

การออกแบบรูปแบบคู่มือการใช้งานเฟรมเวิร์ก  
DESIGN OF FRAMEWORK DOCUMENTATION FORMAT

สุธิดา สัจจรัตนันท์  
SUTHIDA SAJJARITANANT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547

ISBN 974-15-1269-4

การออกแบบรูปแบบคู่มือการใช้งานเฟรมเวิร์ก

DESIGN OF FRAMEWORK DOCUMENTATION FORMAT



สุธิดา สัจจรัตนันท์  
SUTHIDA SAJJARITANANT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาวิทยาการคอมพิวเตอร์  
บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547

ISBN 974-15-1269-4

เลขหมู่.....

เลขทะเบียน 56715

วัน,เดือน,ปี 4 ก.ค. 2548



# DESIGN OF FRAMEWORK DOCUMENTATION FORMAT

SUTHIDA SAJJARITANANT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF SCIENCE IN COMPUTER SCIENCE  
SCHOOL OF GRADUATE STUDIES  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

ISBN 974-15-1269-4

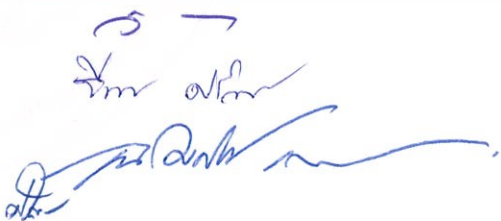
COPYRIGHT 2004

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์      การออกแบบรูปแบบคู่มือการใช้งานเฟรมเวิร์ก  
DESIGN OF FRAMEWORK DOCUMENTATION FORMAT  
ชื่อนักศึกษา            นางสาวสุธิดา สัจจิตานนท์  
รหัสประจำตัว            45064601  
ปริญญา                    วิทยาศาสตรมหาบัณฑิต  
สาขาวิชา                วิทยาการคอมพิวเตอร์  
อาจารย์ผู้ควบคุมวิทยานิพนธ์    ผศ.ดร.ศรัณย์      อินทโกสุม

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.วีระ	บุญจริง	
ผศ.ดร.จิรพร	ศรีสวัสดิ์	
ดร.เฉลิมศักดิ์	เลิศวงศ์เสถียร	
ผศ.ดร.ศรัณย์	อินทโกสุม	

วัน/เดือนปี ที่สอบ 28 ตุลาคม 2547 เวลา 13.00-15.00 น.

สถานที่สอบ ณ อาคารจุฬารณวลัยลักษณ์ 1 ห้อง 219

บัณฑิตวิทยาลัยรับรองแล้ว  
  
(ผศ.ดร.จารุวัตร เจริญสุข)  
คณบดีบัณฑิตวิทยาลัย

วันที่.....18.....เดือน.....พฤษภาคม.....พ.ศ.....2547.....



Thesis Title	Design of Framework Documentation Format
Student	Ms. Suthida Sajjaritanant
Student ID.	45064601
Degree	Master of Science
Program	Computer Science
Year	2004
Thesis Advisor	Asst. Prof. Dr. Sarun Intakosum

### ABSTRACT

The purpose of this research is to design a framework documentation format for application developers to shorten the time to study a new framework, and has a good human computer interaction. Designing of this format is based on two major issues contents, and user interfaces. Various researches on framework documentation and existing framework documents are studied and carefully analyzed to propose what topics should be included and arranged in the contents part. The psychological issues related to human learning factors, user interface design, and web browser user interface design, are studied and properly applied for user interfaces part.

In the experiment there were two test groups. The experimental group used the documentation written using the proposed format whereas the control group used the existing framework documentation. The results show that the experimental group's average development time and satisfaction ratings were significantly higher than the control group's. The former indicates the reduction on time to learn a new framework, the lesser points out that the proposed format can be considered, in some degree, standard for writing framework document that satisfies its users.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ.ดร.ศรัณย์ อินทโกสุม ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบคุณ รศ.ดร.วีระ บุญจริง ผศ.ดร.จีรพร ศรีสวัสดิ์ และดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร คณะกรรมการสอบหัวข้อและโครงร่างวิทยานิพนธ์ที่กรุณาให้คำแนะนำตลอดจนข้อชี้แนะ จนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ ที่คอยตักเตือนและให้คำแนะนำในการทำวิทยานิพนธ์

ขอขอบคุณมูลนิธิเชลล์ 100 ปี ที่ให้ทุนสนับสนุนการทำวิจัยให้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณเพื่อน ๆ นักศึกษาทุกคนที่ช่วยเหลือให้คำแนะนำต่าง ๆ และมีส่วนร่วมในการทดสอบประสิทธิภาพของคู่มือการใช้งานเฟรมเวิร์กจนสำเร็จลุล่วงไปได้ด้วยดี

สำหรับคุณงามความดีอันใดที่เกิดขึ้นจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูบาอาจารย์ที่เคารพทุกท่านที่ได้ถ่ายทอดวิชาความรู้ และประสบการณ์ที่ดีให้แก่ข้าพเจ้า

สุธิดา สัจจิตานันท์

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 สมมติฐานของการศึกษา.....	2
1.4 ขอบเขตการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 เฟรมเวิร์กและงานวิจัยที่เกี่ยวข้องกับการเขียนคู่มือการใช้งานเฟรมเวิร์ก.....	4
2.1 เฟรมเวิร์กเพื่อพัฒนาโปรแกรมประยุกต์.....	4
2.2 ประเภทของผู้เกี่ยวข้องกับเฟรมเวิร์ก.....	4
2.2.1 กลุ่มผู้พัฒนาโปรแกรมประยุกต์.....	5
2.2.2 กลุ่มผู้บำรุงรักษาเฟรมเวิร์ก.....	5
2.2.3 กลุ่มผู้พัฒนาเฟรมเวิร์ก.....	5
2.2.4 กลุ่มผู้ตรวจสอบ.....	5
2.3 งานวิจัยที่เกี่ยวข้องกับการจัดทำคู่มือเฟรมเวิร์ก.....	6
2.3.1 ตัวอย่างข้อสัปดาห์ของโปรแกรมประยุกต์.....	6
2.3.2 เรซิพีและคูกนึ่ง.....	6
2.3.3 รูปแบบการออกแบบ.....	7
2.3.4 คู่มืออ้างอิง.....	7
2.3.5 ภาพรวมของเฟรมเวิร์ก.....	8
2.3.6 อินเทอร์เฟซคอนแทรก.....	8
2.3.7 การเขียนคู่มือในมุมมองด้านไดนามิก.....	8

## สารบัญ (ต่อ)

	หน้า
2.3.8 การเขียนคู่มือในมุมมองของริยูสเคส.....	9
2.3.9 แนวการเขียนคู่มือเพื่อผู้พัฒนาโปรแกรมประยุกต์.....	12
2.3.10 คู่มือการใช้งานเฟรมเวิร์กตามทฤษฎีการสอนแบบมินิมอลลิสต์.....	12
2.3.11 ปัญหาของการใช้งานคู่มือเฟรมเวิร์กในปัจจุบัน.....	13
<b>บทที่ 3 จิตวิทยาและการออกแบบส่วนติดต่อกับผู้ใช้.....</b>	<b>14</b>
3.1 ปัจจัยที่เกี่ยวข้องกับมนุษย์ในการออกแบบส่วนติดต่อกับผู้ใช้.....	14
3.1.1 โครงสร้างความคิดและความจำ.....	14
3.1.2 รูปแบบและระดับของความรู้.....	16
3.1.3 ทักษะและการเรียนรู้.....	16
3.2 หลักการออกแบบคู่มือตามความต้องการของมนุษย์ในเชิงจิตวิทยา.....	17
3.2.1 มีความเรียบง่าย.....	17
3.2.2 ลดภาระความทรงจำระยะสั้น.....	17
3.2.3 สร้างเจตคติที่ดีให้กับผู้ใช้.....	17
3.2.4 มีความคงที่ในการนำเสนอ.....	18
3.2.5 มีทางเลือกในการเข้าถึงข้อมูล.....	18
3.3 หลักการออกแบบส่วนติดต่อกับผู้ใช้.....	18
3.3.1 หลักการนำเสนอข้อมูลบนหน้าจอ.....	18
3.3.2 การกระตุ้นความสนใจของผู้ใช้.....	19
3.3.3 หลักการให้สีในการนำเสนอ.....	20
3.4 การออกแบบส่วนติดต่อกับผู้ใช้บนเว็บเบราว์เซอร์.....	21
3.4.1 การออกแบบให้หน้าเว็บมีความยาวที่เหมาะสม.....	21
3.4.2 ส่วนประกอบของเว็บเพจ.....	21
3.4.3 การเลือกใช้สีสำหรับเว็บ.....	22
3.4.4 การจัดรูปแบบตัวอักษรสำหรับเว็บ.....	22
<b>บทที่ 4 รูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์ก.....</b>	<b>25</b>
4.1 ส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก.....	25

## สารบัญ (ต่อ)

	หน้า
4.1.1 ภาพรวม.....	27
4.1.2 คำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์.....	29
4.1.3 สรุปปรายการคำสั่ง.....	35
4.1.4 อ้างอิง.....	42
4.2 การออกแบบส่วนติดต่อกับผู้ใช้งานสำหรับคู่มือการใช้งานเฟรมเวิร์กบนเว็บ- เบราว์เซอร์.....	43
4.2.1 การออกแบบหน้าเอกสารคู่มือเฟรมเวิร์กให้มีความยาวที่เหมาะสม....	44
4.2.2 การจัดวางส่วนประกอบของคู่มือเฟรมเวิร์กในแต่ละหน้า.....	44
4.2.3 การเลือกใช้สีสำหรับคู่มือการใช้งานเฟรมเวิร์ก.....	45
4.2.4 การจัดรูปแบบตัวอักษรสำหรับคู่มือการใช้งานเฟรมเวิร์ก.....	45
4.3 สรุปรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์ก.....	48
4.3.1 ส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก.....	48
4.3.2 การออกแบบส่วนติดต่อกับผู้ใช้สำหรับคู่มือการใช้งานเฟรมเวิร์กบน เว็บเบราว์เซอร์.....	49
บทที่ 5 วิธีดำเนินการวิจัย.....	51
5.1 ประชากรและกลุ่มตัวอย่าง.....	51
5.2 เครื่องมือที่ใช้ในการวิจัย.....	52
5.3 การดำเนินการวิจัยและเก็บรวบรวมข้อมูล.....	54
5.4 การวิเคราะห์ข้อมูล.....	55
บทที่ 6 ผลการวิเคราะห์ข้อมูล.....	59
6.1 การวิเคราะห์เปรียบเทียบระยะเวลาการพัฒนาโปรแกรมประยุกต์ระหว่าง นักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัท ชั้นไมโครซีเอสเต็มส์.....	59
6.2 การวิเคราะห์เปรียบเทียบความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวิร์ก ระหว่างคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัท ชั้นไมโครซีเอสเต็มส์.....	61

## สารบัญ (ต่อ)

	หน้า
บทที่ 7 การอภิปรายผลการทดลอง.....	66
7.1 ระยะเวลาในการพัฒนาโปรแกรมประยุกต์จากการใช้งานคู่มือเฟรมเวิร์ก.....	66
7.2 ความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวิร์ก.....	66
7.2.1 ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก.....	67
7.2.2 ด้านส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก.....	68
บทที่ 8 สรุปผลการวิจัยและข้อเสนอแนะ.....	69
8.1 สรุปผลการวิจัย.....	69
8.2 ข้อเสนอแนะ.....	69
เอกสารอ้างอิง.....	71
ภาคผนวก ก. แบบสอบถามวัดระยะเวลาการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งาน คู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์.....	74
ภาคผนวก ข. แบบสอบถามความพอใจในรูปแบบคู่มือการใช้งานเฟรมเวิร์ก.....	76
ภาคผนวก ค. คู่มือลงรหัสสำหรับวิเคราะห์ข้อมูลด้วยเครื่องคอมพิวเตอร์.....	79
ภาคผนวก ง. ตัวอย่างคู่มือการใช้งานเฟรมเวิร์กที่ได้จากงานวิจัย (จาวาคอเล็กชันเฟรมเวิร์ก).....	82
ภาคผนวก จ. ตัวอย่างคู่มือการใช้งานเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์ (จาวาคอเล็กชันเฟรมเวิร์ก).....	91
ประวัติผู้เขียน.....	98

# สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงคู่มือที่มีอยู่ในปัจจุบันที่สามารถอธิบายวิธีสเคลสในหมวดการเลือก การประกอบ และการขยาย.....	10
5.1 ตารางแจงแจงกลุ่มตัวอย่าง.....	51
6.1 แสดงค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของระยะเวลา (ชั่วโมง) ที่ใช้ในการพัฒนาโปรแกรมของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยและคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์.....	59
6.2 ผลการเปรียบเทียบความแตกต่างของค่าเฉลี่ยระยะเวลา (ชั่วโมง) ที่ใช้ในการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์.....	60
6.3 แสดงค่ามัธยฐานคะแนนความพอใจในแต่ละข้อจากการใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยและคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์.....	62
6.4 ผลการเปรียบเทียบความแตกต่างของค่ามัธยฐานคะแนนความพอใจโดยรวมในการใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์.....	63
6.5 ผลการเปรียบเทียบความแตกต่างของค่ามัธยฐานคะแนนความพอใจในแต่ละหัวข้อในการใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์.....	64

# สารบัญรูป

รูปที่	หน้า
2.1 แสดงความสัมพันธ์ระหว่างงานพื้นฐานทางด้านวิศวกรรมเฟรมเวอร์คกับข้อมูล การจัดทำคู่มือพื้นฐานของเฟรมเวอร์ค.....	11
3.1 แสดงโครงสร้างระบบความจำของมนุษย์.....	15
4.1 แสดงส่วนประกอบของเนื้อหาหลักในคู่มือเฟรมเวอร์ค.....	26
4.2 แสดงส่วนประกอบของเนื้อหาในแต่ละเรขีพี.....	27
4.3 แสดงตัวอย่างการเขียนภาพรวมของลิสต์อินเทอร์เฟซ.....	28
4.4 แสดงตัวอย่างการเขียนคำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์ของ ลิสต์อินเทอร์เฟซ.....	30
4.5 แสดงตัวอย่างการเขียนคำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์ของ อาเรย์ลิสต์.....	33
4.6 แสดงตัวอย่างการสรุปรายการคำสั่งอย่างย่อของลิสต์อินเทอร์เฟซ.....	36
4.7 แสดงตัวอย่างการสรุปรายการคำสั่งอย่างละเอียดของลิสต์อินเทอร์เฟซ.....	39
4.8 แสดงตัวอย่างส่วนอ้างอิงของลิสต์อินเทอร์เฟซ.....	43
ง1 ตัวอย่างหน้าเมนูหลัก.....	84
ง2 ตัวอย่างหน้าภาพรวม.....	85
ง3 ตัวอย่างหน้าตัวอย่างโปรแกรมประยุกต์.....	86
ง4 ตัวอย่างหน้าคำอธิบายการใช้งานส่วนประกอบภายในเฟรมเวอร์ค.....	87
ง5 ตัวอย่างเนื้อหาในแต่ละเรขีพี.....	88
ง6 คำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์ในอาเรย์ลิสต์.....	90
ง7 แสดงตัวอย่างการสรุปรายการคำสั่งอย่างย่อของอาเรย์ลิสต์.....	91
ง8 แสดงตัวอย่างการสรุปรายการคำสั่งอย่างละเอียดของอาเรย์ลิสต์.....	92
จ1 หน้าเมนูหลัก.....	93
จ2 การเขียนส่วนท้ายของคู่มือ.....	93
จ3 หน้าภาพรวม.....	94
จ4 หน้าคำอธิบายการใช้งานส่วนประกอบภายในเฟรมเวอร์ค.....	95
จ5 เนื้อหาในแต่ละเรขีพี.....	96
จ6 แสดงการสรุปรายการคำสั่งอย่างย่อของอาเรย์ลิสต์.....	97
จ7 แสดงการสรุปรายการคำสั่งอย่างละเอียดของอาเรย์ลิสต์.....	98

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

การใช้งานเฟรมเวิร์กเพื่อการพัฒนาโปรแกรมประยุกต์ (application framework) เป็นการนำหลักการของการนำกลับมาใช้ใหม่มาใช้ เพื่อช่วยให้การพัฒนาโปรแกรมประยุกต์ทำได้อย่างรวดเร็ว ถูกต้องและมีประสิทธิภาพมากขึ้น แต่ผลดังกล่าวจะได้รับประโยชน์สูงสุดก็ต่อเมื่อผู้ใช้งานเฟรมเวิร์กมีความเข้าใจวิธีการและรายละเอียดในการใช้งานเฟรมเวิร์กเป็นอย่างดี และสามารถนำมาสร้างเป็นโปรแกรมประยุกต์ตามความต้องการได้ วิธีการหนึ่งที่จะทำให้เข้าใจเฟรมเวิร์กได้ คือ การศึกษาจากคู่มือการใช้งานเฟรมเวิร์ก (framework document) ดังนั้นคู่มือการใช้งานเฟรมเวิร์กจึงเป็นสิ่งสำคัญมากอย่างหนึ่งในการพัฒนาโปรแกรมประยุกต์จากเฟรมเวิร์กให้ประสบผลสำเร็จ อย่างไรก็ตามคู่มือการใช้งานเฟรมเวิร์กที่ใช้อยู่ในปัจจุบันส่วนใหญ่จะมีลักษณะเป็นรายการคำสั่งที่เฟรมเวิร์กเตรียมไว้ให้ และอธิบายส่วนที่ต้องการให้ผู้พัฒนาโปรแกรมเขียนเพิ่มเติมให้สมบูรณ์ โดยที่ไม่ได้ให้ความสำคัญกับรูปแบบของการเขียนมากนัก เช่น การเรียงลำดับหัวข้อการเขียนคู่มือ สีของตัวอักษร ขนาดตัวอักษรที่ใช้ในการเขียนคู่มือ เป็นต้น ซึ่งคู่มือในลักษณะดังกล่าวอาจทำให้ผู้ใช้งานโดยเฉพาะผู้พัฒนาโปรแกรมประยุกต์ที่ไม่มีประสบการณ์ ไม่สามารถทำความเข้าใจและใช้งานเฟรมเวิร์กอย่างเต็มประสิทธิภาพในระยะเวลาอันรวดเร็ว นอกจากนี้ผู้พัฒนาเฟรมเวิร์กยังนิยมเขียนคู่มือโดยใช้รูปแบบของตัวเอง เป็นผลให้ผู้พัฒนาโปรแกรมประยุกต์ต้องเรียนรู้และทำความเข้าใจกับรูปแบบของคู่มือใหม่ทุกครั้งก่อนการใช้งานเฟรมเวิร์ก

จากงานวิจัยที่เกี่ยวข้องกับการจัดทำคู่มือการใช้งานเฟรมเวิร์ก ยังไม่มีงานวิจัยใดที่มีรูปแบบการเขียนคู่มือเฟรมเวิร์กที่ชัดเจน ส่วนใหญ่จะให้หลักการของการจัดทำคู่มือเฟรมเวิร์กโดยสังเขป หรือนำเสนอวิธีการต่าง ๆ ที่นำมาใช้ในการจัดทำรูปแบบคู่มือเฟรมเวิร์กได้ โดยไม่ได้วิเคราะห์ว่าวิธีการใดเหมาะสมที่จะนำมาใช้พัฒนาคู่มือเฟรมเวิร์กสำหรับผู้พัฒนาโปรแกรมประยุกต์ และยิ่งไปกว่านั้นยังไม่มีงานวิจัยใดที่สนใจวิเคราะห์ส่วนติดต่อกับผู้เข้ามาเป็นประเด็นหนึ่งในการจัดทำรูปแบบคู่มือการใช้งานเฟรมเวิร์ก จากปัญหาทั้งหมดที่กล่าวมานี้จึงมีแนวคิดที่จะทำวิจัยเพื่อกำหนดรูปแบบและวิธีเขียนคู่มือการใช้งานเฟรมเวิร์กสำหรับผู้พัฒนาโปรแกรมประยุกต์ โดยรูปแบบของคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยนี้ จะทดสอบประสิทธิภาพโดยสร้างเป็นคู่มือเฟรมเวิร์กที่นำเสนอบนเว็บเบราว์เซอร์ ซึ่งเป็นโปรแกรมที่ผู้ใช้งานสามารถใช้งานเอกสารที่มีลักษณะเชื่อมโยงได้เป็นอย่างดี ทั้งยังเป็นโปรแกรมที่ติดตั้งอยู่กับเครื่องคอมพิวเตอร์ส่วนใหญ่

งานวิจัยนี้จะออกแบบรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์ก โดยอ้างอิงจากงานวิจัยที่เกี่ยวข้องกับการจัดทำคู่มือเฟรมเวิร์กที่มีมา ทฤษฎีการออกแบบส่วนติดต่อกับผู้ใช้และจิตวิทยาที่เกี่ยวข้องกับการเรียนรู้ของมนุษย์ในการออกแบบส่วนติดต่อกับผู้ใช้ มาประยุกต์ใช้ในการออกแบบรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์ก ซึ่งคู่มือเฟรมเวิร์กที่จัดทำขึ้นจากรูปแบบที่ได้จากงานวิจัยนี้จะช่วยให้ผู้พัฒนาโปรแกรมประยุกต์สามารถพัฒนาโปรแกรมประยุกต์โดยใช้เฟรมเวิร์กได้อย่างสะดวก รวดเร็วและมีประสิทธิภาพยิ่งขึ้น นอกจากนี้ยังสามารถใช้เป็นมาตรฐานในการจัดทำคู่มือการใช้งานเฟรมเวิร์ก ซึ่งจะช่วยลดเวลาในการจัดทำคู่มือและเวลาในการศึกษาการใช้งานเฟรมเวิร์กอีกทางหนึ่งด้วย

## 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อออกแบบรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์กที่เป็นมาตรฐานสำหรับผู้พัฒนาโปรแกรมประยุกต์ เพื่อช่วยลดระยะเวลาในการศึกษาการใช้งานเฟรมเวิร์ก และมีส่วนปฏิสัมพันธ์ที่ดีกับผู้ใช้

## 1.3 สมมติฐานของการศึกษา

รูปแบบการจัดทำคู่มือการใช้งานเฟรมเวิร์กที่ได้จากการวิจัยนี้ จะช่วยให้ผู้จัดทำคู่มือเฟรมเวิร์กมีมาตรฐานในการเขียนคู่มือการใช้งานเฟรมเวิร์กที่สามารถลดระยะเวลาในการพัฒนาโปรแกรมประยุกต์ และสร้างความพอใจให้กับผู้ใช้งานในกลุ่มผู้พัฒนาโปรแกรมประยุกต์มากกว่าคู่มือการใช้งานเฟรมเวิร์กที่มีอยู่ได้

## 1.4 ขอบเขตการวิจัย

วิทยานิพนธ์นี้มีขอบเขตของการวิจัย ดังต่อไปนี้

1.4.1 กลุ่มผู้ใช้งานคู่มือการใช้งานเฟรมเวิร์ก คือ กลุ่มผู้พัฒนาโปรแกรมประยุกต์ที่จะนำคู่มือนี้ไปใช้ในการพัฒนาโปรแกรมประยุกต์ โดยไม่รวมถึงกลุ่มผู้ออกแบบและผู้พัฒนาเฟรมเวิร์กที่ต้องการรายละเอียดเกี่ยวกับโครงสร้างในการออกแบบเฟรมเวิร์กอย่างลึกซึ้ง

1.4.2 ในการทดสอบประสิทธิภาพการใช้งานคู่มือเฟรมเวิร์ก จะเลือกจาวาคอเล็คชันเฟรมเวิร์ก (Java Collection Framework) มาเป็นเฟรมเวิร์กตัวอย่าง ในการวิจัยจะเลือกเพียงบางส่วนมาจัดทำเป็นคู่มือตามรูปแบบที่ทำการออกแบบไว้เท่านั้น

1.4.3 กลุ่มผู้ทดลองใช้งานคู่มือเฟรมเวิร์ก คือ นักศึกษาปริญญาตรีและโทสาขาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่สามารถพัฒนาโปรแกรมด้วยภาษาจาวาได้เป็นอย่างดี จำนวน 27 คน

1.4.4 ในการวิจัยจะไม่ครอบคลุมถึงการเพิ่มเติมหรือเปลี่ยนแปลงคู่มืออันเนื่องมาจากการเปลี่ยนแปลงที่เกิดขึ้นกับเฟรมเวิร์กในอนาคต

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ทำให้ผู้เขียนคู่มือมีแนวทางและรูปแบบการเขียนคู่มือที่เป็นมาตรฐาน

1.5.2 ทำให้ผู้ใช้งานเฟรมเวิร์กเข้าใจขั้นตอนและวิธีการใช้งานเฟรมเวิร์กในการพัฒนาโปรแกรมประยุกต์ได้ง่ายและเร็วขึ้น

1.5.3 ทำให้ผู้พัฒนาโปรแกรมประยุกต์ใช้งานเฟรมเวิร์กกันมากขึ้น เนื่องจากผู้พัฒนาโปรแกรมประยุกต์สามารถทำความเข้าใจและใช้งานคู่มือเฟรมเวิร์กในการพัฒนาเฟรมเวิร์กได้อย่างรวดเร็ว

## บทที่ 2

# เฟรมเวิร์กและงานวิจัยที่เกี่ยวข้องกับการเขียน คู่มือการใช้งานเฟรมเวิร์ก

ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องเพื่อใช้ในการจัดทำคู่มือการใช้งานเฟรมเวิร์ก โดยจะอธิบายถึงความหมายของเฟรมเวิร์ก ประเภทของผู้เกี่ยวข้องกับเฟรมเวิร์ก และงานวิจัยที่เกี่ยวข้องกับการเขียนคู่มือการใช้งานเฟรมเวิร์กสำหรับผู้พัฒนาโปรแกรมประยุกต์

### 2.1 เฟรมเวิร์กเพื่อพัฒนาโปรแกรมประยุกต์ (Application Framework)

เฟรมเวิร์ก [3] [8] [10] เป็นโปรแกรมกึ่งสำเร็จรูปที่นำเทคโนโลยีของการนำกลับมาใช้ใหม่มาช่วยในการออกแบบและจัดทำซอฟต์แวร์เพื่อเพิ่มคุณภาพซอฟต์แวร์ จุดประสงค์หลักของการใช้งานเฟรมเวิร์ก คือ ช่วยลดระยะเวลาและช่วยประหยัดงบประมาณในการผลิตซอฟต์แวร์ จอห์นสันและฟุตเต้ [13] ได้ให้คำนิยามเฟรมเวิร์กไว้ว่า “เฟรมเวิร์กเป็นโปรแกรมประยุกต์ที่อาศัยหลักการของการนำกลับมาใช้ใหม่ โดยผู้ใช้สามารถเพิ่มเติม แก้ไขบางส่วนเพื่อให้กลายเป็นโปรแกรมประยุกต์ที่สมบูรณ์ตามความต้องการ” เฟรมเวิร์กจะประกอบด้วยคลาสที่มีความสัมพันธ์กัน ทั้งคลาสที่เป็นนามธรรมและคลาสรูปธรรม โดยคลาสรูปธรรมจะเรียกอีกอย่างหนึ่งว่า โฟรเซนสปอต (frozen spots) [15] เป็นส่วนที่มีการเขียนโค้ดไว้เรียบร้อยแล้วพร้อมจะนำไปใช้ได้ และคลาสนามธรรมหรือเรียกอีกอย่างว่า ฮอตสปอต (hot spot) [16] เป็นส่วนที่ผู้พัฒนาโปรแกรมประยุกต์สามารถปรับแต่งให้เป็นไปตามความต้องการของโปรแกรมประยุกต์ที่กำลังพัฒนาอยู่ ลักษณะที่สำคัญอีกอย่างหนึ่งของเฟรมเวิร์ก คือ มีการควบคุมการใช้งานแตกต่างจากการเขียนโปรแกรมแบบดั้งเดิม กล่าวคือ ผู้พัฒนาโปรแกรมแบบดั้งเดิมจะต้องเป็นผู้เขียนโครงสร้างและแนวทางในการประมวลผลเองทั้งหมด ส่วนการใช้เฟรมเวิร์กในการพัฒนาโปรแกรมประยุกต์นั้น ผู้พัฒนาโปรแกรมสามารถเขียนเพิ่มเติมส่วนที่หายไปจากเฟรมเวิร์กเท่านั้น เพื่อให้กลายเป็นโปรแกรมประยุกต์ที่สมบูรณ์ตามความต้องการ

### 2.2 ประเภทของผู้เกี่ยวข้องกับเฟรมเวิร์ก

ในการเขียนคู่มือเฟรมเวิร์กให้มีประสิทธิภาพนั้น สิ่งสำคัญอย่างหนึ่งที่ต้องนำมาพิจารณา คือ ความต้องการรายละเอียดในคู่มือเฟรมเวิร์กของกลุ่มผู้เกี่ยวข้องกับเฟรมเวิร์กที่ต่างกัน โมฮัมเมต อี ฟายาดและคณะ [19] ได้แบ่งประเภทของผู้เกี่ยวข้องกับเฟรมเวิร์กออกเป็น 4 กลุ่มดังต่อไปนี้

### 2.2.1 กลุ่มผู้พัฒนาโปรแกรมประยุกต์ (Application Developer)

เป็นผู้ที่นำเฟรมเวิร์กมาพัฒนาเพื่อให้ได้เป็นโปรแกรมประยุกต์ตามความต้องการ โดยในการพัฒนาโปรแกรมประยุกต์ให้เป็นไปตามความต้องการนั้นผู้พัฒนาโปรแกรมประยุกต์จำเป็นต้องมีความรู้ในเรื่องโดเมนของปัญหาและวิธีการใช้งานเฟรมเวิร์ก ดังนั้นคู่มือเฟรมเวิร์กที่ผู้พัฒนาโปรแกรมประยุกต์ต้องการ ควรเน้นถึงขั้นตอนและวิธีการในการใช้งานเฟรมเวิร์กมากกว่ารายละเอียดเกี่ยวกับการออกแบบ และสถาปัตยกรรมของเฟรมเวิร์ก

### 2.2.2 กลุ่มผู้บำรุงรักษาเฟรมเวิร์ก (Framework Maintainer)

เป็นผู้ที่มีหน้าที่เกี่ยวกับการบำรุงรักษาและพัฒนาเฟรมเวิร์ก ซึ่งจำเป็นต้องมีความรู้ความเข้าใจเกี่ยวกับเฟรมเวิร์กในหลาย ๆ ด้าน เช่น รายละเอียดในการออกแบบเฟรมเวิร์ก โครงสร้างความสัมพันธ์ของเฟรมเวิร์กและโดเมนของโปรแกรมประยุกต์ เป็นต้น ดังนั้นคู่มือที่ผู้บำรุงรักษาเฟรมเวิร์กต้องการควรบอกถึงรายละเอียดในการออกแบบ สถาปัตยกรรมโดยรวมของเฟรมเวิร์ก วัตถุประสงค์ของการใช้งาน นอกจากนี้คู่มือที่ใช้อย่างดียังต้องมีความยืดหยุ่นในการเปลี่ยนแปลงเพิ่มเติมสอทสปอทที่เกิดขึ้นจากการแก้ไขหรือปรับปรุงเฟรมเวิร์กได้

### 2.2.3 กลุ่มผู้พัฒนาเฟรมเวิร์ก (Framework Developer)

ในการพัฒนาเฟรมเวิร์กผู้พัฒนาเฟรมเวิร์กจะหาแนวคิดจากเฟรมเวิร์กที่มีอยู่ ซึ่งอาจจะเป็นเฟรมเวิร์กในโดเมนอื่นก็ได้ เพื่อนำมาพัฒนาเป็นเฟรมเวิร์กตามความต้องการ โดยผู้พัฒนาเฟรมเวิร์กจะศึกษารูปแบบการออกแบบ (design pattern) เพื่อสร้างเป็นสอทสปอทที่มีความยืดหยุ่น คู่มือที่ผู้พัฒนาเฟรมเวิร์กต้องการจะเป็นคู่มือที่มีรายละเอียดพื้นฐาน เช่น โดเมนของปัญหาที่สามารถนำเฟรมเวิร์กไปประยุกต์ใช้ได้ วัตถุประสงค์การใช้งานของเฟรมเวิร์ก เป็นต้น จนถึงข้อมูลที่มีรายละเอียดในการออกแบบ คล้ายกับคู่มือที่กลุ่มผู้บำรุงรักษาเฟรมเวิร์กต้องการ

### 2.2.4 กลุ่มผู้ตรวจสอบ (Verifier)

เป็นกลุ่มบุคคลที่ทำหน้าที่ตรวจสอบคุณสมบัติของระบบเพื่อให้ถูกต้องตรงตามความต้องการ การตรวจสอบทำได้โดยการนำข้อกำหนดรายละเอียด (specification) ของเฟรมเวิร์ก เช่น นำส่วนของสอทสปอทและโพสเซนสอทมาตรวจสอบความถูกต้อง (verification) คู่มือที่บุคคลกลุ่มนี้ต้องการเป็นคู่มือที่เกี่ยวข้องกับการออกแบบเฟรมเวิร์ก ข้อกำหนดรายละเอียด และวิธีการใช้งานเฟรมเวิร์ก

