำหนดรายละเอียดต่าง ๆ ในการบันทึกเสียงของผู้ใช้ การทำงานจะไปเรียกเมทอด RecordSetting จาก setting ซึ่งเป็นออบเจ็คของ RecordingPart สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + r

- เมนูไอเท็ม About ทำหน้าที่ในการแสดงรายละเอียดของผู้สร้างและพัฒนาโปรแกรม สามารถเรียกใช้งานด้วยคีย์บอร์ดโดยกดปุ่ม Alt + a

ลิสของเพลง การทำงานในส่วนลิสของเพลงนั้นจะรับไฟล์มาจากไฟล์ที่ถูกเลือกจากเมนู Open file โดยไฟล์ต่าง ๆ นั้นจะมาแสดงยังลิสบ็อก จากนั้นเมื่อมีการเลือกไฟล์ในลิสบ็อก ค่าอินเด็กซ์ของไฟล์ที่ถูกเลือกจะถูกส่งมาที่ออบเจ็กต์ playlist เพื่อใช้ในการเล่นไฟล์ต่อไป

จากนั้นจะมีการสร้างพาเนล (panel) สำหรับสร้างปุ่มควบคุมต่าง ๆ โดยกำหนดให้มีพื้นหลังเป็นสีดำ และตำแหน่งของพาเนลอยู่ทางด้านล่างสุดในส่วนอินเทอร์เฟซของโปรแกรม

ปุ่มต่าง ๆ ของโปรแกรม ประกอบด้วย

- ปุ่มสำหรับเล่นไฟล์ ทำหน้าที่สำหรับเล่นไฟล์ที่ถูกเลือกจากลิสของเพลง การทำงานในส่วนนี้เมื่อมี ActionEvent เข้ามาจะไปเรียกเมทอด playAction เพื่อทำการเล่นไฟล์ต่อไป
- ปุ่มสำหรับบันทึกเสียงพร้อมกับเล่นไฟล์เพลงไปด้วย ทำหน้าที่ในการเริ่มบันทึกเสียงของผู้ใช้พร้อมกับเล่นไฟล์ที่ถูกเลือกจากลิสของเพลงไปพร้อม ๆ กัน โดยก่อนที่จะทำการบันทึกได้นั้นจะต้องทำการกำหนดรายละเอียดต่าง ๆ ในการบันทึกเสียงที่เมนู SettingRecord ก่อน การทำงานในส่วนนี้เมื่อมี ActionEvent เข้ามาจะไปเรียกเมทอด playAction และ captureAction พร้อม ๆ กัน
- ปุ่มสำหรับหยุดการเล่นไฟล์และหยุดการบันทึกเสียง ทำหน้าที่ในการหยุดการเล่นไฟล์ที่ทำการเล่นอยู่ และ/หรือ หยุดการบันทึกเสียงของผู้ใช้ การทำงานในส่วนนี้เมื่อมี ActionEvent เข้ามาจะไปเรียกเมทอด StopAction เพื่อทำการหยุดการเล่นไฟล์
- ปุ่มสำหรับบันทึกเสียง ทำหน้าที่ในการเริ่มบันทึกเสียงของผู้ใช้ โดยก่อนที่จะทำการบันทึกได้นั้นจะต้องทำการกำหนดรายละเอียดต่าง ๆ ในการบันทึกเสียงที่เมนู SettingRecord ก่อน การทำงานในส่วนนี้เมื่อมี ActionEvent เข้ามาจะไปเรียกเมทอด CaptureAction
- ปุ่มสำหรับคัดลอกไฟล์ลงยังเครื่อง ทำหน้าที่ในการแปลงไฟล์จากรูปแบบแฉัดเป็นเอ็มเปคแล้วบันทึกไฟล์ลงยังเครื่อง การทำงานในส่วนนี้จะไปเรียกใช้เมทอด copyAction

เมทอดต่าง ๆ ประกอบด้วย

- เมทอด playAction ทำหน้าที่ส่งไฟล์ซึ่งเป็นพารามิเตอร์ที่ถูกระบุโดยค่าอินเด็กซ์ของไฟล์ที่ถูกเลือกจากลิสของเพลง ไปทำงานยังเมทอด makePlayer ซึ่งอยู่ในคลาส PlayerPart ผ่านทางออบเจ็กต์ toPlay
- เมทอด stopAction ทำหน้าที่ตรวจสอบก่อนว่ามีการเล่นไฟล์อยู่หรือไม่ ถ้ามีก็จะทำการเรียกเมทอด removePlayerComponent ซึ่งอยู่ในคลาส PlayerPart ผ่านทางออบเจ็กต์ toStop จากนั้นจะทำการตรวจสอบว่ามีการบันทึกเสียงอยู่หรือไม่ ถ้ามีก็จะทำการเรียกเมทอด stop และ close ซึ่งอยู่ภายในคลาส Processor ผ่านทางออบเจ็กต์ process

- เมθοธ captureAction ทำหน้าที่เรียกใช้เมθοธ getprocess และ getdatasink จากคลาส RecordingPart ผ่านทางออบเจ็ค toRecord โดยจะคืนค่า processor และ DataSink กลับมายังออบเจ็ค process และ dataSink ตามลำดับ
- เมθοธ copyAction ทำหน้าที่เรียกเมθοธ openFile โดยส่งพารามิเตอร์สตรงซึ่งเป็นไทม์เท็ลของหน้าตา่ openfile ดังกล่าว และเรียกเมθοธ setSong ซึ่งอยู๋ในคลาส CopyPart ผ่านทางออบเจ็ค toCopy

2.PlayerPart

Class PlayerPart เป็นคลาสที่ทำหน้าที่ในการจัดการเกี่ยวกับการเล่นไฟล์ทั้งหมดที่ความสามารถของโปรแกรมทำได้ โดยภายในคลาสจะมีเมธอดที่ทำหน้าที่ต่างๆรองรับอยู่ได้แก่

- เมθοธ ShowErrorMessage ทำหน้าที่ในการแสดงข้อความเตือนในกรณีที่เกิดข้อผิดพลาดขึ้นในการทำงานภายในโปรแกรม
- เมθοธ GetFile ทำหน้าที่ในการเปิดไฟล์ที่จะนำมาเล่นภายในโปรแกรมโดยใช้เมธอดของคลาส JFileChooser
- เมθοธ MakePlayer ทำหน้าที่ในการสร้างเพลเยอร์ที่รองรับการเล่นไฟล์ภายในโปรแกรมในแต่ละครั้ง โดยจะทำการลบเพลเยอร์ตัวเดิมทิ้งก่อนที่จะสร้างตัวใหม่ขึ้นมา โดยใช้เมธอด RemovePlayerComponent และจะมีอินเตอร์เฟสซึ่งทำหน้าที่เป็นListenerสำหรับตรวจสอบสถานะของไฟล์ที่กำลังทำงานด้วย
- เมθοธ RemovePlayerComponent ทำหน้าที่ในการลบคอมโพเนนต์ที่ใช้ในการควบคุมการเล่นไฟล์มีเดียสตรีมออกจากหน้าจอของโปรแกรม
- เมθοธ GetMediaComponents ทำหน้าที่ในการนำคอมโพเนนต์ที่ใช้ในการสร้างและควบคุมภาพและเสียง จากเพลเยอร์ของเจเอ็มเอฟมาใช้

PlayerEventHandler เป็นคลาสที่ทำหน้าที่ในการรองรับEvent ของเพลเยอร์ โดยจะมีเมธอดดังนี้คือ

- เมθοธ RealizeComplete เป็นเมธอดที่ทำหน้าที่ในการทำให้เพลเยอร์อยู่ในสถานะ Realized โดยทำการPrefetch
- เมθοธ PrefetchComplete เป็นเมธอดที่ทำหน้าที่ในการทำให้เพลเยอร์อยู่ในสถานะ Prefetched ที่พร้อมจะเข้าสู่สถานะ Start
- เมθοธ EndOfMedia เป็นเมธอดที่ทำหน้าที่ในการตรวจสอบการสิ้นสุดของไฟล์มีเดียสตรีมที่นำมาเล่นภายในโปรแกรม

3.RecordingPart

RecordingPart เป็นคลาสที่ทำหน้าที่ในการบันทึกเสียงของผู้ร้องลงในเครื่องคอมพิวเตอร์ของผู้ใช้เอง ซึ่งประกอบไปด้วยเมธอดต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมททอด RecordSetting ทำหน้าที่ในการกำหนดเลือกชนิดของอุปกรณ์อินพุตที่จะใช้ในการบันทึกเสียง เลือกฟอร์แมตของไฟล์ และทำการสร้างออฟเจ็กต์ทำหน้าที่ในการควบคุมไฟล์ฟอร์แมตนั้นๆ
- เมททอด GetSelectedDeviceIndex ทำหน้าที่ในการตัวชี้ตำแหน่งของรายการอุปกรณ์อินพุตที่ต้องการออกมา
- เมททอด GetSelectedFormat ทำหน้าที่ในการนำเอารูปแบบของไฟล์ที่เลือกไว้เตรียมสำหรับการนำไปทำงานต่อ
- เมททอด SetDeviceFormat ทำหน้าที่ในการกำหนดรูปแบบของไฟล์ตามที่ได้เลือกไว้ ในขณะที่โปรแกรมในส่วนการบันทึกเสียงนี้ทำงาน
- เมททอด CaptureSaveFile ทำหน้าที่ในการบันทึกเสียงที่ต้องการของผู้ใช้ในขณะร้องในออฟเจ็กต์ inSource ของคลาส DataSource โดยไฟล์ที่ได้จะเป็นไฟล์ในรูปแบบ .wav
- เมททอด MakeDataWriter ทำหน้าที่ในการเขียนไฟล์ที่ได้จากออฟเจ็กต์ inSource เก็บลงไว้ในเครื่องคอมพิวเตอร์ของผู้ใช้โดยผ่านออฟเจ็กต์ outSource
- เมททอด GetSaveFile ทำหน้าที่ในการเขียนไฟล์ลงในเครื่องคอมพิวเตอร์ของผู้ใช้ โดยเมทอดนี้จะโดนเรียกจากเมทอด MakeDataWriter
- เมททอด Getprocess ทำหน้าที่ในการเริ่มการทำงาน ให้แต่ละProcessorของแต่ละเทวด

4. CopyPart

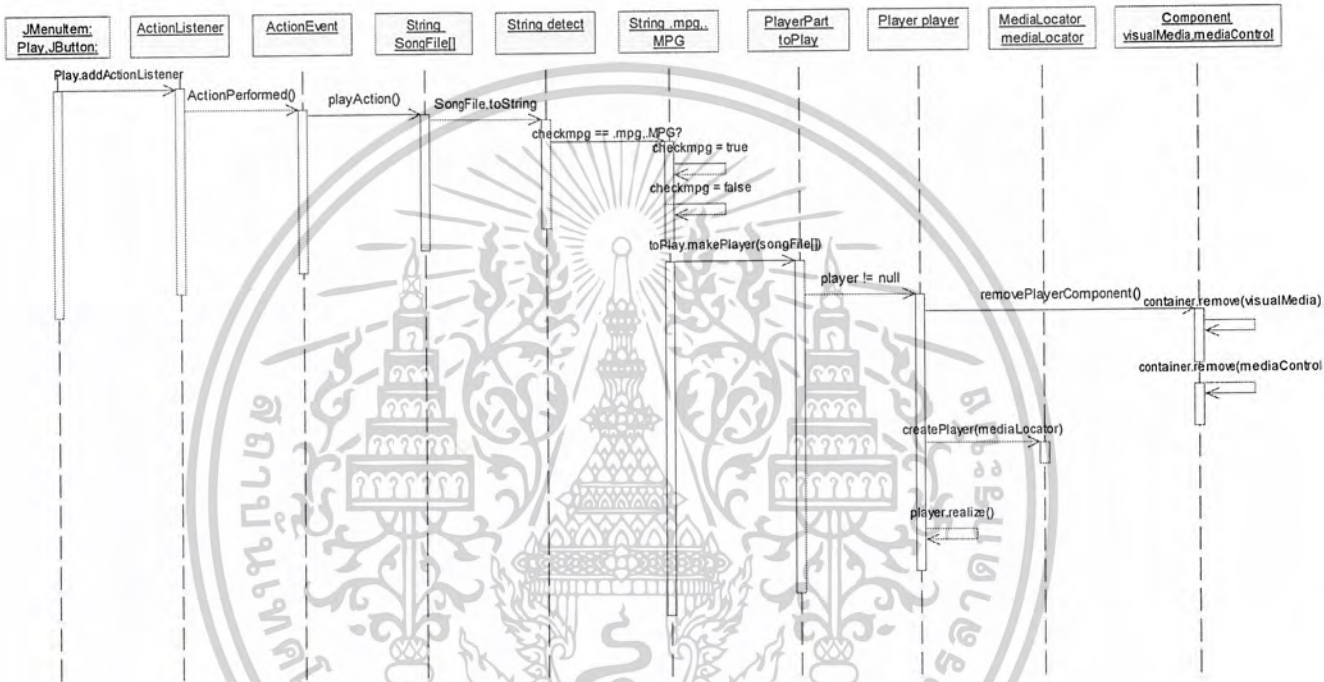
เป็นส่วนของการคัดลอกไฟล์ลงยังเครื่อง (Copy part) ในส่วนนี้จะมีหน้าที่ทำการนำไฟล์ที่ถูกเลือกโดยผู้ใช้ซึ่งไฟล์อยู่ในรูปแบบของ.Dat มาทำการแปลงให้อยู่ในรูปแบบของ .Mpeg ซึ่งจะเรียกใช้โปรแกรม Dat2Mpeg ในการแปลงข้อมูล และเมื่อได้ไฟล์ที่อยู่ในรูปแบบของ.Mpeg แล้วก็จะทำการบันทึกไฟล์นั้นลงยังเครื่องซึ่งได้ทำการระบุไดเรกทอรีไว้ที่ C:\ อธิบายรายละเอียดได้ดังนี้เริ่มต้น จะเป็นการเลือกไฟล์อินพุตโดยทำการเลือกผ่าน JfileChooser ซึ่งไฟล์ที่ถูกเลือกนั้นจะถูกเก็บไว้ที่ออบเจ็กต์ filerec ต่อจากนั้นจะทำการเรียกใช้เมทอด manageString เพื่อทำหน้าที่ในการตรวจสอบความถูกต้องของรูปแบบของไฟล์ โดยเริ่มต้นจะทำการนำชื่อไฟล์ที่ถูกเลือกมาเก็บไว้ยังออบเจ็กต์ detect ซึ่งมีรูปแบบเป็น String จากนั้นจะทำการตรวจสอบนามสกุลของชื่อไฟล์ที่รับมามีนามสกุลเป็น .dat หรือ .DAT หรือไม่ ถ้ามีก็จะสามารถทำงานในขั้นตอนต่อไปได้ ถ้าไม่ก็จะแสดง MessageDialog ว่าไฟล์ที่เลือกนั้นไม่ถูกต้องและจะออกจากการทำงานของโปรแกรมในส่วนนี้ แต่ถ้าไฟล์ที่เลือกนั้นถูกต้องตามเงื่อนไขก็จะทำงานในขั้นตอนต่อไป เมททอด setSong มีหน้าที่รับชื่อไฟล์ใหม่ที่จะบันทึกลงเครื่องจากผู้ ใช้ โดยจะมี Input Dialog แสดงขึ้นมาเพื่อถามชื่อเพลงที่ผู้ใช้งานต้องการบันทึกลงยังเครื่อง โดยผู้ใช้นั้นเพียงแค่กรอกเฉพาะชื่อไฟล์เท่านั้น จากนั้นชื่อไฟล์จะถูกเก็บไว้ที่ออบเจ็กต์ songname และจะมีการเพิ่มนามสกุลของไฟล์คือ .mpg ให้โดยอัตโนมัติ ต่อจากนั้นก็จะมีการกำหนดไดเรกทอรีปลายทางสำหรับบันทึกไฟล์ที่ได้จากการแปลงเป็น .mpg เรียบร้อยแล้ว ซึ่งจะมีการเรียกใช้เมทอด cutstring เพื่อใช้สำหรับตัดตัวอักษรทั้งหน้าและหลังอย่างละ 1

ตัวเพื่อให้ได้ไดเรกทอรีที่มีความถูกต้องและสมบูรณ์ เมื่อกำหนดรายละเอียดต่าง ๆ ครบแล้วก็จะทำการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาริงานเพื่อการศึกษาเท่านั้น เมื่อนุญตใดนำไปใช้ประโยชน์ใดๆ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

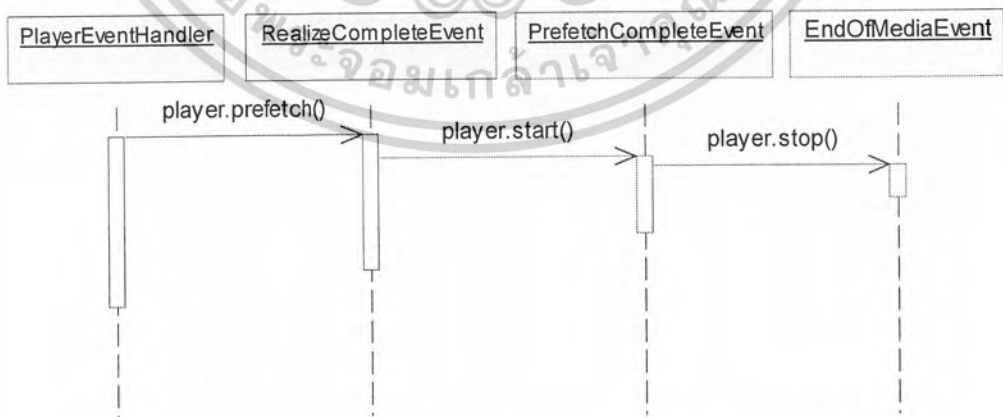
เรียกใช้โปรแกรม dat2mpg ซึ่งโปรแกรมนี้จะทำหน้าที่ในการแปลงไฟล์ที่มีรูปแบบจาก.dat เป็น .Mpeg ในการใช้งานโปรแกรมนั้นก็ต้องทำการรันโปรแกรมพร้อมกับส่งค่าพารามิเตอร์ไป 2 ค่า คือตำแหน่งไฟล์ต้นทางที่ต้องการจะแปลง และ ตำแหน่งไฟล์ปลายทางที่ได้ทำการแปลงแล้ว เมื่อได้ทำการรันเรียบร้อยแล้วจะมีการคืนค่าของโปรเซส p กลับมา ถ้าค่า exitValue เท่ากับ 0 หมายความว่าโปรแกรมการแปลงไฟล์นั้นได้ทำงานสำเร็จและสมบูรณ์ แต่ถ้ามีค่าไม่เท่ากับ 0 ก็หมายความว่าโปรแกรมที่เรียกใช้นั้นทำงานอย่างไม่สมบูรณ์

ขั้นตอนการทำงานของโปรแกรมจำแนกตาม ความสามารถของโปรแกรม

1. การเล่นไฟล์มีเดียสตรีม



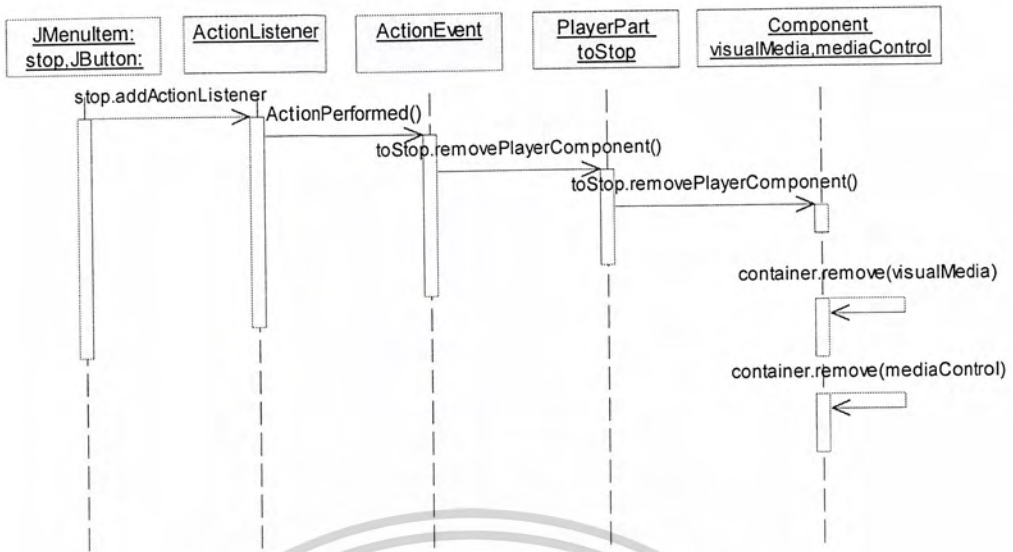
รูปที่ 4-3 แสดงขั้นตอนการทำงานของการเล่นไฟล์มีเดียสตรีม