จากกลุ่มผู้ที่เกี่ยวข้องกับเฟรมเวิร์กที่กล่าวมาข้างต้นจะเห็นว่าแต่ละกลุ่มต้องการรายละเอียดของคู่มือเฟรมเวิร์กที่ต่างกัน จึงเป็นไปได้ยากที่จะจัดทำคู่มือเฟรมเวิร์กที่สามารถสนอง

ความต้องการของทุกกลุ่ม ดังนั้นวิทยานิพนธ์ฉบับนี้จึงเน้นการจัดทำคู่มือการใช้งานเฟรมเวิร์กสำหรับผู้พัฒนาโปรแกรมประยุกต์ เนื่องจากกลุ่มผู้พัฒนาโปรแกรมประยุกต์เป็นกลุ่มผู้ใช้งานส่วนใหญ่ที่มีโอกาสได้ใช้งานคู่มือเฟรมเวิร์กบ่อยที่สุด และถ้าผู้พัฒนาโปรแกรมประยุกต์ในกลุ่มนี้ขาดคู่มือการใช้งานเฟรมเวิร์กที่มีประสิทธิภาพแล้ว การพัฒนาโปรแกรมประยุกต์อาจต้องใช้เวลาเนิ่นนานขึ้นและมีประสิทธิภาพลดลงด้วย

## 2.3 งานวิจัยที่เกี่ยวข้องกับการจัดทำคู่มือเฟรมเวิร์ก

โมฮัมเมด อี ฟายาดและคณะ [19] ได้แบ่งคู่มือเฟรมเวิร์กสำหรับผู้พัฒนาโปรแกรมประยุกต์ออกเป็นประเภทหลักๆ ดังนี้

### 2.3.1 ตัวอย่างซอสโค้ดของโปรแกรมประยุกต์

เป็นคู่มือที่เป็นตัวอย่างซอสโค้ดของโปรแกรมประยุกต์ที่สร้างขึ้นจากเฟรมเวิร์ก รูปแบบที่ใช้ในการจัดทำคู่มือประเภทนี้ไม่มีหลักการที่แน่นอน เนื่องจากตัวอย่างของโปรแกรมประยุกต์ถูกสร้างขึ้นในระหว่างการพัฒนาเฟรมเวิร์กโดยมีวัตถุประสงค์ที่ต่างกัน เช่น บางตัวอย่างถูกสร้างขึ้นเพื่อใช้เป็นแบบในการพัฒนาเฟรมเวิร์ก บางตัวอย่างถูกสร้างขึ้นเพื่อทดสอบความถูกต้องของเฟรมเวิร์กก่อนการใช้งานจริง เป็นต้น สปราร์ค [18] กล่าวว่า ตัวอย่างเหล่านี้ยากเกินไปสำหรับผู้เริ่มใช้งานเพื่อการพัฒนาโปรแกรมประยุกต์ และได้แนะนำให้อธิบายฮอตสปอตเพิ่มเติมให้มากกว่านี้ โดยเริ่มตั้งแต่ตัวอย่างที่ง่ายก่อนแล้วค่อยเพิ่มเติมไปจนถึงตัวอย่างในระดับที่ยากขึ้น

ถึงแม้ว่าตัวอย่างซอสโค้ดของโปรแกรมประยุกต์ที่สร้างขึ้นจากเฟรมเวิร์กนี้เป็นคู่มือที่ไม่มีรูปแบบที่เหมาะสมในการใช้งานเพื่อการเรียนรู้ หรือการสืบค้นข้อมูลตัวอย่างให้ได้ตามต้องการ แต่ซอสโค้ดของโปรแกรมประยุกต์สามารถนำมาเป็นส่วนหนึ่งของคู่มือในการเพิ่มประสิทธิภาพการใช้งานได้

### 2.3.2 เรายี่และคูกบุ๊ก (Recipe and Cookbooks)

เรายี่เป็นคู่มือที่แสดงตัวอย่างพื้นฐานในการใช้งานเฟรมเวิร์ก ที่เขียนในลักษณะภาษาที่ไม่เป็นทางการ โดยมีตัวอย่างซอสโค้ดเป็นส่วนประกอบและอาจจะมีรูปภาพประกอบหรือไม่มีก็ได้ ถึงแม้เรายี่จะมีรูปแบบที่ไม่เป็นทางการ แต่ก็มักเขียนเป็นโครงสร้างตามลำดับ เช่น เริ่มต้นด้วยส่วนวัตถุประสงค์ของการใช้งาน ตามด้วยขั้นตอนการใช้งานเรายี่ ส่วนอ้างอิงถึงเรายี่อื่นและลงท้ายด้วยตัวอย่างซอสโค้ด เป็นต้น

คูกบุ๊กเป็นคู่มือที่นำเรายี่หลายเรายี่มารวมกัน โดยส่วนใหญ่จะมีสารบัญของเรายี่เตรียมไว้ให้ในหน้าแรก หรือใช้หน้าแรกของเรายี่แสดงเป็นภาพรวมของคูกบุ๊ก ตัวอย่างของคู่มือประเภทนี้ ได้แก่ [2] [9] และ [20]

ในการทำให้คู่มือมีโครงสร้างที่ดีขึ้น จอห์นสัน [12] ได้เสนอภาษารูปแบบ (Pattern Language) ที่ไม่เป็นทางการขึ้นเพื่อใช้เขียนคู่มือเฟรมเวิร์กในรูปแบบภาษารวมชาติ คู่มือเฟรมเวิร์กนี้จะประกอบไปด้วยกลุ่มของรูปแบบ ซึ่งในแต่ละรูปแบบจะประกอบด้วยคำอธิบายถึงปัญหาที่เกิดขึ้นในโดเมนของปัญหานั้น วัตถุประสงค์ วิธีการใช้งาน วิธีการแก้ปัญหา การออกแบบ และตัวอย่างโปรแกรมประยุกต์ แต่อย่างไรก็ตามภาษารูปแบบของจอห์นสันยังมีรูปแบบที่ยังไม่แน่นอน ที่เสนอเพียงส่วนประกอบที่ควรจะมีในคู่มือการใช้งานเฟรมเวิร์กเท่านั้น ดังนั้นการศึกษาถึงรูปแบบที่มีความชัดเจน และเหมาะสมกับผู้ใช้ในกลุ่มผู้พัฒนาโปรแกรมประยุกต์จึงเป็นสิ่งสำคัญอย่างยิ่งที่วิทยานิพนธ์ฉบับนี้ จะนำภาษารูปแบบไปปรับปรุงให้มีรายละเอียดชัดเจนมากยิ่งขึ้น

### 2.3.3 รูปแบบการออกแบบ (Design Pattern)

รูปแบบการออกแบบ [14] จะเสนอทางแก้ปัญหาให้กับปัญหาที่อาจจะเกิดขึ้นในบริบท (context) ที่ให้ไว้ รูปแบบการออกแบบต่าง ๆ จะมาจากประสบการณ์ของการออกแบบในระดับของสถาปัตยกรรมย่อยที่เกี่ยวกับการกำหนดความสัมพันธ์ระหว่างคลาสและออบเจกต์ที่เกี่ยวข้องกันในปัญหานั้น คำอธิบายในคู่มือชนิดนี้จะประกอบไปด้วยปัญหา บริบทของปัญหานั้น ทางแก้ปัญหาและผลลัพธ์ที่คาดว่าจะเกิดขึ้นหลังจากการแก้ปัญหา โดยปัญหาที่เกิดขึ้นอาจนำเสนอในลักษณะของตัวอย่าง ทางแก้ปัญหาจะอธิบายถึงออบเจกต์และคลาสที่มีส่วนร่วมกันในการออกแบบ หน้าที่และการทำงานร่วมกัน คู่มือเฟรมเวิร์กประเภทนี้อาจเตรียมตัวอย่างของการแก้ปัญหาที่นำไปใช้กับบริบทหนึ่งที่น่าจะแน่นอนไว้ให้ นอกจากนี้การวิเคราะห์ประโยชน์ ข้อดีและข้อเสียของการใช้งานรูปแบบแต่ละอัน ถือเป็นคำอธิบายที่สำคัญอย่างหนึ่งในคู่มือชนิดนี้

ลาโจอีและเคลเลอร์ [14] ได้นำรูปแบบการออกแบบไปช่วยในการเขียนเรซีพี โดยให้รูปแบบการออกแบบไปอธิบายประเด็นความสัมพันธ์ของสถาปัตยกรรมการออกแบบที่มีรายละเอียดมากกว่าเรซีพีแบบเดิม ซึ่งจะทำให้ผู้พัฒนาโปรแกรมมีความเข้าใจเฟรมเวิร์กและสามารถใช้งานเฟรมเวิร์กได้อย่างมีประสิทธิภาพยิ่งขึ้น

### 2.3.4 คู่มืออ้างอิง (Reference Manual)

คู่มืออ้างอิงสำหรับโปรแกรมเชิงวัตถุจะประกอบไปด้วยคำอธิบายของแต่ละคลาส ตัวแปรโกลบอล ค่าคงที่ และแบบชนิด (type) โดยทั่วไปแล้วคำอธิบายของคลาสจะบอกถึงวัตถุประสงค์ หรือหน้าที่ของคลาส บทบาทของสมาชิกข้อมูลแต่ละตัว และข้อมูลที่เกี่ยวข้องกับแต่ละเมทอด (method) คำอธิบายของเมทอดจะอธิบายถึงฟังก์ชันการทำงาน เงื่อนไขก่อนและหลังการใช้เมทอด และชี้ให้เห็นว่าสมาชิกข้อมูลใดที่มีผลกระทบหรือใช้งานเมทอดนี้ได้บ้าง

สำหรับคู่มือเฟรมเวิร์กคำอธิบายจะเพิ่มเติมในปัจจุบันที่เกี่ยวข้องกับบทบาทของคลาสหรือเมทอดที่จัดให้เกิดความยืดหยุ่นกับฮอตสโปท เช่น คลาสใดที่ต้องการให้ผู้พัฒนาโปรแกรมสร้างสับคลาสย่อยเพิ่มเติม หรือเมทอดใดที่ต้องการให้ผู้พัฒนาโปรแกรมเขียนทับ เป็นต้น อย่างไรก็ตาม บัทเลอร์ จี และ ดีนอมมี พี [5] ได้ให้ความเห็นว่าคู่มืออ้างอิงนี้ไม่เหมาะกับการใช้เพื่อการเรียนรู้การใช้งานเฟรมเวิร์ก เนื่องจากมีการแสดงรายการคำสั่งที่เรียงลำดับตัวอักษร โดยไม่ได้ให้รายละเอียดของวิธีใช้งานและตัวอย่างประกอบไว้ด้วย

### 2.3.5 ภาพรวมของเฟรมเวิร์ก (Framework Overview)

การกำหนดภาพรวมให้กับเฟรมเวิร์กถือเป็นขั้นแรกที่จะช่วยให้ผู้พัฒนาโปรแกรมประยุกต์เลือกใช้งานเฟรมเวิร์กให้ตรงกับความต้องการในการสร้างโปรแกรมประยุกต์ ซึ่งนั่นก็คือการกำหนดโดเมนหรือขอบเขตของเฟรมเวิร์กว่าครอบคลุมถึงส่วนใด เช่น การบอกว่าส่วนใดของเฟรมเวิร์กถูกกำหนดไว้ชัดเจนไม่สามารถเปลี่ยนแปลงได้และส่วนใดสามารถจะปรับเปลี่ยนแก้ไขได้ เป็นต้น การเขียนคู่มือแบบนี้อาจนำเสนอในลักษณะตัวอย่างโปรแกรมประยุกต์ง่าย ๆ

### 2.3.6 อินเทอร์เฟซคอนแทรก (Interface Contract)

เป็นคู่มือที่เป็นสัญญาหรือข้อตกลงที่มีในเฟรมเวิร์ก โดยอินเทอร์เฟซคอนแทรกของคลาสจะมีข้อกำหนดอยู่ 2 อย่าง คือ คลาสอินเทอร์เฟซที่ใช้ในการกำหนดโครงสร้างของคลาสว่าต้องประกอบด้วยเมทอดอะไรบ้าง และคลาสอินวาเรียน (class invariants) ที่เป็นข้อตกลงด้านความหมายซึ่งจะกำหนดเงื่อนไขของคลาสที่ต้องเป็นจริง โดยผู้พัฒนาโปรแกรมประยุกต์จำเป็นต้องทราบข้อมูลเกี่ยวกับคลาสทั้งสองนี้เพื่อสามารถพัฒนาเป็นโปรแกรมประยุกต์ตามความต้องการได้

นอกจากคู่มือประเภทหลัก ๆ ที่ โมฮัมเมด อี ฟายาดและคณะ ได้กล่าวไว้แล้วยังมีงานวิจัยที่เกี่ยวข้องกับการจัดทำคู่มือเฟรมเวิร์กอื่น ๆ อีกได้แก่

### 2.3.7 การเขียนคู่มือในมุมมองด้านไดนามิก (Dynamic View)

ควินซู จีนและฟางเซียว ซู [17] ได้เขียนบทความกล่าวถึงคุณสมบัติของคู่มือเฟรมเวิร์กที่จะทำให้สามารถใช้งานเฟรมเวิร์กได้อย่างมีประสิทธิภาพ คือ คู่มือเฟรมเวิร์กจะต้องมีรูปแบบการเขียนที่เรียบง่ายและมีความถูกต้อง นอกจากนี้ยังต้องคำนึงถึงความแตกต่างของจุดประสงค์การใช้งานของผู้ที่เกี่ยวข้องกับเฟรมเวิร์กแต่ละประเภท โดยได้เสนอแนวทางการจัดทำคู่มือเฟรมเวิร์กแบบผสมผสาน โดยชี้ให้เห็นปัญหาของการเขียนคู่มือแบบเดิมที่มักมีแต่มุมมองทางด้านสแตติก (static view) โดยลืมความสำคัญของมุมมองทางด้านไดนามิกที่อาจจะเป็นส่วนสำคัญที่

ทำให้ผู้ใช้เข้าใจหน้าที่การทำงานและความสัมพันธ์ระหว่างคลาสได้ดียิ่งขึ้น ในบทความนี้ได้เสนอแนวทางการเขียนคู่มือในมุมมองไดนามิก เช่น เทคนิคแผนภาพ (diagram technic) ภาพเคลื่อนไหว การเชื่อมโยงข้อความหลายมิติ (hypertext link) เป็นต้น มาช่วยในการเขียนคู่มือเฟรมเวิร์กให้มีประสิทธิภาพ แต่อย่างไรก็ตามบทความนี้ยังเป็นเพียงแนวคิดไม่มีหลักการในการเขียนที่แน่นอน เช่น การเรียงลำดับหัวข้อ การผสมผสานของคู่มือเฟรมเวิร์กที่จะผสมผสานคู่มือชนิดใดบ้าง เป็นต้น นอกจากนี้ยังไม่มีการจัดแสดงตัวอย่างให้เห็นอย่างชัดเจน

### 2.3.8 การเขียนคู่มือในมุมมองของรียูสเคส (Reuse Case)

บัทเลอร์ จีและคณะ [4] ได้ทำการศึกษาคู่มือการใช้งานเฟรมเวิร์กในมุมมองของรียูสเคส โดยแบ่งรียูสเคส เป็น 6 กลุ่มได้แก่

2.3.8.1 การเลือก (Selecting) คือ การเลือกเฟรมเวิร์กให้เหมาะสมกับโปรแกรมประยุกต์ที่ต้องการ

2.3.8.2 การประกอบ (Composing) คือ การเลือกไลบรารี (library) ที่เหมาะสมมาประกอบลงไปในแต่ละฮอตสปอต หรือถ้าเป็นแบล็กบ็อกซ์เฟรมเวิร์กโปรแกรมประยุกต์ก็จะประกอบขึ้นจากสับคลาสที่มีอยู่แล้ว

2.3.8.3 การขยาย (Extending) คือ ฮอตสปอตที่ถูกปรับเปลี่ยนไปในทางที่ผู้ออกแบบเฟรมเวิร์กได้วางไว้ เช่น เพิ่มจำนวนของดาต้าเบสที่สามารถใช้ในการติดต่อให้มากขึ้น เป็นต้น จะสังเกตเห็นว่าการเปลี่ยนแปลงแบบนี้จะเกี่ยวกับจำนวนมากกว่าชนิด

2.3.8.4 ความยืดหยุ่น (Flexing) คือ ฮอตสปอตที่ถูกปรับเปลี่ยนไปในทางที่นอกเหนือจากที่ผู้ออกแบบเฟรมเวิร์กกำหนดไว้ แต่การเปลี่ยนแปลงนั้นก็ต้องเป็นไปตามกฎและข้อบังคับของการออกแบบเช่นเดิม เช่น เปลี่ยนการรับข้อมูลเข้าจากโหมดกราฟฟิกไปใช้โหมดเซนเซอร์หรือโหมดเสียงแทน เป็นต้น จะสังเกตเห็นว่าการเปลี่ยนแปลงแบบนี้จะเกี่ยวกับการเปลี่ยนชนิด

2.3.8.5 การพัฒนา (Evolving) คือการพัฒนาโดยการเพิ่มความยืดหยุ่นให้กับฮอตสปอตที่มีอยู่ หรือเพิ่มฮอตสปอตใหม่เข้าไป

2.3.8.6 การนำความคิด (Mining) คือ การนำความคิดที่อาจจะนำไปใช้ได้กับบริบทอื่น เช่น การนำเฟรมเวิร์กที่ได้ไปใช้กับโดเมนของเฟรมเวิร์กอื่น เป็นต้น

ในบทความนี้ได้วิเคราะห์การจัดทำคู่มือเฟรมเวิร์กที่ใช้กันอยู่ในปัจจุบันในมุมมองของรียูสเคสไว้ว่า รียูสเคสในแต่ละกลุ่มสามารถใช้คู่มือเฟรมเวิร์กที่มีอยู่มาอธิบายได้ โดยถ้าพิจารณาจากการใช้งานของผู้พัฒนาโปรแกรมประยุกต์แล้วจะเห็นว่าการพัฒนาโปรแกรมประยุกต์พื้นฐานจะใช้งานใน 3 หมวดของรียูสเคส นั่นคือ การเลือก การประกอบ และการขยาย ซึ่งสำหรับ

ผู้พัฒนาโปรแกรมประยุกต์ที่ต้องการปรับแต่งโครงสร้างทางสถาปัตยกรรมเพื่อการพัฒนาโปรแกรมประยุกต์ที่มีความซับซ้อนมากอาจต้องใช้ในหมวดความยืดหยุ่นด้วย แต่เนื่องจากเป็นเพียงส่วนน้อยเท่านั้น ดังนั้นในการวิจัยนี้จึงสนใจแค่ 3 หมวดแรกของริยูลเคสของการทำงานเฟรมเวิร์ก บัทเลอร์ จี และคณะ[4] ได้กล่าวถึงคู่มือเฟรมเวิร์กที่มีอยู่ในปัจจุบัน ที่สามารถอธิบายริยูลเคสทั้งสามนี้ได้ ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงคู่มือที่มีอยู่ในปัจจุบันที่สามารถอธิบายริยูลเคสในหมวดการเลือก การประกอบ และการขยาย

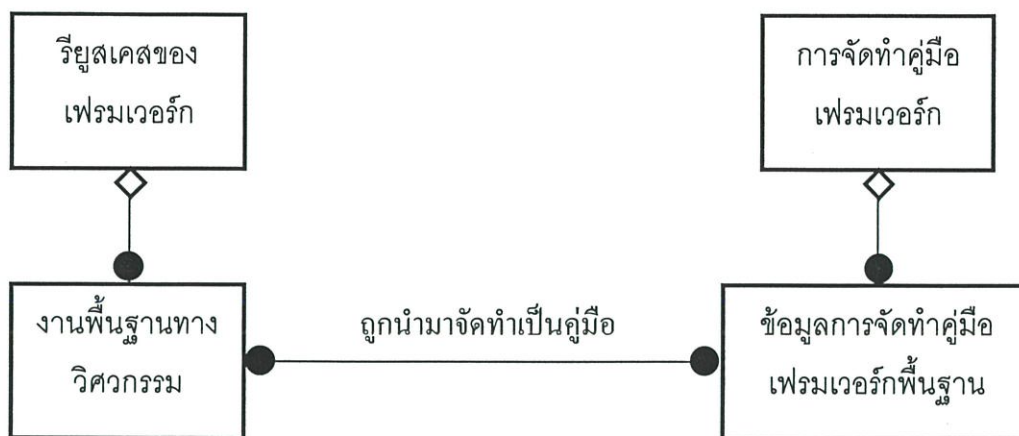
ประเภทของคู่มือเฟรมเวิร์ก	ริยูลเคส		
	การเลือก	การประกอบ	การขยาย
เรซีพี	-	Y	Y
คู่มือ	y	Y	Y
ภาพรวมของเฟรมเวิร์ก	Y	-	-

Y = เป็นวัตถุประสงค์หลักของคู่มือ

y = เป็นวัตถุประสงค์รองของคู่มือ

จากตารางที่ 2.1 แสดงให้เห็นว่าคู่มือเฟรมเวิร์กที่มีอยู่ในปัจจุบันสามารถตอบสนองต่อความต้องการในมุมมองของริยูลเคสได้ ซึ่งผู้พัฒนาโปรแกรมประยุกต์สามารถใช้คู่มือที่ประกอบไปด้วยภาพรวมของเฟรมเวิร์ก คู่มือและเรซีพีในการสนองตอบกับความต้องการในการใช้งานได้อย่างเพียงพอ แต่อย่างไรก็ตามในบทความนี้ไม่ได้กล่าวถึงส่วนประกอบที่ชัดเจนเกี่ยวกับรูปแบบและเนื้อหาภายในคู่มือที่เหมาะสมกับการใช้งานเพื่อการพัฒนาโปรแกรมประยุกต์

บัทเลอร์ จี และคณะ [7] ได้เขียนบทความเพื่อวิเคราะห์การจัดทำคู่มือเฟรมเวิร์กตามความต้องการของใช้งานเฟรมเวิร์กรียูลเคส ในบทความนี้ได้แบ่งริยูลเคสแต่ละกลุ่มออกเป็นงานพื้นฐานทางวิศวกรรมย่อย โดยงานพื้นฐานทางด้านวิศวกรรมเฟรมเวิร์กแต่ละงานจะต้องการข้อมูลการจัดทำคู่มือเฟรมเวิร์กพื้นฐานของตัวเอง เพื่อนำมาจัดทำเป็นคู่มือเฟรมเวิร์กตามต้องการ ดังรูปที่ 2.1



รูปที่ 2.1 แสดงความสัมพันธ์ระหว่างงานพื้นฐานทางด้านวิศวกรรมเฟรมเวิร์กกับข้อมูลการจัดทำคู่มือเฟรมเวิร์กพื้นฐาน

ในบทความนี้ได้แบ่งรายละเอียดออกเป็น 3 กลุ่ม คือ กลุ่มการใช้งานปกติ กลุ่มที่ต้องการใช้งานขั้นสูง และกลุ่มที่ทำการพัฒนาเฟรมเวิร์ก ผู้ใช้งานเฟรมเวิร์กเพื่อพัฒนาโปรแกรมประยุกต์ส่วนใหญ่จะอยู่ในกลุ่มของผู้ใช้งานปกติที่แบ่งออกเป็นรายละเอียดของการเลือก และรายละเอียดของการประกอบ รายละเอียดของการเลือกเป็นกระบวนการที่หาเฟรมเวิร์กที่ตรงกับปัญหาของเรา ซึ่งสามารถแบ่งออกเป็นงานพื้นฐานทางด้านวิศวกรรมย่อย คือ การหาความเข้ากันของข้อมูลสำหรับเฟรมเวิร์กที่มีโดเมนเฉพาะ และการหาความเข้ากันของข้อมูลที่ไม่ขึ้นกับโดเมนใดโดเมนหนึ่ง ส่วนรายละเอียดของการประกอบจะต้องทำการเติมเต็มให้กับฮอตสปอต โดยจะมีงานที่เกี่ยวข้อง 3 งานด้วยกัน คือ ความเข้ากันได้ของอินเทอร์เฟซ เพื่อให้แน่ใจว่าสิ่งที่จะเติมเต็มนั้นสามารถจะเข้าได้กับเมทอดอื่นๆ ในเฟรมเวิร์ก ความเข้ากันได้ในเชิงพฤติกรรม เป็นการตรวจสอบทางความหมายให้แต่ละเมทอดสามารถทำงานได้อย่างถูกต้องตามที่ต้องการ และการลงรหัสตามพฤติกรรม เป็นการเขียนโค้ดเพื่อให้เป็นไปตามพฤติกรรมที่กำหนดไว้ในตอนต้น โดยงานพื้นฐานทางด้านวิศวกรรมย่อยเหล่านี้ต้องการการจัดทำคู่มือเฟรมเวิร์กพื้นฐานมาอธิบาย ซึ่งข้อมูลพื้นฐานที่ใช้ในการอธิบายการใช้งานเฟรมเวิร์กเพื่อพัฒนาเป็นโปรแกรมประยุกต์เหล่านี้ สามารถนำมาวิเคราะห์หาความสัมพันธ์กับคู่มือที่มีอยู่แล้วได้ผลว่า คู่มือคู่มือและเรซีพีเป็นคู่มือที่มีความสมบูรณ์สามารถใช้อธิบายงานพื้นฐานทางวิศวกรรมที่จะใช้ในการพัฒนาโปรแกรมประยุกต์ได้ทั้งหมด ส่วนคู่มือเฟรมเวิร์กอื่น ๆ นั้นสามารถอธิบายงานพื้นฐานทางวิศวกรรมในการพัฒนาโปรแกรมประยุกต์ได้บ้างแต่ไม่ครอบคลุมทั้งหมด ซึ่งอาจจะต้องนำเอาหลาย ๆ คู่มือมารวมกันเพื่อให้สามารถครอบคลุมการใช้งานเฟรมเวิร์กเพื่อพัฒนาโปรแกรมประยุกต์

จากบทความนี้ทำให้เห็นว่าคู่มือคู่มือและเรซีพี้เพียงพอกับความต้องการของผู้ใช้งานปกติ ซึ่งมีรายละเอียดและเนื้อหาที่ครอบคลุมงานพื้นฐานทางวิศวกรรมในการใช้งานเฟรมเวอร์ค เพื่อพัฒนาเป็นโปรแกรมประยุกต์

### 2.3.9 แนวทางการเขียนคู่มือเพื่อผู้พัฒนาโปรแกรมประยุกต์

บัทเลอร์ จี และ ดินอมมี พี [6] ได้เขียนบทความเกี่ยวกับแก้ปัญหาการใช้งานคู่มือเฟรมเวอร์ค เพื่อให้เกิดความรวดเร็วในการพัฒนาโปรแกรมประยุกต์ ในบทความนี้ได้เน้นถึงปัญหาของผู้ใช้งานเฟรมเวอร์คในครั้งแรกที่ต้องใช้เวลาในการเรียนรู้การใช้งานเฟรมเวอร์คค่อนข้างมาก บทความนี้ได้เสนอแนวทางในการจัดทำคู่มือเพื่อผู้พัฒนาโปรแกรมประยุกต์ไว้ ดังนี้

2.3.9.1 มีคำอธิบายภาพรวมของเฟรมเวอร์ค เนื่องจากผู้พัฒนาโปรแกรมประยุกต์จะต้องการทราบบริบทของเฟรมเวอร์ค โดยคำอธิบายภาพรวมควรเอาไว้ในหน้าแรก หรือเรซีพี้แรกของคู่มือการใช้งานเฟรมเวอร์ค

2.3.9.2 มีตัวอย่างโปรแกรมประยุกต์จำนวนหนึ่ง ที่เขียนขึ้นเฉพาะเพื่อประกอบการอธิบายการใช้งานเฟรมเวอร์ค โดยการนำเสนอควรเรียงลำดับจากง่ายไปยาก และค่อย ๆ นำเสนอฮอตสปอท์ใหม่ในเวลาเดียวกัน

2.3.9.3 คัดเลือกข้อมูลและจัดการเก็บข้อมูลเป็นส่วน ๆ ตามการใช้งาน เพื่อลดจำนวนของข้อมูลเอาเฉพาะส่วนที่สำคัญ ๆ ไว้เท่านั้น โดยอาจเขียนเป็นคู่มือที่จัดรูปแบบตามภาษารูปแบบของจอห์นสัน [12]

จากบทความนี้ทำให้เห็นถึงแนวทางการเขียนคู่มือการใช้งานเฟรมเวอร์คเพื่อผู้พัฒนาโปรแกรมประยุกต์ แต่หลักการนี้ก็ยังเป็นเพียงแนวทางที่ยังต้องการเสริมอื่น ๆ ที่มีความชัดเจนในรูปแบบของคู่มือเฟรมเวอร์คมากกว่านี้

### 2.3.10 คู่มือการใช้งานเฟรมเวอร์คตามทฤษฎีการสอนแบบมินิมอลลิสต์ (Minimalist Instruction)

อะเดมาร์ อะกิวอาร์ [1] ได้ทำการจัดทำคู่มือการใช้งานเฟรมเวอร์คที่สามารถประหยัดงบประมาณ โดยใช้หลักการบนทฤษฎีของการสอนแบบมินิมอลลิสต์ ในการอธิบายข้อมูลที่จำเป็นเพียงเล็กน้อยเพื่อให้บรรลุวัตถุประสงค์ในการใช้งาน และกระตุ้นให้ผู้ใช้งานได้ลงมือใช้งานจริง การเขียนคู่มือแบบนี้จะให้การอธิบายหัวข้อเพียงสั้น ๆ จากนั้นก็พยายามให้ผู้ใช้งานได้ใช้ประสบการณ์ที่มีอยู่ก่อนมาเป็นตัวช่วยในการเรียนรู้การใช้งาน การใช้งานเน้นไปที่ความเข้าใจและการใช้งานเฟรมเวอร์คมากกว่าความเข้าใจเกี่ยวกับการออกแบบ ซึ่งข้อดีและข้อเสียของมินิมอลลิสต์จะขึ้นกับโดเมนของปัญหาและความซับซ้อนของเฟรมเวอร์คที่ใช้ รูปแบบของการเขียนคู่มือแบบมินิมอลลิสต์นี้น่าจะเป็นประโยชน์สำหรับผู้ที่มีประสบการณ์ในการใช้งานมาแล้วบ้าง หรือมีความรู้ที่

เกี่ยวข้องมาบ้าง การใช้งานคู่มือชนิดนี้ก็อาจจะมียุทธวิธีน้อยสำหรับเฟรมเวิร์กที่มีความซับซ้อนมาก หรือผู้ที่เริ่มต้นใช้งานเฟรมเวิร์ก

### 2.3.11 ปัญหาของการใช้งานคู่มือเฟรมเวิร์กในปัจจุบัน

ดักกลาส เคิร์ก และคณะ[22] ได้ทำการทดลองเพื่อหาปัญหาของการใช้งานเฟรมเวิร์กที่แท้จริงและหาความสัมพันธ์กับคู่มือเฟรมเวิร์กที่มีอยู่ในปัจจุบัน ซึ่งปัญหาหลักที่พบสามารถแบ่งออกเป็น 4 หมวดด้วยกัน คือ ความเข้าใจเกี่ยวกับฟังก์ชัน ความเข้าใจความสัมพันธ์ระหว่างคลาส การจับคู่ฟังก์ชันที่ตรงกัน และความเข้าใจรายละเอียดเกี่ยวกับสถาปัตยกรรม จากบทความนี้ ดักกลาสได้ศึกษาปัญหาที่พบจากการทดลองกับคู่มือเฟรมเวิร์กในปัจจุบัน พบว่ามีเพียงคู่มือเฟรมเวิร์กแบบภาษารูปแบบเท่านั้นที่สามารถตอบสนองปัญหาใน 4 หมวดได้ทั้งหมด แต่อย่างไรก็ตามถึงแม้ว่าภาษารูปแบบจะมีประสิทธิภาพที่สามารถใช้คู่มือในการใช้งานเฟรมเวิร์กได้ แต่ในเรื่องของรูปแบบการนำเสนอยังไม่ชัดเจน ในตอนท้ายของบทความนี้ได้เสนอให้งานวิจัยในอนาคตจัดทำรูปแบบคู่มือเฟรมเวิร์กให้มีความชัดเจนและมีแนวทางให้ผู้เขียนคู่มือสามารถทำตามได้ง่ายขึ้น

จากงานวิจัยที่เกี่ยวข้องกับการเขียนคู่มือการใช้งานเฟรมเวิร์กข้างต้นจะเห็นว่าแต่ละงานวิจัยมีข้อดีและข้อเสียของการเขียนคู่มือการใช้งานเฟรมเวิร์กที่แตกต่างกันออกไป โดยยังไม่มีงานวิจัยใดที่สามารถแก้ปัญหาเรื่องรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์กที่ทำให้ผู้พัฒนาโปรแกรมประยุกต์พัฒนาโปรแกรมประยุกต์จากการใช้งานเฟรมเวิร์กได้อย่างมีประสิทธิภาพในระยะเวลาไม่มาก นอกจากนี้ในการเขียนคู่มือการใช้งานเฟรมเวิร์กให้มีประสิทธิภาพ ไม่ได้ดูจากการเขียนให้มีเนื้อหาครบถ้วนเป็นไปตามความต้องการเท่านั้น หากแต่ยังต้องจัดการส่วนติดต่อกับผู้ใช้ ให้ผู้ใช้สามารถใช้งานได้ง่ายและสะดวกด้วย เพราะไม่ว่าเนื้อหาที่ใช้ในการเขียนคู่มือจะดีเท่าใดก็ตาม แต่ขาดการจัดรูปแบบการเขียนที่เหมาะสมแล้ว คู่มือนั้นอาจทำให้ผู้พัฒนาโปรแกรมประยุกต์ไม่สามารถใช้งานได้มีประสิทธิภาพเท่าที่ควร งานวิจัยของราฟ อี จอห์นสัน [12] ได้กล่าวถึงรูปแบบของการจัดรูปแบบของการเขียนคู่มือไว้เช่นกัน เช่น ลักษณะของตัวอักษรหรือการจัดรูปแบบที่เป็นรูปแบบเดียวกัน เป็นต้น แต่ก็กล่าวไว้เพียงเล็กน้อยเท่านั้น ดังนั้นในวิทยานิพนธ์ฉบับนี้จะนำเอาส่วนดีของการจัดทำคู่มือเฟรมเวิร์กที่มีอยู่แล้วมาผนวกกับหลักการจัดรูปแบบของส่วนติดต่อกับผู้ใช้ เพื่อให้ได้รูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์กที่เหมาะสมและให้ผู้พัฒนาโปรแกรมประยุกต์สามารถใช้งานคู่มือเฟรมเวิร์กได้อย่างมีประสิทธิภาพยิ่งขึ้น

## บทที่ 3

# จิตวิทยาและการออกแบบส่วนติดต่อกับผู้ใช้

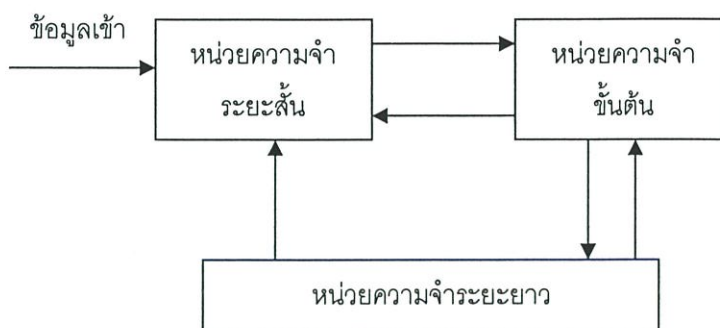
ในการออกแบบคู่มือการใช้งานเฟรมเวิร์ก ถ้าผู้เขียนไม่ได้ออกแบบคู่มือโดยคำนึงถึงการใช้งานของผู้ใช้เป็นหลัก หากแต่คำนึงถึงความชอบของผู้เขียนเอง อาจทำให้คู่มือเฟรมเวิร์กที่ได้ขาดคุณสมบัติของความเป็นมิตรกับผู้ใช้ ดังนั้นในบทนี้จะกล่าวถึงจิตวิทยาที่เกี่ยวข้องกับการออกแบบคู่มือการใช้งานเฟรมเวิร์กอันได้แก่ ปัจจัยเกี่ยวกับผู้ใช้ในออกแบบคู่มือการใช้งานเฟรมเวิร์ก หลักการออกแบบตามความต้องการของผู้ใช้ และเทคนิคในการออกแบบส่วนติดต่อกับผู้ใช้ โดยหลักการต่าง ๆ เหล่านี้สามารถนำมาใช้ประกอบเป็นพื้นฐานในการออกแบบคู่มือการใช้งานเฟรมเวิร์กได้

### 3.1 ปัจจัยที่เกี่ยวข้องกับการเรียนรู้ของมนุษย์ในการออกแบบส่วนติดต่อกับผู้ใช้

ในการทำความเข้าใจองค์ประกอบของมนุษย์ ในด้านกลไกการทำงานที่เกี่ยวข้องกับความคิด ความจำ การเรียนรู้ ความรู้สึกและความต้องการของมนุษย์ ก็เพื่อให้ได้รูปแบบของคู่มือเฟรมเวิร์กที่มีความเหมาะสมและตรงกับความต้องการของผู้ใช้งานมากที่สุด ดร.จรณิต แก้วกังวาล [25] ได้กล่าวถึงปัจจัยที่เกี่ยวข้องกับการเรียนรู้ของมนุษย์ที่สามารถนำมาใช้กับหลักการออกแบบส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก ดังนี้

#### 3.1.1 โครงสร้างความคิดและความจำ

ลักษณะโครงสร้างของระบบความคิดมนุษย์มีหลักการทำงาน คือ มนุษย์จะรับรู้ผ่านประสาทสัมผัส แล้วส่งผ่านมายังหน่วยความจำระยะสั้นเพื่อแบ่งการประมวลผลเป็นหน่วยย่อย ๆ แต่ยังไม่ถูกบันทึก หลังจากนั้นข้อมูลจะส่งผ่านไปยังส่วนที่เรียกกันว่าหน่วยความจำขั้นต้น (working memory) ในบางครั้งอาจเรียกรวมส่วนนี้เป็นหน่วยเดียวกับหน่วยความจำระยะสั้น และสุดท้ายก็ไปเก็บข้อมูลในหน่วยความจำระยะยาว ซึ่งจะจัดเก็บข้อมูลอย่างถาวร ดังรูปที่ 3.1



รูปที่ 3.1 แสดงโครงสร้างระบบความจำของมนุษย์

ข้อมูลที่ส่งมาจากหน่วยความจำระยะสั้นจะค่อยๆ เข้ามาเก็บในหน่วยความจำระยะยาว ดังนั้นหากมีข้อมูลเข้ามามากเกินไป จนหน่วยความจำระยะสั้นไม่สามารถรับได้หมดจะเป็นผลทำให้ข้อมูลบางส่วนสูญหายไป ดังนั้นผู้วิจัยเห็นว่าการเขียนคู่มือเฟรมเวิร์กควรคำนึงถึงขีดจำกัดในเรื่องความจำของมนุษย์ ซึ่งคู่มือเฟรมเวิร์กที่มีการอธิบายข้อมูลเป็นโครงสร้างเข้าใจง่าย จะสามารถช่วยให้ผู้พัฒนาโปรแกรมประยุกต์จำตัวแปรและขั้นตอนการพัฒนาโปรแกรมได้อย่างเป็นระบบโดยไม่สับสน

#### 3.1.1.1 หน่วยความจำระยะสั้น

หน่วยความจำระยะสั้นจะอยู่บริเวณจิตสำนึกของมนุษย์ ซึ่งเป็นตำแหน่งเดียวกับที่มนุษย์ใช้คิดเรื่องราวต่าง ๆ โดยข้อมูลในหน่วยความจำระยะสั้นนี้จะลดลงไปหลังจากรับเข้ามาประมาณ 20 วินาที แต่เราสามารถคงความทรงจำระยะสั้นได้โดยการทวนซ้ำ (rehearsal) แต่ก็ยังไม่สามารถทำให้ความทรงจำนี้อยู่ได้อย่างถาวรตลอดไป ข้อมูลในหน่วยความจำระยะสั้นจะเก็บอยู่ในหน่วยย่อย ๆ ที่เรียกว่า ชังค์ (chunk) ซึ่งปกติหน่วยความจำระยะสั้นจะมีความจุประมาณ 7 หน่วย โดยผู้วิจัยเห็นว่าการเขียนคู่มือการใช้งานเฟรมเวิร์กควรให้ความสำคัญกับความสามารถในหน่วยความจำระยะสั้นด้วย กล่าวคือ การสรุปรายการคำสั่งไม่ควรมีเกิน 7 รายการ เพราะถ้ามีมากกว่านี้อาจทำให้ผู้ใช้สับสน หรือถ้าจำเป็นต้องมีรายการมากกว่า 7 รายการ ก็ควรจัดแบ่งเป็นหมวดหมู่เพื่อให้ผู้ใช้เห็นความสัมพันธ์และจดจำคำสั่งได้ง่ายขึ้น

#### 3.1.1.2 หน่วยความจำระยะยาว

หลังจากข้อความถูกทวนซ้ำในหน่วยความจำระยะสั้นแล้ว ข้อความเหล่านั้นจะถูกถ่ายโอนไปสู่หน่วยความจำระยะยาว ซึ่งในหน่วยความจำระยะยาวนี้มนุษย์จะสามารถจดจำข้อมูลได้ในระยะเวลาที่ยาวนานกว่ามาก แต่ในการเข้าถึงหน่วยความจำระยะยาวนี้อาจมีความยุ่งยากมากกว่า โดยเฉพาะถ้าปราศจากการทวนซ้ำแล้ว อาจจะทำให้เกิดข้อขัดแย้งในการจัดเก็บและดึงข้อมูลขึ้นมาใช้ใหม่ได้ ดังนั้นจึงควรมีการเชื่อมโยงความรู้เดิมมาสู่ความรู้ใหม่ เพื่อให้ข้อมูลอยู่ในหน่วยความจำระยะยาวได้นานและดึงมาใช้ได้ดีขึ้น เช่นเดียวกับการเขียนคู่มือเฟรมเวิร์กที่ควรมี

รูปแบบเดียวกันทั้งเอกสารอาทิเช่น การวางแผน เนื้อหา และจุดเชื่อมโยง ควรวางไว้ในตำแหน่งเดียวกันในทุกหน้า เพื่อให้ผู้พัฒนาโปรแกรมประยุกต์เกิดความคุ้นเคยกับส่วนต่าง ๆ ในคู่มือได้อย่างรวดเร็ว เป็นผลทำให้ผู้พัฒนาโปรแกรมประยุกต์เรียนรู้การใช้งานเฟรมเวิร์กได้เร็วขึ้นด้วย

### 3.1.2 รูปแบบและระดับของความรู้

เมื่อข้อมูลผ่านเข้ามาในส่วนของหน่วยความจำ มนุษย์ไม่ได้เก็บข้อมูลเป็นรูปของข้อมูลดิบ แต่จะเก็บไว้ในรูปแบบเชิงนามธรรมที่เรียกกันว่า ความรู้ (knowledge) ดร.จรณิต แก้วกิงวาล [25] ได้ให้คำจำกัดความของการสร้างความรู้ไว้ คือ “กระบวนการประมวลผลข้อมูลทีผ่สานระหว่างข้อมูลใหม่ที่เข้ามาให้เข้ากับข้อมูลเดิมที่มีอยู่แล้ว” โดยองค์ความรู้ที่เกี่ยวข้องกับระบบคอมพิวเตอร์สามารถแบ่งออกเป็น 2 ระดับ คือ ความรู้ทางด้านความหมาย (semantic) และความรู้ทางด้านโครงสร้าง (syntactic)

3.1.2.1 ความรู้ด้านความหมาย คือ ความรู้ในเชิงความคิดและความเข้าใจในกระบวนการคอมพิวเตอร์ เช่น ความรู้ในเรื่องเทคนิคการจัดการข้อมูลแบบลิงค์ลิสต์ ความรู้ด้านนี้เป็นสิ่งที่ได้รับการเรียนรู้และประสบการณ์

3.1.2.2 ความรู้ด้านโครงสร้าง คือ ความรู้ในโครงสร้างรายละเอียด เช่น จะเขียนคำสั่งเพื่อประกาศรูปแบบและค่าของข้อมูลแบบต่าง ๆ ในภาษาจาวาได้อย่างไร เป็นต้น

ผู้วิจัยมีความเห็นว่าความไม่สอดคล้องประสานระหว่างองค์ความรู้ทั้งสองอย่างนี้อาจจะเกิดขึ้นได้ เนื่องจากวิธีการรับรู้และความสามารถในการจดจำแตกต่างกัน กล่าวคือ ความรู้ด้านความหมาย เป็นสิ่งที่เกิดขึ้นจากประสบการณ์และการเรียนรู้อย่างต่อเนื่อง ข้อมูลที่เข้ามาใหม่จะผสมเข้ากับข้อมูลเดิมที่มีอยู่แล้วในความจำ จนก่อให้เกิดความเข้าใจที่ชัดเจน ส่วนความรู้ด้านโครงสร้างจะเป็นความรู้ที่เกิดจากการใช้ความจำเพียงอย่างเดียว โดยในการเขียนคู่มือที่ทำให้ผู้ใช้เข้าใจความรู้ด้านความหมายได้เป็นสิ่งที่ยากกว่าด้านโครงสร้าง เนื่องจากคู่มือนั้นจะต้องทำให้ผู้ใช้เห็นความสัมพันธ์คลาสภายในเฟรมเวิร์ก และสามารถเชื่อมโยงองค์ความรู้จากการศึกษาจากคู่มือเฟรมเวิร์กไปใช้ในการพัฒนาเป็นโปรแกรมประยุกต์ของตนเอง

### 3.1.3 ทักษะและการเรียนรู้

ในการเรียนรู้สิ่งใหม่ควรทำการฝึกฝนความรู้เดิมจนเกิดเป็นทักษะขึ้น โดยทักษะเป็นการทำงานที่ไม่ต้องหยุดคิดด้วยจิตสำนึก ทำให้สามารถทำงานซับซ้อนได้มากและนานขึ้น ดังนั้นผู้วิจัยจึงมีความเห็นว่าในการจัดทำคู่มือเฟรมเวิร์กควรจะกระตุ้นให้ผู้ใช้ฝึกใช้งานไปด้วยบ่อย ๆ เพื่อให้เกิดเป็นทักษะ หรือทำการกระตุ้นให้ผู้ใช้ระดับกลางกลายเป็นผู้ใช้ที่เชี่ยวชาญได้ด้วยการฝึกปฏิบัตินั่นเอง นอกจากนี้การเขียนคู่มือเฟรมเวิร์กก็ควรจะระมัดระวังเรื่องการเขียนด้วยทักษะเช่นกัน เนื่องจากผู้เขียนคู่มือมักมีทักษะในการใช้งานเฟรมเวิร์กที่มากกว่าผู้อ่าน ดังนั้นการเขียน

คำอธิบายง่าย ๆ ที่ผู้เขียนคิดว่าผู้ใช้ทุกคนน่าจะเข้าใจ เนื่องจากคิดว่าเป็นสิ่งที่รู้กันดีอยู่แล้ว อาจจะทำให้ผู้ใช้สับสนได้ ดังนั้นในการเขียนคู่มือเฟรมเวิร์ก ผู้เขียนจึงควรวางตัวในตำแหน่งของผู้ใช้ ซึ่งยังไม่คุ้นเคยกับการใช้งานเฟรมเวิร์ก แล้วเขียนคู่มือออกมาตามความรู้สึกของผู้ใช้งานเฟรมเวิร์ก

### 3.2 หลักการออกแบบคู่มือตามความต้องการของมนุษย์ในเชิงจิตวิทยา

ดร.จรณิต แก้วกั้งวาล [25] ได้กล่าวถึงความรู้สึกและความต้องการของมนุษย์ขั้นพื้นฐานในเชิงจิตวิทยาที่สามารถนำมาสรุปเป็นลักษณะการเขียนคู่มือเฟรมเวิร์กเพื่อให้เกิดความพึงพอใจและเป็นไปตามความต้องการของผู้ใช้ ได้ดังนี้

#### 3.2.1 มีความเรียบง่าย

ในการเขียนคู่มือที่ทำให้ผู้ใช้เข้าใจได้อย่างรวดเร็ว นั้น ควรเขียนคู่มือบนพื้นฐานแห่งความเรียบง่าย กล่าวคือ ในการอธิบายสิ่งต่าง ๆ ที่อยู่ในคู่มือควรจะสั้นและเรียบง่าย คู่มือที่มีความเรียบง่ายจะทำให้ผู้พัฒนาโปรแกรมเกิดความพึงพอใจ และใช้งานคู่มือเฟรมเวิร์กได้อย่างมีประสิทธิภาพ แต่อย่างไรก็ตามความเรียบง่ายนี้ก็ไม่ควรจะง่ายเกินไป เพราะอาจทำให้ผู้ใช้เกิดความสับสนได้ เช่น การทำเว็บเพจ ลิงค์จะทำให้ผู้ใช้เข้าถึงส่วนอื่น ๆ ได้อย่างรวดเร็ว และง่ายตาย แต่ความง่ายก็อาจทำให้ผู้ใช้หลงทางได้ว่าขณะนี้แล้วผู้ใช้กำลังทำงานอยู่ในส่วนใด เป็นต้น ดังนั้นจะเห็นว่าความง่ายไม่สามารถทำให้เกิดได้ในทุกส่วน ในความเรียบง่ายจะต้องซึ่งนำหน้ากับความง่ายกับองค์ประกอบด้านอื่นด้วย เช่น ความสามารถในการทำงาน หรือลักษณะของผู้ใช้งาน เป็นต้น

#### 3.2.2 ลดภาระความทรงจำระยะสั้น

เนื่องจากมนุษย์มีข้อจำกัดเกี่ยวกับความจำระยะสั้น จึงต้องการจะลดสิ่งที่อยู่ในความจำของตนออกให้เกิดความโล่งใจว่าตนได้ทำงานเสร็จสมบูรณ์ ดังนั้นผู้วิจัยมีความเห็นว่าการนำเสนอเนื้อหาในคู่มือการใช้งานเฟรมเวิร์กควรจะเสนอในแนวทางที่ผู้ใช้สามารถอ่านเนื้อหาไปที่ละน้อย และสามารถอ่านเรื่องที่ต่อเนื่องกันได้เรื่อย ๆ ตามลำดับมากกว่าจะอ่านเรื่องใหญ่เรื่องเดียว ทั้งนี้เพราะเมื่อผู้อ่านได้อ่านเรื่องย่อยๆ จะช่วยทำให้ผู้ใช้เห็นความก้าวหน้าของตัวเองและรู้สึกโล่งใจเมื่อทำความเข้าใจในแต่ละส่วนได้สำเร็จ

#### 3.2.3 สร้างเจตคติที่ดีให้กับผู้ใช้

ดร.จรณิต แก้วกั้งวาล [25] ได้กล่าวไว้ว่า เจตคติของผู้ใช้มีผลต่อการเรียนรู้ของมนุษย์ ซึ่งความวิตกกังวลจะทำให้ความจำในหน่วยความจำระยะสั้นลดลง ผู้วิจัยจึงมีความเห็นว่าการเขียน

คู่มือเฟรมเวิร์กควรสร้างเจตคติที่ดีให้แก่ผู้ใช้งานเฟรมเวิร์ก กล่าวคือ ในการเขียนคู่มือการใช้งานเฟรมเวิร์ก ควรทำให้ผู้ใช้สามารถหาคู่มือได้อย่างง่าย สะดวก และไม่ทำให้ผู้ใช้รู้สึกว่าถูกควบคุมมากเกินไป การจะทำให้ผู้ใช้ลดความกังวลลงอาจทำได้โดย การให้คำอธิบายที่ชัดเจน ใช้คำพูดที่ผู้ใช้คุ้นเคยและผู้ใช้สามารถทำตามได้ไม่ยากนัก ซึ่งเมื่อผู้ใช้ลดความกังวลลงแล้ว พฤติกรรมการใช้งานจะมีคุณภาพดีขึ้นตามมาด้วย

### 3.2.4 มีความคงที่ในการนำเสนอ

การนำเสนอลำดับขั้นตอนอย่างเป็นระบบ มีความคงที่ในรูปแบบในการนำเสนอ เช่น ลักษณะในหน้าจอ เมนูหรือรายการคำสั่งที่แสดงควรจัดวาง ณ ตำแหน่งเดิม เป็นต้น จะทำให้ผู้ใช้สามารถจัดระบบความจำได้ และทำความเข้าใจในเนื้อหาได้โดยง่าย และไม่สับสน

### 3.2.5 มีทางเลือกในการเข้าถึงข้อมูล

เมื่อผู้ใช้เกิดความชำนาญในการใช้มากขึ้น หรือสำหรับผู้ใช้ที่มีความสามารถสูงอยู่แล้ว ย่อมต้องการจะกระโดดข้ามไปยังส่วนของเนื้อหาที่ต้องการได้ทันที ดังนั้นการเขียนคู่มือการใช้งานเฟรมเวิร์กควรทำให้ผู้ใช้สามารถเลือกเส้นทางการเข้าถึงข้อมูลได้ เช่น การใช้ลิงค์เข้ามาช่วยให้สามารถไปในส่วนที่ต้องการได้โดยไม่ต้องผ่านเนื้อหาที่เป็นลำดับตั้งแต่ต้นจนจบ เป็นต้น การเขียนคู่มือเฟรมเวิร์กให้มีทางเลือกผู้วิจัยแนะนำให้ใช้เว็บเบราว์เซอร์ เนื่องจากเป็นซอฟต์แวร์ที่มีความสามารถในการทำเอกสารข้อความหลายมิติ

## 3.3 หลักการออกแบบส่วนติดต่อกับผู้ใช้

### 3.3.1 หลักการนำเสนอข้อมูลบนหน้าจอ

หน้าจอถือเป็นส่วนสำคัญที่เป็นส่วนติดต่อระหว่างผู้ใช้กับคู่มือเฟรมเวิร์กบนสื่ออิเล็กทรอนิกส์ เช่น หน้าจอที่แน่นมากอาจทำให้ผู้ใช้เกิดความหงุดหงิด อึดอัด หรือหน้าจอที่ขาดความคงที่อาจทำให้ผู้ใช้สับสนกับตำแหน่งของข้อมูลได้ง่าย เป็นต้น การออกแบบหน้าจอจึงถือเป็นสิ่งหนึ่งที่ต้องให้ความสำคัญในการออกแบบส่วนติดต่อกับผู้ใช้ สมิตและมอยเซอร์ [21] ได้สรุปเป้าหมายของหลักการจัดแสดงข้อมูลบนหน้าจอไว้ 5 ประการ ดังนี้

#### 3.3.1.1 ความคงที่ในการนำเสนอ

ผู้ออกแบบจำเป็นต้องกำหนดนิยามคำศัพท์ การใช้ตัวย่อ และรูปแบบของภาษา การใช้ภาษาควรมีความคงที่ไม่เปลี่ยนแปลงไปมาจนทำให้ผู้ใช้เกิดความสับสน

#### 3.3.1.2 การเสนอข้อมูลให้ผู้ใช้เข้าใจได้

รูปแบบการนำเสนอควรเป็นสิ่งที่ผู้ใช้คุ้นเคย และสามารถเชื่อมโยงการกระทำหรือสอดคล้องกับส่วนที่ผู้ใช้กำลังทำความเข้าใจได้เป็นอย่างดี นอกจากนี้ความเรียบร้อยในการ

นำเสนอ ได้แก่ การตั้งคอลัมน์ การพิมพ์ตัวอักษรขีดซ้าย การใส่ตัวเลขขีดขวา การตั้งเรียงจุดทศนิยมให้ตรงกัน การเว้นช่วงจังหวะระหว่างวรรคหรือบรรทัด ก็ควรนำมาพิจารณาให้เกิดความเหมาะสมในการอ่านด้วย

### 3.3.1.3 การลดปริมาณสิ่งที่ผู้ใช้ต้องจดจำ

ในการนำเสนอหนึ่งหน้าจอก็ไม่ควรจะมีข้อมูลหรือสิ่งประกอบอื่นที่ผู้ใช้ต้องจดจำมากมายจนเกินขีดจำกัดของหน่วยความจำระยะสั้นของมนุษย์ที่จะสามารถจดจำได้ การนำเสนอจากหน้าจอหนึ่งไปสู่อีกหน้าจอหนึ่งก็ควรมีส่วนที่แจ้งให้ผู้ใช้ทราบว่าตนอยู่ในตำแหน่งใด และควรจะลดขั้นตอนที่ไม่จำเป็นออก เพื่อที่ผู้ใช้จะได้ไม่ลืมคำสั่งที่ตนกำลังจะดำเนินการได้

### 3.3.1.4 ความสมดุลระหว่างการนำเสนอและการกำหนดข้อมูลนำเข้า

รูปแบบและโครงสร้างของข้อมูลควรมีความชัดเจน ให้ผู้ใช้เห็นความสัมพันธ์ของข้อมูลนำเข้าและผลลัพธ์ที่ได้จากการดำเนินการ

### 3.3.1.5 ความคล่องตัวของผู้ใช้ในการควบคุมการนำเสนอ

ผู้ใช้สามารถเลือกดูในส่วนที่ต้องการได้โดยไม่เสียเวลา

## 3.3.2 การกระตุ้นความสนใจของผู้ใช้

โดยหลักการทั่วไป การนำเสนอข้อมูลต่อผู้ใช้ควรจะต้องมีความชัดเจนเข้าใจง่าย แต่ในบางครั้งการนำเสนอก็ต้องกระตุ้นให้ผู้ใช้เกิดความสนใจเป็นพิเศษกับเงื่อนไขหรือข้อมูลที่มีความสำคัญ ดร.จรณ์ิต แก้วกัญวาล [25] ได้กล่าวถึงเทคนิคการนำเสนอไว้ดังนี้

3.3.2.1 ระดับความลึก (intensity) ควรจัดระดับความลึกของข้อมูลไม่เกิน 2 ระดับ เนื่องจากการมีหัวข้อย่อยมากกว่า 2 ระดับจะก่อให้เกิดความยุ่งยากแก่ผู้ใช้ในการจดจำและเข้าใจ

3.3.2.2 การเน้นความสำคัญ (marking) สามารถเลือกใช้การขีดเส้นใต้ ตัวเรืองแสง ตีเส้นกรอบล้อมคำ ชี้อักษร ดอกจัน ขีดเครื่องหมายพิเศษต่างๆ ข้างหน้าข้อความได้ แต่รูปแบบที่ใช้ควรจะต้องมีความคงที่ตลอดการนำเสนอ สำหรับการเขียนคู่มือเฟรมเวิร์กบนเว็บ-เบราว์เซอร์จะมีข้อยกเว้นบางรูปแบบที่ไม่ควรใช้ โดยจะกล่าวในเรื่องส่วนติดต่อกับผู้ใช้บนเว็บ-เบราว์เซอร์ต่อไป

3.3.2.3 ขนาดและแบบอักษร (size and fonts) ไม่ควรใช้ตัวอักษรมากกว่า 4 ขนาดในหน้าเดียวกัน และไม่ควรใช้ตัวอักษรที่แตกต่างกันมากกว่า 3 แบบ เพราะนอกจากจะไม่ทำให้เกิดความสวยงามเพิ่มขึ้นแล้วอาจจะทำให้ผู้ใช้เกิดความสับสนได้

3.3.2.4 การสลับสีและการกระพริบ (inverse video and blinking) ควรเลือกใช้ตัวอักษรปกติ เช่น ตัวอักษรสีขาวบนพื้นดำ กับตัวสลับสี เช่น ตัวอักษรสีดำบนพื้นขาว ตามจังหวะ

ที่เหมาะสมบนหน้าจอ การใช้ตัวกระทบไม่ควรมี ถ้ามีก็ไม่ควรจะมีมากเพราะจะทำให้เสียสายตา ความเร็วในการกระทบไม่ควรเกิน 2-4 เฮิร์ตซ์

3.3.2.5 การใช้สี ไม่ควรใช้มากกว่า 4 สี ในหน้าจอเดียวกัน หากจะใช้มากกว่า 4 สีก็ต่อเมื่อต้องการเรียกร้องความสนใจจากผู้ใช้อย่างเป็นพิเศษ

ทั้งนี้การใช้เทคนิคต่าง ๆ ก็ไม่ควรมีมากเกินไปจนทำให้ผู้ใช้เกิดความสับสน โดยเฉพาะผู้ใช้ระดับต้น จะต้องการการนำเสนอที่เรียบง่ายมีโครงสร้างแน่นอน แต่ในทางกลับกัน ผู้ใช้ที่เป็นผู้เชี่ยวชาญจะต้องการคำอธิบายที่ไม่ยืดเยื้อ หรือได้รับการกระตุ้นเตือนโดยไม่จำเป็น

### 3.3.3 หลักการใช้สีในการนำเสนอ

การใช้สีในการสร้างความประทับใจให้กับผู้ใช้ ทำให้ผู้ใช้สามารถใช้งานระบบได้อย่างมีประสิทธิภาพ แต่การใช้สีที่ผิดวิธีก็อาจมีผลในทางลบได้ ดร.จรณิต แก้วกังวาล [25] ได้กล่าวเกี่ยวกับเกณฑ์พื้นฐานการใช้สีในการจัดจอภาพ ซึ่งสามารถนำมาปรับใช้กับการจัดทำคู่มือการใช้งานเฟรมเวิร์กไว้ ดังนี้

#### 3.3.3.1 ไม่ใช้สีฟุ้งเฟ้อ

การใช้สีควรใช้อย่างมีความหมายว่าสีใดหมายถึงอะไร และคงสีนั้นตลอด เช่น สีฟ้า แสดงข้อความที่เป็นหมายเหตุ สีแดงเป็นการแสดงคำเตือน เป็นต้น

#### 3.3.3.2 จำกัดจำนวนสี

การใช้สีที่หลากหลายอาจทำให้เกิดความเข้าใจผิดได้ เช่น ในเมนูเดียวกันถ้าผู้ออกแบบใช้สีหลายสี ก็จะทำให้ผู้ใช้เข้าใจไปเองว่าตัวเลือกที่มีสีเดียวกัน จะเป็นตัวเลือกที่มีความสัมพันธ์กัน เป็นต้น ในการออกแบบ ไม่ควรใช้สีมากกว่า 4 สีในหน้าจอเดียวกัน และไม่มากกว่า 7 สี สำหรับการแสดงหน้าจอที่ต่อเนื่องกันไป

#### 3.3.3.3 ใช้สีในการสร้างรูปแบบ

บนจอภาพที่นำเสนอสิ่งต่าง ๆ มากมายจนเต็มหน้าจอ ควรจะใช้สีในการจัดกลุ่มสิ่งที่ต้องการนำเสนอออกเป็นชุด

#### 3.3.3.4 ใช้สีที่ตรงกับความคาดหวังของผู้ใช้

การออกแบบจะมีประสิทธิภาพถ้าผู้ออกแบบใช้สีที่ผู้ใช้ทั่วไปคุ้นเคย หรือตั้งความคาดหวังเอาไว้แล้ว เช่น การใช้สีน้ำเงินกับการเชื่อมโยงข้อความหลายมิติ เป็นต้น

#### 3.3.3.5 เลือกจัดสีที่นำเสนออย่างระมัดระวัง

การนำเสนอสีที่เป็นคู่สีตรงกันข้ามบนหน้าจอเดียวกันอาจไม่เป็นผลดีต่อตัวผู้ใช้ เช่น ใช้ตัวหนังสือสีฟ้าบนพื้นหลังสีแดง เป็นต้น เพราะจะทำให้ผู้ใช้ต้องเพ่งสายตามากขึ้น เนื่องจากสีฟ้าจะถูกดูดกลืนหายไปบนสีแดงซึ่งเข้าตามากกว่า นอกจากนี้ยังมีคู่สีตรงข้ามที่ต้องระวังอีก คือ

เหลือกับม่วง และชมพูบานเย็นกับเขียว สีที่คล้ายกันเกินไปก็อาจจะเป็นปัญหาในการอ่านเช่นกัน ทั้งนี้ต้องดูความเหมาะสมในการสื่อความหมายของแต่ละสีที่ใช้ด้วย

### 3.4 หลักการออกแบบส่วนติดต่อกับผู้ใช้บนเว็บเบราว์เซอร์

การออกแบบการจัดทำคู่มือบนสื่ออิเล็กทรอนิกส์ ได้แก่ เว็บเบราว์เซอร์ สามารถนำหลักจิตวิทยาทั่วไปมาใช้ในการออกแบบส่วนติดต่อกับผู้ใช้ได้เช่นกันแต่ไม่ทั้งหมด เนื่องจากเว็บเบราว์เซอร์มีคุณสมบัติและความสามารถในการใช้งานที่แตกต่างจากหน้าจอของการทำงาน โปรแกรมปกติหลายประการ ธวัชชัย ศรีสุเทพ [26] ได้กล่าวเกี่ยวกับเทคนิคจัดทำเว็บเพจ ซึ่งมีหลักการในการออกแบบส่วนติดต่อกับผู้ใช้บนเว็บเบราว์เซอร์ที่สามารถนำมาใช้กับการเขียนคู่มือเฟรมเวอร์กได้ ดังนี้

#### 3.4.1 การออกแบบให้หน้าเว็บมีความยาวที่เหมาะสม

ผู้ใช้สามารถใช้ประโยชน์จากเว็บได้ 2 ทาง คือ ใช้เป็นสื่อโดยตรงในการอ่านบนจอ และใช้ เป็นสื่อกลางในการพิมพ์บนกระดาษเพื่อให้อ่านในภายหลัง ดังนั้นในการกำหนดความยาวของหน้าเว็บก็จะขึ้นอยู่กับการใช้งาน โดยจะสร้างเว็บเพจให้มีขนาดสั้น สำหรับหน้าที่คาดว่าจะถูกอ่านบนหน้าจอ หรือหน้าที่มีกราฟิกขนาดใหญ่มาก และควรจะสร้างเว็บเพจให้มีขนาดยาวเพื่อความ สะดวกในการพิมพ์หรือดาวน์โหลด และสำหรับโครงสร้างของเนื้อหาที่ไม่ควรถูกตัดแบ่งออกจากกัน