รูปที่ 4-4 แสดงสถานะภายในของเพลเยอร์ภายในคลาส PlayerEventHandler

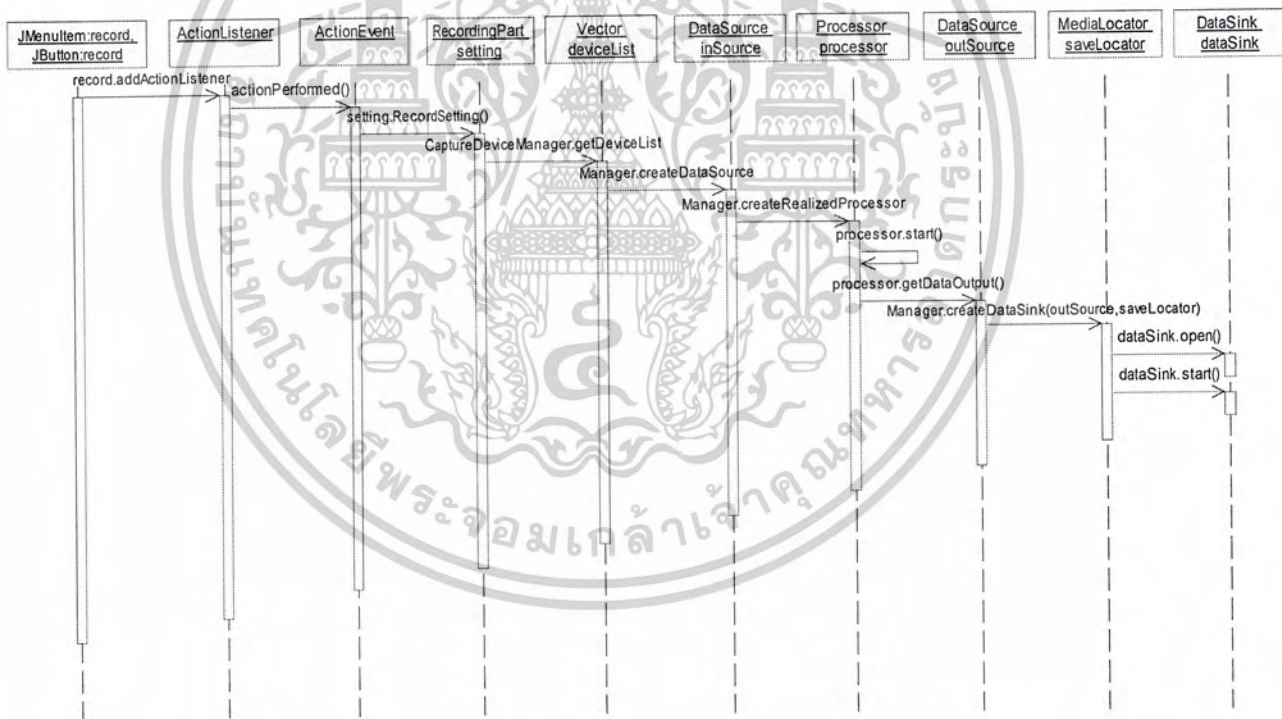
2. การหยุดเล่นไฟล์มีเดียสตรีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 แสดงขั้นตอนการทำงานของการเล่นไฟล์มีเดียสตรีม

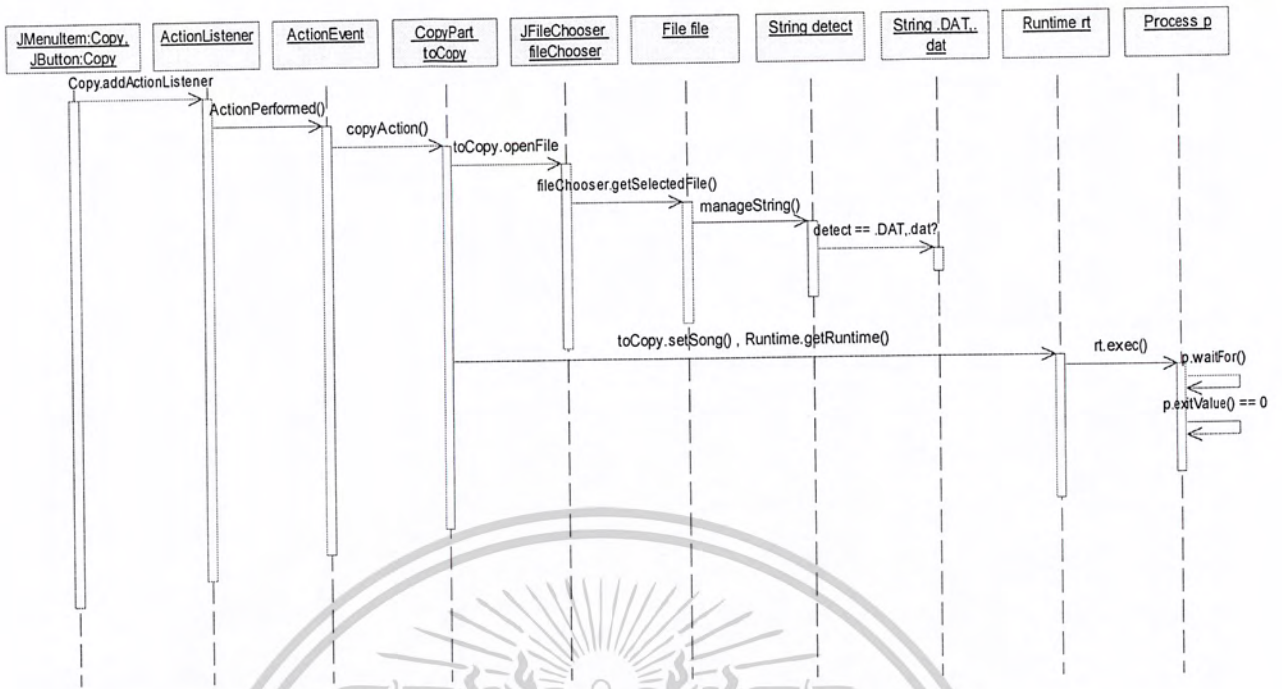
3. การบันทึกเสียงผู้ร้อง



รูปที่ 4-6 แสดงขั้นตอนการทำงานในการบันทึกเสียงของผู้ร้อง

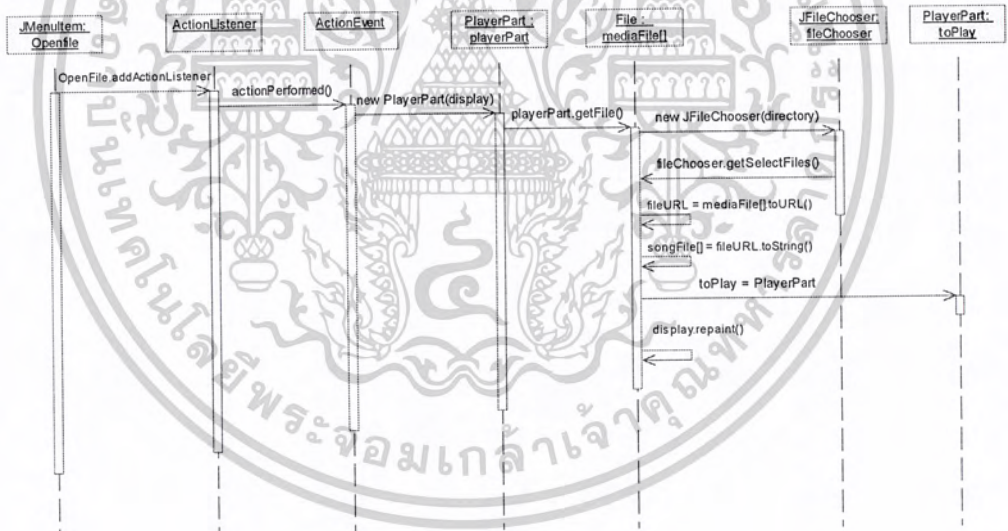
4. การคัดลอกไฟล์จากแผ่นซีดีลงในเครื่องผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-7 แสดงขั้นตอนการคัดลอกไฟล์จากแผ่นวีซีดีเก็บลงในเครื่องของผู้ใช้

5. การเปิดไฟล์มาใส่ไว้ในรายการ



รูปที่ 4-8 แสดงขั้นตอนการเปิดไฟล์มาใส่ไว้ในรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดสอบการทำงานของโปรแกรม

ในการทดสอบความสามารถของโปรแกรม มีรายละเอียดดังนี้

5.1 อุปกรณ์ที่ใช้ในการทดสอบโปรแกรม

- เครื่องคอมพิวเตอร์ระบบปฏิบัติการ WindowsXP Professional Service Pack1
- RAM 256Mb
- CPU 800 MHz
- Harddisk 20Gb
- ไมโครโฟนและลำโพง

5.2 การทำงานและความสามารถของโปรแกรม

จากที่ได้กล่าวถึงความสามารถของโปรแกรมมาแล้วในบทที่ 1 ในบทนี้จะกล่าวถึง การทดลองใช้โปรแกรม จากรูปลักษณะภายนอกของโปรแกรม



รูปที่ 5-1 แสดงรูปลักษณะของโปรแกรม

1. การเล่นไฟล์มีเดียสตรีม

เจเอ็มเอฟสามารถเล่นไฟล์มีเดียสตรีมชนิดต่างๆ ได้ ดังนี้ .aiff, .au, .avi, .mpeg, .mp3, .wav โดยที่พื้นที่ในการแสดงผลจะแสดงภาพก็ต่อเมื่อ ไฟล์นั้นๆเป็นไฟล์จำพวกวีดีโอ ซึ่งในที่นี้ก็คือ ไฟล์ชนิด .mpeg

2. การบันทึกเสียงของผู้ร้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโปรแกรมนี้สามารถบันทึกเสียงของผู้ร้องได้ใน 2 ลักษณะคือ

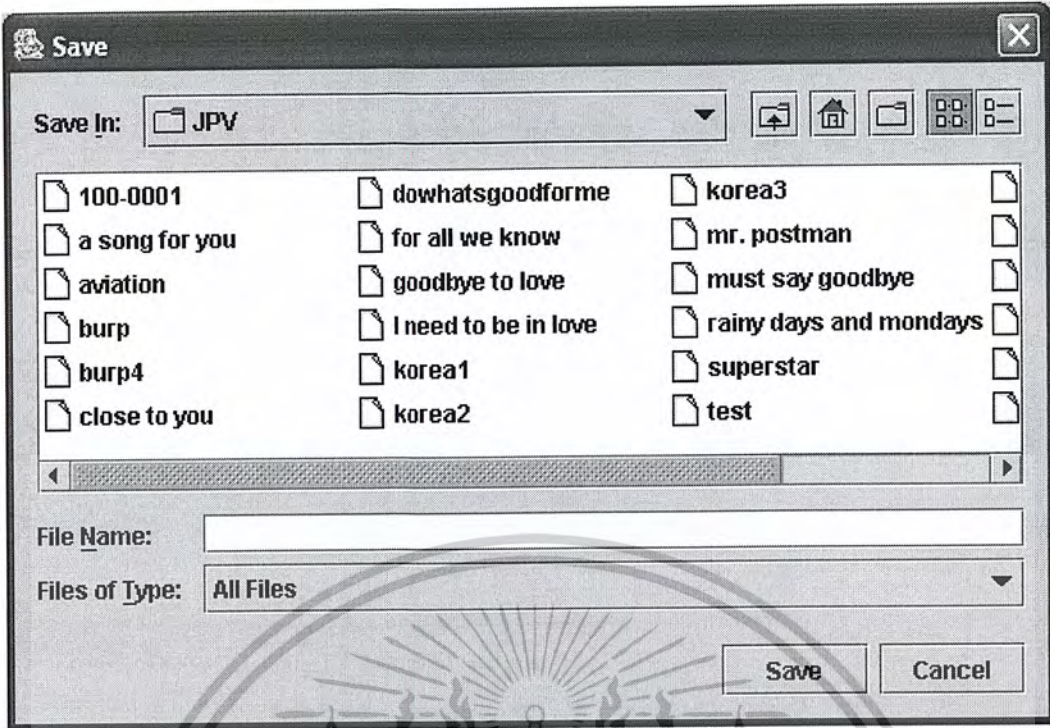
2.1 บันทึกเสียงของผู้ร้องขณะที่เปิดไฟล์มีเดียสตรีม ทำให้ได้ทั้งเสียงของผู้ร้องเองและทำนองเพลงจากมีเดียสตรีมในขณะนั้นด้วย แต่การบันทึกนี้จะได้ไฟล์เอาต์พุตออกมาเป็นไฟล์ในรูปแบบ .mp3 ซึ่งจะไม่มีภาพเหมือนในขณะที่ยังบันทึกกับมีเดียสตรีมจำพวกวีดีโอ

2.2 บันทึกเสียงของผู้ร้องเพียงอย่างเดียว ซึ่งจะได้ไฟล์เอาต์พุตเป็นไฟล์ในรูปแบบ .mp3



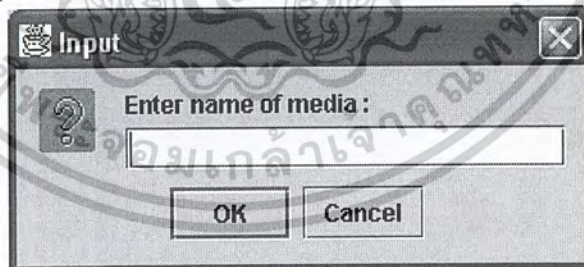
รูปที่ 5-2 แสดงการเลือกใช้ความสามารถในการบันทึกเสียง

ในการบันทึกเสียงของผู้ร้อง โปรแกรมจะมีการกำหนดค่าของอุปกรณ์อินพุตที่ใช้คือ Direct Sound Capture และคุณภาพเสียงที่จะบันทึก คือ Linear 4800 Hz 16 bit Stereo LittleEndian Signed ต่อจากนั้นผู้ใช้จะต้องทำการระบุตำแหน่งและชื่อไฟล์เอาต์พุตที่ต้องการบันทึก

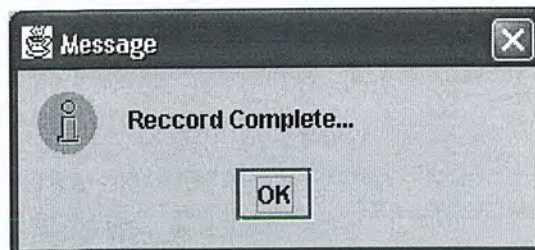


รูปที่ 5-3 แสดงหน้าต่างเพื่อให้ผู้ใช้ระบุตำแหน่งและชื่อไฟล์ที่จะบันทึก

- การคัดลอกไฟล์เพลงจากแผ่นซีดีลงในเครื่องโปรแกรมนี้มีความสามารถในการคัดลอกไฟล์จากแผ่นซีดีซึ่งเป็นไฟล์ในรูปแบบ .dat ลงเก็บไว้ในเครื่อง โดยไฟล์จะถูกเปลี่ยนให้อยู่ในรูปแบบของ .mpg ก่อน โดยเราสามารถเปลี่ยนชื่อไฟล์ได้ตามต้องการ ดังรูปที่ 5-5



รูปที่ 5-4 แสดงขั้นตอนการเปลี่ยนชื่อเพลงโดยการใส่ชื่อที่ต้องการเปลี่ยนลงไปแทน



รูปที่ 5-5 แสดงข้อความที่แจ้งให้ทราบว่า การคัดลอกไฟล์เสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุปและวิจารณ์

6.1 บทสรุปและวิจารณ์

เจเพลเยอร์ว่าไรตี้ ที่ได้ถูกพัฒนาขึ้นนั้น มีความสามารถในการทำงานตรงกับความต้องการของผู้จัดทำคือ โปรแกรมสามารถเล่นไฟล์เพลงที่มีรูปแบบต่าง ๆ คือ mpeg, wav, mp3, aiff, au, avi, midi ได้ โปรแกรมสามารถทำการบันทึกเสียงของผู้ใช้งานได้ โดยผู้ใช้สามารถระบุตำแหน่งและชื่อไฟล์ในการบันทึกได้ซึ่งไฟล์ที่ได้จากการบันทึกนั้นจะมีรูปแบบเป็น mp3 ซึ่งการบันทึกเสียงนั้นสามารถเลือกบันทึกได้ว่าจะบันทึกเสียงของผู้ใช้เพียงอย่างเดียวหรือจะบันทึกเสียงของผู้ใช้ไปพร้อมกับทำนองเพลง โปรแกรมสามารถคัดลอกไฟล์จากไฟล์ซึ่งมีรูปแบบเป็น dat ไปเป็นไฟล์ที่มีรูปแบบเป็น mpeg (แปลงไฟล์) และยังสามารถแก้ไขชื่อไฟล์เก็บลงยังเครื่องได้

6.2 ปัญหาที่เกิด

1. เนื่องจาก JMF นั้นไม่รองรับกับรูปแบบไฟล์ dat ซึ่งเป็นรูปแบบของไฟล์ที่มีการใช้งานอย่างมากในปัจจุบันไม่ว่าจะเป็น ไฟล์คาราโอเกะจากแผ่น VCD ที่มีอยู่ทั่วไปตามท้องตลาด เป็นต้น ทำให้โปรแกรมเจเพลเยอร์ว่าไรตี้ ไม่สามารถรองรับกับรูปแบบไฟล์ดังกล่าวได้ซึ่งเป็นไฟล์ที่มีการนิยมใช้งานอย่างมาก
2. เนื่องจากในขณะที่เล่นไฟล์ที่มีภาพ โปรแกรมจะไม่สามารถแสดงภาพขึ้นมาได้ผู้ใช้จะต้องทำการรีเฟรชหน้าจออยู่เสมอ
3. อินเทอร์เฟซมีการใช้งานที่ลำบาก
4. การเรียกใช้ไฟล์สามารถเรียกได้ทีละ 1 ไฟล์ซึ่งในเวลาที่ต้องการเล่นไฟล์หลาย ๆ ไฟล์ทำให้ไม่สะดวกต่อการใช้งาน

6.3 แนวทางการแก้ปัญหา

ปัญหาในข้อที่ 1 สามารถแก้ไขได้โดยการเรียกใช้โปรแกรม dat2mpg เพื่อทำการแปลงไฟล์จาก dat เป็น mpeg แล้วบันทึกไฟล์ที่ได้ลงยังเครื่อง เพื่อให้โปรแกรมสามารถขยายการรองรับไฟล์ในรูปแบบ dat ได้

ปัญหาในข้อที่ 2 สามารถแก้ไขได้โดยการไปเซ็คขนาดของคอมพิวเตอร์ตามที่ตามขนาดของหน้าจอที่ต้องการแสดง จึงทำให้โปรแกรมสามารถแสดงภาพขึ้นมาได้

ปัญหาในข้อที่ 3 สามารถแก้ไขได้โดยทำการออกแบบอินเทอร์เฟซใหม่ให้มีรูปแบบการใช้งานที่ง่ายขึ้น โดยมีการเพิ่มเมนูของการทำงานต่าง ๆ อย่างครบถ้วน และมีการแยกปุ่มต่าง ๆ ให้ตรงกับการเรียกใช้งานซึ่งการทำงานต่าง ๆ ของปุ่มแสดงด้วยรูปภาพการทำงานซึ่งทำให้ผู้ใช้เข้าใจการทำงานได้มากขึ้น

ปัญหาในข้อที่ 4 สามารถแก้ไขได้โดยทำการสร้างลิสต์บ็อกซ์ขึ้นมารองรับการเรียกไฟล์และยังเซ็คให้สามารถเรียกไฟล์ได้ทีละหลาย ๆ ไฟล์อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 แนวทางการพัฒนา

เนื่องจากโปรแกรมเจเพลเซอร์ว่าไรต์นี่เป็นโปรแกรมที่เน้นทางด้านความบันเทิง ดังนั้นจากคุณสมบัติต่าง ๆ ที่ได้ทำการพัฒนามา จึงอยากจะขอนำเสนอความสามารถอื่น ๆ ที่สามารถพัฒนาเพิ่มต่อไปได้