#### 3.4.2 ส่วนประกอบของเว็บเพจ

ธวัชชัย ศรีสุเทพ [26] ได้แบ่งส่วนประกอบของเว็บเพจแต่ละหน้าออกเป็น 3 ส่วน ได้แก่

##### 3.4.2.1 ส่วนหัว (Header)

เป็นส่วนที่จะดึงดูดความสนใจของผู้ใช้ให้ติดตามเนื้อหาที่เหลือภายในหน้าได้ ดังนั้นในส่วนนี้จะใช้เป็นตัวบอกหัวข้อหลัก หรือชื่อของเนื้อหาในหน้านั้น โดยถ้าหน้านี้มีกราฟิกขนาดใหญ่ อยู่ ขณะรอการแสดงผลของกราฟิกนั้น การมีส่วนหัวจะสามารถดึงดูดความสนใจของผู้ใช้ให้ไม่เกิดความเบื่อหน่ายได้

##### 3.4.2.2 ส่วนเนื้อหา (Body)

ควรจะจัดเนื้อหาให้มีความเป็นระเบียบ เช่น การจัดตัวหนังสือให้ชิดขอบด้านซ้าย เป็นต้น เพื่อการมองหาข้อมูลที่รวดเร็ว พร้อมทั้งทำการจัดรูปแบบตัวอักษรให้เหมาะสม เพื่อให้เนื้อหาดู น่าสนใจและอ่านได้อย่างสะดวก

### 3.4.2.3 ส่วนท้าย (Footer)

เป็นส่วนที่อยู่ด้านล่างสุดของหน้าเว็บ อาจใช้เป็นส่วนสรุปของแต่ละหน้า หรือใช้เป็นส่วนเชื่อมโยงที่ผู้ใช้สามารถเข้าสู่เนื้อหาในส่วนอื่นได้อย่างสะดวก

### 3.4.3 การเลือกใช้สีสำหรับเว็บ

สี เป็นสิ่งที่กำหนดบรรยากาศและความรู้สึกโดยรวมให้กับผู้ใช้ การเลือกใช้สีที่เหมาะสม จะช่วยในการสื่อความหมายให้เนื้อหา และเพิ่มความสวยงามให้กับเว็บได้ แต่ในทางกลับกัน ถ้าการใช้สีไม่เหมาะสม อาจสร้างความลำบากในการอ่านหรือรบกวนสายตาของผู้ใช้ รวมทั้ง อาจจะสื่อความหมายที่ไม่ถูกต้องได้ หลักการใช้สีของเว็บก็คล้ายกับหลักการใช้สีของการออกแบบ หน้าจอทั่วไป แต่ก็มีข้อแม้ในการเลือกสีสำหรับหน้าเว็บเช่นกัน เนื่องจากการแสดงออกของสีในเว็บจะถูกควบคุมโดยระบบสีของเบราว์เซอร์ ซึ่งเมื่อใดที่เว็บเบราว์เซอร์จำเป็นต้องแสดงสีที่ไม่มีอยู่ในชุดสีระบบ เว็บเบราว์เซอร์จะสร้างสีขึ้นมาทดแทน ซึ่งอาจเป็นผลให้ภาพหรือบริเวณของสีนั้นมีลักษณะเป็นจุดกระจายอย่างไม่เป็นระเบียบ โดยธวัชชัย ศรีสุเทพ [26] ได้กล่าวว่าในการใช้สีกับเว็บนั้นควรใช้ชุดสีสำหรับเว็บมีทั้งหมด 216 สี ซึ่งสีเหล่านี้ ผู้ใช้ไม่จำเป็นต้องจำค่าของสีเพื่อจะนำมาใช้กับเว็บ เนื่องจากมีเครื่องมือหนึ่งที่เรียกว่า วิซิบอน (VisiBone) [24] ที่ช่วยให้ผู้สร้างเว็บได้เห็นและเลือกสีที่เข้ากันได้โดยง่าย

### 3.4.4 การจัดรูปแบบตัวอักษรสำหรับเว็บ

ตัวอักษร มีหน้าที่สื่อข้อความถึงผู้ใช้ โดยตัวอักษรแต่ละชนิดจะให้อารมณ์และความรู้สึกที่แตกต่างกัน โดยในเอกสารเว็บนั้นตัวอักษรถือเป็นองค์ประกอบส่วนใหญ่ของแต่ละหน้า ดังนั้นจึงควรให้ความสำคัญกับรูปแบบอักษรด้วย

#### 3.4.4.1 การใช้รูปแบบตัวอักษร

ธวัชชัย ศรีสุเทพ [26] ได้แบ่งรูปแบบตัวอักษรออกเป็น 2 ประเภท คือ

- 1) ตัวอักษรที่สัมพันธ์กับรูปร่าง คือ ตัวอักษรที่มีพื้นที่ตามแนวนอนไม่เท่ากัน โดยจะขึ้นกับรูปร่างของตัวอักษรนั้น ชนิดตัวอักษรที่อยู่ในรูปแบบนี้ได้แก่ Times New Roman, Helvetica Arial, AngsanaUPC และ Cordia New เป็นต้น ตัวอักษรชนิดนี้มักใช้ในส่วนที่เป็น ตัวหนังสือส่วนใหญ่ เช่น ส่วนรายละเอียดของเนื้อหา รายการ เป็นต้น เนื่องจากตัวอักษรที่มีขนาดสัมพันธ์กับรูปร่างนี้อ่านง่ายและใช้พื้นที่หน้ากระดาษน้อยกว่าแบบที่มีความกว้างคงที่

2) ตัวอักษรที่มีขนาดคงที่ คือ ตัวอักษรที่ใช้พื้นที่ในแนวนอนเท่ากันหมด เป็นตัวอักษรที่มีรูปแบบที่เรียบง่ายคล้ายตัวพิมพ์ดีด ชนิดตัวอักษรในรูปแบบนี้ได้แก่ Courier, Monaco, Andale Mono, Courier Mono Thai และ ThaiMono เป็นต้น ตัวอักษรเหล่านี้จะใช้กับส่วนที่แสดงคำสั่งของภาษา

#### 3.4.4.2 ความสะดวกในการอ่าน

การใช้ตัวพิมพ์ใหญ่หรือพิมพ์เล็กในภาษาอังกฤษมีผลอย่างมากกับการอ่าน เนื่องจากผู้อ่านจะจำแนกคำต่าง ๆ ออกด้วยรูปร่างโดยรวม ไม่ใช่การมองตัวอักษรแต่ละตัวแล้วนำมารวมกันเป็นคำในภายหลัง ด้วยเหตุนี้ จึงควรจะใช้ตัวอักษรใหญ่เฉพาะกับตัวอักษรแรกของประโยค และหลีกเลี่ยงการใช้ตัวพิมพ์ใหญ่ทั้งหมด เพราะจะทำให้อ่านยากเนื่องจากคำในรูปแบบตัวพิมพ์ใหญ่ทั้งหมดไม่มีโทนขึ้นลงทำให้ลดส่วนที่สะดุดตาลงไป ส่วนการพิมพ์ตัวเล็กทั้งประโยคจะสร้างความรู้สึกที่ไม่เป็นทางการและความไม่สมบูรณ์ของเนื้อหา และไม่ควรถัดพิมพ์ใหญ่ในตัวอักษรแรกของทุกคำในประโยค เพราะจะทำให้รบกวนรูปแบบโดยรวมของคำ ส่งผลให้อ่านยากขึ้น

#### 3.4.4.3 การจัดตำแหน่ง

การจัดตำแหน่งให้ชิดขอบด้านซ้ายโดยปล่อยให้ขอบด้านขวามีลักษณะไม่สม่ำเสมอ จะให้ความรู้สึกที่ไม่เป็นทางการ แต่ก็อ่านได้ง่ายกว่าการชิดขอบทางขวา ส่วนการจัดตำแหน่งตัวอักษรให้ชิดทั้งขอบซ้ายและขอบขวาจะทำให้มีความรู้สึกเป็นทางการขึ้น แต่ก็ต้องระวังการเกิดช่องว่างรบกวนความสะดวกในการอ่าน เพราะช่วงห่างของตัวอักษรจะเปลี่ยนไป ส่วนการจัดกึ่งกลางจะให้ได้กับข้อมูลที่มีปริมาณไม่มาก และต้องการรูปแบบที่เป็นทางการ

#### 3.4.4.4 การกำหนดความยาวบรรทัด

ถ้าแต่ละบรรทัดมีความยาวมาก ผู้อ่านต้องเคลื่อนสายตาจากปลายบรรทัดไปยังส่วนของต้นบรรทัดใหม่ทำให้ไม่สะดวกในการอ่าน ในขณะที่บรรทัดที่สั้นจนเกินไป ข้อความก็จะถูกแยกจากกันบ่อยซึ่งเป็นการรบกวนผู้อ่านเช่นกัน ธวัชชัย ศรีสุเทพ [26] ได้ทำการศึกษาข้อมูลเกี่ยวกับความยาวของบรรทัด พบว่าสายตามนุษย์สามารถกวาดไปในพื้นที่ประมาณ 4 นิ้วได้โดยไม่ต้องขยับศีรษะ เมื่อเทียบเป็นพื้นที่ในเว็บแล้วก็คือ ประมาณ 350 ถึง 500 พิกเซล

#### 3.4.4.5 การกำหนดขนาดของตัวอักษร

ขนาดตัวอักษรสามารถใช้ในการสร้างลำดับชั้นความสำคัญของข้อมูลได้ กล่าวคือ เนื้อหาที่มีความสำคัญมากควรมีขนาดใหญ่และเห็นชัดกว่าเนื้อหาส่วนอื่น ทำให้ผู้อ่านเข้าใจได้อย่างรวดเร็ว การที่ผู้ใช้สามารถแยกได้ว่าส่วนใดคือหัวเรื่อง ส่วนใดคือหัวข้อย่อยและรายละเอียด จะช่วยให้ผู้ใช้รับข้อมูลได้อย่างสะดวก ชัดเจน และรวดเร็ว ในบางครั้งอาจใช้ตัวอักษรขนาดใหญ่กว่า

ปกติในการเริ่มต้นประโยคแรกของย่อหน้าในการดึงดูดความสนใจของผู้อ่าน และในสวนรายละเอียดของเนื้อหาควรจะใช้ตัวอักษรขนาดเล็ก เนื่องจากจะทำให้บรรจุนี้อหาได้มากในพื้นที่ที่มีอยู่ และช่วยสร้างความต่อเนื่องของเนื้อหาที่ค่อนข้างยาวได้ดีกว่าตัวอักษรขนาดใหญ่

#### 3.4.4.6 การเน้นข้อความให้เด่นชัด

การสร้างความเด่นชัดให้กับข้อความสามารถทำได้หลายวิธี ได้แก่ การทำตัวเข้ม การทำตัวเอียง การขีดเส้นใต้ เป็นต้น ถ้านำมาใช้อย่างเหมาะสมจะสร้างความน่าสนใจให้กับเนื้อหา และส่งเสริมความสวยงามให้กับเว็บเพจได้อีกด้วย โดยการทำตัวเอียงไม่เหมาะสมจะมาใช้เพื่อเน้นข้อความบนเว็บเพจ เนื่องจากจุดบนจอภาพมีความละเอียดต่ำกว่าการพิมพ์มาก ดังนั้นเวลาแสดงผลตัวเอียงจึงมีลักษณะเป็นหยักไม่น่าอ่าน เช่นเดียวกับการเน้นข้อความด้วยการขีดเส้นใต้ที่สามารถใช้เน้นข้อความสำคัญในเอกสารสิ่งพิมพ์ได้ดี แต่ถ้าสำหรับเว็บเพจแล้วการขีดเส้นใต้มีความหมายพิเศษซึ่งแสดงถึงการเชื่อมโยงหลายมิติ ดังนั้นจึงไม่ควรใช้การขีดเส้นใต้ในความหมายอื่น เนื่องจากอาจทำให้ผู้ใช้เกิดความสับสน

#### 3.4.4.7 การใช้สีกับตัวอักษร

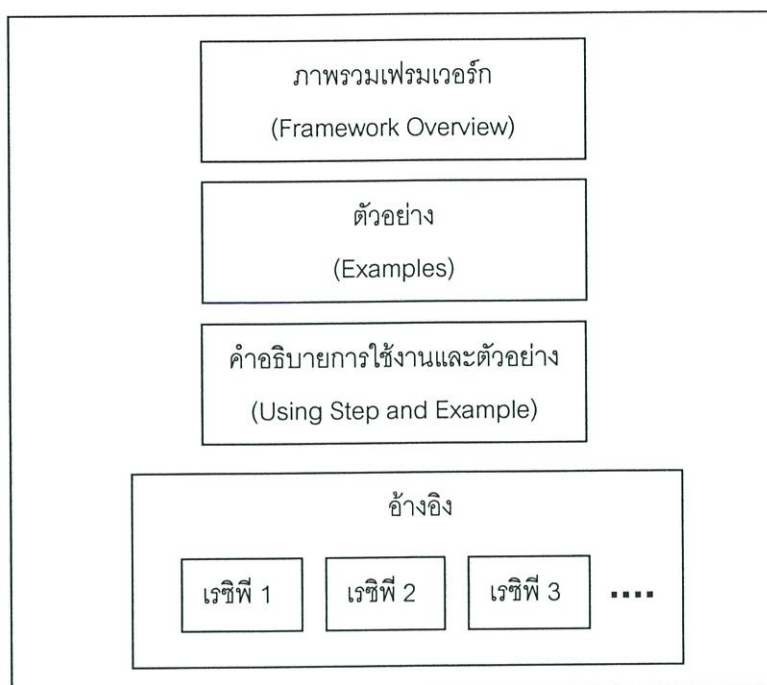
การใช้สีจะช่วยตกแต่งให้โครงสร้างและรูปแบบของตัวอักษรดูดีขึ้น สีได้มีความแตกต่างหรือเด่นชัดมาก ย่อมหมายถึงการเน้นที่มากขึ้นเท่านั้น ในการใช้สีตัวอักษรควรจะใช้อย่างมีความหมายและสม่ำเสมอ เนื่องจากผู้ใช้สามารถเรียนรู้ความแตกต่างของสีได้อย่างรวดเร็ว การใช้สีที่ต่างกันในระดับชั้นความสำคัญของเนื้อหาจะทำให้ผู้ใช้คุ้นเคยและใช้งานได้อย่างสะดวก การใช้สีมีข้อควรระวังสำหรับการแสดงผลทางเครื่องพิมพ์ โดยในส่วนข้อความที่ตั้งใจให้ผู้ใช้แสดงผลทางเครื่องพิมพ์ได้ ควรจะใช้สีที่ทำให้ผู้ใช้เห็นได้อย่างชัดเจน

## รูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์ก

ในบทนี้จะกล่าวถึงรายละเอียดของการออกแบบรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์กสำหรับผู้พัฒนาโปรแกรมประยุกต์ ซึ่งจะนำเอาส่วนดีของการเขียนคู่มือเฟรมเวิร์กที่มีอยู่มาจัดทำเป็นรูปแบบของคู่มือการใช้งานเฟรมเวิร์กแบบใหม่ ให้มีรูปแบบที่ชัดเจนและเหมาะกับการใช้งานของผู้พัฒนาโปรแกรมประยุกต์ จากความต้องการข้อมูลคู่มือพื้นฐานของผู้พัฒนาโปรแกรมประยุกต์ในมุมมองริยูสเคส [4] [7] เห็นได้ว่าการจัดทำคู่มือแบบผสมผสานที่ใช้แนวทางการจัดทำคู่มือที่อธิบายภาพรวม และคู่มือแบบคู่มือกับเรซีพี้ สามารถตอบสนองความต้องการในส่วนขอเนื้อหาให้ผู้พัฒนาโปรแกรมประยุกต์ใช้งานคู่มือเฟรมเวิร์กประกอบในการพัฒนาเป็นโปรแกรมตามความเหมาะสมได้ แต่อย่างไรก็ตามถ้าจะกล่าวถึงรูปแบบของคู่มือการใช้งานเฟรมเวิร์กแบบคู่มือคู่นั้นจะเห็นได้ว่ายังไม่มีการวิจัยใดได้กล่าวถึงรูปแบบของคู่มือชนิดนี้ได้อย่างชัดเจน ผู้เขียนแต่ละคนก็มีรูปแบบการเขียนคู่มือคู่มือที่แตกต่างกันออกไป ดังนั้นในการจัดทำรูปแบบคู่มือการใช้งานเฟรมเวิร์กในวิทยานิพนธ์ฉบับนี้ จะนำเสนอส่วนประกอบและการจัดรูปแบบของคู่มือการใช้งานเฟรมเวิร์กให้มีความชัดเจนมากขึ้น โดยจะอธิบายรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์กออกเป็น 2 ส่วน คือ ส่วนแรกจะอธิบายถึงส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์กเพื่อให้เกิดความครบถ้วนในด้านเนื้อหา สร้างความเป็นเอกภาพให้กับการเขียนคู่มือการใช้งานเฟรมเวิร์ก และส่วนที่สองจะกล่าวถึงการออกแบบส่วนติดต่อกับผู้ใช้ เพื่อให้ผู้ใช้สามารถเรียนรู้การใช้งานเฟรมเวิร์กได้เร็วขึ้น สร้างความพอใจให้กับผู้ใช้และไม่ทำให้ผู้ใช้เกิดความเบื่อหน่ายในการใช้งานคู่มือ โดยผู้พัฒนาคู่มือการใช้งานเฟรมเวิร์กสามารถนำรูปแบบของคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยไปใช้ในการเขียนคู่มือเฟรมเวิร์กทั้งในภาษาไทยและภาษาอังกฤษ

### 4.1 ส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก

การจัดทำคู่มือการใช้งานเฟรมเวิร์กนี้จะจัดให้มีส่วนประกอบทางด้านเนื้อหาให้ครบถ้วนตามความต้องการของผู้พัฒนาโปรแกรมประยุกต์ในมุมมองของริยูสเคส [4] [7] โดยเนื้อหาหลักของคู่มือจะประกอบด้วยภาพรวมของเฟรมเวิร์ก คำอธิบายการใช้งานเฟรมเวิร์กพร้อมด้วยตัวอย่างโปรแกรมประยุกต์ที่พัฒนาขึ้นจากเฟรมเวิร์ก และส่วนอ้างอิงที่จะนำผู้ใช้ไปสู่ส่วนประกอบหรือคลาสต่างๆ ที่เฟรมเวิร์กเตรียมไว้ให้สำหรับการพัฒนาเป็นโปรแกรมประยุกต์ และเนื่องจากตัวอย่างถือเป็นส่วนสำคัญที่ผู้พัฒนาโปรแกรมประยุกต์มักมองหาเป็นอย่างแรก ทำให้รูปแบบการเขียนคู่มือเฟรมเวิร์กจากการวิจัยนี้ แทรกส่วนที่เป็นตัวอย่างลงไปให้ชัดเจนก่อนการอธิบายวิธีการใช้งาน เพื่อให้ผู้ใช้ได้เห็นภาพรวมในมุมมองของการใช้งานที่มากขึ้น ดังรูปที่ 4.1



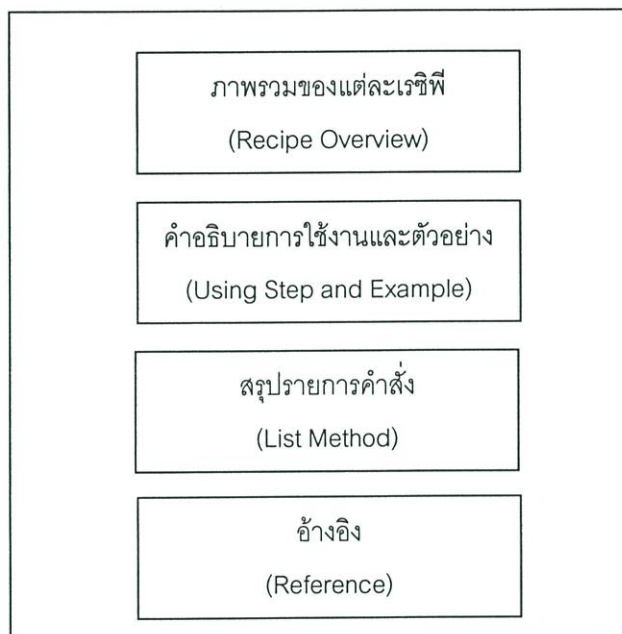
รูปที่ 4.1 แสดงส่วนประกอบของเนื้อหาหลักในคู่มือเฟรมเวิร์ก

ในส่วนของตัวอย่างที่แทรกลงไปใหม่นั้น ไม่จำเป็นต้องมาจากตัวอย่างที่ใช้ในการอธิบายวิธีการใช้งานเพียงอย่างเดียว ผู้เขียนอาจแทรกตัวอย่างใหม่ที่มีส่วนประกอบใหม่ ๆ ลงไป เช่น ตัวอย่างโปรแกรมประยุกต์ที่นำส่วนประกอบของเฟรมเวิร์กหลาย ๆ ส่วนผสมผสานกัน เป็นต้น ซึ่งตัวอย่างเหล่านี้ควรมีการเรียงลำดับจากง่ายไปหายากและควรจัดหมวดหมู่ของตัวอย่างให้เป็นระเบียบโดยไม่ทำให้ผู้พัฒนาโปรแกรมประยุกต์สับสน ดังตัวอย่างคู่มือเฟรมเวิร์กของงานวิจัยในรูป 3 (ภาคผนวก ง.)

และในส่วนสุดท้ายของส่วนประกอบหลักในคู่มือเฟรมเวิร์กก็คือ ส่วนอ้างอิงที่จะนำผู้พัฒนาโปรแกรมประยุกต์เข้าสู่ส่วนประกอบต่าง ๆ ของเฟรมเวิร์ก ซึ่งในส่วนอ้างอิงแต่ละหน้านี้สามารถเรียกอีกอย่างหนึ่งได้ว่าเป็นเรขีพีหนึ่งในเฟรมเวิร์ก โดยในแต่ละเรขีพีของเฟรมเวิร์กจะแสดงรายละเอียดวิธีการใช้งานที่เฉพาะเจาะจงมากขึ้น ในการเรียงลำดับของเรขีพีในการเขียนคู่มือโดยทั่วไปมักจะเรียงลำดับการเขียนจากบนลงล่าง กล่าวคือ เริ่มอธิบายจากคลาสนามธรรมก่อนแล้วค่อยอธิบายไปถึงคลาสรูปธรรมที่สืบทอดจากคลาสนามธรรม ดังจะเห็นได้จากคู่มือของบริษัทซัมซุงโครซีสเต็มส์ในรูป 4 (ภาคผนวก จ.) ที่เริ่มอธิบายการใช้งานคลาสอินเทอร์เฟซก่อนและอธิบายคลาสที่อิมพลีเมนต์แล้วในภายหลัง แต่ในการใช้งานของผู้พัฒนาโปรแกรมประยุกต์ส่วนใหญ่มักไม่เป็นเช่นนั้น ผู้พัฒนาโปรแกรมประยุกต์ส่วนใหญ่มักมองหาสิ่งที่ต้องการจะใช้งานจากคลาสรูปธรรมที่มีมาให้แล้วก่อน ซึ่งถ้าไม่สามารถใช้งานตามความต้องการได้แล้วจึงจะไปอิมพลีเมนต์ในส่วนของคลาสนามธรรมต่อไป ดังนั้นการจัดทำคู่มือเฟรมเวิร์กจากงานวิจัยนี้จะ

เรียงลำดับเร시피ใหม่เป็นจากล่างขึ้นบน ดังรูป 4 เพื่อความสะดวกของผู้พัฒนาโปรแกรมประยุกต์ส่วนใหญ่

ส่วนรูปแบบในเร시피นั้น เร시피ทุกเร시피จะมีส่วนประกอบที่เหมือนกัน ได้แก่ ภาพรวมของแต่ละเร시피 คำอธิบายการใช้งานและตัวอย่าง สรุปรายการคำสั่งทั้งหมดในเร시피นั้น และส่วนอ้างอิงไปยังเร시피อื่นที่มีความสัมพันธ์กัน ดังรูปที่ 4.2



รูปที่ 4.2 แสดงส่วนประกอบของเนื้อหาในแต่ละเร시피

จากภาพจะเห็นว่าส่วนประกอบของการเขียนคู่มือการใช้งานเฟรมเวิร์กสามารถแบ่งได้อย่างชัดเจนออกเป็นดังรูปที่ 4.1 และ 4.2 โดยทั้งสองรูปนี้แตกต่างกันเพียงส่วนประกอบของเนื้อหาหลักในคู่มือเฟรมเวิร์กในรูป 4.1 จะแสดงครั้งเดียวในหน้าแรกๆ ที่ผู้ใช้สามารถเข้าถึงได้ และการรวมตัวอย่างที่มีในแต่ละเร시피ย่อยเข้าด้วยกัน ส่วนในรูป 4.2 จะมีรูปแบบในทุก ๆ เร시피หรือคำอธิบายในทุก ๆ คลาสที่เหมือนกัน ผู้วิจัยได้แสดงตัวอย่างของเนื้อหาในเร시피ไว้ในรูป 5 โดยรายละเอียดในเนื้อหาแต่ละส่วนสามารถอธิบายได้ ดังนี้

#### 4.1.1 ภาพรวม

จากงานวิจัยที่เกี่ยวกับการจัดทำคู่มือการใช้งานเฟรมเวิร์ก [19] [4] [6] ได้ให้ความสำคัญกับภาพรวมของเฟรมเวิร์กที่ทำให้ผู้ใช้เข้าใจพื้นฐานการทำงานโดยรวมของเฟรมเวิร์กได้ ซึ่งผู้วิจัยได้สรุปวัตถุประสงค์ของการเขียนภาพรวมไว้ดังนี้

##### 4.1.1.1 อธิบายโดเมนของปัญหา

##### 4.1.1.2 อธิบายคลาสและความสัมพันธ์ของคลาสภายในเฟรมเวิร์กโดยสังเขป

การอธิบายโดเมนของปัญหา คือ การอธิบายวัตถุประสงค์ของเฟรมเวิร์กว่าออกแบบเพื่อแก้ปัญหาอะไร โดยอาจจะยกตัวอย่างปัญหาหรือรูปแบบของปัญหาที่สามารถใช้เฟรมเวิร์กนี้ในการแก้ปัญหา การอธิบายโดเมนในส่วนนี้จะช่วยแก้ปัญหาการเลือกใช้งานที่ผิดได้ ส่วนการอธิบายคลาสและความสัมพันธ์ของคลาภายในเฟรมเวิร์ก เช่น การแนะนำคลาสในเฟรมเวิร์กว่าคลาสใดเป็นคลาสรูปธรรม คลาสใดที่เป็นคลาสนามธรรม และคลาสเหล่านั้นมีความสัมพันธ์กันอย่างไร เป็นต้น โดยคำอธิบายเหล่านี้จะช่วยให้ผู้พัฒนาได้เห็นภาพรวมของความสัมพันธ์ภายในเฟรมเวิร์กในเบื้องต้นได้ ในส่วนภาพรวมเฟรมเวิร์กอาจจะใช้ภาพ ตาราง หรือไดอะแกรมประกอบให้ผู้ใช้งานได้เห็นภาพความสัมพันธ์ของการใช้งานเฟรมเวิร์กได้ง่ายขึ้น ดังตัวอย่างการเขียนภาพรวมเฟรมเวิร์กในรูป 2 และตัวอย่างการเขียนภาพรวมการใช้งานของลิสต์อินเทอร์เฟซ (List Interface) ในจาวาคอลเลกชันเฟรมเวิร์ก (Java Collection Framework) ในรูปที่ 4.3

#### ขอบเขตและโดเมนของลิสต์อินเทอร์เฟซ

ใช้ในการเก็บข้อมูลที่มีลักษณะเป็นแถวเดี่ยว โดยในการเข้าถึงข้อมูลอาจจะเป็นแบบสุ่มหรือเรียงลำดับก็ได้แล้วแต่การอิมพลีเมนต์ ในลิสต์อินเทอร์เฟซมีคุณสมบัติพื้นฐานที่ควรจะทราบ ดังนี้

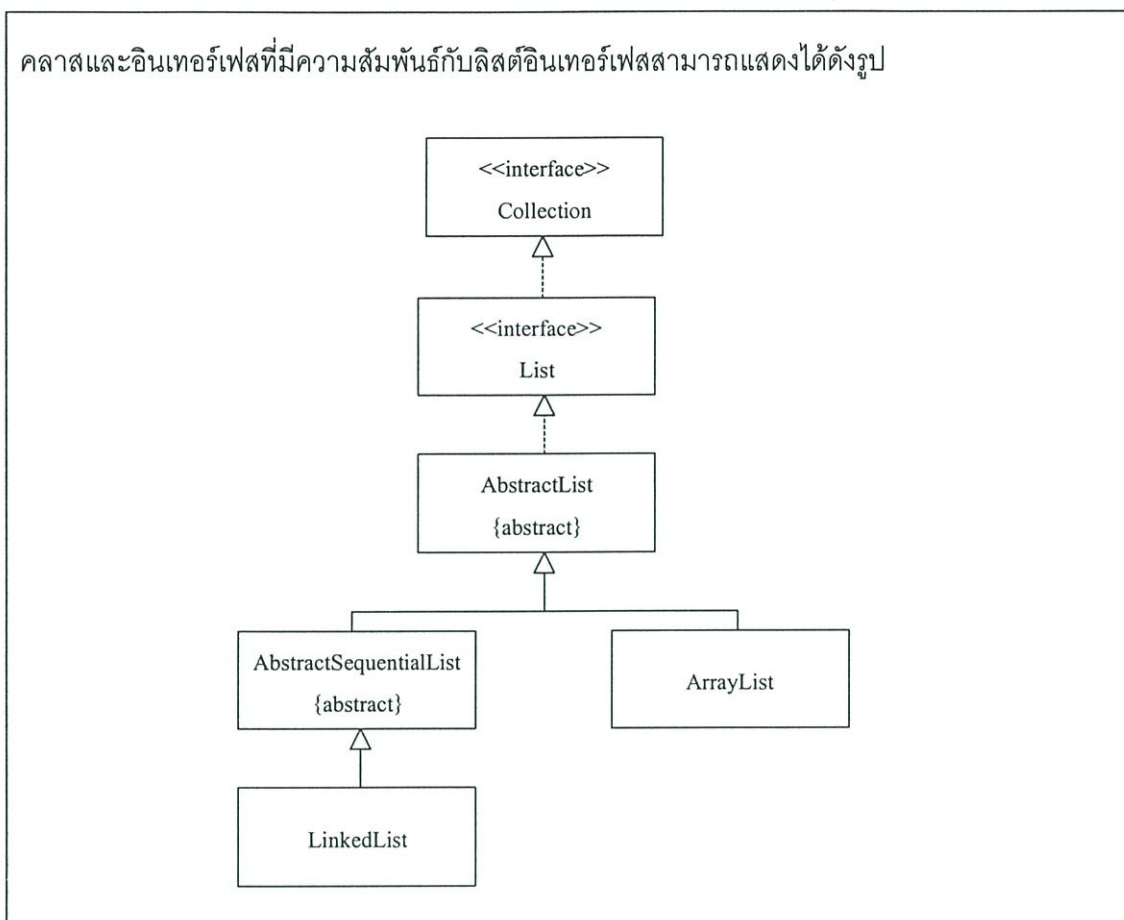
1. สามารถเข้าถึงข้อมูลในลิสต์ได้โดยการบอกตำแหน่งดัชนี (index) ที่แน่นอนของข้อมูล
2. มีข้อมูลซ้ำกันในลิสต์ได้
3. มีตำแหน่งดัชนีเริ่มต้นที่ 0

#### ความสัมพันธ์ของคลาส

1. ลิสต์เป็นอินเทอร์เฟซที่สืบทอดมาจากคอลเลกชันอินเทอร์เฟซ
2. คลาสรูปธรรมที่อิมพลีเมนต์ลิสต์อินเทอร์เฟซ ได้แก่ ออเรย์ลิสต์ และลิงค์ลิสต์
3. คลาสนามธรรมที่อิมพลีเมนต์บางส่วนของลิสต์ ได้แก่ แอบสแทรกทลิสต์ (AbstractList) , แอบสแทรกซีควนเชียลลิสต์ (AbstractSequentialList)

รูปที่ 4.3 แสดงตัวอย่างการเขียนภาพรวมของลิสต์อินเทอร์เฟซ

คลาสและอินเทอร์เฟซที่มีความสัมพันธ์กับลิสต์อินเทอร์เฟซสามารถแสดงได้ดังรูป



รูปที่ 4.3 (ต่อ)

จะเห็นว่าตัวอย่างการเขียนภาพรวมนี้ใช้คลาสไดอะแกรม (class diagram) เข้ามาช่วย ทำให้สามารถเห็นความสัมพันธ์ของลิสต์อินเทอร์เฟซได้ง่ายขึ้น ซึ่งจะต่างจากการเขียนภาพรวมในรูป ๑3 และ ๑5 ของคู่มือของซันไมโครซิสเต็มส์ที่อธิบายเป็นตัวอักษร โดยไม่ใช้คลาสไดอะแกรมเข้ามาช่วย