1. การให้คะแนนในการร้องเพลง ในรูปแบบของการเล่นไฟล์คาราโอเกะ
2. การปรับแต่งเสียงของผู้ร้องให้มีความหลากหลายของเสียงมากยิ่งขึ้น
3. การรับ-ส่งไฟล์กันผ่านระบบเครือข่ายเพื่อความสะดวกในการเรียกใช้ไฟล์ต่าง ๆ
4. เพิ่มฟังก์ชันในการค้นหาเพลงจากคำสำคัญต่าง ๆ
5. ให้สามารถร้องเพลงผ่านเครือข่ายร่วมกันได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

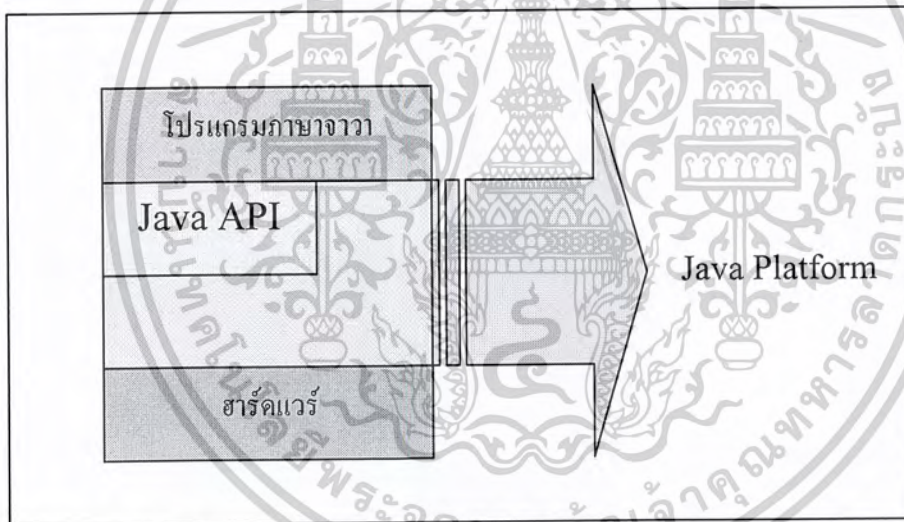
ภาคผนวก ก

ภาษาจาวา

ก.1 ลักษณะของภาษา

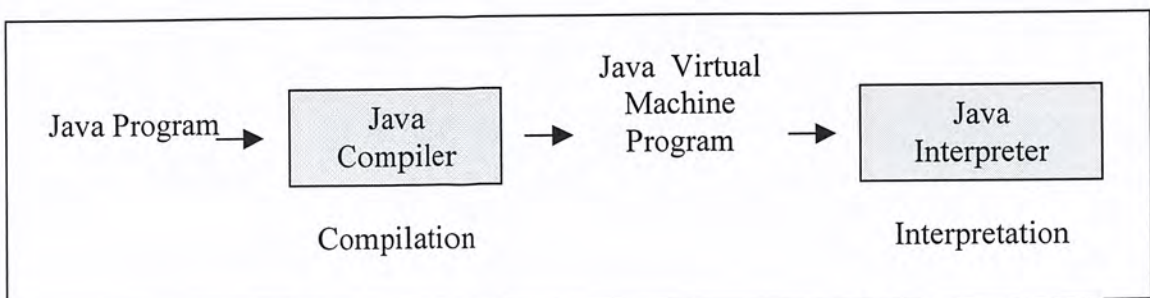
โครงสร้างของภาษาจาวาจะคล้ายคลึงกับภาษาซีพลัสพลัส (C++) มาก ภาษาจาวาจะมีตัวแปลภาษาแบบที่เรียกว่าอินเทอร์พรีเตอร์ โดยที่ชุดคำสั่งของจาวาที่ผ่านการตรวจสอบความถูกต้องของภาษาแล้วจะเก็บอยู่ในรูปที่เรียกว่าเป็นไบนารีโคด (Bytecode) ซึ่งเป็นคำสั่งที่เรียก จาวาเวอร์ชวลแมชชีน (Java Virtual Machine) กล่าวคือไบนารีโคดที่ได้สามารถนำไปประมวลผลได้กับเครื่องคอมพิวเตอร์ทุกประเภทที่มีจาวาอินเทอร์พรีเตอร์ หรือกล่าวได้ว่าเขียนเพียงครั้งเดียวแต่สามารถประมวลผลได้ทุกแห่ง (Write once, run anywhere)

จาวาแพลตฟอร์ม (Java Platform) จะประกอบด้วยสองส่วนคือ จาวาเวอร์ชวลแมชชีน และส่วนเชื่อมประสานในการเขียนโปรแกรม (Java Application Programming Interface :เอพีไอ) จาวาเอพีไอ เป็นชุดคำสั่งที่เอื้ออำนวยความสะดวกให้ผู้พัฒนาโปรแกรมนำไปใช้ได้ โดยเฉพาะส่วนที่เป็นส่วนเชื่อมประสานด้านกราฟฟิก (Graphic User Interface: GUI) ดังนั้นโปรแกรมภาษาจาวาจึงประมวลผลบนจาวาเอพีไอ หรือจาวาเวอร์ชวลแมชชีน ดังรูป



รูปที่ ก-1 จาวาเอพีไอและจาวาเวอร์ชวลแมชชีน ไม่ขึ้นอยู่กับองค์ประกอบทางด้านฮาร์ดแวร์

ภาษาจาวานำความคิดการสร้างเครื่องจักรสมมติมาใช้ เพื่อให้โปรแกรมไม่ขึ้นกับระบบ โดยมีคอมไพเลอร์ทำการแปลภาษาให้เป็นโปรแกรมของจาวาเวอร์ชวลแมชชีน แล้วนำโปรแกรมนั้นมาทำงานด้วยเครื่องจักรสมมติที่จำลองขึ้นโดยจาวาอินเทอร์พรีเตอร์ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

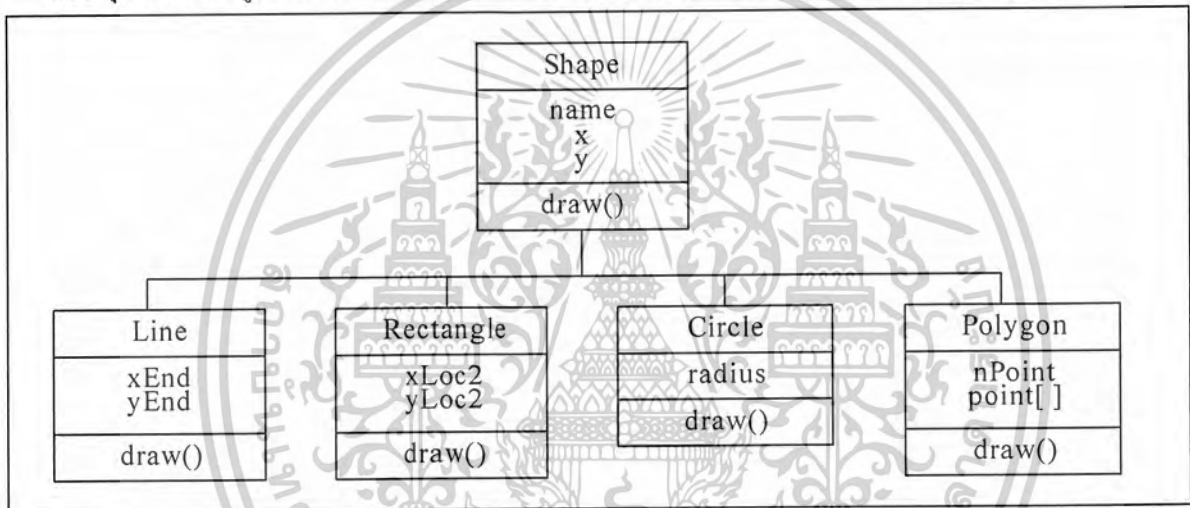
รูปที่ ก-2 แสดงการทำการแปลภาษาให้เป็นโปรแกรมของจาวาเวอร์ซวลแมชชีน

ภาษาจาวาจะถูกคอมไพล์โดยจาวาคอมไพเลอร์ ได้เป็นโปรแกรมของจาวาเวอร์ซวลแมชชีนแล้วสามารถนำไปทำงานบนเครื่องใด ๆ ที่มีจาวาอินเทอร์พรีเตอร์ได้จึงมีคุณสมบัติไม่ขึ้นกับระบบใด ๆ ระบบหนึ่ง

ก.2 ลักษณะการเขียนโปรแกรมด้วยภาษาจาวา

ภาษาจาวาเป็นภาษาเชิงวัตถุ ดังนั้นการออกแบบหรือเขียนโปรแกรมเชิงวัตถุ ภาษาคอมพิวเตอรืต้องมีคุณสมบัติที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ ซึ่งต้องประกอบด้วยหลักการดังนี้

I . ความสามารถในการสืบทอด (Inheritance) ยกตัวอย่างเรื่อง shape ดังรูปซึ่งเป็นตัวอย่างที่ชัดเจนในการสืบทอด กล่าวคือชื่อของรูปทรง และพิกัดจะสืบทอดไปยังรูปทรงอื่น เพราะเป็นคุณสมบัติที่ทุกรูปต้องมี ซึ่งหลักการของการสืบทอดนี้ทำให้เราสามารถสร้างวัตถุอื่น ๆ บนพื้นฐานของวัตถุที่มีอยู่ และสามารถเรียกใช้ฟังก์ชันหรือ ชุดคำสั่งที่มีอยู่แล้วมาใช้งานได้อย่างมีประสิทธิภาพ ลดเวลาในการโปรแกรมใหม่ทุกครั้ง



รูปที่ ก-3 แสดงวัตถุของรูปทรงเรขาคณิต

II . ความสามารถในการเก็บซ่อน (Encapsulation) ข้อมูล โดยทั่วไปของการเขียนชุดคำสั่ง ข้อมูลบางประเภทเป็นข้อมูลที่ไม่ต้องการให้มีการแก้ไข ทั้งนี้เพื่อให้การประมวลผลเป็นไปตามวัตถุประสงค์ที่ต้องการ ภาษาที่ใช้ในการเขียนชุดคำสั่งงานที่ดีควรมีคุณสมบัติที่จะเก็บซ่อนข้อมูลที่ไม่ต้องการให้ผู้ใช้เปลี่ยนแปลง ภาษาจาวาประยุกต์ หลักการของการเก็บซ่อนโดยผ่านคลาส ซึ่งเป็นชุดคำสั่งในการกำหนดข้อมูลและพฤติกรรม หรือ ฟังก์ชันและชุดคำสั่งดังกล่าว ผู้นำคลาสไปใช้งานไม่จำเป็นต้องทราบในรายละเอียดหรือขั้นตอนในการทำงานของคลาส สิ่งที่ต้องทราบคือส่วนที่เป็นส่วนเชื่อมประสาน หรือ ฟังก์ชัน หรือ ตัวแปรอะไรบางอย่างที่เราสามารถเรียกประมวลผลได้ และต้องส่งข้อมูลใด ๆ ให้ฟังก์ชันดังกล่าว ส่วนอื่น ๆ จะถูกเก็บซ่อนด้วยคุณลักษณะของภาษาเองที่ไม่อนุญาตให้มองเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

III . ความสามารถในการแปลงร่าง (Polymorphism) ในการสืบทอดคุณสมบัติ ไม่จำเป็นเสมอไปที่จะสืบทอดคุณสมบัติทุกอย่าง ในบางครั้งอาจต้องมีการแก้ไขชุดคำสั่งเดิม จะทำให้ต้องเสียเวลาในการตามแก้ไขโปรแกรมต่าง ๆ ที่ใช้ฟังก์ชันที่ถูกแก้ไข วิธีการที่ดีคืออนุญาตให้มีการเขียนชุดคำสั่งใหม่ภายใต้ชื่อฟังก์ชันเดิมที่มีอยู่

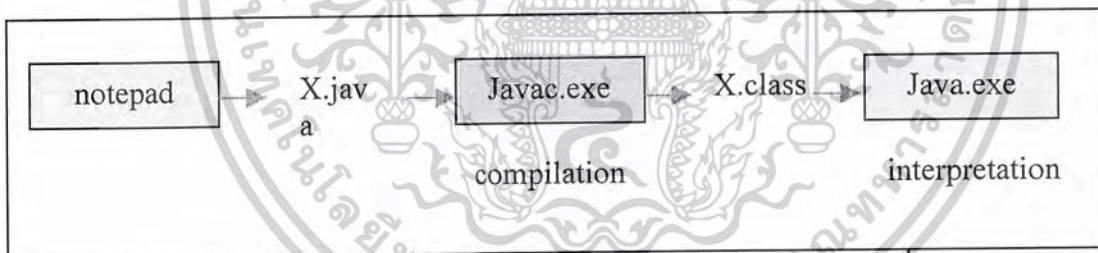
IV . ความสามารถในการจัดการโครงสร้างข้อมูลแบบเชิงนาม หรือแอบสแตรก (Abstract) ความหมายของข้อมูลแบบเชิงนามคือข้อมูลหรือกระบวนการยังไม่เกิดขึ้นจริง ซึ่งในการพัฒนาระบบงานข้อมูลอาจจะยังไม่เกิด หรือยังไม่สามารถที่จะอธิบายได้ว่าขั้นตอนในการทำงานจะเป็นอย่างไร แต่เพื่อไม่ให้งานต้องหยุดชะงัก การสร้างคลาสเป็นแบบเชิงนามก็จะเป็นอีกวิธีหนึ่ง

ก.3 ประเภทของโปรแกรมจาวา

โปรแกรมจาวาแบ่งออกเป็น 2 ประเภทคือ

1. จาวาแอปพลิเคชัน (Java Application) คือโปรแกรมที่สร้างขึ้นด้วยภาษาจาวา เพื่อถูกทำงานโดยจาวาอินเทอร์พรีเตอร์ ต้องเป็นโปรแกรมที่สมบูรณ์มี main() เป็นจุดเริ่มต้น และสามารถควบคุมการดำเนินไปของตัวเองได้ ทำงานภายใต้จาวาอินเทอร์พรีเตอร์ โดยไม่ต้องมีโปรแกรมอื่นช่วย บางคนจึงเรียกว่า เป็นโปรแกรมที่ทำงานด้วยตัวเอง (stand alone program)

ขั้นตอนการสร้างและทำงานจาวาแอปพลิเคชัน มีดังนี้ เริ่มจากใช้เอดิเตอร์เขียนโปรแกรมสำหรับจาวาแอปพลิเคชันเก็บไว้ในไฟล์ที่มีรูปแบบเป็น .java เช่น X.java จากนั้นใช้ javac.exe ทำการคอมไพล์ X.java จะได้ผลลัพธ์เป็นไฟล์ X.class ซึ่งเป็นโปรแกรมจาวาเวอร์ชวลแมชชีนของคลาส X เมื่อต้องการให้ X.class นั้นทำงานก็ส่งให้แก่ java.exe ค้างแสดงในรูป



รูปที่ ก-4 แสดงขั้นตอนการสร้างและทำงานของจาวาแอปพลิเคชัน

2. แอปเพล็ต (Applet) คือโปรแกรมภาษาจาวาที่สร้างขึ้นเพื่อส่งไปกับเอชทีเอ็มแอลเพจ (HTML page) ให้ไปทำงานภายใต้เว็บเบราว์เซอร์ ที่มีจาวาอินเทอร์พรีเตอร์โดยสามารถติดต่อกับผู้ใช้และแสดงผลผ่านทางเว็บเพจ จากในเอชทีเอ็มแอลเพจ จะสามารถเรียกแอปเพล็ตมาทำงานและแสดงผลในพื้นที่ของเพจนั้น โดยที่เว็บเบราว์เซอร์จะเป็นผู้โหลดแอปเพล็ตที่ถูกเรียกมาจากเซิร์ฟเวอร์ที่ให้เอชทีเอ็มแอลเพจนั้นมาและจัดการทำงานให้โดยอัตโนมัติ วิธีนี้ทำให้เราสามารถส่งโปรแกรมไปทำงานบนเครื่องใด ๆ ในระบบอินเทอร์เน็ตโดยใช้เอชทีเอ็มแอลเพจเป็นตัวกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.4 ข้อดีของการพัฒนาโปรแกรมด้วยภาษาจาวา

1. ความง่าย จาวากลับได้ว่าเป็นภาษาที่ง่ายต่อการศึกษา จำนวนชุดคำสั่งที่ต้องศึกษาจะมีจำนวนน้อย ไม่มีคำสั่ง GOTO และที่สำคัญไม่มีตัวชี้ (Pointer) โปรแกรมเมอร์ที่คุ้นกับการเขียนภาษาซี/ซีพลัสพลัส แทบจะไม่ต้องเสียเวลาในการเรียนภาษาจาวา ทั้งนี้เพราะรูปแบบของภาษาจาวาจะเลียนแบบภาษาซี/ซีพลัสพลัสเป็นส่วนใหญ่
2. เป็นภาษาเชิงวัตถุที่สมบูรณ์ คุณสมบัติของภาษาเชิงวัตถุประกอบด้วย ความสามารถในการสืบทอด, ความสามารถในการเก็บซ่อน, ความสามารถในการแปลงร่าง, และความสามารถในการสร้างคลาสแบบเชิงนาม
3. การกันการผิดพลาด (Robust) ภาษาจาวาเป็นภาษาที่เชื่อได้ว่ามีการตรวจสอบข้อมูลทุกตัวที่มีการใช้งานในขั้นตอนของการคอมไพล์ (Strong type) ไม่มีตัวชี้ซึ่งทำให้ง่ายในการตรวจสอบหาข้อผิดพลาด นอกจากนี้ภาษาจาวายังมีหน่วยในการกำจัดสิ่งที่ไม่ใช้ (Garbage Collection) แบบอัตโนมัติโดยเฉพาะในส่วนของจัดการหน่วยความจำหลัก นอกจากนี้ยังมีหน่วยที่เรียกว่าการจัดการความผิดพลาด (Exception Handling) ซึ่งจะจัดการกับความผิดพลาดทั้งหลายอัตโนมัติผู้ใช้ไม่จำเป็นต้องเขียนเพิ่ม
4. ความปลอดภัยในการที่จะประมวลผล (Secure) เป็นอีกคุณสมบัติหนึ่งของภาษาจาวา ทั้งนี้เพราะภาษาจาวาจะประมวลผลที่เครื่องใด หรือมีการกระจายในการทำงานทั่วทั้งเครือข่ายบนอินเทอร์เน็ต ดังนั้นต้องมั่นใจได้ว่าโปรแกรมที่ประมวลผลยังคอมพิวเตอร์เครื่องใดเครื่องหนึ่ง จะไม่มาทำลายทรัพยากรของเครื่องดังกล่าว
5. มัลติเธรด (Multithread) ความสามารถในการประมวลผลหลาย ๆ งานได้พร้อมกัน การประมวลผลผ่านเว็บส่วนใหญ่จะมีหลาย ๆ งานประมวลผลไปพร้อมกัน และโดยเฉพาะงานทางด้านกราฟิกเราสามารถแยกงานออกเป็นงานย่อยหลาย ๆ งานและให้แสดงผลไปพร้อมกัน เช่น การวาดรูปวงกลม อาจให้แสดงไปพร้อมกับการแสดงข้อความ เป็นต้น ภาษาจาวามีหน่วยของโปรแกรมที่สนับสนุนการทำงานแบบมัลติเธรด โดยจะอยู่ใน java.lang
6. ไม่ขึ้นกับโปรแกรมควบคุมระบบ (Architecture-neutral) ทั้งนี้เพราะภาษาจาวาคอมไพล์แล้วได้ไบต์โค้ดซึ่งไบต์โค้ดสามารถที่จะนำไปประมวลผลยังเครื่องใด ๆ ก็ได้และไม่ขึ้นกับโปรแกรมควบคุมระบบอีกด้วย ไบต์โค้ดที่ได้ไม่จำเป็นต้องไปคอมไพล์ใหม่เพื่อให้สามารถประมวลผลบนเครื่องใดเครื่องหนึ่ง หรือโปรแกรมควบคุมระบบใดระบบหนึ่ง
7. เคลื่อนย้ายง่าย (Portable) คือสามารถประมวลผลได้กับคอมพิวเตอร์ทุกประเภท และโปรแกรมควบคุมระบบทุกแบบ
8. การเชื่อมโยงแบบพลวัต (Dynamic link) และการกระจาย (Distributed) หลักการของการเชื่อมโยงแบบพลวัตทำให้สามารถใช้ทรัพยากรทางคอมพิวเตอร์ร่วมกันและผลที่ตามมาคือ ทำให้สามารถประหยัดหน่วยความจำส่วนการกระจายนั้นมาจากหลักการที่ว่าพวกชุดคำสั่งที่เป็นไลบรารี (Library) สามารถดาวน์โหลดมาใช้งานได้โดยไม่ต้องเก็บในรูปของ CD-ROM ซึ่งการให้ดาวน์โหลดจากแหล่งของต้นแบบจะทำให้ได้ชุดของคำสั่งที่ทันสมัย อย่างไรก็ตามหลักการในข้อหลังยังมีข้อจำกัดเนื่องจากความเร็วของระบบเครือข่ายยังไม่เพียงพอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ความรู้พื้นฐานภาษาจาวา