#### 4.1.2 คำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์

ตัวอย่างโปรแกรมประยุกต์เป็นส่วนประกอบสำคัญที่จะทำให้ผู้พัฒนาโปรแกรมประยุกต์เข้าใจเฟรมเวิร์กได้ง่ายขึ้น ซึ่งรูปแบบในการเขียนคู่มือของงานวิจัยหลายงาน [9] [12] ได้รวมตัวอย่างโปรแกรมประยุกต์ไว้ในภาพรวมของเฟรมเวิร์ก ซึ่งอาจทำให้ผู้ที่ต้องการศึกษาเฟรมเวิร์กจากตัวอย่างหาตัวอย่างโปรแกรมประยุกต์ได้ยาก นอกจากนั้นยังทำให้การอธิบายภาพรวมมีความยาวมากจนเกินไป ดังตัวอย่างรูป ๑5 ดังนั้นรูปแบบที่จะจัดทำขึ้นจากการวิจัยนี้จะแยกตัวอย่างออกจากภาพรวม โดยนำไปรวมกับคำอธิบายวิธีการใช้งาน ดังตัวอย่างในรูป 6 เพื่อให้ผู้พัฒนาโปรแกรมได้เข้าใจขั้นตอนการพัฒนาเป็นโปรแกรมได้ง่ายและรวดเร็วขึ้น โดยตัวอย่างที่เสนอไม่ควรจะยากเกินไป แต่ก็ควรมีความครอบคลุมประเด็นการใช้งานในทุกส่วน เพราะถ้า

นำเสนอตัวอย่างที่ยากจนเกินไปจะทำให้ผู้พัฒนาโปรแกรมประยุกต์เสียเวลาในการทำความเข้าใจกับตัวอย่างเพิ่มขึ้น โดยอาจจะใช้คำอธิบายประกอบแทรกในตัวอย่าง เพื่อเป็นการย้ำความเข้าใจและความชัดเจนในการอธิบายอีกครั้งหนึ่ง ถ้าตัวอย่างที่ใช้เป็นตัวอย่างที่สมบูรณ์ก็ควรมีผลลัพธ์ประกอบเพื่อทำให้เกิดความน่าเชื่อถือ และยังทำให้ผู้พัฒนาโปรแกรมประยุกต์สามารถเปรียบเทียบขั้นตอนการทำงานกับผลลัพธ์ในการทำความเข้าใจได้ดีขึ้น ในการเขียนตัวอย่างสามารถแบ่งออกเป็น 2 ประเภท คือ การเขียนตัวอย่างและคำอธิบายการอิมพลีเมนต์คลาสนามธรรม และการเขียนตัวอย่างและคำอธิบายการใช้งานคลาสรูปธรรม

4.1.2.1 การเขียนตัวอย่างและคำอธิบายการอิมพลีเมนต์คลาสนามธรรม ตัวอย่างที่ใช้จะมีวัตถุประสงค์เพื่ออธิบายขั้นตอนในการอิมพลีเมนต์ โดยตัวอย่างแบบนี้ใช้เป็นแนวทางในการประยุกต์เพื่อการอิมพลีเมนต์ ซึ่งหลักการอิมพลีเมนต์ส่วนใหญ่จะขึ้นอยู่กับผู้พัฒนาเอง ตัวอย่างแบบนี้จะให้รายละเอียดหลักในการอิมพลีเมนต์ เช่น คลาสนี้ต้องใช้คลาสที่มีความสัมพันธ์ใดบ้าง ต้องอิมพลีเมนต์คลาสอะไรบ้าง และจะสร้างอินสแตนซ์ (instance) เพื่อมาเรียกใช้เมทอดในคลาสรูปธรรมที่สร้างจากคลาสนามธรรมนี้ได้อย่างไร เป็นต้น ดังรูปที่ 4.4

ในการอิมพลีเมนต์ลิสต์อินเทอร์เฟซมีขั้นตอนการอิมพลีเมนต์ดังต่อไปนี้

1. import java.util.\*
2. ทำการทำการอิมพลีเมนต์ในทุกเมทอดในลิสต์
3. ทำการสร้างเมทอดหลักขึ้นมาเพื่อสร้าง อินสแตนซ์ ที่เรียกใช้เมทอดในลิสต์

```
import java.util.*;

public class TestList1 implements List {
// สร้าง Constructor
    public TestList1(int initialCapacity) {
        if (initialCapacity < 0)
            throw new IllegalArgumentException("Illegal Capacity: "+ initialCapacity);
        this.elementData = new Object[initialCapacity];
    } //Constructor
```

รูปที่ 4.4 แสดงตัวอย่างการเขียนคำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์ของลิสต์อินเทอร์เฟซ

```

//ทำการเพิ่มลิเมนต์เมทอดทั้งหมดใน List Interface
// Precondition : 0 <= index <= size(). Otherwise,
//           IndexOutOfBoundsException will be thrown
// Postcondition : element has been inserted at position index in this List, and
//           all elements that were at positions >= index before this call have
//           been moved to the next higher position.
public void add(int index, Object element) {
    if (index > size || index < 0)
        throw new IndexOutOfBoundsException(
            "Index: "+index+", Size: "+size);
    System.arraycopy(elementData, index, elementData, index + 1,
        size - index);
    elementData[index] = element;
    size++;
} // add method
// Precondition : 0 <= index <= size(). Otherwise,
//           IndexOutOfBoundsException will be thrown
//Postcondition : The elements of c has been inserted into this List, starting at
//           position index, and true has been returned if at least one element
//           from c was add to this List. Otherwise, false has been returned
public boolean addAll(int index, Collection c) {
    if (index > size || index < 0)
        throw new IndexOutOfBoundsException(
            "Index: " + index + ", Size: " + size);
    Object[] a = c.toArray();
    int numNew = a.length;
    int numMoved = size - index;

```

```

        if (numMoved > 0)
            System.arraycopy(elementData, index, elementData, index + numNew,
                               numMoved);

            System.arraycopy(a, 0, elementData, index, numNew);

            size += numNew;

            return numNew != 0;

    } //addAll method
    // ทำการอิมพลิเมนต์เมทอดที่เหลือจนหมด
    :
    // สร้างเมทอดหลักเพื่อสร้าง อินสแตนซ์ ของ TestList1
    public static void main(String[] args) {
        TestList1 Obj = new TestList1()
        ...
    }
} // Test List1 Class

```

**\*\*หมายเหตุ** ในการใช้งานลิสต์อินเทอร์เฟสจะต้องทำการอิมพลิเมนต์ลิสต์อินเทอร์เฟสทุกเมทอด ถ้าไม่อิมพลิเมนต์ทุกเมทอดจะต้องกำหนดให้เป็นคลาสนามธรรม เช่น public abstract class TestList เป็นต้น

#### รูปที่ 4.4 (ต่อ)

4.1.2.2 การเขียนตัวอย่างและคำอธิบายการใช้งานคลาสรูปธรรม การอธิบายจะเน้นไปที่การเรียกใช้งาน ซึ่งจะไม่อธิบายรายละเอียดทางสถาปัตยกรรมมากนัก การยกตัวอย่างสามารถแสดงเป็นตัวอย่างโปรแกรมประยุกต์ที่สมบูรณ์ได้ โดยควรมีการอธิบายถึงขั้นตอนการใช้งานในตอนต้น จากนั้นจึงอธิบายถึงตัวอย่างปัญหาและการใช้งานคลาสรูปธรรมเพื่อสร้างเป็นโปรแกรมประยุกต์ ดังตัวอย่างรูปที่ 4.5

ในการใช้งานคลาสอาเรย์ลิสต์มีขั้นตอนการเรียกใช้ดังต่อไปนี้

1. สร้างอินสแตนซ์ของคลาสอาเรย์ลิสต์ขึ้นมา เช่น

```
ArrayList obj1 = new ArrayList();
```

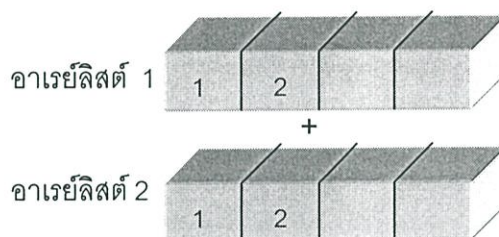
obj1 จะถูกสร้างขึ้นมาเป็นอินสแตนซ์ของอาเรย์ลิสต์

2. เรียกใช้งานเมทอดต่าง ๆ ที่อยู่ในอาเรย์ลิสต์เพื่อสร้างเป็นโปรแกรมประยุกต์ตามต้องการ เช่น

```
obj1.add("element1");
```

element1 จะถูกเพิ่มลงไปในอาเรย์ obj1

ตัวอย่างที่ 1 สร้างอาเรย์ลิสต์ 2 ชุด โดยแต่ละชุดประกอบไปด้วยเลข 1 และ 2 ให้นำตัวเลขในอาเรย์ลิสต์ 1 และ อาเรย์ลิสต์ 2 มาบวกกัน แล้วนำผลลัพธ์ที่ได้ใส่ลงไปใน อาเรย์ลิสต์ 3 จากนั้นนำเลข 9 และ เลข 5 มาใส่ในตำแหน่งที่ 1 ของ อาเรย์ลิสต์ 3 ตามลำดับ



```
public class TestArray {
    public static void main(String[] args) {
        int sum;
        int temp;
        ArrayList obj1 = new ArrayList(); //obj1 is created as instance of ArrayList
        ArrayList obj2 = new ArrayList(); //obj2 is created as instance of ArrayList
        ArrayList obj3 = new ArrayList(); //obj3 is created as instance of ArrayList
    }
}
```

รูปที่ 4.5 แสดงตัวอย่างการเขียนคำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์ของอาเรย์ลิสต์

```

// add 1 and 2 repectively to obj1 and obj2
obj1.add(new Integer (1));
obj1.add(new Integer (2));
obj2.add(new Integer (1));
obj2.add(new Integer (2));
// calculate sum of element in obj1 + obj and put in obj3
for (int i = 0; i< obj1.size() ; i++){
    sum = ((Integer)obj1.get(i)).intValue() + ((Integer)obj2.get
        (i)).intValue();
    obj3.add(new Integer(sum));
}
// add 9 and 5 to index 1 of obj3 respectively
obj3.add(1,new Integer (9));
obj3.add(1,new Integer (5));
// Print all element in Obj3
for (int i = 0; i< obj3.size() ; i++)
    System.out.println("obj3 ["+ i+"]:"+obj3.get(i));
}
}

```

**ผลลัพธ์**

obj3 [0]: 2

obj3 [1]: 5


obj3 [2]: 9

obj3 [3]: 4

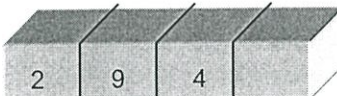
รูปที่ 4.5(ต่อ)

### คำอธิบาย

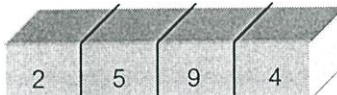
ในโปรแกรมหลังจากนำข้อมูลใน obj1 มาบวกกับ obj 2 แล้วจะได้ค่าใน obj 3 ดังนี้

อาเรย์ลิสต์ 3 

นำเลข 9 เพิ่มลงไปในตำแหน่งที่ 1 จะได้

อาเรย์ลิสต์ 3 

นำเลข 5 เพิ่มลงไปในตำแหน่งที่ 1 จะได้

อาเรย์ลิสต์ 3 

รูปที่ 4.5(ต่อ)

ในการเสนอตัวอย่างและคำอธิบายการใช้งานคลาสรูปธรรม สามารถเสนอได้มากกว่า 1 ตัวอย่าง โดยควรเรียงลำดับตัวอย่างจากง่ายไปหายาก

#### 4.1.3 สรุปรายการคำสั่ง

การแสดงรายการคำสั่งทั้งหมดเป็นส่วนที่มีความสำคัญอีกส่วนหนึ่ง ซึ่งจะช่วยให้เกิดความชัดเจนในองค์ประกอบทั้งหมดของเฟรมเวิร์กทั้งในด้านความหมายและทางด้านโครงสร้าง โดยในการเขียนรายการคำสั่งควรมีการจัดเรียงให้เป็นระบบระเบียบ ซึ่งอาจจะเรียงลำดับตามตัวอักษรหรือตามหมวดหมู่ก็ได้ ขึ้นอยู่กับจำนวนและความสัมพันธ์ของข้อมูล ถ้าข้อมูลมีในจำนวนไม่เกิน 7 รายการ ตามทฤษฎีเกี่ยวกับหน่วยความจำระยะสั้นในหัวข้อที่ 3.1.1.1 หรือไม่ค่อยเข้าพวกกัน การเรียงลำดับสามารถเรียงตามตัวอักษรได้ แต่ถ้าข้อมูลมีปริมาณมากกว่า 7 รายการและมีความคล้ายคลึงกันของข้อมูลก็ควรที่จะจัดเรียงเป็นหมวดหมู่ โดยในแต่ละหมวดหมู่ก็ใช้การเรียงลำดับตามตัวอักษรอีกครั้งหนึ่งเพื่อความสะดวกในการค้นหา และถ้าผู้จัดทำคู่มือเฟรมเวิร์กให้ผู้ใช้สามารถเลือกลักษณะการจัดเรียงได้ตามความต้องการ หรือมีเครื่องมือช่วยในการค้นหาข้อมูลก็จะทำให้ผู้ใช้สามารถใช้งานได้สะดวก รวดเร็วและมีประสิทธิภาพมากขึ้น ในรูปแบบคู่มือเฟรมเวิร์กจากงานวิจัยนี้ ส่วนการสรุปรายการคำสั่งจะแยกออกจากส่วนของคำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์ เนื่องจากรายการคำสั่งมีความยาวค่อนข้างมาก อาจจะทำให้

การอธิบายไม่กระชับ ดังนั้นจึงต้องแยกส่วนของการสรุปรายการคำสั่งออกมาเป็นอีกส่วนหนึ่ง เพื่อให้เกิดความเรียบร้อย ชัดเจน และสะดวกต่อการค้นหา การสรุปรายการคำสั่งมีวัตถุประสงค์หลักเพื่อให้ผู้พัฒนาโปรแกรมได้เห็นถึงคำสั่งทั้งหมดที่เตรียมไว้ให้หรือคำสั่งทั้งหมดที่ผู้พัฒนาโปรแกรมประยุกต์จะต้องทำการอิมพลิเมนต์ให้เกิดเป็นโปรแกรมประยุกต์ที่สมบูรณ์ โดยการเขียนรายการคำสั่งในรูปแบบใหม่ของคุณีมือการใช้งานเฟรมเวิร์ก ผู้วิจัยได้แบ่งการสรุปรายการคำสั่งเป็น 2 ส่วน คือ การสรุปรายการคำสั่งอย่างย่อ และส่วนของตัวอย่างและรายละเอียดในการใช้งาน ดังนี้

4.1.3.1 การสรุปรายการคำสั่งอย่างย่อ ในส่วนนี้จะสรุปรายการคำสั่งทั้งหมดที่ผู้ใช้สามารถเรียกใช้ได้ ซึ่งประกอบไปด้วย

- 1) การประกาศชื่อเมทอด
- 2) คำอธิบายการใช้งานสั้น ๆ

การประกาศชื่อเมทอดจะมีความหมายในตัวอยู่แล้ว ที่จะบอกผู้ใช้งานถึงพารามิเตอร์ที่จะต้องส่งเข้ามามีอะไรบ้าง มีการคืนค่ากลับหรือไม่ ถ้ามีจะคืนค่าเป็นตัวแปรชนิดใด ในส่วนแรกนี้จะช่วยให้ผู้ใช้เลือกใช้เมทอดได้อย่างรวดเร็ว ในการเขียนอาจเรียงลำดับคำสั่งตามตัวอักษรหรือเป็นกลุ่มก็ได้ ขึ้นอยู่กับความเหมาะสมของรายการเมทอดที่มี ดังรูปที่ 4.6

รายการเมทอด	คำอธิบาย
กลุ่มการเพิ่มข้อมูล	
void add(int index, Object element)	เพิ่มข้อมูลในตำแหน่ง index
boolean add(Object ele)	เพิ่มข้อมูลต่อท้ายลิสต์
boolean addAll(Collection c)	เพิ่มข้อมูล Collection c ต่อท้ายลิสต์
boolean addAll(int index, Collection c)	เพิ่มข้อมูลเป็นกลุ่มในตำแหน่ง index
กลุ่มการลบข้อมูล	
Object remove(int index)	ลบข้อมูลในตำแหน่ง index
boolean remove(Object ele)	ลบข้อมูล ele ที่เจอเป็นตัวแรกออกจากลิสต์
boolean removeAll(Collection c)	ลบข้อมูล Collection c ทั้งหมดออกจากลิสต์
boolean retainAll(Collection c)	เก็บเฉพาะข้อมูล Collection c ไว้ในลิสต์

รูปที่ 4.6 แสดงตัวอย่างการสรุปรายการคำสั่งอย่างย่อของลิสต์อินเทอร์เฟส

void clear()	ลบข้อมูลทุกตัวออกจากลิสต์
กลุ่มการค้นหาข้อมูล	
boolean contains(Object ele)	ตรวจสอบว่าในรายการมีข้อมูล ele อยู่หรือไม่
boolean containsAll(Collection c)	ตรวจสอบว่าในรายการมีชุดของข้อมูล c อยู่หรือไม่
boolean equals(Object ele)	เปรียบเทียบข้อมูล ele กับลิสต์ว่ามีค่าเท่ากันหรือไม่
Object get(int index)	ดึงข้อมูลจากตำแหน่ง index
int size()	หาจำนวนข้อมูลในลิสต์
List subList(int fromIndex, int toIndex)	แสดงตำแหน่งข้อมูลในช่วงที่ระบุ
กลุ่มการหาตำแหน่งของข้อมูล	
int indexOf(Object ele)	หาตำแหน่งของข้อมูล ele ในลิสต์
int lastIndexOf(Object ele)	หาตำแหน่งสุดท้ายของข้อมูล ele ในลิสต์
boolean isEmpty()	ตรวจสอบว่ามีข้อมูลอยู่ในลิสต์หรือไม่
กลุ่มการทำซ้ำ	
Object clone()	การคัดลอกลิสต์ ที่หน้าตาเหมือนเดิม
ListIterator listIterator()	จะได้ ListIterator ของข้อมูลในลิสต์
ListIterator listIterator(int index)	จะได้ ListIterator ของข้อมูลในลิสต์ โดยเริ่มจากตำแหน่ง index
Iterator iterator()	แสดงข้อมูลเรียงลำดับ
การเปลี่ยนแปลงข้อมูล	
int hashCode()	ให้ค่าแฮชโค้ดของลิสต์
Object set(int index, Object element)	แทนที่ข้อมูลในตำแหน่ง index ด้วยข้อมูลใหม่
การย้ายข้อมูล	
Object[] toArray()	นำข้อมูลทั้งหมดในรายการไปใส่ในอาเรย์
Object[] toArray(Object[] a)	นำข้อมูลทั้งหมดในรายการไปใส่ในอาเรย์ โดยในการส่งค่าคืนจะเป็นชนิดอาเรย์

รูปที่ 4.6 (ต่อ)

การสรุปรายการคำสั่งอย่างย่อของงานวิจัยจะจัดข้อมูลเป็นกลุ่ม ดังตัวอย่างรูป ๗ ทำให้  
ง่ายต่อการค้นหา ซึ่งต่างกับคู่มือการใช้งานของซันไมโครซิสเต็มส์ ในรูป ๑๖ ที่เรียงรายการคำสั่ง  
ตามลำดับตัวอักษรเท่านั้น

4.1.3.2 ตัวอย่างและรายละเอียดในการใช้งาน ในส่วนที่สองนี้จะมีคำอธิบาย  
ต่าง ๆ ที่เตรียมไว้สำหรับผู้ใช้ที่ต้องการรายละเอียดเพิ่มเติมจากส่วนแรก เช่น รายละเอียดเงื่อนไข  
การเติมเต็มในส่วนฮอตสปอต เพื่อการใช้งานที่ถูกต้องตรงกับความต้องการของผู้พัฒนาโปรแกรม  
ประยุกต์โดยไม่ขัดกับความต้องการของเฟรมเวิร์กที่เตรียมไว้ให้ เป็นต้น ในส่วนนี้จะประกอบไป  
ด้วยคำอธิบายต่าง ๆ ดังนี้

1) เงื่อนไขก่อน (precondition) มีไว้เพื่อตรวจสอบข้อมูลที่เข้ามาให้เป็น  
จริงตามเงื่อนไข ซึ่งในส่วนของเงื่อนไขก่อนนี้ไม่จำเป็นต้องมีในทุกเมทอด โดยจะมีหรือไม่ขึ้นอยู่กับ  
ว่ามีเงื่อนไขที่ต้องการให้เป็นจริงก่อนการทำคำสั่งในรายการนั้นหรือไม่ เช่น ตรวจสอบค่าดัชนีที่  
รับเข้ามาต้องไม่น้อยกว่า 0 เป็นต้น ซึ่งถ้าข้อมูลที่เข้ามาไม่เป็นไปตามเงื่อนไขก่อนแล้ว อาจจะทำให้  
ให้ผลลัพธ์เกิดข้อผิดพลาดได้

2) เงื่อนไขหลัง (postcondition) มีไว้เพื่อบอกถึงผลลัพธ์ที่เกิดขึ้นจาก  
การดำเนินการในเมทอดหรือคำสั่งนั้น และถ้ามีการส่งค่ากลับก็จะบอกด้วยว่าชนิดของตัวแปรที่ส่ง  
ค่ากลับมาว่าเป็นตัวแปรชนิดใด ซึ่งเงื่อนไขหลังนี้ควรมีการอธิบายไว้ในทุกรายการคำสั่งเนื่องจาก  
ผู้ใช้จะได้ทราบแนวทางการอิมพลีเมนต์ที่ถูกต้องได้

3) ตัวอย่างการใช้งานเมทอด มีไว้เพื่อให้ผู้พัฒนาโปรแกรมที่เพิ่งเริ่มใช้  
งานเฟรมเวิร์ก หรือเมทอดเหล่านี้เป็นครั้งแรกทำความเข้าใจและเรียนรู้การใช้งานได้อย่างรวดเร็ว  
โดยในการเขียนตัวอย่างนี้ผู้เขียนอาจจะสร้างส่วนที่สามารถเชื่อมโยงกลับไปยังส่วนของตัวอย่างที่  
มีอยู่แล้ว เพื่อไม่ให้เกิดความซ้ำซ้อนหรือจะสร้างตัวอย่างขึ้นมาใหม่ในกรณีที่ไม่มีตัวอย่างที่ถูก  
สร้างขึ้นก่อนหน้านี้เลยก็ได้

การสรุปรายละเอียดคำสั่งรายการในส่วนที่สองนี้ควรเรียงลำดับรายการเมทอดให้  
เหมือนกับการสรุปรายการคำสั่งอย่างย่อ เพื่อป้องกันความสับสนของการใช้งาน และการอธิบาย  
รายละเอียดในส่วนที่สองนี้จะอธิบายทุก ๆ คลาสที่ได้เขียนไว้ในสรุปรายการคำสั่งอย่างย่อ  
ถึงแม้ว่าบางเมทอดจะมีการอธิบายไว้ในคลาสแม่แล้วก็ตาม ดังตัวอย่างรูป 8 ซึ่งการทำแบบนี้จะ  
ต่างจากคู่มือเฟรมเวิร์กของบริษัทซันไมโครซิสเต็มส์ ดังรูป ๑๗ ที่ให้รายละเอียดเฉพาะรายการ  
เมทอดที่กล่าวถึงเป็นครั้งแรกเท่านั้น ทำให้รายการคำสั่งที่แสดงรายละเอียดมีจำนวนและลำดับไม่  
ตรงกับกรเขียนรายการคำสั่งอย่างย่อ โดยผู้ใช้ต้องการศึกษาในส่วนที่กล่าวมาแล้ว ผู้ใช้จะต้อง  
ลิงค์กลับไปสู่หน้าของคลาสแม่ที่กล่าวถึงเมทอดนั่นเอง โดยผู้วิจัยมีความเห็นว่าการเชื่อมโยง  
กลับไปกลับมาในลักษณะนี้อาจทำให้ผู้ใช้เกิดความสับสนได้ ดังนั้นในการเขียนรายละเอียดของ

ทุกรายการคำสั่งที่มีในแต่ละคลาส โดยผู้ใช้ไม่ต้องให้ลิงค์กลับไปยังคลาสแม่ จะช่วยลดความสับสนของผู้ใช้ได้ ตัวอย่างการเขียนสรุปรายละเอียดคำสั่งดูได้จากรูปที่ 4.7

ในลิสต์อินเทอร์เฟซมีเมทอดที่เพิ่มขึ้นมาจากคอลเล็กชันอินเทอร์เฟซ โดยสามารถแสดงรายละเอียดการใช้งานแบ่งออกเป็นกลุ่มดังนี้

**1. กลุ่มของการเพิ่มข้อมูล**

- add
  - // Precondition :           0 <= index <= size(). มีเช่นนั้นแล้ว
  - //                                 จะถูกดักจับด้วย IndexOutOfBoundsException
  - // Postcondition :         ข้อมูล element จะถูกนำมาแทรกลงในลิสต์ ณ ตำแหน่ง index และ
  - //                                 ข้อมูล element ทุกตัวที่อยู่ในตำแหน่ง >= index ก่อนการเรียก
  - //                                 เมทอดนี้ จะถูกเลื่อนสูงขึ้นในตำแหน่งถัดไป

```
void add(int index, Object element);
```

**ตัวอย่าง** ให้ดูตัวอย่างการใช้งานได้ในเมทอด add(int index, Object element) ของ ArrayList และ LinkedList
- addAll
  - // Precondition :           0 <= index <= size(). มีเช่นนั้นแล้ว
  - //                                 จะถูกดักจับด้วย IndexOutOfBoundsException
  - // Postcondition :         ข้อมูล Collection c จะถูกนำมาแทรกลงในลิสต์ โดยเริ่มแทรกจาก
  - //                                 ตำแหน่ง index ค่า true จะส่งกลับมาเมื่อข้อมูลใน c อย่างน้อย 1
  - //                                 ตัวถูกแทรกลงในลิสต์ ถ้าไม่เช่นนั้นค่าที่ส่งกลับมาจะเป็น false

```
boolean addAll (int index, Collection C);
```

**ตัวอย่าง** ให้ดูตัวอย่างการใช้งานได้ในเมทอด addAll(int index, Object element) ของ ArrayList และ LinkedList

**2. กลุ่มของการลบข้อมูล**

- remove
  - // Precondition :           0 <= index < size().มีเช่นนั้นแล้ว
  - //                                 จะถูกดักจับด้วย IndexOutOfBoundsException

รูปที่ 4.7 แสดงตัวอย่างการสรุปรายการคำสั่งอย่างละเอียดของลิสต์อินเทอร์เฟซ

```
// Postcondition :   ข้อมูล element ณ ตำแหน่ง index จะถูกลบออกจากลิสต์
//                  ข้อมูล element ทุกตัวที่อยู่ในตำแหน่ง < index ก่อนการเรียก
//                  เมทอดนี้ จะถูกเลื่อนต่ำลงในตำแหน่งถัดไป และค่า element ที่
//                  ถูกลบไปจะถูกส่งค่ากลับมา
```

```
Object remove(int index);
```

**ตัวอย่าง** ให้ดูตัวอย่างการใช้งานได้ในเมทอด remove(int index) ของ ArrayList และ

LinkedList

### 3. กลุ่มการค้นหาข้อมูล

- get

```
// Precondition :   0 <= index < size().มิเช่นนั้นแล้ว
//                  จะถูกดักจับด้วย IndexOutOfBoundsException
// Postcondition :   ข้อมูลในตำแหน่ง index จะถูกส่งค่ากลับมา
```

```
Object get(int index);
```

**ตัวอย่าง** ให้ดูตัวอย่างการใช้งานได้ในเมทอด get(int index) ของ ArrayList และ LinkedList

- subList

```
// Precondition :   0 <= fromIndex < toIndex <= size().มิเช่นนั้นแล้ว
//                  จะถูกดักจับด้วย IndexOutOfBoundsException
// Postcondition :   ข้อมูลในลิสต์ตั้งแต่ตำแหน่ง fromIndex จนถึง toIndex จะถูกคืน
//                  ค่ากลับมาเป็นลิสต์
```

```
List subList(int fromIndex, int toIndex);
```

**ตัวอย่าง** ให้ดูตัวอย่างการใช้งานได้ในเมทอด subList(int fromIndex, int toIndex) ของ

ArrayList และ LinkedList

### 4. กลุ่มของการหาตำแหน่งข้อมูล

- indexOf

```
// Postcondition :   ค่า index ของข้อมูล ele ในตำแหน่งแรกที่พบในลิสต์จะถูกส่งค่า
//                  กลับมา แต่ถ้าไม่พบข้อมูล ele ในลิสต์ -1 จะถูกส่งกลับมา
```

```
int indexOf(Object ele);
```

**ตัวอย่าง** ให้ดูตัวอย่างการใช้งานได้ในเมทอด indexOf(Object ele) ของ ArrayList และ

LinkedList

- lastIndexOf
 

// Postcondition :       ค่า index ของข้อมูล ele ในตำแหน่งสุดท้ายที่พบในลิสต์จะถูกส่งค่า  
//                               กลับมา แต่ถ้าไม่พบข้อมูล ele ในลิสต์ -1 จะถูกส่งกลับมา

```
int lastIndexOf(Object ele);
```

*ตัวอย่าง* ให้ดูตัวอย่างการใช้งานได้ในเมทอด lastIndexOf(Object ele) ของ ArrayList และ

LinkedList

## 5. กลุ่มการทำซ้ำ

- clone
 

// Postcondition :       object ของลิสต์ที่ถูกคัดลอกจะถูกส่งกลับ

```
Object clone();
```

*ตัวอย่าง* ให้ดูตัวอย่างการใช้งานได้ในเมทอด clone() ของ ArrayList และ LinkedList
- ListIterator()
 

// Postcondition :       ListIterator ของข้อมูล element ในลิสต์นี้จะถูกส่งค่ากลับมา

```
ListIterator listIterator();
```

*ตัวอย่าง* ให้ดูตัวอย่างการใช้งานได้ในเมทอด listIterator() ของ ArrayList และ LinkedList
- ListIterator(int index)
 

// Postcondition :       ListIterator ของข้อมูล element เริ่มที่ตำแหน่ง index ในลิสต์นี้จะ  
//                               ถูกส่งค่ากลับมา

```
ListIterator listIterator(int index);
```

*ตัวอย่าง* ให้ดูตัวอย่างการใช้งานได้ในเมทอด listIterator(int index) ของ ArrayList และ

LinkedList

## 6. การเปลี่ยนแปลงข้อมูล

- set
 

// Precondition :        $0 \leq \text{index} < \text{size}()$ . มิเช่นนั้นแล้ว  
//                               จะถูกดักจับด้วย IndexOutOfBoundsException

// Postcondition :       ข้อมูล element จะแทนที่ข้อมูลเดิมในลิสต์ ณ ตำแหน่ง index และ  
//                               ข้อมูลในลิสต์จะถูกคืนค่ากลับมา

รูปที่ 4.7 (ต่อ)

Object set(int index, Object element);

**ตัวอย่าง** ให้ดูตัวอย่างการใช้งานได้ในเมทอด set(int index, Object element) ของ ArrayList และ LinkedList

**\*\*หมายเหตุ** ตัวอย่างการใช้งานจะยกตัวอย่างการใช้งานอาเรย์ลิสต์หรือลิงค์ลิสต์ ซึ่งเป็นคลาสที่อิมพลิ-เมนต์อินเทอร์เฟซ มาให้ดูเพื่อให้เห็นภาพชัดเจนในการใช้งาน เนื่องจากตัวอินเทอร์เฟซเองไม่สามารถใช้งานได้จนกว่าจะอิมพลิเมนต์จนเป็นคลาสรูปธรรมแล้ว

#### รูปที่ 4.7 (ต่อ)

จากตัวอย่างจะแบ่งกลุ่มของรายการเมทอดเป็นกลุ่ม แต่ละกลุ่มจะเรียงลำดับเมทอดตามตัวอักษร ซึ่งในทุก ๆ รายการเมทอดจะต้องมีเงื่อนไขหลังที่บอกถึงผลลัพธ์จากการทำงานคำสั่งในเมทอดเสมอ แต่ส่วนของเงื่อนไขก่อนอาจจะมีในเมทอดก็ได้

#### 4.1.4 อ้างอิง

เมื่อผู้พัฒนาโปรแกรมประยุกต์ศึกษาในสิ่งที่ต้องการทราบเรียบร้อยแล้ว อาจต้องการทราบรายละเอียดของส่วนที่เกี่ยวข้องอื่นอีก และเนื่องจากเฟรมเวิร์กมีความซับซ้อนอยู่มาก ดังนั้นเพื่อความต่อเนื่องของเนื้อหาในส่วนประกอบเฟรมเวิร์ก และความเข้าใจที่ดียิ่งขึ้น รูปแบบของคู่มือเฟรมเวิร์กจึงมีความจำเป็นต้องมีส่วนอ้างอิงขึ้น เพื่อบอกกับผู้อ่านว่าควรจะอ่านส่วนใดเป็นส่วนต่อไป หรือถ้าผู้ใช้ต้องการรายละเอียดเพิ่มเติมควรจะอ่านในส่วนใด ซึ่งถ้าเป็นหน้าแรกของคู่มือเฟรมเวิร์ก ส่วนอ้างอิงก็คือส่วนที่บอกว่าผู้ใช้ควรจะเข้าไปอ่านเป็นหน้าแรก โดยในภาพรวมก็ได้กล่าวเกี่ยวกับคลาสที่มีความสัมพันธ์กับคลาสปัจจุบันเช่นกัน แต่เพื่อให้ผู้อ่านสามารถอ่านเนื้อหาที่ต่อเนื่องได้โดยไม่ต้องกลับไปอ่านในส่วนของภาพรวมใหม่ ซึ่งอาจทำให้ผู้ใช้เกิดความสับสนในการย้อนกลับไปกลับมาได้ จึงได้มีส่วนอ้างอิงในตอนท้ายของแต่ละเรซีพี้ เพื่อสะดวกและความต่อเนื่องในการใช้งานคู่มือเฟรมเวิร์ก ดูตัวอย่างส่วนอ้างอิงดังรูปที่ 4.8

เมทอดของลิสต์อินเทอร์เฟซที่สืบทอดมาจากคอลเล็กชันอินเทอร์เฟซ ได้แก่

add(object ele), addAll(Collection c), clear(), contains(object ele), containsAll( Collection c), equals(Object ele), hashCode(), isEmpty(), iterator(), remove(object ele), removeAll(Collection c), retainAll(Collection c), size(), toArray(), toArray(Object[] a)

คลาสที่อิมพลีเมนต์ลิสต์อินเทอร์เฟซ ได้แก่

1. AbstractList // เป็นคลาสที่อิมพลีเมนต์บางส่วนของลิสต์
2. AbstractSequentialList // เป็นคลาสที่สืบทอดจากแอบสแทรกทลิสต์ เพื่อการเข้าถึงข้อมูล  
// แบบโดยลำดับ
3. ArrayList // เป็นคลาสรูปแบบที่อิมพลีเมนต์ลิสต์อินเทอร์เฟซ
4. LinkedList // เป็นคลาสรูปแบบที่อิมพลีเมนต์ลิสต์อินเทอร์เฟซ

#### รูปที่ 4.8 แสดงตัวอย่างส่วนอ้างอิงของลิสต์อินเทอร์เฟซ

นอกจากส่วนอ้างอิงคู่มือเฟรมเวิร์กจากงานวิจัยที่ได้กล่าวมาแล้ว งานวิจัยนี้ยังเพิ่มส่วนเชื่อมโยงที่ด้านข้างของคู่มือในทุกหน้า ทำให้ผู้ใช้สามารถไปยังเรซีพี้อื่นในเฟรมเวิร์กได้ง่ายขึ้น ดังตัวอย่างคู่มือในภาคผนวก ง.