ในการออกแบบโปรแกรมนั้นมีเป้าหมายเพื่อที่จะให้โปรแกรมสามารถทำงานได้อย่างสมบูรณ์ โดยพัฒนาขึ้นตามหลักการของออบเจ็กต์โอเรียนเตด (Object Oriented) และคุณสมบัติของภาษาจาวา ซึ่งมีรายละเอียดของทฤษฎีต่าง ๆ ดังนี้

ข.1 เอ็กเซปชัน แฮนดลิ่ง (Exception handling)

โดยปกติการเขียนโปรแกรมเราไม่สามารถหลีกเลี่ยงความผิดพลาดที่อาจเกิดขึ้นได้ ในภาษาจาวา จะแบ่งความผิดพลาดดังกล่าวออกเป็น 2 ชนิดคือ เออเรอร์ (Error) และ เอ็กเซปชัน (Exception) โดยความผิดพลาดประเภทเออเรอร์ เป็นความผิดพลาดชนิดร้ายแรงซึ่งเมื่อเกิดขึ้นแล้วจะทำให้โปรแกรมหยุดการทำงานทันที (Terminate) ส่วนความผิดพลาดประเภทเอ็กเซปชัน เป็นความผิดพลาดที่สามารถควบคุมให้โปรแกรมทำงานต่อไปได้ ส่วนกระบวนการที่ใช้ในการป้องกันหรือตรวจจับเอ็กเซปชัน ที่เกิดขึ้นเรียกว่า “เอ็กเซปชัน แฮนดลิ่ง”

เอ็กเซปชันเป็นออบเจ็กต์ อย่างหนึ่งที่ตัวแปรภาษาสร้างขึ้นเมื่อมีข้อผิดพลาดเกิดขึ้นกับ โปรแกรมเพื่อใช้ตรวจสอบสถานะที่ผิดปกติของโปรแกรม หรือการทำงานที่ไม่ถูกต้องของโปรแกรม โดย เอ็กเซปชัน จะถูกส่ง (throw) ออกมาจากโปรแกรมที่มีความผิดปกติ เกิดขึ้นซึ่งสามารถนำเอ็กเซปชันดังกล่าวไปตรวจสอบเพื่อควบคุมให้โปรแกรมสามารถทำงานต่อไปได้อย่างปกติ

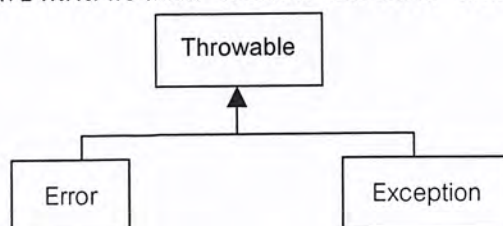
ประโยชน์ของการตรวจสอบข้อผิดพลาดก็คือ จะไม่ทำให้โปรแกรมหยุดการทำงานเมื่อเกิดสิ่งผิดปกติขึ้นเพราะสามารถตรวจสอบข้อผิดพลาดได้จึงสามารถสั่งให้โปรแกรมทำงานต่อไปได้ทันที คลาส เอ็กเซปชันเป็นซัพคลาสของ “Throwable” ซึ่งสามารถส่งต่อไปยังโปรแกรมอื่นได้

คลาส “Throwable” ใช้สำหรับสร้างออบเจ็กต์ เพื่อใช้ตรวจสอบข้อผิดพลาดของโปรแกรมในขณะที่ทำงานซึ่ง คลาส “Throwable” มีคอนสตรัคเตอร์ (Constructor) 2 ตัวดังนี้

```
Throwable();
Throwable(String message);
```

คอนสตรัคเตอร์ตัวที่มีพารามิเตอร์ (parameter) เป็น สตริง (String) ใช้สำหรับเก็บข้อความเกี่ยวกับข้อผิดพลาดนั้น ๆ (Exception Message) และเมื่อต้องการแสดงข้อความดังกล่าวก็สามารถเรียกใช้ได้ด้วยเมทอด “getMessage()”

โดยปกติเราไม่ใช่ คลาส “Throwable” ตรวจสอบข้อผิดพลาดของโปรแกรมโดยตรงแต่จะขยาย (Inherit) คลาส นี้ออกอีก 2 คลาส คือ คลาส “เออเรอร์” และ คลาส “เอ็กเซปชัน”



รูปที่ ข-1 แสดงการอินเฮริทจาก คลาส “Throwable”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.1.1 การตรวจสอบข้อผิดพลาดโดยใช้คำสั่ง try, catch และ finally

โดยปกติขณะที่โปรแกรมจาวา กำลังรันอยู่หากเกิดข้อผิดพลาดขึ้นบางอย่างจาวาอินเทอร์พรีเตอร์ จะสร้างออบเจกต์ของ คลาส “เออเรอร์” หรือ “เอ็กเซปชัน” ขึ้นทันทีและส่ง ออบเจกต์ ที่สร้างขึ้นไปที่ ตำแหน่งในประโยคที่เกิดข้อผิดพลาดขึ้นโดยปกติเราจะใช้คำสั่ง try, catch และ finally เป็นตัวควบคุมใน ส่วนที่คิดว่าจะสามารถเกิดข้อผิดพลาดขึ้นได้ คำสั่ง try, catch และ finally มีรูปแบบดังนี้

```
Try {
    //คำสั่งทั่วไปรวมทั้งคำสั่งที่คาดว่าจะสามารถเกิดเอ็กเซปชันได้
}
[catch(Type1 id1){
    //คำสั่งที่ใช้จัดการกรณีที่เกิดเอ็กเซปชันของ Type1
}]
[catch(Type2 id2){
    //คำสั่งที่ใช้จัดการกรณีที่เกิดเอ็กเซปชันของ Type2
}]
[finally{
    //คำสั่งใช้ทำงานเมื่อทำงานใน try,catch เสร็จ
}]
```

โดยที่ Type1, Type2 หมายถึง ชื่อ เอ็กเซปชันคลาส ที่ใช้ในการตรวจสอบ (รวมถึง คลาส ที่ขยาย ออก จาก คลาส “เอ็กเซปชัน” ด้วย)
id1, id2 หมายถึง ชื่อ ออบเจกต์ ของ Type1 และ Type2 ตามลำดับ (สามารถกำหนดเป็น ชื่ออะไรก็ได้)

การทำงานของ block คำสั่ง try, catch และ finally สามารถอธิบายได้ดังนี้

- ในบล็อกของทั้ง 3 คำสั่งจะต้องมีเครื่องหมาย ‘{’ และ ‘}’ เสมอ
- สามารถกำหนดบล็อกคำสั่ง catch กับบล็อก try ได้
- อาจจะไม่มียกคำสั่ง catch หรือ finally ก็ได้แต่อย่างน้อยต้องมี 1 บล็อกคำสั่งเพื่อใช้งานร่วมกับบล็อกคำสั่ง try
- คำสั่งในบล็อกคำสั่ง finally จะทำงานเสมอไม่ว่าจะเกิดข้อผิดพลาดใด ๆ หรือไม่ก็ตาม
- การทำงานในบล็อกคำสั่ง try จะทำงานจนกว่าจะพบคำสั่ง break , continue , return หรือมี เอ็กเซปชันเกิดขึ้น โดยหากมีเอ็กเซปชัน เกิดขึ้นโปรแกรมก็จะเปลี่ยนการทำงานไปที่ block คำสั่ง catch ทันทีและจะไม่กลับมาทำงานหลังคำสั่งที่ทำให้เกิดเอ็กเซปชันอีก
- ภายใต้ออกคำสั่ง try คือคำสั่งทั่วไป หรือคำสั่งที่อาจจะเกิดเอ็กเซปชัน ได้ ซึ่งถ้าหากมี เอ็กเซปชัน เกิดขึ้นโปรแกรมก็จะเข้ามาทำงานภายใต้ออกคำสั่ง catch ที่มีพารามิเตอร์ ตรงกับเอ็กเซปชัน ที่เกิดขึ้นทันทีโดยเมื่อทำงานภายในบล็อกคำสั่ง catch ดังกล่าวเสร็จแล้วก็จะไปทำงานในบล็อกคำสั่ง finally ทันที (ไม่ทำที่ catch อื่น ๆ) แต่หากไม่มีเอ็กเซปชัน เกิดขึ้นก็จะเข้ามาทำงานที่บล็อกคำสั่ง finally (ถ้าได้กำหนดไว้) แต่หากไม่มีบล็อก คำสั่ง finally ก็จะไปทำงานที่คำสั่งถัดจากบล็อก คำสั่ง catch ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง แสดงการตรวจจับ NoPlayerException และ IOException

```

try{
    player = Manager.createPlayer(mediaLocator);
    player.addControllerListener(new PlayerEventHandler());
    player.realize();
    System.out.println("player.realize");

    Checked = true;
    return Checked;
}
catch (NoPlayerException noPlayerException) {
    noPlayerException.printStackTrace();
    return false;
}
catch (IOException ioException){
    ioException.printStackTrace();
    return false;
}
}
}

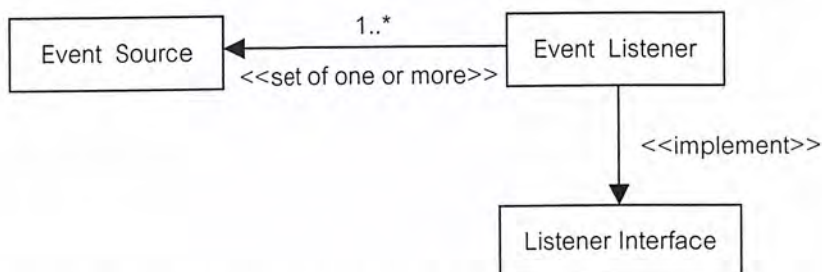
```

จากตัวอย่างเป็นการแสดงการตรวจจับเอ็กเซปชันที่เกิดขึ้นกรณีที่ไม่มีเพลเยอร์(Player) ที่สามารถทำงานได้และ กรณีไอโอ โอเปอเรชัน (I/O Operation) ไม่สามารถทำงานได้เสร็จสมบูรณ์ ซึ่งกรณีทั้งสองที่เกิดขึ้นจะทำให้ไม่สามารถทำงานได้ ดังนั้นจึงมีเอ็กเซปชันเกิดขึ้นคือ NoPlayerException และ IOException

ข.2 อีเวนต์ แอนด์ลิง (Event Handling)

เมื่อผู้ใช้กระทำต่อคอมโพเนนต์ (Component) ไม่ว่าจะเป็นการคลิกเมาส์การกดปุ่มหรือการพิมพ์ข้อความบนเท็กซ์ฟิลด์ (Text Filed) เป็นต้น จะทำให้เกิดเหตุการณ์ หรืออีเวนต์ ขึ้นกับคอมโพเนนต์ ต่าง ๆ เหล่านั้น โดยแต่ละเหตุการณ์ที่เกิดขึ้นจะถูกส่งไปให้โปรแกรม (Class) ที่เป็นลิสเทนเนอร์ (Listener) เพื่อเรียกใช้เมทอดหรือ ตัวแปร สำหรับจัดการกับเหตุการณ์ที่เกิดขึ้น ได้อย่างเหมาะสม

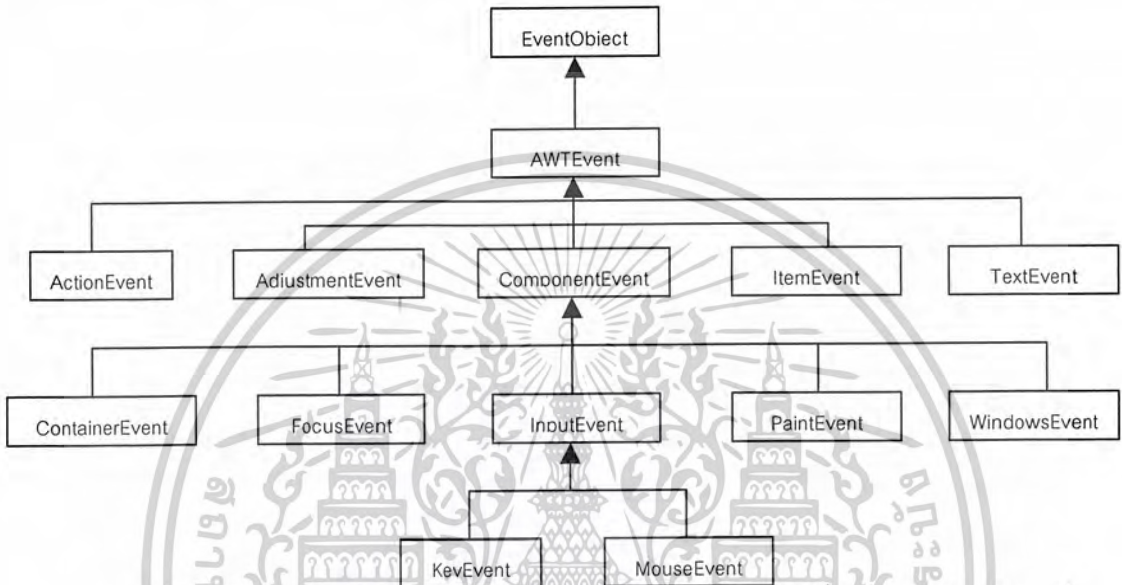
ออบเจกต์ของคอมโพเนนต์ ที่ถูกทำให้เกิดเหตุการณ์จะเรียกว่า อีเวนต์ซอร์ส (Event Source) โดยปกติอีเวนต์ซอร์ส จะไม่สามารถตอบโต้กับเหตุการณ์ที่เกิดขึ้นได้เอง แต่ถ้าต้องการจะต้องขึ้นทะเบียน (register)อีเวนต์ซอร์สให้เป็นอีเวนต์ลิสเทนเนอร์ ก่อน การขึ้นทะเบียนจะต้องเลือกใช้ลิสเทนเนอร์อินเทอร์เฟซ ที่เหมาะสมกับแต่ละเหตุการณ์ ซึ่งหลังจากขึ้นทะเบียนเรียบร้อยแล้วอีเวนต์ซอร์ส จะมีสถานะเป็นตัวรับฟังเหตุการณ์เรียกว่า อีเวนต์ลิสเทนเนอร์ (Event Listener) ดังรูป



รูปที่ ข-2 แสดงความสัมพันธ์ระหว่างอีเวนต์ซอร์ส, อีเวนต์ลิสเทนเนอร์ และ ลิสเทนเนอร์อินเทอร์เฟซ
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปอีเวนต์ซอสซ์ ซึ่งเป็นคอมโพเนนท์ สามารถเกิดเหตุการณ์ได้มากกว่า 1 เหตุการณ์ (สามารถสร้างอีเวนต์ลิสเทนเนอร์ได้หลายตัวต่อ 1 อีเวนต์ซอสซ์) โดยถูกลงทะเบียนเป็นคอมโพเนนท์ไว้เป็นอีเวนต์ลิสเทนเนอร์ถ้ามีเหตุการณ์ที่ขึ้นทะเบียนไว้เกิดขึ้นอีเวนต์ลิสเทนเนอร์ก็จะไปเรียกใช้ (Implement) ลิสเทนเนอร์อินเทอร์เฟสที่เกี่ยวข้องมาทำงานโดยอัตโนมัติ

คลาส ต่าง ๆ ที่ใช้ในอีเวนต์ แชนด์ลิงของเอดับบิวทีอีเวนท์ (AWT Event) จัดเก็บอยู่ในแพ็คเกจ "java.awt.event" มีโครงสร้างการถ่ายทอด คลาส ดังรูป



รูปที่ ข-3 แสดงความสัมพันธ์ของคลาส ต่าง ๆ ที่ใช้ในอีเวนต์แชนด์ลิงของเอดับบิวทีอีเวนท์

ทุกคลาสที่ใช้สำหรับอีเวนต์แต่ละประเภทจะมีอีเวนต์ลิสเทนเนอร์อินเทอร์เฟส ไว้สำหรับให้โปรแกรม (Class) ลิสเทนเนอร์ทำการอิมพลีเมนต์อินเทอร์เฟสดังกล่าว เพื่อให้โปรแกรมสามารถรับอีเวนต์แต่ละชนิดที่เกิดจากซอสซ์ได้ อีเวนต์ลิสเทนเนอร์อินเทอร์เฟส จัดเก็บอยู่ในแพ็คเกจ "java.awt.event" และ "javaw.swing.event" ขึ้นอยู่กับว่าเป็นอีเวนท์ ประเภทใด ทุก ๆ อีเวนต์ลิสเทนเนอร์อินเทอร์เฟสจะขยายมาจากอินเทอร์เฟส "อีเวนต์ลิสเทนเนอร์" ซึ่งจัดเก็บอยู่ในแพ็คเกจ "java.util"

ชื่อของอีเวนต์ลิสเทนเนอร์อินเทอร์เฟสจะมีรูปแบบดังนี้

<Event Type>Listener

เช่น คลาส "ChangeEvent" จะมีอีเวนต์ลิสเทนเนอร์อินเทอร์เฟสเป็น "ChangeListener" ซึ่งมีเพียง 1 แออสแต็กเมทอดเท่านั้นคือ

Public void stateChanged(ChangeEvent e)

โดยโปรแกรมที่เป็นลิสเทนเนอร์ของ "ChangeEvent" จะต้องทำการอิมพลีเมนต์ "ChangeListener" และเมทอดนี้ ซึ่งเมื่อเกิดขึ้นกับซอสซ์ที่ขึ้นทะเบียนไว้ซอสซ์ ดังกล่าวก็จะส่งอินสแตนส์ ของคลาส "ChangeEvent" ออกมาโดยอัตโนมัติและเรียกใช้เมทอด "stateChange()" ทันที

ดังนั้นหากโปรแกรม (Class) ใดต้องการทำหน้าที่เป็นリスเทนเนอร์ ที่ต้องการรับอีเวนต์ประเภทใดจะต้องปฏิบัติดังนี้ (โปรแกรมリスเทนเนอร์สามารถรับหลายอีเวนต์ได้)

- ทำการอิมพลิเมนต์อีเวนต์リスเทนเนอร์สำหรับ <Event Type>ที่จะรับเข้ามาจัดการจากซอสซ์ใด ๆ
- ขึ้นทะเบียนให้ซอสซ์ (คอมโพเนนท์) เพื่อให้รู้จักโปรแกรมที่ทำหน้าที่เป็นリスเทนเนอร์ โดยเรียกใช้เมทอด “add <Event Type>Listener()” ของซอสซ์ นั้น ๆ

โปรแกรม (Class) ใดที่ต้องการทำหน้าที่เป็นリスเทนเนอร์ที่ต้องการตรวจจับอีเวนต์แต่ละประเภทจะต้องทำการอิมพลิเมนต์อีเวนต์リスเทนเนอร์อินเทอร์เฟซตาม <Event Type> นั้น ๆ แต่บางอีเวนต์リスเทนเนอร์Interface ประกอบด้วย Abstract Method ซึ่งโปรแกรม (Class) Listener ที่ทำการอิมพลิเมนต์อินเทอร์เฟซดังกล่าวก็จะต้องทำการอิมพลิเมนต์ทุก แอบสแตกเมทอดด้วยซึ่งในบางครั้งเราต้องการใช้งานเพียงบางเมทอด เท่านั้นก็อาจจะทำให้เกิดความไม่สะดวกในการใช้งานขึ้น เพื่อแก้ปัญหานี้จึงมีการสร้างคลาส ที่อิมพลิเมนต์อีเวนต์リスเทนเนอร์อินเทอร์เฟซบางตัวไว้ในแพ็คเกจ “java.awt.event” และ “javax.swing.event” (ขึ้นอยู่กับอีเวนต์) โดยคลาสเหล่านี้ทำการอิมพลิเมนต์เมทอดในอินเทอร์เฟซทุกตัวไว้แต่เป็น เมทอดที่ว่างเปล่า เราเรียกคลาสเหล่านี้ว่า อแดปเตอร์ (Adapter) ซึ่งมีรูปแบบดังนี้

<Event Type>Adapter

โปรแกรม (Class) ที่เป็นリスเทนเนอร์สามารถใช้งานอแดปเตอร์คลาส ได้โดยการขยายมาจากอแดปเตอร์คลาส ของ <Event Type> ได้ทันที แล้วทำการ โอเวอร์ไรด์เฉพาะเมทอดที่ต้องการใช้งานเท่านั้น ส่วนเมทอดอื่น ๆ ซึ่งเป็นเมทอดว่างจะได้รับการถ่ายทอดมาจากอัตโนมัติ

ข.2.1 เมาส์リスเทนเนอร์ (MouseListener), เมาส์โมชันリスเทนเนอร์ (MouseMotionListener) และ เมาส์อีเวนต์ (MouseEvent)

เป็นการทำงานเกี่ยวข้องกับเมาส์ได้แก่ เมาส์คลิก (Mouse Clicked), เมาส์เอนเทอร์ (Mouse Entered), เมาส์เอ็กซิต (Mouse Exited), เมาส์เพลส (Mouse Pressed), เมาส์รีลีส (Mouse Released), เมาส์มูบ (Mouse Moved) และ เมาส์ดรั๊ก (Mouse Dragged)

เมาส์リスเทนเนอร์เป็นอีเวนต์リスเทนเนอร์อินเทอร์เฟซใช้สำหรับให้คอมโพเนนท์ขึ้นทะเบียนเพื่อตรวจจับเมาส์อีเวนต์ ที่เกิดจากการกระทำกับเมาส์ ทั่วไปคือ เมาส์คลิก, เมาส์เอนเทอร์, เมาส์เอ็กซิต, เมาส์เพลส, เมาส์รีลีส ในเมาส์リスเทนเนอร์อินเทอร์เฟซ มีเมทอดอยู่ 5 เมทอดคือ

```
public void mouseClicked(MouseEvent e) //ทำงานเมื่อกดเมาส์
public void mouseEntered(MouseEvent e) //ทำงานเมื่อนำเมาส์เข้าไปในเขตพื้นที่ของคอมโพเนนท์
public void mouseExited(MouseEvent e) //ทำงานเมื่อนำเมาส์ออกจากพื้นที่ของคอมโพเนนท์
public void mousePressed(MouseEvent e) //ทำงานเมื่อกดเมาส์
public void mouseReleased(MouseEvent e) //ทำงานเมื่อปล่อยเมาส์
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมาส์โมชันลิสเทนเนอร์ เป็นอีเวนท์ลิสเทนเนอร์ใช้สำหรับให้คอมโพเนนท์ขึ้นทะเบียนเพื่อตรวจจับเมาส์อีเวนท์ที่ไม่สามารถตรวจจับได้ในเมาส์ลิสเทนเนอร์ คือ เมาส์มูบ์ว และ เมาส์ดรั๊ก

เมาส์อีเวนท์ ใช้สำหรับจัดการกับเมาส์อีเวนท์เมื่อมีการทำงานต่างๆ กับเมาส์เช่น การคลิก การดรั๊ก เป็นต้น โดยสามารถนำมาใช้งานได้กับทุก ๆ คอมโพเนนท์ในขณะที่คอมโพเนนท์ที่ขึ้นทะเบียนไว้มีไฟกัส อยู่ก็สามารถส่งเมาส์อีเวนท์ออกมาได้ทันที

ตัวอย่าง แสดงการใช้เมาส์อีเวนท์ และ เมาส์ลิสเทนเนอร์กับคอมโพเนนท์

```
songList = new JList(songFile);
songList.setVisibleRowCount(3);
songList.setFixedCellHeight(15);
songList.setFixedCellWidth(200);

songList.setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
songScroll = new JScrollPane(songList);
MouseListener mouseListener = new MouseAdapter() {
    public void mouseClicked(MouseEvent e) {
        if (e.getClickCount() == 2) {
            int index = songList.locationToIndex(e.getPoint());
            System.out.println("Double clicked on Item " + index);
            playAction();
        }
    }
};

songList.addMouseListener(mouseListener);
songList.addListSelectionListener(new ListSelectionListener() {
    public void valueChanged(ListSelectionEvent event) {
        int a = 0;
        a = songList.getSelectedIndex();
        playList = a;
    }
});
```

ลักษณะการทำงานของโปรแกรมมีดังนี้

เมื่อคลิกเมาส์ที่บริเวณ “songList” จำนวน 2 ครั้ง ก็จะทำการแสดงข้อความของตำแหน่งที่เมาส์นั้นคลิกและเล่นไฟล์ในตำแหน่งที่เมาส์นั้นคลิกเช่นกัน

ข.3 ออบเจ็กต์

ข.3.1 การสร้างออบเจ็กต์

เมื่อกำหนดได้แล้วว่าต้องการจะสร้างออบเจ็กต์ให้มีโครงสร้างหลัก ๆ (Class) เป็นอย่างไรจากนั้นก็ถึงขั้นตอนของการสร้างออบเจ็กต์ จากโครงสร้างที่ได้กำหนดไว้ ซึ่งการสร้างออบเจ็กต์ในภาษาจาวาทำได้ 3 วิธีคือ

วิธีที่ 1 สร้างออบเจ็กต์ โดยกำหนดคลาสของออบเจ็กต์ ที่จะสร้างก่อน มี 2 ขั้นตอน คือ

- 1) กำหนดว่าออบเจ็กต์นี้จะให้อยู่ใน คลาส ไค

รูปแบบ ClassName ObjectName

ตัวอย่าง Circle MyCircle;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของออบเจกต์ชื่อ “MyCircle” จะถูกสร้างขึ้นโดยมีคุณลักษณะของ คลาส ชื่อ “Circle”

- 2) สร้างออบเจกต์เพื่อจัดสรรหน่วยความจำให้ออบเจกต์นี้

รูปแบบ `ObjectName = new ClassName()`

ตัวอย่าง `MyCircle = new Circle();`

วิธีที่ 2 วิธีลัดโดยใช้คำสั่งเดียว

รูปแบบ `ClassName ObjectName = new ClassName()`

ตัวอย่าง `Circle MyCircle = new Circle();`

วิธีที่ 3 กรณีที่ต้องการสร้างออบเจกต์ใหม่มากกว่า 1 ออบเจกต์สามารถใช้เครื่องหมายคอมม่า (,)

ช่วยได้เหมือน การประกาศตัวแปรหลาย ๆ ตัว (ตัวอย่างเช่น `int I,j,k,l;`)

รูปแบบ `ClassName ObjectName_1 = new ClassName(),`

`ObjectName_2 = new ClassName();`

ตัวอย่าง `Circle MyCircle_1 = new Circle(), MyCircle_2 = new Circle();`

ข.3.2 การเรียกใช้แอตทริบิวต์ (Attribute) และเมทอดของออบเจกต์

การเรียกใช้แอตทริบิวต์

รูปแบบ `ObjectName.Attribute`

ตัวอย่าง `MyCircle.radius`

ความหมาย จะได้ค่ารัศมี (radius) ของ ออบเจกต์ ชื่อ “MyCircle”

การเรียกใช้เมทอด

รูปแบบ `ObjectName.method`

ตัวอย่าง `MyCircle.findArea()`

ความหมาย จะได้ค่าพื้นที่วงกลมของออบเจกต์ ชื่อ “MyCircle”

การสร้างและเรียกใช้แบบลัด ปกติจะสร้างและเรียกใช้เมทอดหรือแอตทริบิวต์ผ่านชื่อออบเจกต์

เช่น

`Circle MyCircle = new Circle();`

`MyCircle.findArea();`

แต่ถ้าไม่ต้องการเรียกใช้ผ่าน ออบเจกต์ ก็สามารถทำได้ โดยการไม่กำหนดชื่อออบเจกต์ เช่น

`new Circle().radius;`

`new Circle().findArea;`

ซึ่งเป็นการสร้างออบเจกต์ ใหม่โดยไม่กำหนดชื่อ แต่จะต้องเรียกใช้เมทอดหรือแอตทริบิวต์ พร้อม

กันไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.4 คอนสตรัคเตอร์เมทอด

ในการเขียนโปรแกรมเมื่อออบเจ็ค โค้ด ๆ ถูกสร้างขึ้นมาจากได้โครงสร้างของคลาส หนึ่งโปรแกรมจะต้องไปเรียกใช้ เมทอด ที่มีชื่อเดียวกันกับชื่อ คลาส ทันที (ถ้ามีการสร้างเมทอดลักษณะนี้ไว้) การกำหนดการทำงานที่มีลักษณะเช่นนี้เรียกว่า “คอนสตรัคเตอร์” (Constructor) และในภาษาจาวาจะเรียก เมทอด ที่มีชื่อเดียวกับชื่อคลาสว่า “คอนสตรัคเตอร์เมทอด” (Constructor Method) ทั้งนี้ในการสร้าง คอนสตรัคเตอร์เมทอด มีข้อกำหนดดังนี้

คอนสตรัคเตอร์เมทอด ต้องมีชื่อเดียวกับชื่อคลาส

เฮดเดอร์ของคอนสตรัคเตอร์เมทอด ต้องไม่มีค่า รูปแบบของข้อมูลส่งกลับ(Return_data_type) หรือแม้กระทั่งคีย์เวิร์ด “void”

ดังนั้นถ้าชื่อเมทอด โค้ด ๆ ใน คลาส เป็นชื่อเดียวกันกับชื่อ คลาส และมีข้อกำหนดตรงกับข้างต้น ทั้ง 2 ข้อ แสดงว่าเมทอดนั้นถูกกำหนดให้เป็น “คอนสตรัคเตอร์เมทอด” ของคลาสที่มีชื่อเดียวกันนั้น

ตัวอย่าง การสร้าง คอนสตรัคเตอร์

```
public class Interface extends JFrame {
    public Interface() {}
    .....
}
```

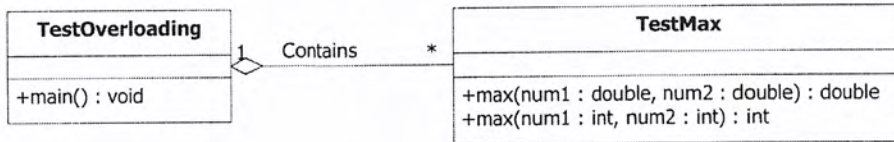
โดยทั่วไปโปรแกรมใหญ่ ๆ ใน คลาส หนึ่งๆ มักจะมีการกำหนด คอนสตรัคเตอร์ ไว้มากกว่า 1 คอนสตรัคเตอร์ โดยจะขึ้นอยู่กับค่าของพารามิเตอร์ที่ส่งมา ดังนั้นจึงสามารถนำลักษณะที่เรียกว่า โอเวอร์โหลดคิง (Overloading) มาใช้งานร่วมกับคอนสตรัคเตอร์ได้

- หากไม่มีการกำหนดคอนสตรัคเตอร์ไว้ ตัวแปลภาษาจาวาจะสร้างคอนสตรัคเตอร์ว่างดังตัวอย่างข้างต้นมาให้โดยอัตโนมัติ
- การทำงานของคอนสตรัคเตอร์ จะเริ่มต้นเมื่อโปรแกรมมีการสร้างออบเจ็คใหม่ขึ้นจากคลาส ที่ระบุ โดยโปรแกรมจะกระโดดไปทำงานที่คลาส นั้นและนิยามค่าของตัวแปรระดับคลาส (แอตทริบิว) ของคลาสนั้นก่อนจากนั้นจึงมาทำงานที่คอนสตรัคเตอร์ เมื่อเสร็จแล้วจึงกลับไปทำงานที่คำสั่งที่อยู่ต่อจากคำสั่งที่ใช้ในการสร้างออบเจ็คนั้น ๆ

ข.5 โอเวอร์โหลดคิงเมทอด

ในภาษาจาวาเราสามารถสร้างเมทอดภายในคลาสเดียวกันที่มีชื่อเดียวกันได้หลาย ๆ เมทอด โดยมีข้อกำหนดว่าเมทอด ที่มีชื่อเดียวกันนี้จะต้องมีชนิดหรือจำนวนของอาร์กิวเมนต์ (argument) ไม่เหมือนกันในแต่ละเมทอด และเมื่อมีการเรียกใช้เมทอดดังกล่าวตัวแปลภาษาจะตรวจสอบชนิดหรือจำนวนพารามิเตอร์ลิสต์ ที่ส่งไปว่าตรงกับชนิดหรือจำนวนอาร์กิวเมนต์ของเมทอด ตัวใด ตัวแปลภาษาที่

จะเรียกเมทอด ตัวนั้นขึ้นมาทำงานเองอัตโนมัติ ลักษณะการทำงานเช่นนี้เรียกว่า “โอเวอร์โหลดดิ่ง”(Overloading)



รูปที่ ข-4 แสดงถึงโอเวอร์โหลดดิ่งเมทอด และการใช้งาน

```

public class TestOverloading {
    public static void main (String srgs[]){
        TestMax FindMax;
        FindMax = new TestMax();
        System.out.println ("ข้อมูลที่มีค่ามากที่สุดระหว่าง 3 และ 4 คือ" + FindMax.max (3,4));
        System.out.println ("ข้อมูลที่มีค่ามากที่สุดระหว่าง 4.7 และ 5.9 คือ" + FindMax.max (4.7,5.9));
    } //end method main
} //class "TestOverloading"

class TestMax {
    public double max (double num1, double num2){
        if (num1 > num2)
            return num1;
        else
            return num2;
    } // method max สำหรับใช้กับข้อมูลชนิด double
    public int max (int num1, int num2){
        if (num1 > num2)
            return num1;
        else
            return num2;
    } //method max สำหรับใช้กับข้อมูลชนิด integer
} //class "TestMax"
  
```

Output :

ข้อมูลที่มีค่ามากที่สุดระหว่าง 3 และ 4 คือ 4

ข้อมูลที่มีค่ามากที่สุดระหว่าง 4.7 และ 5.9 คือ 5.9

จากตัวอย่างจะเห็นว่าเมทอด “max” ที่ถูกสร้างภายในคลาส ชื่อ “TestMax” เป็น โอเวอร์โหลดดิ่งเมทอด ที่สามารถรับพารามิเตอร์ลิสที่มีชนิดของข้อมูลเป็นอินทีเจอร์ (Integer) และสามารถรับ พารามิเตอร์ลิสที่มีชนิดของข้อมูลเป็นดับเบิล (double) ได้ซึ่งจะถูกเรียกใช้ผ่านออบเจ็ค “FindMax”

ในการเรียกใช้ครั้งแรกส่งตัวเลข 3 และ 4 ไปให้เมทอด “max()” โดยโปรแกรมจะทำการตรวจสอบได้ว่าชนิดข้อมูลของพารามิเตอร์ลิสที่ส่งไปเป็นอินทีเจอร์ ดังนั้น โปรแกรมจะไปเรียกใช้เมทอด ที่มีอาร์กิวเมนต์เป็นอินทีเจอร์ขึ้นมาทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเรียกใช้ครั้งที่ สองส่งตัวเลข 4.7 และ 5.9 ไปให้เมทอด “max()” โดยคำสั่งซึ่งมีชนิดข้อมูลของพารามิเตอร์ลิสเป็นดับเบิลดั่งนั้น โปรแกรมจึงไปเรียกใช้เมทอดที่มีชนิดอาร์กิวเมนต์เป็นดับเบิล

ข.6 การใช้ “instanceof”

ในบางครั้งเราอาจจำเป็นต้องทราบชนิดของออบเจ็กต์ในระหว่างการประมวลผล ทั้งนี้เพื่อตรวจสอบว่าออบเจ็กต์นั้นเป็นชนิดที่ตรงกันกับที่ต้องการหรือไม่ ถ้าไม่ตรงจะได้ไม่ประมวลผลและส่งค่ากลับไปแจ้งให้ผู้ใช้ทราบ แต่ถ้าตรงกับชนิดที่ต้องการก็จะได้ประมวลผลต่อไป เช่น ซูเปอร์คลาส “A” มี ซับคลาส “B” และ “C” ดังนั้น “B” และ “C” จะใช้ค่าของกันและกันไม่ได้ ดังนั้นจึงจำเป็นต้องตรวจสอบว่าออบเจ็กต์ที่จะใช้งานนั้นอยู่ในคลาสใด มิฉะนั้น โปรแกรมจะผิดพลาดได้มักจะนำไปใช้กับเรื่องการเขียนโปรแกรมแบบอีเวนท์ ไดรฟ์ (Events Driven) หรือการตรวจสอบเหตุการณ์ต่าง ๆ ของคอมพิวเตอร์

รูปแบบ



รูปที่ ข-5 แสดงความเป็นอินสแทนซ์ของคลาสต่าง ๆ

```

public class InstanceOf{
    public static void main (String args[]){
        A a = new A();
        B b = new B();
        C c = new C();
        D d = new D();
        if(a instanceof A)
            System.out.println("aเป็น instanceof ของ A");
        if(b instanceof B)
            System.out.println("bเป็น instanceof ของ B");
        if(c instanceof C)
            System.out.println("cเป็น instanceof ของ C");
        if(c instanceof A)
            System.out.println("cเป็น instanceof ของ A");
        if(a instanceof C) //คำสั่งนี้ไม่เป็นจริง
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        System.out.println("a เป็น instanceof ของ C");

//ทุกออบเจกต์จะเป็นสมาชิกของคลาสที่ชื่อ "Object" โดยอัตโนมัติ
    if(a instanceof Object)
        System.out.println("a เป็น instanceof ของ Object");
    if(b instanceof Object)
        System.out.println("b เป็น instanceof ของ Object");
    if(c instanceof Object)
        System.out.println("c เป็น instanceof ของ Object");
    if(d instanceof Object)
        System.out.println("d เป็น instanceof ของ Object");
    }
} // class InstanceOf

class A{
    int i,j;
}

class B{
    int i,j;
}

class C extends A{
    int k;
}

class D extends A{
    int k;
}

```

Output:

```

a เป็น instanceof ของ A
b เป็น instanceof ของ B
c เป็น instanceof ของ C
c เป็น instanceof ของ A
a เป็น instanceof ของ Class Object
b เป็น instanceof ของ Class Object
c เป็น instanceof ของ Class Object
d เป็น instanceof ของ Class Object

```

จากโปรแกรมสังเกตได้ว่า ออบเจกต์ที่สร้างจากซบคลาสย่อมเป็นอินสแตนซ์ของซูเปอร์คลาสเสมอ แต่ออบเจกต์ที่สร้างจาก ซูเปอร์คลาส ไม่เป็นอินสแตนซ์ของซบคลาส และ ทุก ๆ ออบเจกต์ที่สร้างขึ้นจะต้องเป็นอินสแตนซ์ของคลาสออบเจกต์เสมอ

ข.7 การใช้คีย์เวิร์ด “super”

คีย์เวิร์ด “super” ใช้สำหรับการทำงานใน Subคลาส ซึ่งมีประโยชน์ 2 ประการคือ

1. เพื่อเรียกใช้สมาชิกของ ซูเปอร์คลาส ในกรณีที่มีซบคลาสและซูเปอร์คลาสใช้สมาชิก (ตัวแปร, เมธอด) ชื่อเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

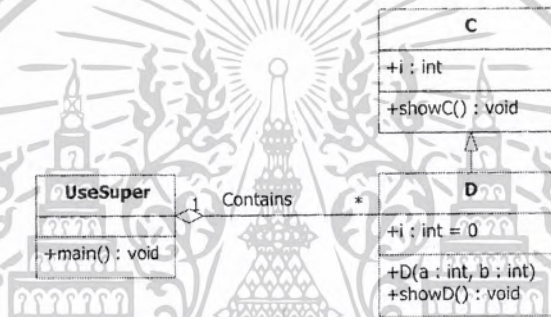
2. ชั้นคลาสสามารถเรียกใช้คอนสตรัคเตอร์เมทอดของซูเปอร์คลาสได้เลย โดยที่ชั้นคลาส ไม่จำเป็นต้องสร้างคอนสตรัคเตอร์ของตนเองขึ้นมาใหม่ (คอนสตรัคเตอร์จะไม่ถูกสืบทอดจากการอินเฮอริทด้วยคำสั่ง “extends” แต่จะเรียกใช้ได้โดยคีย์เวิร์ด “super”) ทำให้ประหยัดเวลาและไม่ซ้ำซ้อน

ข.7.1 การใช้คีย์เวิร์ด “super” เพื่อเรียกใช้สมาชิกของ ซูเปอร์คลาส

ใช้กล่าวอ้างถึงสมาชิกในซูเปอร์คลาสในกรณีที่ต้องการอ้างถึงสมาชิกของซูเปอร์คลาสภายในชั้นคลาส มีรูปแบบดังนี้

super.member

โดยที่ “member” หมายถึง สมาชิกของคลาส (ตัวแปร,เมทอด)



รูปที่ ข-6 แสดงความสัมพันธ์ของคลาสต่าง ๆ