นอกจากการจัดทำส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์กให้มีความชัดเจนแล้ว สิ่งสำคัญอีกอย่างหนึ่งในการเขียนคู่มือการใช้งานเฟรมเวิร์ก คือ คำอธิบายรายละเอียดที่ใช้ในเนื้อหาควรมีความสั้นและกระชับ เป็นไปตามทฤษฎีของการสอนแบบมินิมอลลิสต์ [1] เพื่อให้ผู้อ่านใช้เวลาในการอ่านให้น้อยที่สุดและได้เนื้อหาที่ครบถ้วนตามต้องการ

#### 4.2 การออกแบบส่วนติดต่อกับผู้ใช้สำหรับคู่มือการใช้งานเฟรมเวิร์กบนเว็บเบราว์เซอร์

หลังจากได้รูปแบบของส่วนประกอบในคู่มือการใช้งานเฟรมเวิร์กแล้ว อีกส่วนที่ควรให้ความสำคัญไม่ยิ่งหย่อนไปกว่าในส่วนแรก คือ ส่วนติดต่อกับผู้ใช้ ซึ่งในอดีตการเขียนคู่มือเฟรมเวิร์กส่วนใหญ่ ผู้เขียนแต่ละคนจะมีรูปแบบการเขียนที่เป็นของตัวเอง ทำให้รูปแบบที่ได้มีความหลากหลาย ในขณะที่ผู้ใช้งานคู่มือเฟรมเวิร์กต้องการความเป็นมาตรฐานในรูปแบบของคู่มือ เพื่อที่จะไม่ต้องเรียนรู้รูปแบบของคู่มือใหม่ทุกครั้ง และนอกจากนี้รูปแบบการเขียนคู่มือเฟรมเวิร์กในอดีตบางส่วนยังให้ความสำคัญกับผู้ใช้ไม่มากนัก ทำให้คู่มือเฟรมเวิร์กที่ได้ขาดความเป็นมิตร

กับผู้ใช้ และทำให้ผู้ใช้เกิดความเบื่อหน่ายในการใช้งานคู่มือในที่สุด จากสาเหตุดังกล่าวนี้ ผู้วิจัย จึงพิจารณาส่วนติดต่อกับผู้ใช้มาเป็นส่วนหนึ่งในการจัดทำรูปแบบคู่มือการใช้งานเฟรมเวิร์กด้วย หลักการออกแบบคู่มือเฟรมเวิร์กในส่วนนี้ ผู้วิจัยจะเน้นที่การออกแบบส่วนติดต่อกับผู้ใช้ สำหรับคู่มือการใช้งานเฟรมเวิร์กบนสื่ออิเล็กทรอนิกส์อื่นได้แก่ เว็บเบราว์เซอร์ โดยผู้วิจัยจะใช้ หลักจิตวิทยา และหลักการออกแบบส่วนติดต่อกับผู้ใช้บนเว็บเบราว์เซอร์ในบทที่ 3 มาปรับใช้ให้ เกิดความเหมาะสมกับการเขียนคู่มือเฟรมเวิร์ก ซึ่งมีรายละเอียดสำคัญที่เกี่ยวกับรูปแบบการ เขียนคู่มือเฟรมเวิร์ก ดังนี้

#### 4.2.1 การออกแบบหน้าเอกสารคู่มือเฟรมเวิร์กให้มีความยาวที่เหมาะสม

ตามทฤษฎีในหัวข้อที่ 3.4.1 ความยาวที่เหมาะสมของคู่มือการใช้งานเฟรมเวิร์กในแต่ละ หน้าขึ้นอยู่กับผู้ใช้เป็นส่วนใหญ่ว่าจะใช้อ่านบนหน้าจอ หรือใช้พิมพ์บนกระดาษ โดยผู้วิจัยได้ พิจารณาแล้วว่าหน้าแรกของคู่มือเฟรมเวิร์กจะเป็นหน้าของเมนูหลักที่ควรมีความยาวไม่มากนัก และผู้ใช้สามารถเห็นรายการทั้งหมดภายในหน้าจอเดียว ดังรูป ง1 ส่วนหน้าที่เป็นตัวอย่ าง โปรแกรมประยุกต์ รายการเมทอด สามารถสร้างให้ยาวมากกว่า 1 หน้าจอได้ เพื่อความต่อเนื่อง และสามารถแสดงผลหน้าเหล่านี้ออกมาทางกระดาษได้

#### 4.2.2 การจัดวางส่วนประกอบของคู่มือเฟรมเวิร์กในแต่ละหน้า

##### 4.2.2.1 ส่วนหัว (Header)

ผู้วิจัยเสนอว่าในส่วนหัวของทุก ๆ หน้าควรจะประกอบด้วยแถบเมนู เพื่อให้ผู้ใช้สามารถ เข้าไปในส่วนต่าง ๆ ของคู่มือเฟรมเวิร์กได้ง่ายและรวดเร็วจากทุกหน้า โดยผู้ใช้ควรมีการเน้นสี ให้แตกต่างจากส่วนอื่น เช่น ใช้พื้นหลังที่แตกต่างจากส่วนเนื้อหา เป็นต้น เพื่อแยกส่วนของเมนูให้ ผู้ใช้สังเกตเห็นได้อย่างชัดเจน และถัดลงมาจากเมนูควรจะเป็นส่วนที่บอกให้ผู้ใช้ทราบว่าขณะนี้ ผู้ใช้อยู่ในส่วนใดของคู่มือเฟรมเวิร์ก เพื่อความสะดวกของผู้ใช้ในการย้อนกลับไปอ่านในส่วนที่ ผ่านมา และป้องกันความสับสนของผู้ใช้ดูตัวอย่างส่วนหัวของคู่มือได้ในรูป ง1-ง5 โดยถ้า เปรียบเทียบกับส่วนหัวของคู่มือซีเอ็มไอทีเอ็มเอส ในรูป ง1 ง2 และ ง4 จะเห็นว่าได้บอกเพียงชื่อ เรื่องของแต่ละหน้า โดยไม่มีส่วนที่ทำให้ผู้ใช้สามารถไปในส่วนประกอบอื่นของเฟรมเวิร์กได้ เช่นเดียวกับรูปแบบที่ได้จากงานวิจัย

##### 4.2.2.2 ส่วนเนื้อหา (Body)

ในส่วนเนื้อหานี้จะประกอบไปด้วยกรอบของเนื้อหา และส่วนที่สามารถเชื่อมโยงไปสู่ ส่วนประกอบของเฟรมเวิร์ก โดยภายในกรอบของเนื้อหาจะใส่รายละเอียดเกี่ยวกับเฟรมเวิร์กที่ ได้กล่าวมาแล้วในหัวข้อที่ 4.1 เช่น ภาพรวม คำอธิบายการใช้งาน ตัวอย่าง การสรุปรายการคำสั่ง และส่วนอ้างอิง เป็นต้น โดยการจัดตัวอักษรจะจัดให้ชิดขอบด้านซ้าย มีการย่อหน้า และเว้น

บรรทัด เพื่อให้เกิดความสมดุลกับส่วนของเนื้อหาที่ไม่มากหรือนั่นจนเกินไปจนทำให้ผู้ใช้เกิดความเบื่อหน่าย นอกจากนี้ยังทำให้เกิดความเรียบง่ายและการมองหาข้อมูลที่รวดเร็ว

#### 4.2.2.3 ส่วนท้าย (Footer)

เป็นส่วนที่อยู่ด้านล่างสุดของหน้าเอกสารคู่มือเฟรมเวิร์ก ที่จะใช้แสดงข้อความหลายมิติที่สามารถเชื่อมโยงไปสู่ส่วนถัดไป หรือส่วนก่อนหน้าในคู่มือ และส่วนล่างสุดแสดงเมนูเช่นเดียวกับส่วนหัวอีกครั้งหนึ่งแต่จะมีรูปแบบที่เรียบง่ายกว่า เพื่ออำนวยความสะดวกให้ผู้ใช้สามารถเชื่อมโยงเนื้อหาไปสู่ในส่วนอื่นได้อย่างสะดวก โดยไม่ต้องเลื่อนหน้าจอกลับไปที่ส่วนบนสุดอีกครั้ง ดังรูป ง1 ในส่วนนี้จะคล้ายกับส่วนท้ายของคู่มือของบริษัทซัมซุงโมบายล์ในรูป ง2 ต่างกันเพียงคู่มือที่ได้จากงานวิจัยมีส่วนท้ายที่มีความเรียบง่ายกว่ามาก ไม่มีส่วนที่รบกวนสายตาผู้ใช้จนเกินความจำเป็น

#### 4.2.3 การเลือกใช้สีสำหรับคู่มือการใช้งานเฟรมเวิร์ก

การเลือกใช้สีให้กับคู่มือการใช้งานเฟรมเวิร์กจะอยู่บนพื้นฐานของการใช้ชุดสีสำหรับเว็บ 216 สี ตามที่ ธวัชชัย ศรีสุเทพ [26] ได้กล่าวไว้ เพื่อการเห็นสีที่ตรงกันของผู้เขียนและผู้ใช้งานคู่มือเฟรมเวิร์ก ผู้เขียนคู่มือสามารถเลือกใช้สีที่เข้ากันได้จากเครื่องมือวิซิโบน [24] คู่มือการใช้งานเฟรมเวิร์กเป็นเอกสารที่ต้องการความเชื่อถือที่มากกว่าการเขียนเว็บทั่วไป ดังนั้นจึงไม่ควรเลือกใช้สีที่ดูฉูดฉาดในการเขียน โดยผู้วิจัยแนะนำให้ใช้สีโทนดำ-เทาในการเขียนคู่มือ โดยดูตัวอย่างได้จากคู่มือที่ได้จากงานวิจัยในภาคผนวก ง. เนื่องจากสีในโทนนี้จะทำให้คู่มือเฟรมเวิร์กที่ได้ดูน่าเชื่อถือ และไม่มีการสะท้อนของสีที่รบกวนการอ่านของผู้ใช้ แต่อย่างไรก็ตาม ผู้เขียนสามารถใช้โทนสีอื่นที่เป็นสีโทนเย็นในการเขียนคู่มือตามความเหมาะสมได้

#### 4.2.4 การจัดรูปแบบตัวอักษรสำหรับคู่มือการใช้งานเฟรมเวิร์ก

เนื่องจากตัวอักษรเป็นองค์ประกอบส่วนใหญ่ในคู่มือเฟรมเวิร์ก ดังนั้นจึงควรให้ความสำคัญกับการใช้ตัวอักษรด้วย โดยรูปแบบตัวอักษรที่กล่าวมาแล้วในหัวข้อที่ 3.4.4 สามารถนำมาปรับใช้กับการเขียนคู่มือการใช้งานเฟรมเวิร์กได้ ดังต่อไปนี้

##### 4.2.4.1 ประเภทของแบบอักษรที่เหมาะสมกับเนื้อหาในคู่มือเฟรมเวิร์ก

ผู้วิจัยแนะนำให้ใช้แบบอักษรที่สัมพันธ์กับรูปร่างในส่วนของเนื้อหาและคำอธิบายส่วนใหญ่ในคู่มือ ซึ่งแบบอักษรลักษณะนี้มีให้เลือกมากมาย ผู้เขียนควรพิจารณาถึงสัดส่วนที่เหมาะสมที่ไม่แคบหรือกว้างจนเกินไปให้ผู้ใช้สามารถอ่านได้ง่าย ในการเขียนตัวอย่างคู่มือเฟรมเวิร์กที่ได้จากการวิจัย ผู้วิจัยใช้แบบอักษร Microsoft Sans Serif ดังรูป ง1-ง4 เนื่องจากตัวอักษรประเภทนี้ที่มีลักษณะที่อ่านง่าย เป็นทางการและแสดงผลได้ทั้งอักษรภาษาไทยและอักษรภาษาอังกฤษ

ในส่วนของตัวอย่างโปรแกรม และชื่อเมทอดในส่วนสรุปรายการอย่างย่อผู้วิจัยแนะนำให้ใช้แบบอักษรที่มีขนาดคงที่ เนื่องจากในแต่ละบรรทัดของตัวอย่างโปรแกรมและชื่อเมทอดมีความยาวไม่มากนัก ผู้ใช้สามารถอ่านโค้ดโปรแกรมด้วยแบบอักษรนี้ได้ง่ายขึ้นกว่าตัวอักษรแบบสลับพันธ์กับรูปร่างในส่นหมายเหตุ (comment) ของตัวอย่างโปรแกรมจะใช้ตัวอักษรที่มีขนาดคงที่ด้วย เพื่อให้รูปแบบตัวอักษรเป็นไปในแนวทางเดียวกัน การเขียนตัวอย่างคู่มือเฟรมเวิร์กที่ได้จากงานวิจัย ผู้วิจัยเลือกใช้แบบอักษร Courier ดังรูป 6 และ 7 เนื่องจากแบบอักษรนี้มีลักษณะที่อ่านง่าย และมีความกว้างที่เหมาะสม ซึ่งต่างจากคู่มือของชันไมโครซิสเต็มส์ในภาคผนวก จ. ที่ใช้ตัวอักษรแบบเดียวกันหมดในทุกส่วนของคู่มือ

#### 4.2.4.2 การใช้ตัวพิมพ์และการเว้นวรรคเพื่อความสะดวกในการอ่าน

การใช้ข้อความภาษาอังกฤษในการเขียนคู่มือการใช้งานเฟรมเวิร์กบนเว็บเบราว์เซอร์สามารถใช้หลักการเดียวกันกับเอกสารอื่น ๆ ทั่วไปได้ กล่าวคือ ควรใช้ตัวอักษรใหญ่เฉพาะกับตัวอักษรแรกของประโยค เพื่อความสะดวกในการอ่านเนื้อหาได้อย่างรวดเร็ว ส่วนหลักการใช้ข้อความภาษาไทยนั้น ผู้เขียนควรจะมีการเว้นวรรคระหว่างประโยคให้เหมาะสมกับจังหวะในการอ่าน เพื่อให้ผู้อ่านสามารถอ่านได้โดยไม่ติดขัด

#### 4.2.4.3 การจัดตำแหน่งเนื้อหาในคู่มือเฟรมเวิร์ก

ผู้วิจัยแนะนำให้จัดตำแหน่งของเนื้อหาแบบชิดขอบด้านซ้ายโดยปล่อยให้ขอบด้านขวามีลักษณะไม่สม่ำเสมอ เนื่องจากการจัดตำแหน่งแบบนี้ มีระยะห่างระหว่างตัวอักษรที่เป็นธรรมชาติ ทำให้ผู้อ่านสามารถอ่านได้อย่างสะดวกและรวดเร็วมากที่สุด

#### 4.2.4.4 ความกว้างของบรรทัดในคู่มือการใช้งานเฟรมเวิร์ก

ในคู่มือการใช้งานเฟรมเวิร์ก ผู้วิจัยแนะนำให้ใช้ความกว้างของข้อความไม่รวมแถบด้านข้างเท่ากับ 500 ถึง 550 พิกเซล เนื่องจากเนื้อหาของคู่มือเฟรมเวิร์กมีความยาวค่อนข้างมาก ถ้าผู้เขียนกำหนดความยาวบรรทัดน้อยกว่า 500 พิกเซล จะทำให้ความยาวของเนื้อหาดูยาวขึ้นมาก อาจเป็นผลให้ผู้อ่านเกิดความเบื่อหน่ายในการศึกษาเฟรมเวิร์กจากคู่มือได้ และถ้ากำหนดความกว้างของบรรทัดมากเกินไปก็จะทำให้เกิดปัญหาในขณะที่กวาดสายตาในการอ่าน ที่ผู้อ่านต้องเคลื่อนสายตาจากปลายบรรทัดไปยังส่วนของต้นบรรทัดใหม่ทำให้ไม่สะดวกในการอ่าน ส่วนความกว้างของคู่มือทั้งหมดก็ไม่ควรมีความกว้างเกิน 780 พิกเซล เนื่องจากในปัจจุบันยังมีผู้ใช้งานคอมพิวเตอร์ด้วยความละเอียด 800 x 600 พิกเซลอยู่ ซึ่งถ้ากำหนดความกว้างของเอกสารคู่มือให้มีขนาดมากกว่า 780 พิกเซล อาจทำให้เกิดปัญหาการเห็นหน้าจอที่ไม่ครบถ้วนก็ได้

#### 4.2.4.5 ขนาดของตัวอักษร

เนื่องจากเอกสารที่ใช้เว็บเบราว์เซอร์เป็นสื่อในการนำเสนอ จำเป็นต้องใช้ตารางเข้ามาจัดตำแหน่งข้อความเป็นส่วนใหญ่ ทำให้ขนาดของอักษรมีผลต่อการจัดหน้าให้เกิดความสวยงาม

อย่างมาก ซึ่งถ้าผู้ใช้สามารถปรับขนาดตัวอักษรตามต้องการ เช่น ถ้าตัวอักษรขนาด 14 พิกเซล มีความยาวพอดีกับความกว้างของเซลล์ตารางหนึ่ง แต่ผู้ใช้กำหนดขนาดตัวอักษรให้มีขนาดใหญ่ขึ้นเป็น 16 พิกเซล ซึ่งทำให้บรรทัดที่ใช้เปลี่ยนจาก 1 บรรทัด กลายเป็น 2 บรรทัด เป็นต้น ถ้าสิ่งเหล่านี้เกิดขึ้นกับหัวข้อหรือส่วนสำคัญกับเนื้อหาแล้ว จะสร้างความรู้สึกไม่เสร็จสมบูรณ์หรือไม่เรียบร้อยให้กับผู้ใช้งานคู่มือ ดังนั้นผู้วิจัยแนะนำให้ผู้เขียนคู่มือเฟรมเวอร์กกำหนดขนาดของตัวอักษรที่แน่นอน และสามารถใช้นาขนาดตัวอักษรในการสร้างลำดับชั้นความสำคัญของข้อมูล ผู้วิจัยแนะนำให้ใช้นาขนาดตัวอักษรทั่วไปเท่ากับ 14 พิกเซล และใช้ตัวอักษรที่มีขนาดใหญ่ขึ้นตามลำดับความสำคัญของหัวข้อ โดยแต่ละระดับของหัวข้อควรมีขนาดต่างกัน 2 พิกเซลขึ้นไป เนื่องจากผู้ใช้จะสามารถแยกความแตกต่างของหัวข้อแต่ละระดับได้อย่างชัดเจน อย่างไรก็ตามผู้เขียนอาจเลือกใช้วิธีการเน้นข้อความแบบอื่น ตามหัวข้อที่ 4.2.4.6 ในการกำหนดลำดับความสำคัญของหัวข้อ เช่น การใช้สี การทำตัวหนา เป็นต้น โดยไม่จำเป็นต้องเพิ่มขนาดของหัวข้อก็ได้

#### 4.2.4.6 การเน้นข้อความให้เด่นชัด

การเน้นข้อความให้เด่นชัดในการเขียนคู่มือการใช้งานเฟรมเวอร์กทำได้หลายวิธี โดยข้อความที่ต้องการเน้นจะต้องเป็นข้อความที่มีความสำคัญหรือเป็นส่วนข้อความที่ผู้เขียนต้องการแสดงให้ผู้ใช้สังเกตเห็นได้ง่ายกว่าข้อความทั่วไป สำหรับการเน้นข้อความบนเว็บเบราว์เซอร์ผู้เขียนสามารถเลือกใช้วิธีการต่าง ๆ ได้แก่ การทำตัวหนา การกำหนดขนาดตัวอักษรใหญ่ขึ้น การทำสีตัวอักษร และการทำสีพื้นหลัง โดยการกำหนดขนาดตัวอักษรผู้เขียนสามารถใช้หลักการตามหัวข้อที่ 4.2.4.5 และการใช้สีกับตัวอักษรตามหัวข้อที่ 4.2.4.7 นอกจากนี้การใช้สีตัวอักษรในการเน้นข้อความควรจะใช้ชุดสีเดียวกับส่วนประกอบอื่นของคู่มือ ไม่ควรจะใช้สีตัวอักษรที่มีสีที่ตัดกันจนเกินไป ส่วนการเน้นข้อความด้วยวิธีอื่น เช่น การทำตัวเอียง สามารถใช้ในการเน้นข้อความได้เช่นกัน แต่ไม่ควรใช้กับข้อความที่มีขนาดใหญ่มากนัก เนื่องจากจะทำให้เห็นลักษณะรอยหยักอักษรที่ชัดเจนมากขึ้น การเน้นข้อความด้วยการขีดเส้นใต้ควรหลีกเลี่ยงในการเขียนคู่มือเฟรมเวอร์กที่ใช้แสดงบนสื่อเว็บเบราว์เซอร์ เนื่องจากอาจจะทำให้ผู้ใช้เกิดความสับสนกับลักษณะของข้อความหลายมิติได้ แต่ถ้าเป็นเอกสารแบบอื่นที่ไม่ใช่เอกสารเว็บผู้เขียนสามารถเน้นข้อความด้วยการขีดเส้นใต้ได้ตามปกติ เนื่องจากลักษณะการเน้นข้อความมีให้เลือกใช้หลายแบบ ผู้เขียนควรเลือกลักษณะการเน้นข้อความให้กับข้อความที่อยู่ในกลุ่มเดียวกัน หรือมีระดับความสำคัญเท่ากันให้เหมือนกันทั้งเอกสาร โดยดูตัวอย่างการเน้นข้อความของคู่มือที่ได้จากงานวิจัยในภาคผนวก ง. ซึ่งถ้าเปรียบเทียบกับคู่มือของบริษัทซัมไมโครซิสเต็มส์ เช่น รูป ๑4 เป็นต้น จะเห็นว่ามี การเน้นข้อความด้วย ตัวเอียง ตัวหนา ที่มากเกินไป ทำให้ข้อความดูสับสน

#### 4.2.4.7 การใช้สีกับตัวอักษร

การใช้สีตัวอักษรในคู่มือเฟรมเวิร์กผู้วิจัยแนะนำให้ใช้สีดำกับส่วนใหญ่ของตัวอักษรและเน้นข้อความในโทนสีเทา-ดำตามหลักการที่กล่าวมาแล้วในหัวข้อที่ 4.2.3 ยกเว้นส่วนที่แสดงโปรแกรมตัวอย่าง เนื่องจากต้องการใช้สีในการเน้นให้เห็นถึงส่วนประกอบต่าง ๆ ในตัวอย่างโปรแกรม โดยแต่ละสีจะให้ความหมายที่แตกต่างกัน เช่น สีเขียว หมายถึง หมายเหตุของโปรแกรม สีแดง หมายถึง ชื่อคลาส เป็นต้น โดยการใช้สีในตัวอย่างโปรแกรมประยุกต์ของตัวอย่างคู่มือการใช้งานเฟรมเวิร์กในภาคผนวก ง. อ้างอิงมาจากการใช้สีในการเขียนโปรแกรมจาวาด้วยโปรแกรมอีดิทพลัส (EditPlus) ซึ่งผู้เขียนคู่มืออาจจะใช้สีโดยอ้างอิงจากโปรแกรมอื่น หรือหลักการอื่นก็ได้ แต่การใช้สีจะต้องมีความหมายที่เหมือนกันทั้งคู่มือ นอกจากนี้การใช้สียังต้องมีข้อควรระวังในการแสดงผลทางเครื่องพิมพ์ด้วย ผู้วิจัยแนะนำให้ใช้พื้นหลังที่มีสีอ่อนและตัวอักษรสีเข้ม เพื่อให้ผู้ใช้สามารถพิมพ์ข้อความออกมาทางเครื่องพิมพ์ได้อย่างชัดเจน โดยจากคู่มือชั้นไมโครซีเอสเต็มสีในภาคผนวก จ. จะเห็นว่าการใช้สีกับตัวอักษรไม่ได้เน้นในส่วนของตัวอย่างหรือโค้ดของโปรแกรมเลย ทำให้ผู้ใช้สังเกตเห็นตัวอย่างจากคู่มือชั้นไมโครซีเอสเต็มสีได้ยาก

### 4.3 สรุปรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์ก

#### 4.3.1 ส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก

คู่มือการใช้งานเฟรมเวิร์กจะต้องมีหัวข้อที่เป็นส่วนประกอบหลักและการเรียงลำดับ ดังนี้

4.3.1.1 ภาพรวมเฟรมเวิร์ก ที่อธิบายโดเมนของปัญหาและความสัมพันธ์ของคลาสภายในเฟรมเวิร์กโดยสังเขป

4.3.1.2 ตัวอย่างการใช้งานเฟรมเวิร์ก เป็นการรวบรวมโปรแกรมประยุกต์ในคู่มือเฟรมเวิร์กเพื่อความสะดวกในการค้นหา โดยอาจจะเพิ่มเติมตัวอย่างใหม่ลงไปก็ได้

4.3.1.3 คำอธิบายการใช้งานพร้อมด้วยตัวอย่างโปรแกรมประยุกต์ ที่จัดให้เป็นหมวดหมู่ เช่น หมวดของคลาสรูปธรรม หมวดของคลาสนามธรรม เป็นต้น

4.3.1.4 ส่วนอ้างอิงที่นำไปสู่เรขีย่อย ๆ ในเฟรมเวิร์ก โดยในแต่ละเรขีย่อยจะประกอบด้วยหัวข้อดังต่อไปนี้

1) ภาพรวมของแต่ละเรขีย่อย ใช้อธิบายโดเมนของปัญหาและความสัมพันธ์ของคลาสที่เกี่ยวข้องกับเรขีย่อยนี้

2) คำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์ ซึ่งถ้าเป็นการเขียนตัวอย่างและคำอธิบายการใช้งานคลาสนามธรรม คำอธิบายและตัวอย่างจะใช้เป็นแนวทางในการอิมพลีเมนต์ ส่วนถ้าเป็นการเขียนตัวอย่างและคำอธิบายการใช้งานคลาสรูปธรรม

คำอธิบายและตัวอย่างจะเน้นไปที่การใช้งาน โดยควรยกตัวอย่างที่เป็นโปรแกรมประยุกต์ที่สมบูรณ์ พร้อมทั้งผลลัพธ์เพื่อประกอบการทำความเข้าใจ

3) สรุปรายการคำสั่ง ที่ประกอบด้วย การสรุปรายการคำสั่งอย่างย่อ ตัวอย่างและรายละเอียดในการใช้งาน

4) อ้างอิง ใช้อ้างอิงไปสู่เรขีฟี่อื่นที่มีความสัมพันธ์กัน

4.3.2 การออกแบบส่วนติดต่อกับผู้ใช้สำหรับคู่มือการใช้งานเฟรมเวิร์กบนเว็บเบราว์เซอร์ คู่มือควรมีรูปแบบของส่วนติดต่อกับผู้ใช้สำหรับคู่มือเฟรมเวิร์กบนเว็บเบราว์เซอร์ โดยสรุป ดังนี้

4.3.2.1 หน้าเมนูหลักไม่ควรมีความยาวเกิน 1 หน้าจอคอมพิวเตอร์

4.3.2.2 ส่วนหัว ที่ประกอบไปด้วย แถบเมนูที่เห็นได้ชัดเจน และส่วนบอกตำแหน่งที่อยู่ของผู้ใช้ในคู่มือ

4.3.2.3 ส่วนเนื้อหา ที่ประกอบไปด้วยกรอบของเนื้อหาที่มีความกว้าง 500 ถึง 550 พิกเซล และกรอบของส่วนอ้างอิงทางซ้ายมือของกรอบเนื้อหา โดยความกว้างโดยรวมของทั้งสองส่วนรวมกันต้องไม่เกิน 800 พิกเซล

4.3.2.4 ส่วนท้าย ที่ประกอบไปด้วยข้อความหลายมิติที่สามารถเชื่อมโยงไปสู่เนื้อหาในส่วนถัดไป หรือส่วนก่อนหน้าในคู่มือ และตามด้วยส่วนเมนูอย่างง่ายในตอนท้ายของทุกหน้า

4.3.2.5 การเลือกใช้สีในคู่มือการใช้งานเฟรมเวิร์ก

1) ใช้สีโทนเทา-ดำในการเขียนคู่มือ หรือสีในโทนเย็นอื่นซึ่งผู้ใช้สามารถเลือกสีที่เข้ากันได้จากเครื่องมือวิชัน

2) การใช้สีกับตัวอักษรให้เน้นโทนเทา-ดำเช่นเดียวกับส่วนอื่นของคู่มือ ยกเว้นของตัวอย่างโปรแกรมประยุกต์ที่จะใช้สีในการสื่อความหมาย เช่น สีแดง หมายถึง ข้อผิดพลาด เป็นต้น

4.3.2.6 การจัดรูปแบบตัวอักษรในคู่มือการใช้งานเฟรมเวิร์ก

1) ใช้แบบอักษรที่สัมพันธ์กับรูปร่าง เช่น Microsoft Sans Serif เป็นต้น ในส่วนของเนื้อหาและคำอธิบายส่วนใหญ่ในคู่มือ

2) ใช้แบบอักษรที่มีขนาดคงที่ เช่น Courier เป็นต้น ในส่วนของตัวอย่างโปรแกรม และชื่อเมทอดในส่วนสรุปรายการอย่างย่อ

3) ใช้ขนาดตัวอักษรทั่วไปเท่ากับ 14 พิกเซล

4) ใช้ตัวอักษรที่มีขนาดใหญ่ขึ้นตามลำดับความสำคัญ โดยแต่ละระดับควรมีขนาดต่างกันอย่างน้อย 2 พิกเซล

5) การเน้นข้อความทำได้หลายวิธี ได้แก่ การทำตัวหนา การทำสีตัวอักษร การทำสีพื้นหลังตัวอักษร การกำหนดขนาดให้ใหญ่ขึ้น เป็นต้น ควรหลีกเลี่ยงการขีดเส้นใต้และการทำตัวเอียง

จากรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์กในบทนี้ ผู้วิจัยได้นำมาใช้พัฒนาเป็นคู่มือจาวาคอเล็กชันเฟรมเวิร์ก (ภาคผนวก ง.) ซึ่งการทดสอบประสิทธิภาพ ทำโดยเปรียบเทียบระยะเวลาการพัฒนาโปรแกรมประยุกต์และความพอใจในการใช้งานคู่มือเฟรมเวิร์กกับคู่มือของบริษัทซันไมโครซิสเต็มส์ (ภาคผนวก จ.) โดยมีวิธีดำเนินการทดลองและผลการวิเคราะห์ข้อมูลในบทที่ 5 และ 6 ตามลำดับ

## บทที่ 5

### วิธีดำเนินการวิจัย

ในการวิจัยครั้งนี้ใช้ระเบียบวิธีการวิจัยเชิงทดลอง (Experimental Design) เพื่อเปรียบเทียบระยะเวลาในการพัฒนาโปรแกรมประยุกต์และความพอใจในการใช้งานคู่มือเฟรมเวิร์กระหว่างคู่มือการใช้งานเฟรมเวิร์กที่จัดทำตามรูปแบบที่ได้จากงานวิจัยกับคู่มือการใช้งานเฟรมเวิร์กของบริษัทชั้นนำโครซิสเต็มส์