```

public class UseSuper{
    public static void main(String []largs){
        D subOD = new D(3,4);
        subOD.showD();
    }
} // class UseSuper

class C{
    int i = 0;
    void showC(){
        System.out.println("i ในซูเปอร์คลาส : "+i);
    }
} // class C

class D extends C{
    int i = 0;
    D(int a,int b){
        super.i = a; // อ้างถึงตัวแปร i ของซูเปอร์คลาส
        i = b; // อ้างถึงตัวแปร i ของชั้นคลาส
    }

    void showD(){
        super.showC(); //ใช้super อ้างถึงเมทอดของซูเปอร์คลาส -ในที่นี้จะไม่ใช่super. ก็ได้
        System.out.println("iในชั้นคลาส: "+i);
    }
}
  
```

Output:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปะสิ่งเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นทำงานที่เมทอด “main” โดยสร้างออบเจ็กต์ชื่อ “subOD” ขึ้นจากคลาส “D” ซึ่งเป็นซัพคลาสของคลาส “C” เมื่อ คลาส “D” ถูกเรียกใช้ ก็จะไปเรียกคอนสตรัคเตอร์ “D” และส่งพารามิเตอร์ไปทำงานด้วย (3 และ 4) โดยนำตัวเลข 3 ซึ่งเก็บในตัวแปร a ไปใส่ในตัวแปร i ของคลาส “C” (super.i = a) และนำตัวเลข 4 ซึ่งเก็บในตัวแปร b ไปใส่ในตัวแปร i ของคลาส D (i = b)

ภายในเมทอด “showD()” ของคลาส “D” มีการเรียกใช้เมทอด “showC()” ของซูเปอร์คลาส ซึ่งความเป็นจริงแล้วภายในซัพคลาสสามารถที่จะเรียกใช้เมทอดของซูเปอร์คลาส ได้โดยไม่ต้องระบุคำสั่ง “super” ก็ได้ แต่ใช้คำสั่ง “super” จะมีประโยชน์อย่างมากเมื่อเมทอดของซูเปอร์คลาส และ ซัพคลาส มีชื่อเดียวกัน เราสามารถใช้คีย์เวิร์ด “super” แยกแยะว่าจะเรียกใช้เมทอดของคลาสซูเปอร์คลาส หรือ ซัพคลาส

เรียกใช้เมทอด “showD()” ของออบเจ็กต์ “subOD” เพื่อแสดงค่า i ทั้งสองคลาส

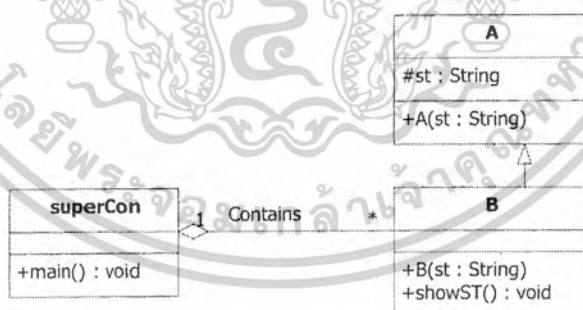
ข.7.2 การใช้คีย์เวิร์ด “super” เพื่อเรียกใช้คอนสตรัคเตอร์เมทอดของซูเปอร์คลาส

ถ้าซูเปอร์คลาสมีการสร้างคอนสตรัคเตอร์ไว้แล้วซัพคลาสจะสามารถใช้คีย์เวิร์ด “super” เพื่อเรียกใช้คอนสตรัคเตอร์ ได้เลยโดยไม่ต้องสร้างขึ้นมาใช้งานเอง

รูปแบบ

super(argument)

โดยมีเงื่อนไขว่าคีย์เวิร์ด “super” นี้จะต้องเป็นคำสั่งแรกที่ใช้งานภายในคอนสตรัคเตอร์ของซัพคลาสเท่านั้น ไม่สามารถเรียกใช้คอนสตรัคเตอร์ของซูเปอร์คลาส ได้จากส่วนอื่นของซัพคลาส



รูปที่ ข-7 แสดงการเรียกใช้ คอนสตรัคเตอร์ของซูเปอร์คลาส

```

public class superCon{
public static void main (String []args){
    new B("Welcome to Bangkok,Thailand!").showST();
}
}
class A{
protected String st;
A(String st){
    This.st = st;
}
}
// class A
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

class B extends A{
    B(String st){
        super(st); //เรียกใช้คอนสตรัคเตอร์ของซูเปอร์คลาส
    }

    public void showST(){
        System.out.println("Message ที่ส่งไปยังคอนสตรัคเตอร์ คือ "+st);
    }
}
//class B

```

Output :

Message ที่ส่งไปยัง Constructor คือ Welcome to Bangkok,Thailand

ภายใน ซูเปอร์คลาส “A” มี คอนสตรัคเตอร์เมทอด โดยจะรับเอาอาร์กิวเมนต์เข้ามาแล้วนำค่าไปเก็บไว้ในตัวแปร “st” ซึ่งมีชนิดการเข้าถึงเป็นโพรเทค (protected) ในคอนสตรัคเตอร์เมทอดของซับคลาส “B” มีการเรียกใช้คีย์เวิร์ด “super” เพื่อเรียกใช้คอนสตรัคเตอร์ของซูเปอร์คลาส “A” ซึ่งเมื่อมาถึงคำสั่งนี้ตัวแปลภาษาจาวาก็จะเรียกใช้ คอนสตรัคเตอร์ของซูเปอร์คลาสทันที

การสร้างออบเจกต์จากคลาส “B” พร้อมกับเรียกใช้เมทอด “showST()” โดยมีการทำงานตามลำดับดังนี้

- เมื่อพบ “extends” โปรแกรมจะไปทำความเข้าใจกับสมาชิกในคลาส “A” ทั้งหมด แต่ไม่ได้ทำงานใดๆ
- สร้างออบเจกต์ของคลาส “B” ดังนั้น จึงต้องทำงานตามคอนสตรัคเตอร์ของคลาส “B” แต่คอนสตรัคเตอร์ของคลาส “B” กำหนดให้ไปทำงานที่คอนสตรัคเตอร์ของซูเปอร์คลาส คือ คลาส “A” แทน โดยการรับค่าตัวแปรสตริงที่ส่งเข้ามา มาเก็บไว้ในตัวแปรของคลาส (this.)
- ทำงานในเมทอด “showST()”

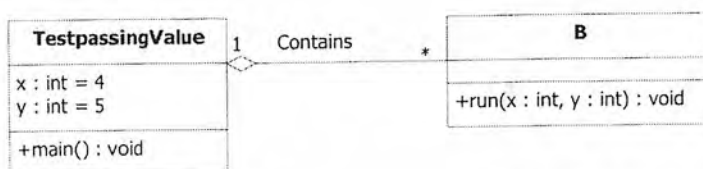
ข.8 การส่งผ่านตัวแปร

การส่งผ่านตัวแปร คือ การส่งข้อมูลผ่านตัวแปร (อาร์กิวเมนต์) ไปให้กับพารามิเตอร์ของเมทอดที่สร้างขึ้น เพื่อนำค่าดังกล่าวไปใช้งานอย่างใดอย่างหนึ่งในเมทอดซึ่งจะมีอยู่ 2 ลักษณะคือ การส่งผ่านโดยใช้ค่า (Pass by Value) และการส่งผ่านโดยการอ้างอิง (Pass by Reference)

ข.8.1 การส่งผ่านตัวแปรด้วยค่า (Pass by Value)

เป็นการส่งผ่านค่าตัวแปรทั่วไป โดยคัดลอกข้อมูลจากตัวแปรหนึ่งไปยังอีกตัวแปรหนึ่งเพื่อทำกระบวนการใด ๆ ที่ต้องการ ดังตัวอย่างสังเกตบรรทัด “Tb.run(x,y)” เป็นการส่งผ่านค่าข้อมูลของตัวแปร x และ y ของ คลาส “TestpassingValue” ไปทำกิจกรรมภายในเมทอดของออบเจกต์ที่สร้างขึ้นจาก คลาส “B” ซึ่งค่าข้อมูลของ x และ y ภายใน คลาส “TestpassingValue” ไม่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-8 แสดงการส่งผ่านค่าให้กับเมทอดแบบการส่งผ่านด้วยค่า (Pass by Value)

```

public class TestpassingValue{
    static int x=4,y=5;
    public static void main(String args[]){
        System.out.println("ค่า x ก่อนส่งไปให้เมทอด run =" +x);
        System.out.println("ค่า y ก่อนส่งไปให้เมทอด run =" +y);
        B Tb=new B(); Tb.run(x,y);
        System.out.println("ค่า x หลังส่งไปให้เมทอด run =" +x);
        System.out.println("ค่า y หลังส่งไปให้เมทอด run =" +y);
    }
} //class TestpassingValue

class B{
    public void run(int x,int y){
        x+=10;
        y+=10;
        System.out.println("ค่า x ที่ส่งมาบวกด้วย 10 =" +x);
        System.out.println("ค่า y ที่ส่งมาบวกด้วย 10 =" +y);
    }
} //class B
  
```

Output:

```

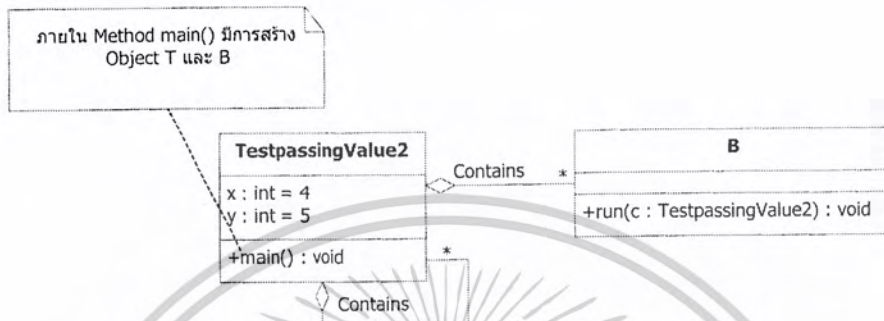
ค่า x ก่อนส่งไปให้เมทอด run =4
ค่า y ก่อนส่งไปให้เมทอด run =5
ค่า x ที่ส่งมาบวกด้วย 10 =14
ค่า y ที่ส่งมาบวกด้วย 10 = 15
ค่า x หลังส่งไปให้เมทอด run = 4
ค่า y หลังส่งไปให้เมทอด run = 5
  
```

จากตัวอย่างภายในเมทอด “main” มีการแสดงค่าของตัวแปร x และ y หลังจากนั้นจะเรียกใช้เมทอด “run” ของออบเจ็กต์ “Tb” (ออบเจ็กต์ของคลาส “B”) พร้อมกับส่งค่าของตัวแปร x และ y เป็นพารามิเตอร์ลิสต์ให้กับเมทอดดังกล่าวด้วย หลังจากนั้นภายในเมทอด “run” ของออบเจ็กต์ “Tb” จะมีการนำค่าอาร์กิวเมนต์ต่าง ๆ ที่รับเข้ามา มาทำการคำนวณค่าใหม่โดยการเพิ่มค่าของอาร์กิวเมนต์ทั้งสองขึ้นอีก 10 แล้วแสดงผลออกทางจอภาพ เมื่อทำงานในเมทอด “run” เสร็จแล้วโปรแกรมจะเข้ามาทำงานต่อ ซึ่งก็คือการแสดงผลของตัวแปร x และ y อีกครั้งหนึ่ง จะเห็นว่าค่าของตัวแปร x และ y ก่อนและหลังส่งค่าไปทำงานในเมทอด “run” มีค่าเท่าเดิม

จะเห็นว่าค่าของตัวแปร x และ y ในเมทอด “main” ก่อนและหลังส่งค่าไปให้เมทอด “run” ไม่มีการเปลี่ยนแปลง ถึงแม้ภายในเมทอดนั้น ๆ จะนำค่าดังกล่าวไปเปลี่ยนค่าใหม่ก็ตาม เพราะการส่งค่าให้กับออบเจ็กต์ของเมทอดเป็นเพียงการคัดลอกค่าไปเท่านั้นจึงไม่มีผลกระทบต่อค่าของตัวแปรต้นฉบับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.8.2 การส่งผ่านด้วยการอ้างอิง (Pass by Reference)

เป็นการส่งผ่านค่าโดยใช้ออบเจ็กเป็นตัวรับส่งข้อมูล ซึ่งวิธีนี้จะมีพอยเตอร์ (Pointer) ซึ่งไปยังตัวแปรของออบเจ็กที่ต้องการอ้างอิงถึง ดังนั้นถ้าค่าของตัวแปรของออบเจ็กต้นฉบับเปลี่ยนแปลงไป ค่าของตัวแปรของออบเจ็กที่ถูกอ้างอิงถึงก็จะเปลี่ยนแปลงตามไปด้วยเช่นกัน การส่งผ่านค่าโดยใช้ออบเจ็กนั้นจะสามารถอ้างอิงถึงกันได้หมด เรียกการส่งผ่านโดยวิธีการนี้ว่า “การส่งผ่านด้วยการอ้างอิง”



รูปที่ ข-9 แสดงการส่งผ่านตัวแปรโดยใช้ออบเจ็กอ้างอิง (Reference)

```

public class TestpassingValue2{
    static int x=4,y=5;
    public static void main(Strings args[]){
        TestpassingValue2 T = new TestpassingValue2();
        System.out.println("ค่า x ก่อนส่งไปให้เมทอด run =" +T.x);
        System.out.println("ค่า y ก่อนส่งไปให้เมทอด run =" +T.y);
        B b = new B();
        b.run(T);
        System.out.println("ค่า x หลังส่งไปให้เมทอด run =" +T.x);
        System.out.println("ค่า y หลังส่งไปให้เมทอด run =" +T.y);
    }
}
//end method main
//class TestpassingValue2

class B{
    public void run (TestpassingValue2 c){
        c.x+=10;
        c.y+=10;
        System.out.println("ค่า x ที่ส่งมาบวกด้วย 10="+c.x);
        System.out.println("ค่า y ที่ส่งมาบวกด้วย 10="+c.y);
    }
}
//end method run
//class B
  
```

Output:

```

ค่า x ก่อนส่งไปให้เมทอด run = 4
ค่า y ก่อนส่งไปให้เมทอด run = 5
ค่า x ที่ส่งมาบวกด้วย 10 = 14
ค่า y ที่ส่งมาบวกด้วย 10 = 15
ค่า x หลังส่งไปให้เมทอด run = 14
ค่า y หลังส่งไปให้เมทอด run = 15
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเมธอด “main()” แสดงค่าตัวแปร x และ y หลังจากนั้นส่งออบเจ็กต์ “T” ไปเป็นพารามิเตอร์ให้กับเมธอด “run()” (อาร์กิวเมนต์ของเมธอด “run()” มีชนิดเป็นคลาส “TestpassingValue2”) ซึ่งภายในเมธอด “run()” จะมีการคำนวณค่าตัวแปรของออบเจ็กต์ที่ส่งมาโดยการเพิ่มค่าขึ้นอีก 10 ซึ่งในที่นี้ออบเจ็กต์ที่ส่งเข้ามาประกอบด้วยสองตัวแปรคือ x และ y หลังจากคำนวณค่าตัวแปรของออบเจ็กต์เสร็จแล้วก็แสดงค่าดังกล่าวออกทางจอภาพหลังจากทำงานภายในเมธอด “run” เสร็จแล้ว โปรแกรมก็จะเข้ามาทำงานที่เมธอด “main()” อีกครั้งโดยมีการแสดงค่าตัวแปร x และ y ของออบเจ็กต์ “T” อีกครั้ง ซึ่งจะเห็นว่าค่าตัวแปร x และ y ของออบเจ็กต์ “T” เปลี่ยนแปลงไปโดยจะมีค่าเท่ากับค่าที่มีการคำนวณใหม่ภายในเมธอด “run()”

จะเห็นว่าค่าตัวแปร x และ y ก่อนและหลังส่งออบเจ็กต์ “T” ไปให้กับเมธอด “run” มีค่าไม่เท่ากันเพราะการส่งพารามิเตอร์เป็นออบเจ็กต์นั้น เป็นการอ้างอิงถึงตัวแปรในพื้นที่หน่วยความจำเดียวกันและเมื่อกระทำการเปลี่ยนแปลงใด ๆ กับข้อมูลของตัวแปรตัวหนึ่งก็จะมีผลกระทบต่ออีกตัวแปรหนึ่งที่อ้างอิงด้วย

ข.9 สวิงค์ คอมโพเนนต์ (Swing Component)

ภายในแพ็คเกจ “swing” ประกอบด้วย คอมโพเนนต์อีกมากมายเพื่อใช้สำหรับออกแบบจิวไอส์ (GUIs) ขึ้นมาใช้งาน ส่วนประกอบของคอมโพเนนต์ของสวิงค์ มีการสืบทอด(Inherit)และการอิมพลีเมนต์มาจาก คลาสและอินเทอร์เฟสมากมาย โดยมาจากการสืบทอดคอมโพเนนต์ต่าง ๆ ดังต่อไปนี้

คอมโพเนนต์ของสวิงมีอยู่มากมายและจะขึ้นต้นด้วยตัวอักษรเจ (J) เช่น JButton , JLabel และ JPanel เป็นต้น โดยทุกคอมโพเนนต์จะสืบทอดลักษณะมาจากคลาส "JComponent" แต่ JFrame , JApplet และ JDialog สืบลักษณะมาจากคลาส "Frame" , "Applet" , "Dialog" ตามลำดับและถึงแม้ว่า สวิงคอมโพเนนต์จะเป็นคอมโพเนนต์ ที่พัฒนาต่อมาจากเอดับบลิวิตคอมโพเนนต์แต่ก็ยังสามารถนำ เอดับบลิวิตคอมโพเนนต์บางตัวมาใช้ทำงานร่วมกับสวิงคอมโพเนนต์ได้เช่นกัน

สิ่งที่คอมโพเนนต์ของสวิงได้มาจากคอนเทนเนอร์ (Container) และ JComponent มีอยู่มากมายทั้ง เมตธอด , ตัวแปร และคุณสมบัติของคอมโพเนนต์นั้น ๆ โดยสำคัญมีดังนี้

คุณสมบัติที่ได้จาก JComponent

1. คลาสและเมตธอดที่สนับสนุนการจัดเลเอาท์ (Layout) ต่าง ๆ
2. คลาสและเมตธอดที่สนับสนุนการวาดคอมโพเนนต์
3. คลาสและเมตธอดที่สนับสนุนการจัดการกับเหตุการณ์ (Event)

คุณสมบัติที่ได้จากคอนเทนเนอร์

1. เกี่ยวกับการเพิ่มคอมโพเนนต์ลงในคอนเทนเนอร์
2. วิธีในการจัดวางเลเอาท์ของคอมโพเนนต์

คอมโพเนนต์ในสวิงแบ่งออกเป็นชนิดต่าง ๆ ได้ดังตาราง

ชนิดของ Component	Component
Atomic controls	JLabel , JButton , JCheckBox , JRadioButton , JToggleButton , JComboBox , JScrollBar , JSeparator , JSlider , JSpinner , JProgressBar
Complex data components	JTable , JTree , JList , JFileChooser , JColorChooser , JOptionPane
Text components	JTextField , JFormattedTextField , JPasswordField , JTextArea , JTextPane , JEditorPane
Menu components	JMenuBar , JMenu , JPopupMenu , JMenuItem , JCheckboxMenuItem , JRadioButtonMenuItem
Layout containers	JPanel , JScrollPane , JSplitPane , JTabbedPane , JDesktopPane , JToolBar
Top-level window component (Top-level Swing Component)	JFrame , JDialog , JWindow , JInternalFrame , JApplet

ตารางที่ ๑-1 แสดงคอมโพเนนต์ชนิดต่าง ๆ ในสวิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.10 การเรียกใช้โปรแกรมอื่นด้วยจาวา

ในการเรียกใช้โปรแกรมอื่นที่มีนามสกุลเป็น “exe” จาวาได้รองรับการเรียกใช้โปรแกรมอื่นโดยมีรูปแบบการเรียกใช้ดังตัวอย่างต่อไปนี้

ตัวอย่าง การเรียกใช้โปรแกรม Acrobat 5.0 เพื่อเปิดไฟล์ CONTROL\HP LaserJet 4L ผ่านทางจาวา

```
String command = "C:\\Program Files\\Adobe\\Acrobat
5.0\\Reader\\AcroRd32.exe /t "+selectedDocument+" \\\\CONTROL\\HP LaserJet
4L";
Runtime rn = Runtime.getRuntime();
Process process = rn.exec(command);
process.waitFor();
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

รายละเอียดขอโค้ดหลักของโปรแกรม

```

package control;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.net.*;
import javax.media.*;
import java.util.*;

import javax.swing.event.*;
import javax.swing.border.*;
import javax.media.protocol.*;
import javax.media.format.*;
import javax.media.control.*;
import javax.media.datasink.*;

public class Interface extends JFrame {
    //Variable declarations
    //*****
    //      JComponents      //
    //*****
    private Container display;
    private JPanel buttonPanel, listPanel;
    private JButton Play, Stop, Capture, Copy, FromURL, PlayRec;
    private JMenuBar menubar;
    private JMenu
menuFile, menuView, menuPlay, menuAbout, submenutaskbar, submenuscreen;
    private JMenu subDevice, subFormat;
    private ButtonGroup taskGroup, fullGroup, deviceGroup, formatGroup;
    private JList songList;
    private JScrollPane songScroll;

    //*****
    //      Objects      //
    //*****
    private PlayerPart playerPart, toStop, toPlay;
    private RecordingPart setting, toRecord;
    private CopyPart toCopy;
    //*****
    //      Others      //
    //*****
    private File mediaFile[] = new File[20] ;
    private URL fileURL;
    private boolean Close;
    private String songFile[] = new String[40];
    private int i=1, playList=0;
    private String directory = "c://JPV";
    //-----
    //Record filed-----
    private RecordingPart processor;
    private Processor process;
    private boolean checkrec = false;
    //-----
    public Interface(){
        super("J Player Variety");
        display = getContentPane();
        File f = new File(directory);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

f.mkdir();

//-----
//                               add MenuBar
//-----
menubar = new JMenuBar();
menuFile = new JMenu("File");
menuView = new JMenu("View");
menuPlay = new JMenu("Play");
menuAbout = new JMenu("About");
//-----Create MenuItem to MenuFile-----
//=====Action for Open File menu=====
JMenuItem OpenFile = new JMenuItem("Open File..");
OpenFile.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent event) {
            PlayerPart playerPart = new PlayerPart(display);

            mediaFile = playerPart.getFile();
            for(int c=0;c<=mediaFile.length;c++){
                if (mediaFile[c] != null){

                    try{
                        fileURL = mediaFile[c].toURL();
                        songFile[i] = fileURL.toString();
                        i++;
                    }
                    catch(MalformedURLException badURL){
                        badURL.printStackTrace();
                        playerPart.showMessageDialog("Bad URL");
                    }
                    toPlay = playerPart;
                    display.repaint();
                }
            }
        }
    });
OpenFile.setMnemonic('O');
menuFile.add(OpenFile);
//*****//
menuFile.addSeparator();
//*****//
//=====--End Action for Open File menu=====
JMenuItem clear = new JMenuItem("Clear List");
clear.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent event) {
            for(int a=1;a<=songFile.length;a++){
                songFile[a] = null;
                display.repaint();
            }
        }
    });
clear.setMnemonic('c');
menuFile.add(clear);
//    display.repaint();
//=====Action for Exit menu=====
JMenuItem exit = new JMenuItem("Exit");
exit.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent event) {
            { System.exit(0);
            }
        }
    });
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

);
exit.setMnemonic('x');
menuFile.add(exit);
menuFile.setMnemonic('F');
//=====End Action for Exit
menu=====
//-----Create MenuItem to MenuPlay-----
JMenuItem play = new JMenuItem("Play");
play.setMnemonic('l');
play.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent event)
            { playAction();
            }
    }
);

menuPlay.add(play);

//-----
JMenuItem stop = new JMenuItem("Stop");
stop.setMnemonic('t');
stop.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event)
        { stopAction();
        }
});

menuPlay.add(stop);

//-----
menuPlay.addSeparator();
JMenu subRecord = new JMenu("Record");

JMenuItem setRecord = new JMenuItem("Enable Recording");
setRecord.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event){
        RecordingPart setting = new RecordingPart();
        setting.RecordSetting();
        toRecord = setting;
    }
});

//=====End Action for Setting>Record>Save to File menu=====
setRecord.setMnemonic('s');
subRecord.add(setRecord);
subRecord.addSeparator();
JMenuItem record = new JMenuItem(">>Record without song");
record.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event)
        { captureAction();
        }
});

record.setMnemonic('e');
subRecord.add(record);

//-----
JMenuItem recordWithSong = new JMenuItem(">>Record with song");
recordWithSong.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event)
        { playAction();
        captureAction();
        }
});

recordWithSong.setMnemonic('w');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

subRecord.add(recordWithSong);
menuPlay.add(subRecord);
//-----
JMenuItem copy = new JMenuItem("Copy");
copy.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event)
        { copyAction();
          }
});

copy.setMnemonic('o');
menuPlay.add(copy);
//-----

menuPlay.setMnemonic('P');
//-----
//-----Create MenuItem to MenuSetting-----

menuAbout.setMnemonic('a');
//===== Action for Setting>About menu=====
JMenuItem subAbout = new JMenuItem("About");
subAbout.addActionListener( new ActionListener(){
    public void actionPerformed(ActionEvent event){
//        display.setBackground(Color);
JOptionPane.showMessageDialog(null, "                J Player Variety\n"+
    "Advisor : Dr.Surin Kititornkul\n"+
    "    Presented by\n"+
    "    Panumas Wongsalert\n"+
    "    Wipawee Wishayashakorn\n"+
    "Faculty of Computer Engineering\n"+
    "                KMITL\n",
    "J Player Variety",JOptionPane.WARNING_MESSAGE);
    }
});
subAbout.setMnemonic('a');
menuAbout.add(subAbout);

//=====End Action for Setting>About menu=====
//-----
//                Add JMenuItem to JMenuBar
//-----

menubar.add(menuFile);
menubar.add(menuPlay);
menubar.add(menuAbout);

menubar.setBorderPainted();
setJMenuBar(menubar);

//-----
//                End Of Menu Bar
//-----

//-----
//                JList Panel display list of file selected.
//-----

songFile[0] = new String("List of Songs...");
songList = new JList(songFile);
songList.setVisibleRowCount(3);
songList.setFixedCellHeight(15);
songList.setFixedCellWidth(200);

songList.setSelectionBackground(Color.orange);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

songList.setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
songScroll = new JScrollPane(songList);
MouseListener mouseListener = new MouseAdapter() {
    public void mouseClicked(MouseEvent e) {
        if (e.getClickCount() == 2) {
            int index = songList.locationToIndex(e.getPoint());
            System.out.println("Double clicked on Item " + index);
            playAction();
        }
    }
};
songList.addMouseListener(mouseListener);

songList.addListSelectionListener(new ListSelectionListener(){
    public void valueChanged(ListSelectionEvent event){
        int a = 0;
        a = songList.getSelectedIndex();
        // System.out.println("a =" + a);
        playList = a;
    }
});

//-----
//                               End of list.
//-----
//-----
//                               Panel for playing capabilities = "buttonPanel"
//-----
//====="buttonPanel"====
buttonPanel = new JPanel();
buttonPanel.setBackground(Color.black); //add Panel Part1 to
Container
display.add(buttonPanel, BorderLayout.SOUTH); //Position of buttons

//Decorate Buttons style
Border b = new BevelBorder(BevelBorder.RAISED);

/////Play Button/////
Icon Iplay = new ImageIcon("play.gif");
Play = new JButton(Iplay);
Play.setToolTipText("Play file");
Play.setBackground(Color.white);
Play.setBorder(b);
//=====Action for Play button=====
Play.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event){
        playAction();
    }
});
//=====Action for Play and Record button=====
//=====
// Play and Record
Icon IplayRec = new ImageIcon("playrecord.gif");
PlayRec = new JButton(IplayRec);
PlayRec.setToolTipText("Play and Record your voice.");
PlayRec.setBackground(Color.white);
PlayRec.setBorder(b);
//=====Action for Play button=====
PlayRec.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event){
        playAction();
        captureAction();
    }
});

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

});

//=====END Action for Play and Record
button=====

/////Stop Button/////
Icon Istop = new ImageIcon("stop.gif");
Stop = new JButton(Istop);
Stop.setToolTipText("Stop.");
Stop.setBackground(Color.white);
Stop.setBorder(b);
//===== Action for Stop button=====
Stop.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent event){
        stopAction();
    }
});
//=====END Action for Stop button=====

//Capture button >>> Record sound and Save file
Icon Icapture = new ImageIcon("record.gif");
Capture = new JButton(ICapture);
Capture.setToolTipText("Record your voice and save.");
Capture.setBorder(b);
Capture.setBackground(Color.white);
//=====Action for Capture button=====
Capture.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent event){
            captureAction();
        }
    }
);
//=====End Action for Capture button=====

/////Copy button /////
//>>> copy file from CD,Rename and save to system.
Icon Icopy = new ImageIcon("copy.gif");
Copy = new JButton(ICopy);
Copy.setToolTipText("Copy Media file into your system");
Copy.setBorder(b);
Copy.setBackground(Color.white);
//===== Action for Copy button=====
Copy.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        copyAction();
    }
});
//=====End Action for Copy button=====

//=====END Action for OpenURL=====
buttonPanel.add(Play);
buttonPanel.add(PlayRec);
buttonPanel.add(Stop);
buttonPanel.add(Capture);
buttonPanel.add(Copy);
// buttonPanel.add(FromURL);
buttonPanel.add(songScroll);

setSize(1024,738);
setVisible(true);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----
//-----Play Action Part-----
public void playAction(){
    boolean Stop = false;
    System.out.println("songFile[playList] =" + songFile[playList]);
//-----Return .MPG or not-----
    String mpg1,mpg2,mpg3;
    mpg1 = ".mpg";
    mpg2 = ".MPG";
    mpg3 = ".mpeg";
    int lengname;
    String detect;
    detect = songFile[playList].toString();
    lengname = detect.length();
//get file to string then get string to char and compare .mpg/.MPG //
char checkmpg[] = new char[4];
    detect.getChars(lengname-4, lengname, checkmpg, 0);
    String tmp1 = new String(checkmpg);
    System.out.println("tmp1 =" + tmp1);

    if (tmp1.equals(mpg1) || tmp1.equals(mpg2) || tmp1.equals(mpg3))
        {toPlay.setIsMpeg(true); }
    else {toPlay.setIsMpeg(false);}

//-----
    Stop = toPlay.makePlayer(songFile[playList]);
    Close = Stop;
    toStop = toPlay;
}
//-----
//-----Stop Action Part-----
public void stopAction(){
    if (Close){
        toStop.removePlayerComponent();
        System.out.println("That's right.");}

    if(checkrec){
        process.stop();
        process.close();
        checkrec = false;
        System.out.println( "checkrec = " + checkrec);
        System.out.println ( "Capture closed." );
    }
    else System.out.println("Bad Action");

}
//-----
private DataSink dataSink;
//-----Capture Action Part-----
public void captureAction(){
    System.out.println ( "Capture open." );
    process = toRecord.getprocess();
    dataSink = toRecord.getdatasink();

    checkrec = true;
    System.out.println("checkrec = " + checkrec);
}
//-----
//-----Copy Action Part-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void copyAction(){
    CopyPart toCopy = new CopyPart();
    toCopy.openFile("Search File *.dat for Record to Karaoke Temp.");
    toCopy.setSong();
}
//-----