#### 5.1 ประชากรและกลุ่มตัวอย่าง

กลุ่มตัวอย่างที่ใช้ในการวิจัยครั้งนี้เป็นนักศึกษาที่กำลังศึกษาอยู่ในสาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จำนวนทั้งหมด 27 คน แจกแจงได้ตามตารางที่ 5.1

ตารางที่ 5.1 ตารางแจกแจงกลุ่มตัวอย่าง

กลุ่มตัวอย่าง	จำนวน
นักศึกษาปริญญาตรีชั้นปีที่ 4	7
นักศึกษาปริญญาโทชั้นปีที่ 1	6
นักศึกษาปริญญาโทชั้นปีที่ 2	10
นักศึกษาปริญญาโทชั้นปีที่ 3	4
รวม	27

กลุ่มตัวอย่างทั้งหมดมีคุณสมบัติขั้นต้น ดังนี้

5.1.1 สามารถอ่านและเข้าใจภาษาอังกฤษได้อย่างดี

5.1.2 มีความสามารถในการพัฒนาโปรแกรมด้วยภาษาจาวาได้

ในการทดลองนี้จะสมมติให้กลุ่มตัวอย่างทุกกลุ่มมีความเท่าเทียมกัน เนื่องจากกลุ่มมาจากสิ่งแวดล้อมเดียวกัน และมีความรู้พื้นฐานใกล้เคียงกัน

## 5.2 เครื่องมือที่ใช้ในการวิจัย

5.2.1 โปรแกรมสำเร็จรูปทางสถิติ SPSS FOR WINDOWS รุ่น 10.0.7

5.2.2 คู่มือการใช้งานเฟรมเวิร์ก ได้แก่

5.2.2.1 คู่มือการใช้งานคอเล็กชันเฟรมเวิร์กที่ได้จากการวิจัย

5.2.2.2 คู่มือการใช้งานคอเล็กชันเฟรมเวิร์กของบริษัทชั้นไมโครซิสเต็มส์

5.2.3 แบบสอบถาม ใช้วัดระยะเวลาในการพัฒนาโปรแกรมประยุกต์ ซึ่งมีเนื้อหาเกี่ยวกับการสร้างลิสต์ด้วยจาวาคอเล็กชันเฟรมเวิร์ก

5.2.4 แบบสอบถาม ใช้วัดความพอใจของกลุ่มควบคุมและกลุ่มทดลองหลังการใช้งานคู่มือเฟรมเวิร์กในการพัฒนาโปรแกรมประยุกต์ใน 2 ประเด็น คือ ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์กและด้านส่วนติดต่อกับผู้ใช้ โดยแบบสอบถามแต่ละข้อใช้ตรวจสอบความพอใจในรูปแบบคู่มือเฟรมเวิร์ก ดังนี้

5.2.4.1 ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก มีทั้งหมด 7 ข้อ ได้แก่

ข้อที่ 1 มีหัวข้อในการแสดงเนื้อหาที่เพียงพอต่อการพัฒนาโปรแกรมประยุกต์ ในข้อนี้ใช้ความพอใจเกี่ยวกับหัวข้อด้านส่วนประกอบตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.1 ส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก

ข้อที่ 2 มีการเรียงลำดับส่วนประกอบของคู่มือที่ง่ายต่อการใช้งาน ในข้อนี้ใช้วัดความพอใจเกี่ยวกับการเรียงลำดับส่วนประกอบของคู่มือที่ได้เสนอไว้ตามรูปที่ 4.1 แสดงส่วนประกอบของเนื้อหาหลัก และ รูปที่ 4.2 แสดงส่วนประกอบของเนื้อหาในแต่ละเรขีพี

ข้อที่ 3 ผู้ใช้สามารถค้นหารายการคำสั่งในคู่มือเฟรมเวิร์กได้รวดเร็ว ในข้อนี้ใช้วัดความพอใจเกี่ยวกับความเร็วในการค้นหาข้อมูลรายการคำสั่งที่ผู้ใช้องการตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.1.3 สรุปรายการคำสั่ง

ข้อที่ 4 มีส่วนที่ทำให้เห็นภาพรวมของเฟรมเวิร์กได้ง่าย ในข้อนี้ใช้วัดความพอใจเกี่ยวกับส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์กในด้านการเขียนภาพรวมตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.1.1 ภาพรวม

ข้อที่ 5 มีส่วนที่ทำให้ผู้ใช้เห็นความสัมพันธ์ของส่วนประกอบภายในเฟรมเวิร์กได้ง่าย ในข้อนี้ใช้วัดความพอใจของผู้ใช้เกี่ยวกับการทำความเข้าใจความสัมพันธ์ของส่วนประกอบภายในเฟรมเวิร์ก ตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.1.1 ภาพรวม

ข้อที่ 6 มีตัวอย่างโปรแกรมประยุกต์ที่เหมาะสม ในข้อนี้ใช้วัดความพอใจเกี่ยวกับส่วนประกอบของคู่มือเฟรมเวิร์กทางด้านตัวอย่างโปรแกรมประยุกต์ใน

ส่วนประกอบหลักของคู่มือและความเพียงพอของตัวอย่างโปรแกรมประยุกต์ตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.1.2 คำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์

ข้อที่ 7 มีคำอธิบายหรือส่วนที่เชื่อมโยงไปยังเนื้อหาอื่นได้ง่าย ไม่สับสน ในข้อนี้ใช้วัดความพอใจของผู้ใช้เกี่ยวกับการเชื่อมโยงเนื้อหาไปสู่ส่วนประกอบอื่นของเฟรมเวิร์กตามรูปแบบที่เสนอไว้ในหัวข้อ 4.1.4 อ้างอิง

5.2.4.2 ด้านส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก มีทั้งหมด 9 ข้อ ได้แก่

ข้อที่ 8 มีส่วนหัวของแต่ละหน้าที่เหมาะสม ในข้อนี้ใช้วัดความพอใจในส่วนหัวของเอกสารคู่มือที่ออกแบบตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.2.1 การจัดวางส่วนหัวในแต่ละหน้า

ข้อที่ 9 มีส่วนท้ายของเอกสารที่เหมาะสม ในข้อนี้ใช้วัดความพอใจในส่วนท้ายของเอกสารคู่มือที่ออกแบบตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.2.3 การจัดวางส่วนท้ายในแต่ละหน้า

ข้อที่ 10 มีการจัดตัวอักษรในเนื้อหาที่เหมาะสม ในข้อนี้ใช้วัดความพอใจเกี่ยวกับการจัดตัวอักษรในเนื้อหาของคู่มือที่ออกแบบตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.2.2 การจัดวางส่วนเนื้อหาในแต่ละหน้า และ 4.2.4.3 การจัดตำแหน่งเนื้อหาในคู่มือ

ข้อที่ 11 มีขนาดตัวอักษรในเนื้อหาที่เหมาะสม ในข้อนี้ใช้วัดความพอใจเกี่ยวกับขนาดตัวอักษรในเนื้อหาของคู่มือที่ออกแบบตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.4.5 ขนาดของตัวอักษร

ข้อที่ 12 มีความกว้างบรรทัดของเนื้อหาที่เหมาะสม ในข้อนี้ใช้วัดความพอใจเกี่ยวกับความกว้างบรรทัดของเนื้อหาของคู่มือที่ออกแบบตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.4.4 ความกว้างของบรรทัดในคู่มือการใช้งานเฟรมเวิร์ก

ข้อที่ 13 มีการใช้สีที่อ่านได้ง่าย ในข้อนี้ใช้วัดความพอใจเกี่ยวกับการใช้สีที่ช่วยในการอ่านที่ออกแบบตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.3 การเลือกใช้สีสำหรับคู่มือการใช้งานเฟรมเวิร์ก และ 4.2.4.7 การใช้สีกับตัวอักษร

ข้อที่ 14 มีการใช้สีในการเน้นข้อความที่เหมาะสม ในข้อนี้ใช้วัดความพอใจเกี่ยวกับการใช้สีในการเน้นข้อความตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.4.6 การเน้นข้อความให้เด่นชัด

ข้อที่ 15 มีการใช้ขนาดตัวอักษรในการเน้นข้อความที่เหมาะสม ในข้อนี้ใช้วัดความพอใจเกี่ยวกับขนาดตัวอักษรที่ใช้ในการเน้นข้อความของคู่มือเฟรมเวิร์กตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.4.5 ขนาดของตัวอักษร และ 4.2.4.6 การเน้นข้อความให้เด่นชัด

ข้อที่ 16 มี การใช้แบบอักษรที่เหมาะสมกับแต่ละส่วนของเนื้อหา ในข้อนี้ใช้วัดความพอใจเกี่ยวกับแบบอักษรที่ใช้ในแต่ละส่วนของเนื้อหาในคู่มือเฟรมเวอร์ก ตามรูปแบบที่เสนอไว้ในหัวข้อที่ 4.2.4.1 ประเภทของตัวอักษรที่เหมาะสมกับเนื้อหาในคู่มือเฟรมเวอร์ก

### 5.3 การดำเนินการวิจัยและเก็บรวบรวมข้อมูล

ผู้วิจัยได้ดำเนินการเก็บรวบรวมข้อมูลโดยมีวัตถุประสงค์ 2 ประการ ได้แก่ การเปรียบเทียบระยะเวลาการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวอร์กจากงานวิจัยกับคู่มือเฟรมเวอร์กของบริษัทชั้นไมโครซีสเต็มส์ และการเปรียบเทียบความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวอร์กระหว่างคู่มือเฟรมเวอร์กจากงานวิจัยกับคู่มือเฟรมเวอร์กของบริษัทชั้นไมโครซีสเต็มส์ โดยมีขั้นตอนดำเนินการวิจัย ดังนี้

5.3.1 เปรียบเทียบระยะเวลาการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวอร์กจากงานวิจัยกับคู่มือเฟรมเวอร์กของบริษัทชั้นไมโครซีสเต็มส์

5.3.1.1 ให้กลุ่มตัวอย่างจับฉลากแบ่งเป็นกลุ่มควบคุมและกลุ่มทดลอง ดังนี้

1) กลุ่มทดลอง คือ กลุ่มที่ใช้งานคู่มือเฟรมเวอร์กที่ได้จากการวิจัย จำนวน 14 คน

2) กลุ่มควบคุม คือ กลุ่มที่ใช้งานคู่มือเฟรมเวอร์กของบริษัทชั้นไมโครซีสเต็มส์ จำนวน 13 คน

5.3.1.2 สอนความรู้พื้นฐานเกี่ยวกับเฟรมเวอร์กให้กับกลุ่มตัวอย่างทั้ง 2 กลุ่ม

5.3.1.3 จับเวลาการพัฒนาโปรแกรมประยุกต์ของทั้ง 2 กลุ่ม

5.3.2 เปรียบเทียบความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวอร์กระหว่างคู่มือเฟรมเวอร์กจากงานวิจัยกับคู่มือเฟรมเวอร์กของบริษัทชั้นไมโครซีสเต็มส์

ให้กลุ่มตัวอย่างที่ได้พัฒนาโปรแกรมประยุกต์ตามขั้นตอนการดำเนินการวิจัยที่ 5.3.1 เรียบร้อยแล้ว ตอบแบบสอบถามเพื่อวัดความพอใจในการใช้งานคู่มือเฟรมเวอร์ก ดังนี้

1) กลุ่มทดลอง ตอบแบบสอบถามวัดความพอใจในการใช้งานคู่มือเฟรมเวอร์กที่ได้จากการวิจัย จำนวน 14 คน

2) กลุ่มควบคุม ตอบแบบสอบถามวัดความพอใจในการใช้งานคู่มือเฟรมเวอร์กของบริษัทชั้นไมโครซีสเต็มส์ จำนวน 13 คน

หมายเหตุ เนื่องจากการใช้งานคู่มือเฟรมเวอร์กก่อนและหลังมีผลต่อการเรียนรู้การใช้งานเฟรมเวอร์กของกลุ่มตัวอย่าง ดังนั้นจึงไม่สามารถเปรียบเทียบประสิทธิภาพการใช้งานระหว่าง

คู่มือเฟรมเวิร์กที่ได้จากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นนำไมโครซีเอสเต็มส์จากผู้ทดลองคนเดียว

5.3.3 วิเคราะห์เปรียบเทียบผลการทดลองที่ได้จากการจับเวลาการพัฒนาโปรแกรมประยุกต์และการตอบแบบสอบถามเกี่ยวกับความพอใจในการใช้งานคู่มือเฟรมเวิร์กโดยวิธีทางสถิติ t-test Independent และ สถิติ Mann-Whitney U Test ตามลำดับ

## 5.4 การวิเคราะห์ข้อมูล

งานวิจัยนี้ได้นำข้อมูลมาวิเคราะห์ด้วยโปรแกรม SPSS FOR WINDOWS รุ่น 10.0.7 โดยมีวิธีทางสถิติ ดังนี้

### 5.4.1 สถิติทดสอบค่าที (t-test)

ใช้สถิตินี้เพื่อวิเคราะห์เปรียบเทียบระยะเวลาการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นนำไมโครซีเอสเต็มส์

#### ข้อกำหนดเบื้องต้น

- 1) กลุ่มตัวอย่างแต่ละกลุ่มได้มาโดยการสุ่มและเป็นอิสระต่อกัน
- 2) กลุ่มตัวอย่างแต่ละกลุ่มได้มาจากประชากรที่มีการแจกแจงเป็นโค้งปกติ
- 3) ขนาดของกลุ่มตัวอย่างมีขนาดเล็ก และไม่ทราบค่าความแปรปรวน

$$(n_1 < 30 \text{ และ } n_2 < 30)$$

#### สมมติฐาน

$$H_0: \mu_1 \geq \mu_2$$

$$H_1: \mu_1 < \mu_2$$

เมื่อ  $\mu_1$  และ  $\mu_2$  แทน ค่าเฉลี่ยระยะเวลาการพัฒนาโปรแกรมของกลุ่มทดลอง และกลุ่มควบคุม ตามลำดับ

## สถิติทดสอบ

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$$

$$df = \frac{\left[ \frac{S_1^2}{n_1} + \frac{S_2^2}{n_2} \right]^2}{\frac{\left[ \frac{S_1^2}{n_1} \right]^2}{n_1 - 1} + \frac{\left[ \frac{S_2^2}{n_2} \right]^2}{n_2 - 1}}$$

เมื่อ	t	แทน	ค่าพิจารณาใน t-distribution
	$\bar{X}_1$	แทน	เวลาเฉลี่ยในการพัฒนาโปรแกรมประยุกต์ของกลุ่มทดลอง
	$\bar{X}_2$	แทน	เวลาเฉลี่ยในการพัฒนาโปรแกรมประยุกต์ของกลุ่มควบคุม
	df	แทน	ชั้นแห่งความเป็นอิสระ
	$S_1^2$	แทน	ค่าความแปรปรวนของกลุ่มทดลอง
	$S_2^2$	แทน	ค่าความแปรปรวนของกลุ่มควบคุม
	$n_1$	แทน	จำนวนนักศึกษาในกลุ่มทดลอง
	$n_2$	แทน	จำนวนนักศึกษาในกลุ่มควบคุม

## การตัดสินใจ

จะปฏิเสธ  $H_0$  ถ้า ค่า p-value ที่ได้มีค่าน้อยกว่าค่าความเชื่อมั่นที่กำหนด โดยในการวิจัยนี้จะวัดความเชื่อมั่นที่ 95 เปอร์เซนต์ ดังนั้นจะปฏิเสธ  $H_0$  และยอมรับ  $H_1$  ถ้าค่า p-value < 0.05

## 5.4.2 สถิติทดสอบของวิลคอกซ์ แมนวิทนี (Mann-Whitney U Test)

สถิตินี้ใช้เพื่อวิเคราะห์เปรียบเทียบความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวิร์ก ระหว่างคู่มือการใช้งานเฟรมเวิร์กจากงานวิจัยกับคู่มือการใช้งานเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

## ข้อกำหนดเบื้องต้น

- 1) ข้อมูลประกอบด้วยตัวอย่างสุ่ม ด้วยค่า  $X_1, X_2, \dots, X_n$  จากกลุ่มทดลอง และตัวอย่างสุ่มอีก 1 ชุด ด้วยค่า  $Y_1, Y_2, \dots, Y_n$  จากกลุ่มควบคุม
- 2) ตัวอย่าง 2 ชุดนี้ เป็นอิสระต่อกัน
- 3) ค่าตัวแปรสุ่มมีค่าต่อเนื่อง
- 4) มาตรการวัดอย่างน้อยเป็นแบบเรียงลำดับ (ordinal scale)

## สมมติฐาน

$$H_0: M_x \leq M_y$$

$$H_1: M_x > M_y$$

เมื่อ  $M_x$  และ  $M_y$  แทน ค่ามัธยฐานความพอใจในการใช้งานเฟรมเวิร์กของกลุ่มทดลอง และกลุ่มควบคุม ตามลำดับ

## สถิติทดสอบ

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - S_1$$

$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - S_2$$

$$Z = \frac{U - n_1 n_2 / 2}{\sqrt{n_1 n_2 (n_1 + n_2 + 1) / 12}}$$

เมื่อ	$U_1$	แทน	ค่า U ของกลุ่มทดลอง
	$U_2$	แทน	ค่า U ของกลุ่มควบคุม
	$U$	แทน	ค่าที่น้อยกว่าระหว่าง $U_1$ กับ $U_2$
	$S_1$	แทน	ผลรวมลำดับที่ของกลุ่มทดลอง
	$S_2$	แทน	ผลรวมลำดับที่ของกลุ่มควบคุม
	$n_1$	แทน	จำนวนกลุ่มทดลอง
	$n_2$	แทน	จำนวนกลุ่มควบคุม

### การตัดสินใจ

จะปฏิเสธ  $H_0$  ถ้า ค่า p-value ที่ได้มีค่าน้อยกว่าค่าความเชื่อมั่นที่กำหนด โดยในการวิจัยนี้จะวัดความเชื่อมั่นที่ 95 เปอร์เซ็นต์ ดังนั้นจะปฏิเสธ  $H_0$  และยอมรับ  $H_1$  ถ้าค่า p-value  $< 0.05$

## บทที่ 6

### ผลการวิเคราะห์ข้อมูล

จากข้อมูลที่รวบรวมได้ในบทที่ 5 ผู้วิจัยได้นำมาวิเคราะห์ด้วยโปรแกรมสำเร็จรูปทางสถิติ SPSS FOR WINDOWS รุ่น 10.0.7 ในหัวข้อต่อไปนี้

1. การวิเคราะห์เปรียบเทียบระยะเวลาการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์
2. การวิเคราะห์เปรียบเทียบความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวิร์กระหว่างคู่มือการใช้งานเฟรมเวิร์กจากงานวิจัยกับคู่มือการใช้งานเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

#### 6.1 การวิเคราะห์เปรียบเทียบระยะเวลาการพัฒนาโปรแกรมประยุกต์ระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

จากการจับเวลาการพัฒนาโปรแกรมของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัย และคู่มือของบริษัทชั้นไมโครซีเอสเต็มส์ สามารถแสดงผลการทดลองเป็นค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐาน ดังตารางที่ 6.1

ตารางที่ 6.1 แสดงค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของระยะเวลา (ชั่วโมง) ที่ใช้ในการพัฒนาโปรแกรมของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยและคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

	คู่มือชั้นไมโครซีเอสเต็มส์		คู่มือจากงานวิจัย	
	$\bar{X}$	SD.	$\bar{X}$	SD.
เวลาในการพัฒนาโปรแกรมประยุกต์	1:53	0.23	1:22	0.25

การเปรียบเทียบค่าเฉลี่ยของระยะเวลาในการพัฒนาโปรแกรมประยุกต์ระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์ ทำได้โดยการทดสอบด้วยค่าที ซึ่งมีสมมติฐานและผลการทดสอบ ดังนี้

### สมมติฐานที่ทดสอบ

$H_0$  : ระยะเวลาเฉลี่ยของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยในการพัฒนาโปรแกรมไม่น้อยกว่าค่าเฉลี่ยระยะเวลาของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์

$H_1$  : ระยะเวลาเฉลี่ยของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยในการพัฒนาโปรแกรมน้อยกว่าค่าเฉลี่ยระยะเวลาของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์

ตารางที่ 6.2 ผลการเปรียบเทียบความแตกต่างของค่าเฉลี่ยระยะเวลา (ชั่วโมง) ที่ใช้ในการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์

t	df	p-value
3.233	25	0.006

เมื่อพิจารณาจากตารางที่ 6.2 ค่า p-value มีค่าเท่ากับ 0.006 ซึ่งน้อยกว่า 0.05 (ระดับนัยสำคัญทางสถิติที่ 0.05) ดังนั้น จึงยอมรับ  $H_1$  ที่ว่า ระยะเวลาเฉลี่ยของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยในการพัฒนาโปรแกรมประยุกต์น้อยกว่าค่าเฉลี่ยของระยะเวลาของนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีสเต็มส์

## 6.2 การวิเคราะห์เปรียบเทียบความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวิร์ก ระหว่างคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

การวิเคราะห์จะเปรียบเทียบค่ามัธยฐานของคะแนนการตอบแบบสอบถามของคู่มือการใช้งานเฟรมเวิร์กที่ได้จากงานวิจัยเปรียบเทียบกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์ว่าแตกต่างกันหรือไม่ ซึ่งในแบบสอบถามมีทั้งหมด 16 ข้อแบ่งเป็น 2 หมวด ได้แก่ ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก 7 ข้อ (ข้อ 1 – 7) และด้านส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก 9 ข้อ (ข้อ 8 – 16) โดยคะแนนความพอใจจากแบบสอบถามมีความหมาย ดังนี้

5	หมายถึง	มากที่สุด
4	หมายถึง	มาก
3	หมายถึง	ปานกลาง
2	หมายถึง	น้อย
1	หมายถึง	น้อยที่สุด

จากการตอบแบบสอบถามเรื่องความพอใจในการใช้งานคู่มือเฟรมเวิร์กสามารถแสดงผลการทดลองเป็นค่ามัธยฐาน ดังตารางที่ 6.3

ตารางที่ 6.3 แสดงค่ามัธยฐานคะแนนความพอใจในแต่ละข้อจากการใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยและคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

แบบสอบถาม	ค่ามัธยฐาน	
	คู่มือชั้นไมโครซีเอสเต็มส์	คู่มือจากงานวิจัย
ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก		
1. มีหัวข้อในการแสดงเนื้อหาที่เพียงพอต่อการพัฒนาโปรแกรมประยุกต์	3.00	4.00
2. มีการเรียงลำดับส่วนประกอบของคู่มือที่ง่ายต่อการใช้งาน	2.00	4.00
3. สามารถค้นหาข้อมูลในคู่มือเฟรมเวิร์กได้ง่าย	2.00	4.00
4. มีส่วนที่ทำให้เห็นภาพรวมของเฟรมเวิร์กได้ง่าย	2.00	5.00
5. มีส่วนที่ทำให้ผู้ใช้เห็นความสัมพันธ์ของส่วนประกอบภายในเฟรมเวิร์กได้ง่าย	2.00	4.00
6. มีตัวอย่างโปรแกรมประยุกต์ที่เหมาะสม	1.00	4.50
7. มีคำอธิบายหรือส่วนที่เชื่อมโยงไปยังเนื้อหาอื่นได้ง่าย ไม่สับสน	2.00	4.00
ด้านส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก		
8. มีส่วนหัว (header) ของแต่ละหน้าที่เหมาะสม	2.00	4.50
9. มีส่วนท้ายของเอกสาร (footer) ที่เหมาะสม	2.00	4.00
10. มีการจัดตัวอักษรในเนื้อหาที่เหมาะสม	2.00	5.00
11. มีขนาดตัวอักษรในเนื้อหาที่เหมาะสม	3.00	5.00
12. มีความกว้างบรรทัดของเนื้อหาที่เหมาะสม	2.00	4.50
13. มีการใช้สีที่อ่านได้ง่าย	3.00	4.50
14. มีการใช้สีในการเน้นข้อความที่เหมาะสม	3.00	5.00
15. มีการใช้ขนาดตัวอักษรในการเน้นข้อความที่เหมาะสม	3.00	4.50
16. มีการใช้แบบอักษร (font) ที่เหมาะสมกับแต่ละส่วนของเนื้อหา	3.00	4.50

จากคะแนนความพอใจของการตอบแบบสอบถามในทุกข้อ สามารถนำมาเปรียบเทียบค่ามัธยฐานของคะแนนความพอใจโดยรวมของคู่มือการใช้งานเฟรมเวิร์กที่ได้จากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์ โดยใช้สถิติทดสอบ Mann-Whitney U Test ซึ่งมีสมมติฐานและผลการทดสอบ ดังนี้

### สมมติฐานที่ทดสอบ

- $H_0$  : ค่ามัธยฐานคะแนนความพอใจโดยรวมของการใช้งานคู่มือเฟรมเวิร์กที่ได้จากการวิจัยมีค่าไม่มากกว่าค่ามัธยฐานคะแนนความพอใจในการใช้งานคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์
- $H_1$  : ค่ามัธยฐานคะแนนความพอใจโดยรวมของการใช้งานคู่มือเฟรมเวิร์กที่ได้จากการวิจัยมีค่ามากกว่าค่ามัธยฐานคะแนนความพอใจในการใช้งานคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

ตารางที่ 6.4 ผลการเปรียบเทียบความแตกต่างของค่ามัธยฐานคะแนนความพอใจโดยรวมในการใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

U	Z	p-value
1669.000	-17.140	0.000

เมื่อพิจารณาข้อมูลจากตารางที่ 6.4 พบว่า p-value มีค่าเท่ากับ 0.000 ซึ่งน้อยกว่า 0.05 (ระดับนัยสำคัญทางสถิติที่ 0.05) ดังนั้น จึงยอมรับ  $H_1$  ที่ว่า ค่ามัธยฐานคะแนนความพอใจโดยรวมของการใช้งานคู่มือเฟรมเวิร์กที่ได้จากการวิจัยมีค่ามากกว่าค่ามัธยฐานคะแนนความพอใจในการใช้งานคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

เพื่อแสดงให้เห็นถึงปัจจัยในแต่ละข้อของคู่มือที่ได้จากงานวิจัยที่มีผลต่อการใช้งานคู่มือเฟรมเวิร์กเทียบกับคู่มือของบริษัทชั้นไมโครซีเอสเต็มส์ ผู้วิจัยจึงได้เปรียบเทียบค่ามัธยฐานของคะแนนความพอใจจากการตอบแบบสอบถามในแต่ละข้อ ด้วยสถิติทดสอบ Mann-Whitney U Test โดยมีสมมติฐานและผลการทดสอบ ดังนี้

### สมมติฐานที่ทดสอบ

- $H_0$  : ค่ามัธยฐานคะแนนความพอใจในแต่ละข้อของการใช้งานคู่มือเฟรมเวิร์กที่ได้จากการวิจัยไม่มากกว่าค่ามัธยฐานคะแนนความพอใจในแต่ละข้อของการใช้งานคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์
- $H_1$  : ค่ามัธยฐานคะแนนความพอใจในแต่ละข้อของการใช้งานคู่มือเฟรมเวิร์กที่ได้จากการวิจัยมากกว่าค่ามัธยฐานคะแนนความพอใจในแต่ละข้อของการใช้งานคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

ตารางที่ 6.5 ผลการเปรียบเทียบความแตกต่างของค่ามัธยฐานคะแนนความพอใจในแต่ละหัวข้อในการใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นนำโครซีเอสเต็มส์

แบบสอบถาม	U	Z	p-value
ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก			
1. มีหัวข้อในการแสดงเนื้อหาที่เพียงพอต่อการพัฒนาโปรแกรมประยุกต์	14.000	-3.924	0.000
2. มีการเรียงลำดับส่วนประกอบของคู่มือที่ง่ายต่อการใช้งาน	11.500	-4.033	0.000
3. สามารถค้นหาข้อมูลในคู่มือเฟรมเวิร์กได้ง่าย	0.500	-4.627	0.000
4. มีส่วนที่ทำให้เห็นภาพรวมของเฟรมเวิร์กได้ง่าย	1.000	-4.578	0.000
5. มีตัวอย่างโปรแกรมประยุกต์ที่เหมาะสม	0.000	-4.544	0.000
6. มีส่วนที่ทำให้ผู้ใช้เห็นความสัมพันธ์ของส่วนประกอบภายในเฟรมเวิร์กได้ง่าย	3.000	-4.408	0.000
7. มีคำอธิบายหรือส่วนที่เชื่อมโยงไปยังเนื้อหาอื่นได้ง่าย ไม่สับสน	6.000	-4.249	0.000
ด้านส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก			
8. มีส่วนหัว (header) ของแต่ละหน้าที่เหมาะสม	5.500	-4.286	0.000
9. มีการจัดตัวอักษรในเนื้อหาที่เหมาะสม	2.000	-4.430	0.000
10. มีขนาดตัวอักษรในเนื้อหาที่เหมาะสม	0.000	-4.563	0.000
11. มีความกว้างบรรทัดของเนื้อหาที่เหมาะสม	2.500	-4.419	0.000
12. มีส่วนท้ายของเอกสาร (footer) ที่เหมาะสม	9.000	-4.129	0.000
13. มีการใช้สีที่อ่านได้ง่าย	8.000	-4.181	0.000
14. มีการใช้สีในการเน้นข้อความที่เหมาะสม	16.000	-3.771	0.000
15. มีการใช้ขนาดตัวอักษรในการเน้นข้อความที่เหมาะสม	11.500	-4.000	0.000
16. มีการใช้แบบอักษร (font) ที่เหมาะสมกับแต่ละส่วนของเนื้อหา	3.500	-4.393	0.000

เมื่อพิจารณาข้อมูลจากตารางที่ 6.5 พบว่าแบบสอบถามแต่ละข้อมีค่า p-value น้อยกว่า 0.05 (ระดับนัยสำคัญทางสถิติที่ 0.05) ดังนั้น จึงยอมรับ  $H_1$  ที่ว่า ค่ามัธยฐานของคะแนนความพอใจในแต่ละข้อ ได้แก่ ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก (ข้อ 1 – 7) และด้านส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก (ข้อ 8 – 16) ของการใช้งานคู่มือเฟรมเวิร์กที่ได้จากการวิจัยมีค่ามากกว่าค่ามัธยฐานของคะแนนความพอใจในการใช้งานคู่มือเฟรมเวิร์กของบริษัทชันไมโครซิสเต็มส์

## การอภิปรายผลการทดลอง

จากการออกแบบรูปแบบการใช้งานคู่มือเฟรมเวิร์กสำหรับผู้พัฒนาโปรแกรมประยุกต์ที่ผสมผสานระหว่างรูปแบบการเขียนคู่มือการใช้งานเฟรมเวิร์กที่มีมากับทฤษฎีทางจิตวิทยาเกี่ยวกับการเรียนรู้ของมนุษย์ในการออกแบบส่วนติดต่อกับผู้ใช้ ผู้วิจัยได้นำรูปแบบที่ได้จากการวิจัยนี้มาเขียนเป็นคู่มือการใช้งานคอเล็กชั่นเฟรมเวิร์ก และทำการทดสอบประสิทธิภาพด้วยการจับเวลาการพัฒนาโปรแกรมประยุกต์และเก็บข้อมูลแบบสอบถามเกี่ยวกับความพอใจในการใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยเปรียบเทียบกับคู่มือเฟรมเวิร์กของบริษัทชันไมโครซิสเต็มส์ สามารถนำมาอภิปรายผลการทดลองได้ ดังนี้

### 7.1 ระยะเวลาในการพัฒนาโปรแกรมประยุกต์จากการใช้งานคู่มือเฟรมเวิร์ก

จากการทดลองนำนักศึกษาที่ไม่เคยพัฒนาโปรแกรมประยุกต์ด้วยจาวาคอเล็กชั่นเฟรมเวิร์ก มาใช้งานคู่มือเฟรมเวิร์กในการพัฒนาโปรแกรมประยุกต์ พบว่า กลุ่มที่ใช้งานคู่มือเฟรมเวิร์กที่เขียนขึ้นจากรูปแบบที่ได้จากงานวิจัยใช้เวลาการพัฒนาโปรแกรมเฉลี่ย 1 ชั่วโมง 22 นาที ซึ่งน้อยกว่ากลุ่มที่ใช้งานคู่มือเฟรมเวิร์กของบริษัทชันไมโครซิสเต็มส์ที่ใช้เวลาในการพัฒนาโปรแกรมเฉลี่ย 1 ชั่วโมง 53 นาที อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

จากผลการทดลองนี้พบว่า ถึงแม้ว่าระยะเวลาที่ใช้ในการพัฒนาโปรแกรมประยุกต์จากคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยจะดีกว่าคู่มือของบริษัทชันไมโครซิสเต็มส์ แต่เวลาที่ใช้ในการพัฒนาโปรแกรมประยุกต์ยังมากอยู่ ซึ่งอาจเกิดจากกลุ่มผู้ทดลองมีข้อจำกัดในการพิมพ์และความเร็วในการอ่าน ซึ่งถ้าการทดลองในอนาคตสามารถลดข้อจำกัดเหล่านี้ออกไปได้ ผู้วิจัยเชื่อว่ารูปแบบคู่มือที่ได้จากงานวิจัยจะสามารถลดระยะเวลาในการศึกษาการใช้งานเฟรมเวิร์กได้มากขึ้น