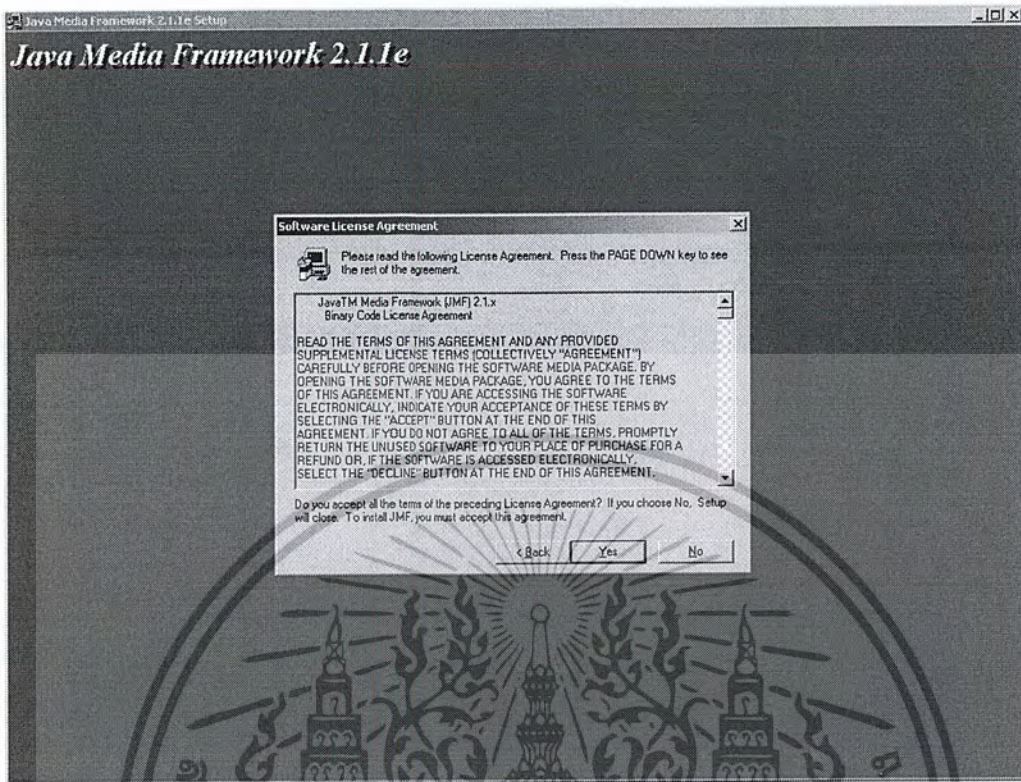
public static void main(String args[]){
    Interface application = new Interface();
    //application.setLocation(200,150);
    application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}

```



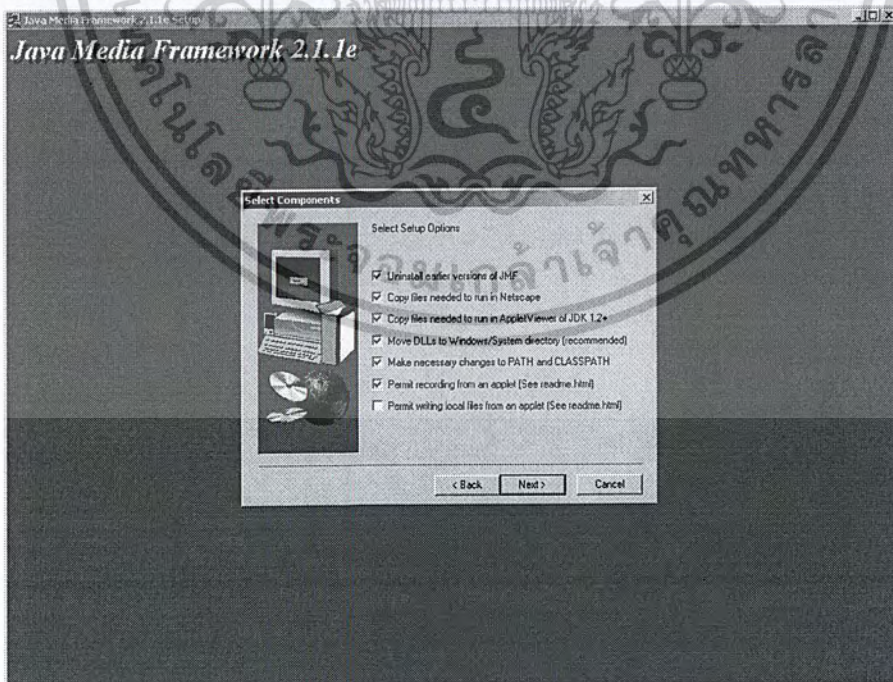
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ต่อจากนั้นจะเป็นการตอบข้อตกลงเพื่อให้ปฏิบัติตาม คลิกที่ YES ดังรูป



รูปที่ 3- แสดงข้อตกลงของโปรแกรม

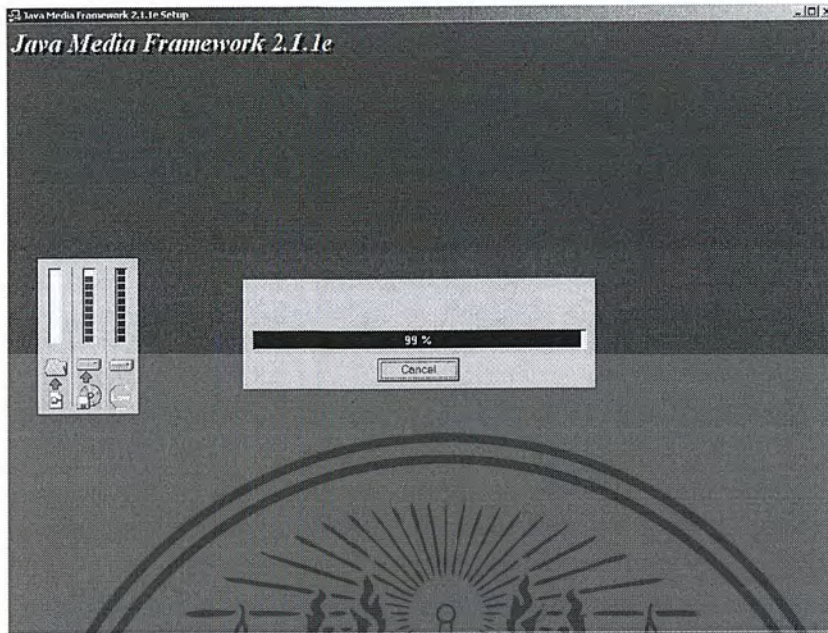
4. จากนั้น โปรแกรมจะให้เลือกอบช้ัน (option) ต่าง ๆ ให้คลิกดังรูป



รูปที่ 4- แสดงอบช้ันต่าง ๆ ของโปรแกรมการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ต่อจากนั้นระบบจะทำการติดตั้งโปรแกรมและแสดงหน้าต่างดังรูป

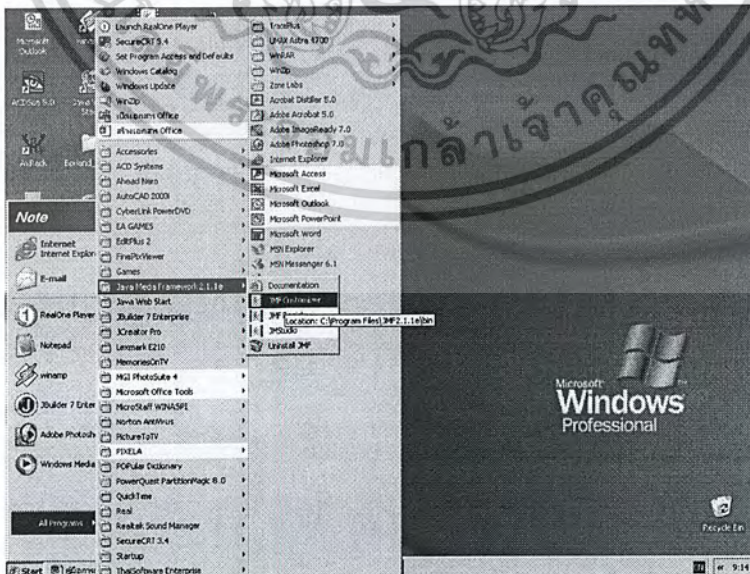


รูปที่ ๕-5 แสดงการทำงานในการติดตั้งโปรแกรม

6. จากนั้นจะปรากฏหน้าต่างถามว่าให้ทำการรีสตาร์ท (restart) เครื่องใหม่หรือไม่ เลือกรีสตาร์ทใหม่ เมื่อเครื่องทำการรีสตาร์ทใหม่ก็จบการติดตั้งโปรแกรม

๖.2 การตั้งค่า JMF Customizer

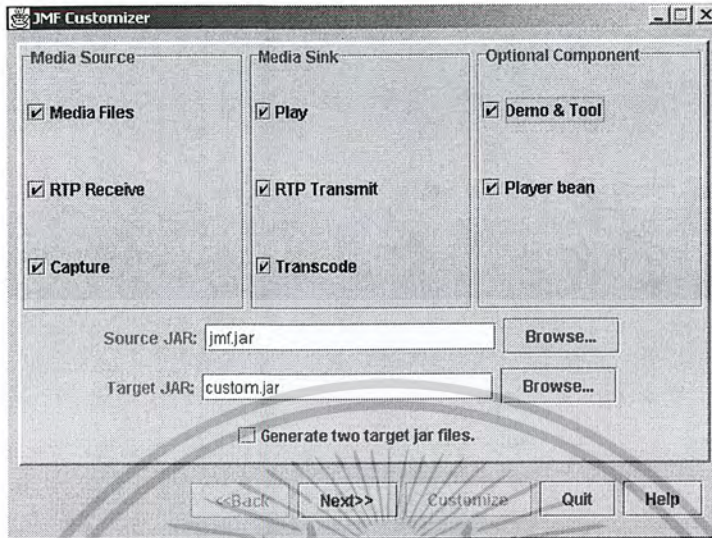
1. คลิกที่ start แล้วเลือกที่ All program ต่อจากนั้นเลือกที่โปรแกรม JMF 2.1.1e แล้วเลือก JMF Customizer ดังรูป



รูปที่ ๖.2 -1 แสดงตำแหน่งของโปรแกรมการตั้งค่า

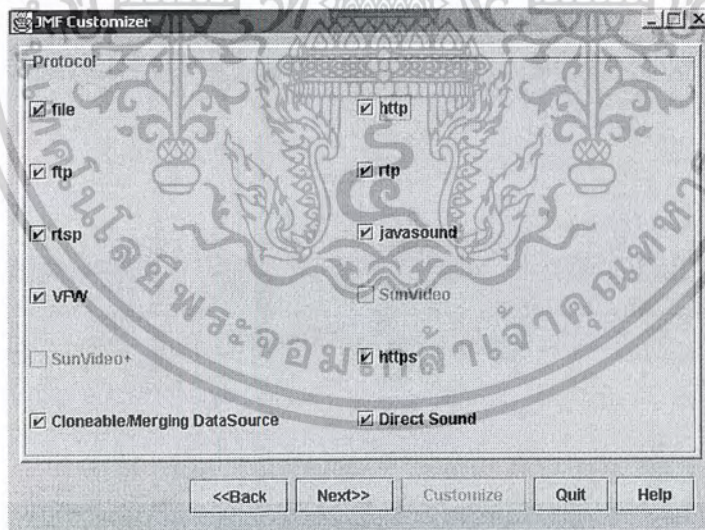
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คลิกเลือกอบชั้นต่าง ๆ ดังรูป แล้วคลิก Next



รูปที่ ง.2 -2 แสดงการเลือกอบชั้นต่าง ๆ

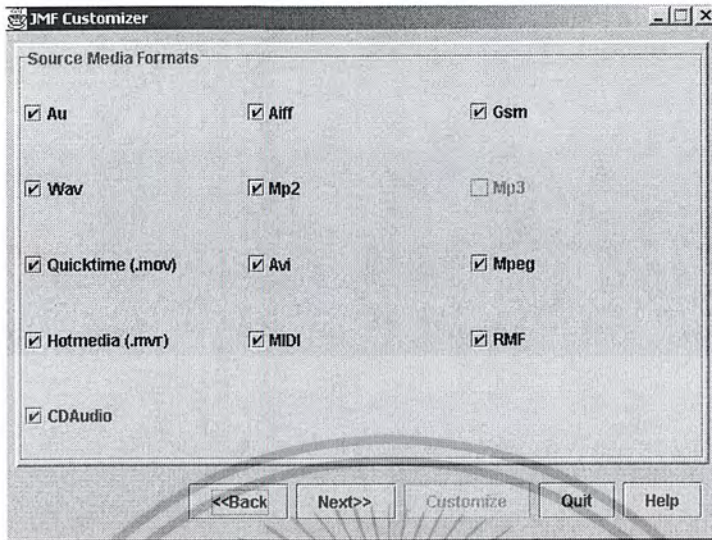
คลิก Next



รูปที่ ง.2 -3 แสดงการเลือกอบชั้นต่าง ๆ

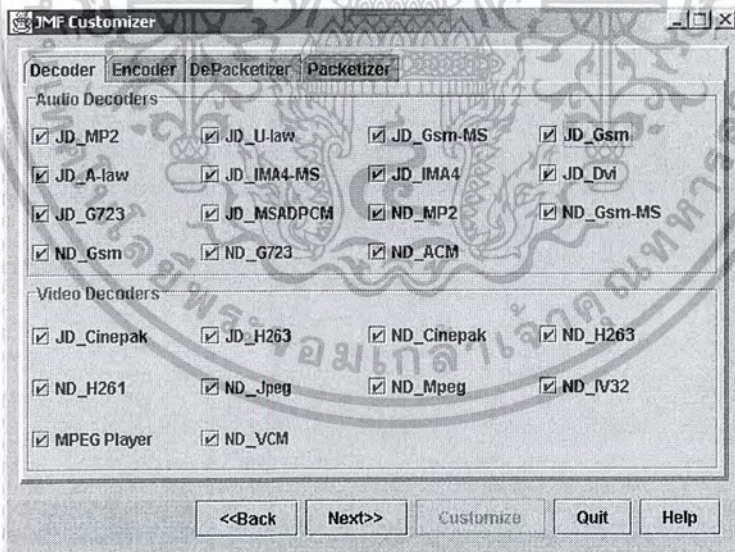
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลิก Next



รูปที่ ง.2 -4 แสดงการเลือกอบนชั้นต่าง ๆ

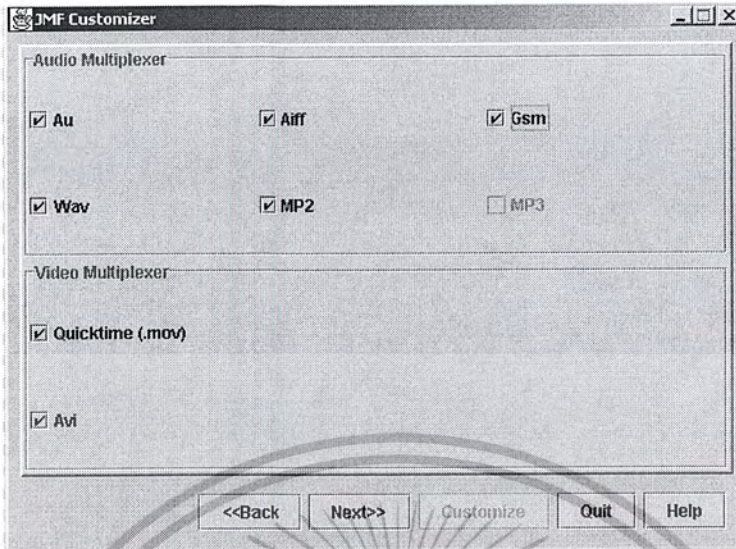
คลิก Next



รูปที่ ง.2 -5 แสดงการเลือกอบนชั้นต่าง ๆ

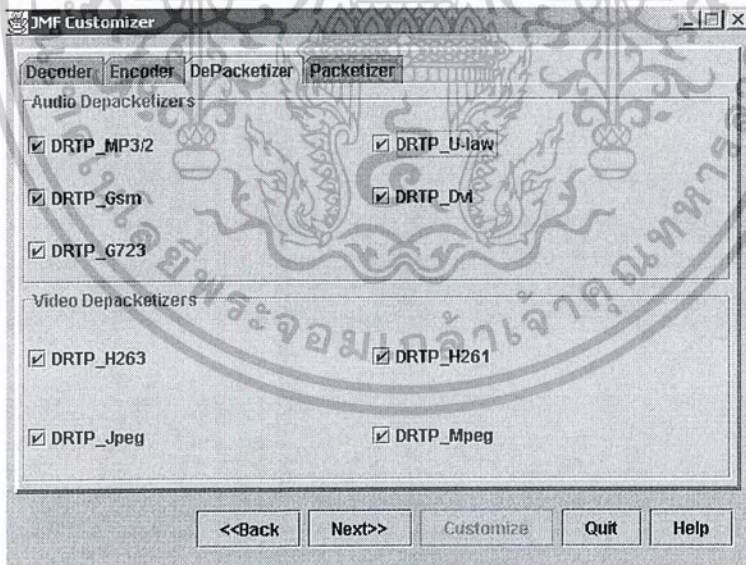
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลิก Next



รูปที่ 3.2-6 แสดงการเลือกอบชั้นต่าง ๆ

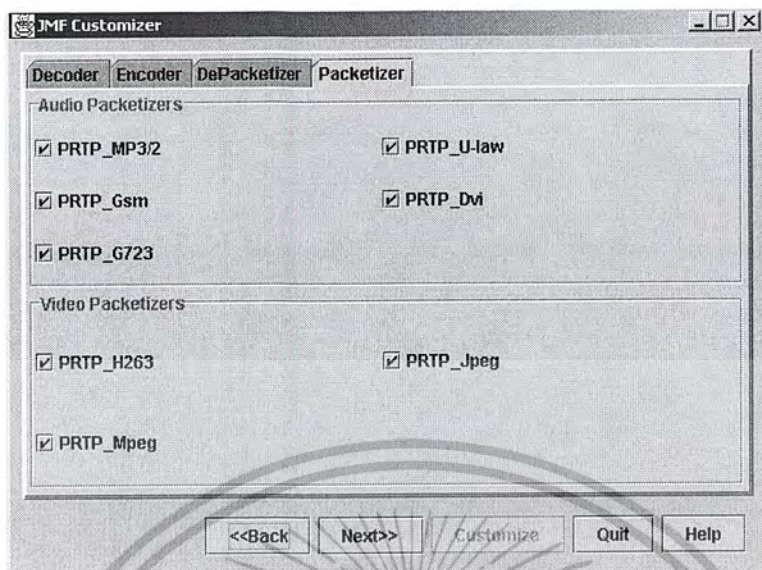
คลิก Next



รูปที่ 3.2-7 แสดงการเลือกอบชั้นต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลิก Next



รูปที่ 3.2 -8 แสดงการเลือกอบชั้นต่าง ๆ

เมื่อเลือกอบชั้นต่าง ๆ แล้วคลิก Customize



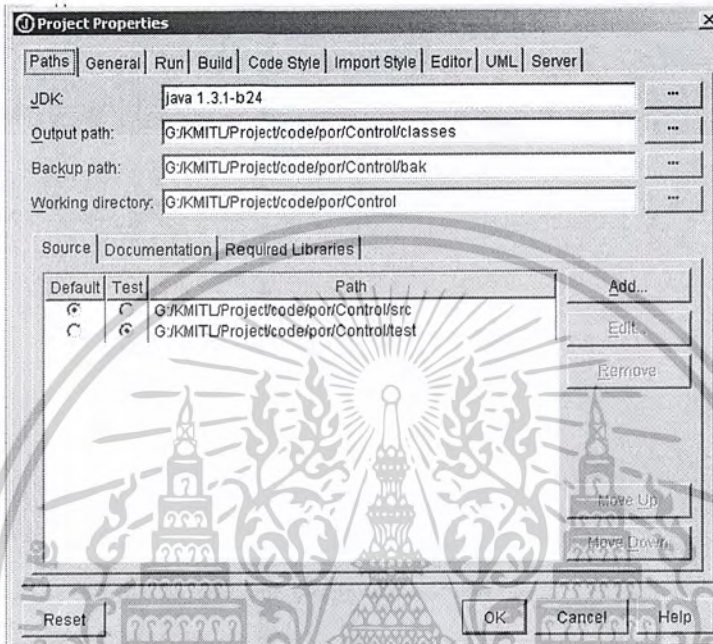
รูปที่ 3.2 -9 แสดงการเลือกอบชั้นต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง.3 การตั้งค่า Library ภายใน Jbuilder 7

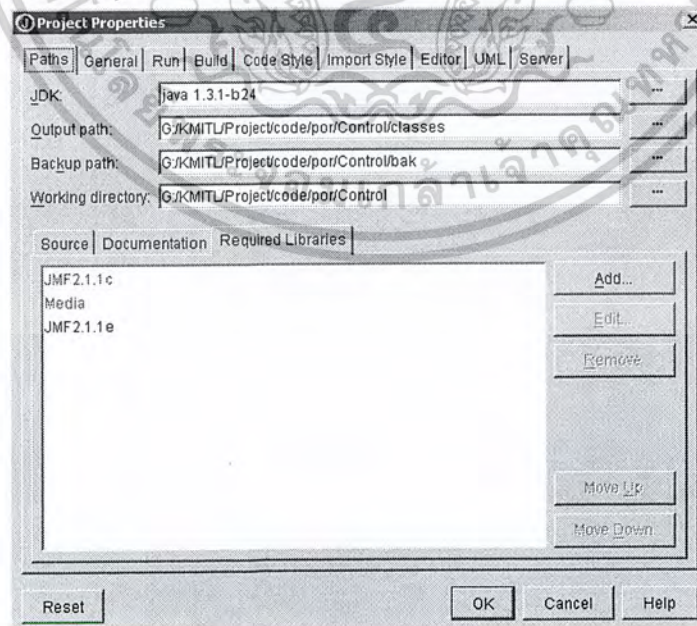
เมื่อทำการติดตั้ง Jbuilder 7.0 เรียบร้อยแล้วก่อนจะทำการรัน (run) โปรแกรมได้จะต้องทำการ add Library ของ JMF ก่อนดังนี้

1. เลือกที่ Project แล้วเลือก Project Properties
2. ต่อจากนั้นคลิกที่แถบ Required Libraries ดังรูป



รูปที่ ง.3-1 แสดงการ add Libraries

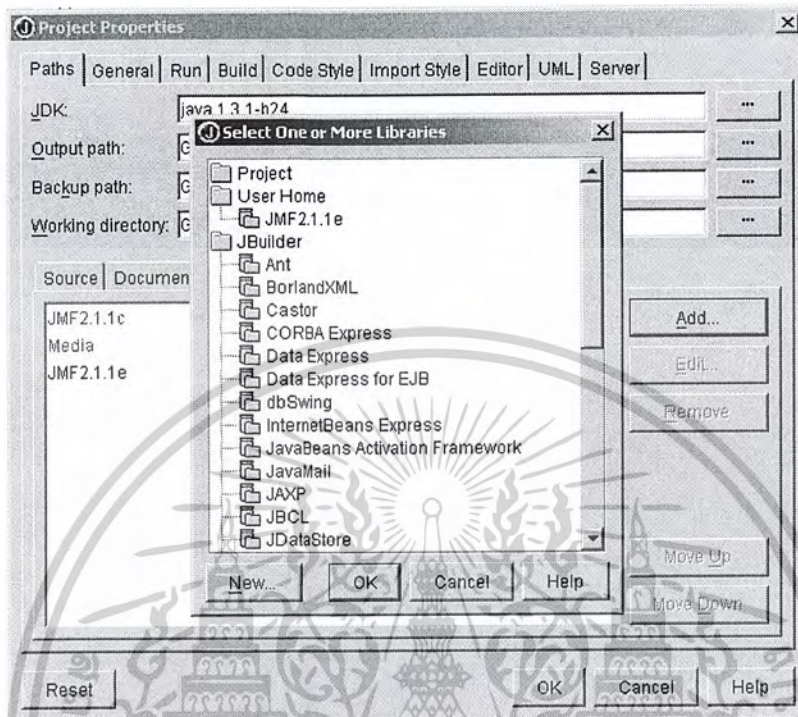
3. จะแสดงหน้าต่างดังรูป



รูปที่ ง.3-2 แสดงการ add Libraries

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คลิกที่ add เพื่อทำการเพิ่ม Library จะปรากฏหน้าต่างดังรูป



รูปที่ 3.3-3 แสดงการ add Libraries

5. ทำการเลือก Library JMF 2.1.1e ถ้าไม่พบดังรูปให้คลิกที่ new แล้วระบุตำแหน่งที่ลง JMF 2.1.1e

ง.4 การใช้โปรแกรม dat2mpg

dat2mpg เป็นโปรแกรมที่ใช้สำหรับแปลงไฟล์ที่มีนามสกุล dat ให้เป็นไฟล์ที่มีนามสกุล mpg ในการเรียกใช้โปรแกรมจะต้องพิมพ์คำสั่ง โดยมีรูปแบบดังต่อไปนี้

C:\>dat2mpg (ไฟล์.dat ที่ต้องการแปลง) (ไฟล์ที่ได้เมื่อแปลงแล้ว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง

```

C:\WINDOWS\System32\cmd.exe

Directory of C:\

05.MPG                1.wav
1111.MPG              123.MPG
2b2p.zip              3333.MPG
[43010325]            43010325.zip
4444.MPG              6666.MPG
888.MPG               ASPI.LOG
AUTOEXEC.BAT         bridge.jpg
CONFIG.SYS            [DeitelMessengerCaseStudy]
diagram.png          [Documents and Settings]
[FBBM]                [j2sdk1.4.1_02]
[JBuilder7]          [KaraokeTemp]
[Macromedia Flash v5.0 FINAL] [MVASPI]
[network]             [network3]
[POPDICT]             [Program Files]
[test]                test.jpg
testclient.txt        [tftp]
tftp.rar              [TFTPclient]
[TFTPserver]          [WINDOWS]
                    18 File(s)    348,677,111 bytes
                    18 Dir(s)    3,155,439,616 bytes free

C:\>dat2mpg c:\test.dat c:\output.mpg

```

รูปที่ ง.4-1 แสดงตัวอย่างการเรียกใช้โปรแกรม

```

C:\WINDOWS\System32\cmd.exe

AUTOEXEC.BAT         bridge.jpg
CONFIG.SYS            [DeitelMessengerCaseStudy]
diagram.png          [Documents and Settings]
[FBBM]                [j2sdk1.4.1_02]
[JBuilder7]          [KaraokeTemp]
[Macromedia Flash v5.0 FINAL] [MVASPI]
[network]             [network3]
[POPDICT]             [Program Files]
[test]                test.jpg
testclient.txt        [tftp]
tftp.rar              [TFTPclient]
[TFTPserver]          [WINDOWS]
                    18 File(s)    348,677,111 bytes
                    18 Dir(s)    3,155,439,616 bytes free

C:\>dat2mpg c:\test.dat c:\output.mpg
Source file:c:\test.dat      Target File:c:\output.mpg
Size of Source file:43006364 with 18285 Frames
Sync. char is found!
Truncate 18000 of 18285 frames, Time to stop: 0 min: 0 sec
*****
Total 18255 frames is read and left 0 byte
Total 60 Mpeg frames error!

C:\>

```

รูปที่ ง.4-2 แสดงตัวอย่างการเมื่อโปรแกรมได้ทำงานเสร็จสมบูรณ์แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Sun Microsystem, Inc. (1998-99) : “ Java™ Media Framework API Guide ”, Sun Microsystem, Inc., 1999 JMF 2.0 FCS, pp. 1-106
- [2] Ali Bahrami (1999) : “ OBJECT ORIENTED SYSTEMS DEVELOPMENT ”, Irwin/McGraw-Hill, 1999, pp. 89-118
- [3] Harvey and Paul Deitel (2002) : “ JAVA™ HOW TO PROGRAM FOURTH EDITION ”, Prentice-Hall, Inc, 2002
- [4] กิตติ ภัคดีวัฒนะกุล, ศิริวรรณ อัมพรคณัย (2544) : “ Object – Oriented ฉบับพื้นฐาน ”, KTP Comp & Consult, 2544
- [5] กิตติ ภัคดีวัฒนะกุล (2544), กิตติพงษ์ กลมกล่อม : “ UML วิเคราะห์และออกแบบระบบเชิงวัตถุ ”, KTP Comp & Consult, 2544, pp. 119-131
- [6] อาจารย์อภิเนตร อุนากุล (2543) : “ OBJECT-ORIENTED ANALYSIS AND DESIGN ”, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543
- [7] ดร.สุสิทธิ์ จรัสกุลชัย (2543) : “ การเขียนโปรแกรมเชิงวัตถุภาษาจาวา ”, สำนักพิมพ์มหาวิทยาลัยเกษตรศาสตร์, 2543
- [8] ดร. วีระศักดิ์ ชิงถาวร (2545) : “ JAVA Programming Volume I ”, ซีเอ็ดยูเคชั่น, 2545
- [9] ดร. วีระศักดิ์ ชิงถาวร (2543) : “ Fundamental of JAVA Programming Volume 2 ”, ซีเอ็ดยูเคชั่น, 2543, pp. 117-307
- [10] กิตติ ภัคดีวัฒนะกุล (2546) : “ คัมภีร์ JAVA เล่ม 1 ”, KTP Comp & Consult, 2546
- [11] ณัฐคณัย สุขรัตน์ (2546) : “ คัมภีร์ JAVA เล่ม 2 ”, KTP Comp & Consult, 2546, pp. 25-314
- [12] ทวีศักดิ์ กาญจนสุวรรณ (2546) : “ Multimedia ฉบับพื้นฐาน ”, KTP Comp & Consult, 2546, pp. 179-229

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้