### 7.2 ความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวิร์ก

จากการวิเคราะห์ข้อมูลแบบสอบถามวัดความพอใจในการใช้งานคู่มือเฟรมเวิร์กของกลุ่มตัวอย่างที่ใช้งานคู่มือเฟรมเวิร์กที่ได้จากงานวิจัยและกลุ่มตัวอย่างที่ใช้งานคู่มือเฟรมเวิร์กของบริษัทชันไมโครซิสเต็มส์ พบว่า กลุ่มตัวอย่างแรกมีความพอใจทั้งโดยรวมและรายข้อมากกว่ากลุ่มตัวอย่างที่สอง อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 โดยสามารถสรุปหัวข้อจากแบบสอบถามเพื่อเชื่อมโยงไปยังรูปแบบคู่มือเฟรมเวิร์กที่น่าเสนอไว้ในงานวิจัยนี้ได้ ดังนี้

### 7.2.1 ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก มีทั้งหมด 7 ข้อ ได้แก่

ข้อที่ 1 มีหัวข้อในการแสดงเนื้อหาที่เพียงพอต่อการพัฒนาโปรแกรมประยุกต์ จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.1 ส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก

ข้อที่ 2 มีการเรียงลำดับส่วนประกอบของคู่มือที่ง่ายต่อการใช้งาน จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในรูปที่ 4.1 แสดงส่วนประกอบของเนื้อหาหลัก และ 4.2 แสดงส่วนประกอบของเนื้อหาในแต่ละเรขีพี

ข้อที่ 3 มีความสามารถค้นหาข้อมูลในคู่มือเฟรมเวิร์กได้ง่าย จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.1.3 สรุปรายการคำสั่ง

ข้อที่ 4 มีส่วนที่ทำให้เห็นภาพรวมของเฟรมเวิร์กได้ง่าย จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.1.1 ภาพรวม

ข้อที่ 5 มีส่วนที่ทำให้ผู้ใช้เห็นความสัมพันธ์ของส่วนประกอบภายในเฟรมเวิร์กได้ง่าย จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.1.1 ภาพรวม

ข้อที่ 6 มีตัวอย่างโปรแกรมประยุกต์ที่เหมาะสม จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.1.2 คำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์

ข้อที่ 7 มีคำอธิบายหรือส่วนที่เชื่อมโยงเนื้อหาอื่นได้ง่าย ไม่สับสน จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.1.4 อ้างอิง

### 7.2.2 ด้านส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก ได้แก่

ข้อที่ 8 มีส่วนหัวของแต่ละหน้าที่เหมาะสม จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.2.1 การจัดวางส่วนหัวในแต่ละหน้า

ข้อที่ 9 มีส่วนท้ายของเอกสารที่เหมาะสม จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.2.3 การจัดวางส่วนท้ายในแต่ละหน้า

ข้อที่ 10 มีการจัดตัวอักษรในเนื้อหาที่เหมาะสม จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.2.2 การจัดวางส่วนเนื้อหาในแต่ละหน้า และ 4.2.4.3 การจัดตำแหน่งเนื้อหาในคู่มือ

ข้อที่ 11 มี ขนาดตัวอักษรในเนื้อหาที่เหมาะสม จากแบบสอบถามข้อนี้ สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.2.5 ขนาดของตัวอักษร

ข้อที่ 12 มีความกว้างบรรทัดของเนื้อหาที่เหมาะสม จากแบบสอบถาม ข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.4.4 ความกว้างของ บรรทัดในคู่มือการใช้งานเฟรมเวิร์ก

ข้อที่ 13 มีการใช้สีที่อ่านได้ง่าย จากแบบสอบถามข้อนี้สามารถบ่งชี้ถึง ความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.3 การเลือกใช้สีสำหรับคู่มือการใช้งานเฟรม- เวิร์ก และ 4.2.4.7 การใช้สีกับตัวอักษร

ข้อที่ 14 มีการใช้สีในการเน้นข้อความที่เหมาะสม จากแบบสอบถาม ข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.4.6 การเน้นข้อความ ให้เด่นชัด

ข้อที่ 15 มีการใช้ขนาดตัวอักษรในการเน้นข้อความที่เหมาะสม จาก แบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.4.5 ขนาด ของตัวอักษร และ 4.2.4.6 การเน้นข้อความให้เด่นชัด

ข้อที่ 16 มีการใช้แบบอักษรที่เหมาะสมกับแต่ละส่วนของเนื้อหา จาก แบบสอบถามข้อนี้สามารถบ่งชี้ถึงความสำคัญของรูปแบบที่ได้นำเสนอไว้ในหัวข้อที่ 4.2.4.1 ประเภทของตัวอักษรที่เหมาะสมกับเนื้อในคู่มือเฟรมเวิร์ก

## บทที่ 8

# สรุปผลการวิจัยและข้อเสนอแนะ

### 8.1 สรุปผลการวิจัย

จากวัตถุประสงค์ของงานวิจัยในการออกแบบรูปแบบคู่มือการใช้งานเฟรมเวิร์กที่เป็นมาตรฐานสำหรับผู้พัฒนาโปรแกรมประยุกต์ เพื่อช่วยลดระยะเวลาในการศึกษาการใช้งานเฟรมเวิร์ก และมีส่วนปฏิสัมพันธ์ที่ดีกับผู้ใช้ คู่มือเฟรมเวิร์กที่พัฒนาขึ้นจะพิจารณาใน 2 ประเด็น ได้แก่ ส่วนของเนื้อหาและส่วนติดต่อกับผู้ใช้ โดยในส่วนของเนื้อหาผู้วิจัยได้วิเคราะห์ข้อดีและข้อเสียของคู่มือเฟรมเวิร์กที่มีอยู่ในปัจจุบัน และความต้องการข้อมูลคู่มือพื้นฐานของผู้พัฒนาโปรแกรมประยุกต์ในมุมมองวิจัย [4] [7] จนได้ข้อสรุปถึงหัวข้อที่จำเป็นต้องมีในส่วนของเนื้อหาและการเรียงลำดับของหัวข้อเหล่านั้นในหัวข้อที่ 4.3 ทางด้านส่วนติดต่อกับผู้ใช้ ผู้วิจัยได้นำความรู้ทางด้านจิตวิทยาที่เกี่ยวข้องกับการเรียนรู้ของมนุษย์ การออกแบบส่วนติดต่อกับผู้ใช้ และการออกแบบส่วนติดต่อกับผู้ใช้นบนเว็บเบราว์เซอร์ มาประยุกต์ใช้ให้เข้ากับรูปแบบของคู่มือการใช้งานเฟรมเวิร์ก การทดสอบผู้วิจัยได้เลือกคู่มือของบริษัทซัมไมโครซิสเต็มส์เป็นตัวแทนของคู่มือเฟรมเวิร์กที่มีอยู่ เนื่องจากคู่มือของบริษัทซัมไมโครซิสเต็มส์เป็นคู่มือที่มากับซอฟต์แวร์ของบริษัทซัมไมโครซิสเต็มส์ และกลุ่มผู้พัฒนาโปรแกรมประยุกต์ใช้งานกันอย่างแพร่หลาย การทดสอบได้แบ่งกลุ่มตัวอย่างออกเป็น 2 กลุ่ม คือ กลุ่มควบคุมที่ใช้งานคู่มือเฟรมเวิร์กของบริษัทซัมไมโครซิสเต็มส์ และกลุ่มทดลองที่ใช้งานคู่มือเฟรมเวิร์กที่พัฒนาขึ้น จากการทดสอบสรุปได้ว่ากลุ่มทดลองใช้เวลาในการพัฒนาโปรแกรมประยุกต์น้อยกว่ากลุ่มควบคุมอย่างมีนัยสำคัญ และกลุ่มทดลองมีความพอใจในคู่มือเฟรมเวิร์กมากกว่ากลุ่มควบคุมอย่างมีนัยสำคัญ จากผลการทดลองในส่วนของเวลาในการพัฒนาโปรแกรมประยุกต์สามารถบ่งชี้ได้ว่าคู่มือดังกล่าวสามารถลดระยะเวลาในการศึกษาการใช้งานเฟรมเวิร์ก และผลจากการทดสอบความพอใจสามารถบ่งชี้ได้ว่า รูปแบบการเขียนคู่มือเฟรมเวิร์กที่เสนอจากงานวิจัยนี้สามารถนำมาเป็นมาตรฐานในการพัฒนาคู่มือการใช้งานเฟรมเวิร์กที่ตอบสนองความพอใจของผู้ใช้ได้ในระดับหนึ่ง

### 8.2 ข้อเสนอแนะ

8.2.1 รูปแบบที่ได้จากการวิจัยนี้ สามารถพัฒนาต่อให้เป็นเครื่องมือสำหรับสร้างคู่มือเฟรมเวิร์กที่ช่วยให้ผู้เขียนคู่มือเฟรมเวิร์กพัฒนาคู่มือได้รวดเร็ว และแก้ไขเปลี่ยนแปลงเนื้อหาได้ง่าย

8.2.2 การทดสอบควรทำการเปรียบเทียบกับคู่มือเฟรมเวิร์กและกลุ่มผู้ทดสอบที่มีความหลากหลายมากขึ้น เพื่อปรับปรุงรูปแบบให้ดีขึ้น

8.2.3 รูปแบบที่ได้จากการวิจัยนี้สามารถนำไปพัฒนาต่อให้ใช้ได้กับผู้เกี่ยวข้องกับเฟรมเวิร์กทุกกลุ่ม เช่น มีฐานข้อมูลรูปแบบคู่มือเฟรมเวิร์กกลางที่ใช้งานได้กับทุกกลุ่ม และมีฐานข้อมูลเฉพาะสำหรับผู้ที่เกี่ยวข้องกับเฟรมเวิร์กแต่ละกลุ่ม เป็นต้น

8.2.4 งานวิจัยนี้ยังไม่ได้ทำการวิเคราะห์ว่าปัจจัยใดเป็นปัจจัยที่มีอิทธิพลต่อความพอใจของผู้พัฒนาโปรแกรมประยุกต์ ซึ่งผู้วิจัยเชื่อว่าการวิเคราะห์ข้อมูลดังกล่าวจะเป็นประโยชน์ต่อผู้พัฒนาคู่มือเฟรมเวิร์กให้ตระหนักว่าปัจจัยข้อใดเป็นปัจจัยที่ควรจะให้มีความสำคัญเป็นพิเศษ

## เอกสารอ้างอิง

- [1] Ademar A. "A Minimalist Approach to Framework Documentation" OOPSLA 2000 Companion Minneapolis, Minnesota, 2000. pp.143-144.
- [2] Adobe System. Postscript Language – Tutorial and Cookbook. Reading : Addison-Wesley. 1985.
- [3] Apple Computer : MacApp Programmer's Guide. 1989.
- [4] Butler G. Peter G. and Ferhat K. "A Reuse Case Perspective on Documenting Frameworks." Software Engineering Conference, 1998. pp. 94 –101.
- [5] Butler G. and Denommee P. Documenting Framework to Assist Application Developers. In Object Application Frameworks. New York : John Wiley and Sons, 1999.
- [6] Butler G. and Denommee P. "Documenting Framework.", Concordia University. [Online]. Available : <http://www.cs.concordia.ca/~faculty/gregb/home.html> 2000.
- [7] Butler G. Rudolf K.K. and Hafedh M. "A Framework for Framework Documentation." ACM Computing Surveys (CSUR), 2000.
- [8] Codeonie W., De Hondt K., Steyaert P. and Vercammen A. Evolving Custom-Made Application into Domain-Specific Frameworks. In Communications of the ACM. 1997.
- [9] Krasner G.E. and Pope S.T. "A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80." Journal of Object-Oriented Programming, 1988.
- [10] Froehlich G., Hoover H.J., Liu L. and Sorenson P. "Hooking into Object-Oriented Application Frameworks." Proceedings of the 19th International Conference on Software Engineering, 1997. pp. 491-501.
- [11] Gamma E. Richard, H. Ralph J. and John V. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley. 1995.
- [12] Johnson R. "Documenting Frameworks Using Patterns." Proceedings of OOPSLA 1992. New York: ACM/SIGPLAN. 1992. pp.63-76.

- [13] Johnson R.E. and Brian F. "Designing Reusable Classes." *Journal of Object-Oriented Programming*. 1(5), June/July 1988. pp.22-35.
- [14] Lajoie R. and Keller R.K. "Design and Reuse in Object-Oriented Frameworks: Patterns, Contracts, and Motifs in Concert." In *Object-Oriented Technology for Database and Software Systems*. Singapore: World Scientific Publishing. 1995. pp.295-312.
- [15] Parsons D., Rashid A., Speck A. and Telea A. "A Framework for Object Oriented Framework Design." *IEEE Computer*. June 1999. pp.141 – 151.
- [16] Pree W. *Design Patterns for Object-Oriented Software Development*. Reading, MA: Addison-Wesley, 1995.
- [17] Qinxue C. and Fangxiao X. "Framework Issue:Framework Documentation." University of Nevada , Redo. [Online]. Available : [http://www.cs.unr.edu/~chen\\_q/docu.html](http://www.cs.unr.edu/~chen_q/docu.html)
- [18] Sparks S. Benner K. and Faris C. "Managing Object-Oriented Framework Reuse." *IEEE Computer*, Theme Issue on Managing Object-Oriented Software Development, Mohamed E. Fayad and Marshall Cline, editors. 29(9) : 1996. pp.52-61.
- [19] Mohamed E.F. Douglas C.S. and Ralph E.J. *Building Application Frameworks Object-oriented Foundations of Framework Design*. New York : John Wiley & Sons, 1999.
- [20] Sandro P. "Jacaranda Cookbook." [Online]. Available : <http://a.die.supsi.ch/~pedrazz/jacaranda/documentation/cookbook/cookbook.html>, 2000.
- [21] Smith S.L. and Moizer J.N. *Design Guidelines for User – System Interface Software*. Report ESD-TR-84-190, The MITRE Coop. VA: National Technical Information Service, 1984.
- [22] Douglas K. "Defining the Problems of Framework Reuse." *Computer Software and Applications Conference*, 2002. pp.623 – 626.
- [23] Sun Microsystems : Java™ 2 SDK, Standard Edition Documentation Version 1.4.0. 2002.
- [24] Webmaster's Color Laboratory. Available: <http://www.visibone.com/colorlab/index.html> 2004.

- [25] จรณิต แก้วกังวาล. วิศวกรรมซอฟต์แวร์ หลักการออกแบบพัฒนาระบบเชิงวิศวกรรม และองค์ประกอบมนุษย์. กรุงเทพฯ : ซีเอ็ดยูเคชั่น. 2540.
- [26] ธวัชชัย ศรีสุเทพ. คัมภีร์ Web Design. กรุงเทพฯ : โปรวิชั่น. 2544.

ภาคผนวก ก.

แบบสอบ

วัดระยะเวลาการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งานคู่มือ  
เฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของ  
บริษัทชันไมโครซิสเต็มส์

คำแนะนำ ให้ผู้ทดสอบพัฒนาโปรแกรมประยุกต์จากคอลเล็กชันเฟรมเวิร์กตามโจทย์ที่กำหนดให้ต่อไปนี้ โดยใช้คู่มือการใช้งานเฟรมเวิร์กที่ส่งได้

ข้อ 1. จงสร้างคลาสใหม่ที่ชื่อว่า Stack1 โดยใช้คลาส LinkedList ช่วย โดยคลาสใหม่นี้จะต้องมีเมทอดที่สามารถเรียกใช้งานแบบ Stack 6 เมทอดดังต่อไปนี้

1. Constructor ที่เป็น Stack ว่าง

```
public stack()
```

2. เมทอดที่คืนค่าขนาดของ Stack หรือจำนวนข้อมูลภายใน Stack

```
public size()
```

3. เมทอดที่ตรวจสอบว่า Stack มีข้อมูลอยู่หรือไม่ โดยจะคืนค่าเป็น true ถ้าไม่มีข้อมูลอยู่ และ False ถ้ามีข้อมูลอยู่

```
public boolean isEmpty()
```

4. เมทอดที่ใช้ใส่ข้อมูลลงในตำแหน่งบนสุดของ Stack และคืนค่าข้อมูลที่ใส่ลงไปด้วย

```
public Object push(Object element);
```

5. เมทอดที่นำข้อมูลในตำแหน่งบนสุดออกจาก Stack โดยมีเงื่อนไขว่า Stack จะต้องไม่ว่าง ถ้า Stack ว่างจะต้องโยน exception ออกมาด้วย

```
public Object pop()
```

6. เมทอดที่ใช้อ่านข้อมูลในตำแหน่งบนสุดของ Stack โดยไม่ลบออกจาก Stack โดยมีเงื่อนไขว่า Stack จะต้องไม่ว่าง ถ้า Stack ว่างจะต้องโยน exception ออกมาด้วย

```
public Object peek()
```

ข้อ 2. จงใช้ ArrayList Class เพื่อพัฒนาโปรแกรมตามขั้นตอนต่อไปนี้

1. สร้าง ArrayList1 โดยเพิ่มข้อมูลลงไปใน ArrayList1 ตามลำดับดังนี้

```
Jane, Sandy, Ken, Nuk
```

2. สร้าง ArrayList2 โดยการคัดลอกข้อมูลจาก ArrayList1

3. เพิ่มข้อมูล Sara ลงไปต่อจาก Jane ใน ArrayList2 ให้ได้ผลลัพธ์ ดังนี้

```
Jane, Sara, Sandy, Ken, Nuk
```

4. พิมพ์ข้อมูลทั้งหมดใน ArrayList1 และ ArrayList2 ออกมาโดยใช้ method ListIterator ให้ได้ผลลัพธ์ ดังนี้

```
ArrayList1 : Jane, Sandy, Ken, Nuk
```

```
ArrayList2 : Jane, Sara, Sandy, Ken, Nuk
```

ภาคผนวก ข.

แบบสอบถาม

ความพอใจในรูปแบบคู่มือการใช้งานเฟรมเวิร์ก

## คำชี้แจง

1. ผู้ตอบแบบสอบถามฉบับนี้ คือ นักศึกษาปริญญาตรีและโท สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ใช้งานคู่มือคอเล็กชันเฟรมเวอร์กจากการวิจัยในการพัฒนาโปรแกรมประยุกต์เรียบร้อยแล้ว
2. แบบสอบถามมีวัตถุประสงค์เพื่อให้ผู้ใช้งานคู่มือเฟรมเวอร์กได้แสดงความคิดเห็นเกี่ยวกับความพอใจในด้านส่วนประกอบและส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวอร์ก
3. การแสดงความคิดเห็นมีด้วยกัน 5 ระดับ ซึ่งหมายถึงความพอใจในแต่ละระดับ ดังนี้
 

5	หมายถึง	มากที่สุด
4	หมายถึง	มาก
3	หมายถึง	ปานกลาง
2	หมายถึง	น้อย
1	หมายถึง	น้อยที่สุด

แบบสอบถามเพื่อการวิจัย
------------------------

เรื่อง ความพอใจในรูปแบบคู่มือการใช้งานเฟรมเวิร์ก

คำชี้แจง กรุณาขีดเครื่องหมาย ✓ ลงในช่องว่างตามความคิดเห็นของท่าน

ท่านคิดว่าคู่มือคอเล็กชั่นเฟรมเวิร์กที่จัดทำขึ้นจากการวิจัย สร้างความพอใจในการทำงานของท่านมากน้อยเพียงใด

มากที่สุด ← → น้อยที่สุด

ข้อ	รูปแบบของคู่มือการใช้งานเฟรมเวิร์ก	ระดับความคิดเห็น				
		5	4	3	2	1
1.	ด้านส่วนประกอบของคู่มือการใช้งานเฟรมเวิร์ก					
	มีหัวข้อในการแสดงเนื้อหาที่เพียงพอต่อการพัฒนาโปรแกรมประยุกต์	....	....	....	....	....
	มีการเรียงลำดับส่วนประกอบของคู่มือที่ง่ายต่อการใช้งาน	....	....	....	....	....
	ผู้ใช้สามารถค้นหารายการคำสั่งในคู่มือเฟรมเวิร์กได้อย่างรวดเร็ว	....	....	....	....	....
	มีส่วนที่ทำให้เห็นภาพรวมของเฟรมเวิร์กได้ง่าย	....	....	....	....	....
	มีส่วนที่ทำให้ผู้ใช้เห็นความสัมพันธ์ของส่วนประกอบภายในเฟรมเวิร์กได้ง่าย	....	....	....	....	....
	มีตัวอย่างโปรแกรมประยุกต์ที่เหมาะสม	....	....	....	....	....
2.	มีคำอธิบายหรือส่วนที่เชื่อมโยงไปยังเนื้อหาอื่นได้ง่าย ไม่สับสน	....	....	....	....	....
	ด้านส่วนติดต่อกับผู้ใช้ของคู่มือการใช้งานเฟรมเวิร์ก					
	มีส่วนหัว (header) ของแต่ละหน้าที่เหมาะสม	....	....	....	....	....
	มีส่วนท้ายของเอกสาร (footer) ที่เหมาะสม	....	....	....	....	....
	มีการจัดตัวอักษรในเนื้อหาที่เหมาะสม	....	....	....	....	....
	มีขนาดตัวอักษรในเนื้อหาที่เหมาะสม	....	....	....	....	....
	มีความกว้างบรรทัดของเนื้อหาที่เหมาะสม	....	....	....	....	....
3.	มีการใช้สีที่อ่านได้ง่าย	....	....	....	....	....
	มีการใช้สีในการเน้นข้อความที่เหมาะสม	....	....	....	....	....
	มีการใช้ขนาดตัวอักษรในการเน้นข้อความที่เหมาะสม	....	....	....	....	....
	มีการใช้แบบอักษร (font) ที่เหมาะสมกับแต่ละส่วนของเนื้อหา	....	....	....	....	....
		....	....	....	....	....

ภาคผนวก ค.

คู่มือลงรหัสสำหรับวิเคราะห์ข้อมูลด้วยเครื่องคอมพิวเตอร์

ส่วนที่ 1 การลงข้อมูลเวลาการพัฒนาโปรแกรมระหว่างนักศึกษาที่ใช้งานคู่มือเฟรมเวิร์กจากงานวิจัยกับคู่มือเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

ชื่อตัวแปร	ความหมายของตัวแปร	รหัสค่าตัวแปร	สดมภ์ที่
Doc	ประเภทของคู่มือ	d1 = คู่มือของบริษัทชั้นไมโครซีเอสเต็มส์ d2 = คู่มือที่ได้จากงานวิจัย	1
Time	เวลาที่ใช้ในการพัฒนาโปรแกรมประยุกต์	ไม่มี	2

ส่วนที่ 2 การลงข้อมูลคะแนนที่ได้จากการตอบแบบสอบถามวัดความพอใจในการใช้งานคู่มือการใช้งานเฟรมเวิร์กระหว่างคู่มือการใช้งานเฟรมเวิร์กจากงานวิจัยกับคู่มือการใช้งานเฟรมเวิร์กของบริษัทชั้นไมโครซีเอสเต็มส์

ชื่อตัวแปร	ความหมายของตัวแปร	รหัสค่าตัวแปร	สดมภ์ที่
Doc	ประเภทของคู่มือ	d1 = คู่มือของบริษัทชั้นไมโครซีเอสเต็มส์ d2 = คู่มือที่ได้จากงานวิจัย	1
ITEM	คะแนนความพอใจการใช้งานคู่มือเฟรมเวิร์ก	5 = มากที่สุด 4 = มาก 3 = ปานกลาง 2 = น้อย 1 = น้อยที่สุด	2
Freq1	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 1	-	3
Freq2	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 2	-	4
Freq3	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 3	-	5
Freq4	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 4	-	6

ชื่อตัวแปร	ความหมายของตัวแปร	รหัสค่าตัวแปร	สดมภ์ที่
Freq5	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 5	-	7
Freq6	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 6	-	8
Freq7	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 7	-	9
Freq8	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 8	-	10
Freq9	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 9	-	11
Freq10	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 10	-	12
Freq11	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 11	-	13
Freq12	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 12	-	14
Freq13	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 13	-	15
Freq14	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 14	-	16
Freq15	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 15	-	17
Freq16	ความถี่ความพอใจในแต่ละระดับของคำถามข้อที่ 16	-	18
FreqAll	ความถี่ความพอใจในแต่ละระดับโดยรวม (ข้อ 1 – 16)	-	19

ภาคผนวก ง.

ตัวอย่างคู่มือการใช้งานเฟรมเวิร์กที่ได้จากงานวิจัย  
(จาวาคอเล็กชันเฟรมเวิร์ก)



# Collections Framework

[Home](#) : [Overview](#) : [Example](#) : [Explanation](#) : [Contact me](#) : [About me](#)

[Home](#)

Search :  [go!!](#)

## Java Collection Framework

---

### Collection Implementations

- [ArrayList](#)
- [LinkedList](#)
- [HashSet](#)
- [TreeSet](#)
- [HashMap](#)
- [TreeSet](#)

### Abstract Implementations

- [AbstractCollection](#)
- [AbstractList](#)
- [AbstractSet](#)
- [AbstractSequentialList](#)
- [AbstractMap](#)

### Collection Interface

- [Collection](#)
- [List](#)
- [Set](#)
- [SortedSet](#)
- [Map](#)
- [SortedMap](#)

## Java Collections Framework

The collections framework is a unified architecture for representing and manipulating collections, allowing them to be manipulated independently of the details of their representation. It reduces programming effort while increasing performance. It allows for interoperability among unrelated APIs, reduces effort in designing and learning new APIs, and fosters software reuse.

### Overview

- [Overview](#) - An overview of the Collections framework.

### Example

- [Example](#) - The examples and step for using Java Collection Framework class and interfaces.

### Reference

- [API Reference](#) - An annotated outline of the classes and interfaces comprising the collections framework

[Next >>>](#)

[Home](#) | [Overview](#) | [Example](#) | [Explanation](#) | [Contact me](#) | [About me](#)

Copyright (c) 2004 All Rights Reserved.

รูปที่ ง1 ตัวอย่างหน้าเมนูหลัก

Home : Overview : Example : Explanation : Contact me : About me

Overview

Java Collection Framework

Collection Implementations

- ArrayList
- LinkedList
- HashSet
- TreeSet
- HashMap
- TreeSet

Abstract Implementations

- AbstractCollection
- AbstractList
- AbstractSet
- AbstractSequentialList
- AbstractMap

Collection Interfaces

- Collection
- List
- Set
- SortedSet
- Map
- SortedMap

## Java Collections Framework Overview

A collection is an object that represents a group of objects. A collections framework is a part of `java.util` and unified architecture for representing and manipulating collections, allowing them to be manipulated independently of the details of their representation.

The Java Collection Framework consists basically of a hierarchy with interfaces at every level except the lowest, and the lowest level having implementations of those interfaces.

This is a relation of class in the Java Collection Framework

```

classDiagram
    class Collection["<<interface>> Collection"]
    class List["<<interface>> List"]
    class Set["<<interface>> Set"]
    class Map["<<interface>> Map"]
    class SortedSet["<<interface>> SortedSet"]
    class SortedMap["<<interface>> SortedMap"]
    class ArrayList
    class LinkedList
    class HashSet
    class TreeSet
    class HashMap
    class TreeMap

    Collection ..|> List
    Collection ..|> Set
    Collection ..|> Map
    List --> ArrayList
    List --> LinkedList
    Set --> SortedSet
    SortedSet --> HashSet
    SortedSet --> TreeSet
    Map --> HashMap
    Map --> TreeMap
    SortedMap --> TreeMap
  
```

### Collection Interfaces

There are six collection interfaces. The most basic interface is `Collection`. Three interfaces extend `Collection` is `Set`, `List` and `SortedSet`. The other two collection interfaces, `Map` and `SortedMap`, do not extend `Collection`, as they represent mappings rather than true collections. However, these interfaces contain collection-view operations, which allow them to be manipulated as collections.

### Collection Implementations

Class that implement the collection interfaces typically have names of the form `<implementation-style><interface>`  
 Implementation of `List` interface is `ArrayList` and `LinkedList`.  
 Implementation of `Set` interface is `HashSet` and `Tree set`.  
 Implementation of `Map` interface is `HashMap` and `TreeMap`.

The general-purpose implementations support all of the optional operations in the collection interfaces, and have no restrictions on the elements they may contain.

รูปที่ ๒ ตัวอย่างหน้าภาพรวม

Home : Overview : **Example** : Explanation : Contact me : About me

▶ **Example**

Java Collection Framework

Collection Implementations

- [ArrayList](#)
- [LinkedList](#)
- [HashSet](#)
- [TreeSet](#)
- [HashMap](#)
- [TreeSet](#)

Abstract Implementations

- [AbstractCollection](#)
- [AbstractList](#)
- [AbstractSet](#)
- [AbstractSequentialList](#)
- [AbstractMap](#)

Collection Interface

- [Collection](#)
- [List](#)
- [Set](#)
- [SortedSet](#)
- [Map](#)
- [SortedMap](#)

## Java Collections Framework Example

This page will give you many examples that make you understand more about Java Collection Framework. In each example, you will see comments in the program for make you understand better in detail. But examples in this page just tell you about overview how to use framework if you want to understand in step and want a complete explanation you can see at Explanation page.

### Examples of Collection Implemented Class

#### ArrayList

• <a href="#">TestArray1.java</a>	This is an example of ArrayList. This will give you some idea how to add two array together.
• <a href="#">TestArray2.java</a>	This is an easy example of ArrayList for using add and remove method in ArrayList.

#### LinkedList

• <a href="#">TestLinkedList1.java</a>	This is an example of LinkedList. This example is very similar to TestArray1.java but we use addFirst and addLast with this. This Example will show how to display LinkedList by using ListIterator.
• <a href="#">TestLinkedList2.java</a>	This is another example of LinkedList for using more method in LinkedList.

### Examples of Abstract Class

#### AbstractList

• <a href="#">TestAbstractList1.java</a>	This is an example how to extend AbstractList. The implementation is a modifiable list that implement get(int index), size() and set(int index, Object element) methods.
• <a href="#">TestAbstractList2.java</a>	This is another example of AbstractList implementation. This example will give you more methods implementation.
• <a href="#">Test2.java</a>	This example show how to use class that extend AbstractList. There are some methods that you don't have to implement but you can use such as many methods in this example.

### Examples of implement interface

#### List

• <a href="#">TestList1.java</a>	This example is implement some methods in List interface. This example just give an idea how to implement but it has not completed yet, so the programmer have to implement the rest of the code before using.
----------------------------------	--

รูปที่ ๓3 ตัวอย่างหน้าตัวอย่างโปรแกรมประยุกต์

Home : Overview : Example : **Explanation** : Contact me : About me

► **Explanation Home**

Java Collection Framework

The Collections framework consists of :

Collection Implementations	Collection Implementation
<ul style="list-style-type: none"> <li>• <a href="#">ArrayList</a></li> <li>• <a href="#">LinkedList</a></li> <li>• HashSet</li> <li>• TreeSet</li> <li>• HashMap</li> <li>• TreeMap</li> </ul>	<p><b>ArrayList</b> Resizable-array implementation of the List interface. (Essentially an unsynchronized Vector.) The best all-around implementation of the List interface.</p> <p><b>LinkedList</b> Doubly-linked list implementation of the List interface. May provide better performance than the ArrayList implementation if elements are frequently inserted or deleted within the list. Useful for queues and double-ended queues (deques).</p> <p><b>HashSet</b> Hash table implementation of the Set interface. The best all-around implementation of the Set interface.</p> <p><b>TreeSet</b> Red-black tree implementation of the SortedSet interface.</p> <p><b>HashMap</b> Hash table implementation of the Map interface. (Essentially an unsynchronized Hashtable that supports null keys and values.) The best all-around implementation of the Map interface.</p> <p><b>TreeMap</b> Red-black tree implementation of the SortedMap interface.</p>
Abstract Implementations	Abstract Implementations
<ul style="list-style-type: none"> <li>• <a href="#">AbstractCollection</a></li> <li>• <a href="#">AbstractList</a></li> <li>• <a href="#">AbstractSet</a></li> <li>• <a href="#">AbstractSequentialList</a></li> <li>• <a href="#">AbstractMap</a></li> </ul>	<p><b>AbstractCollection</b> Skeletal implementation of a collection that is neither a set nor a list (such as a "bag" or multiset).</p> <p><b>AbstractList</b> Skeletal implementation of a set.</p> <p><b>AbstractSet</b> Skeletal implementation of a list backed by a random-access data store (such as an array).</p> <p><b>AbstractSequentialList</b> Skeletal implementation of a list backed by a sequential-access data store (such as a linked list).</p> <p><b>AbstractMap</b> Skeletal implementation of a map.</p>
Collection Interface	Collection Interface
<ul style="list-style-type: none"> <li>• <a href="#">Collection</a></li> <li>• <a href="#">List</a></li> <li>• <a href="#">Set</a></li> <li>• <a href="#">SortedSet</a></li> <li>• <a href="#">Map</a></li> <li>• <a href="#">SortedMap</a></li> </ul>	<p><b>Collection</b> A group of objects. No assumptions are made about the order of the collection (if any), or whether it may contain duplicate elements.</p> <p><b>List</b> Ordered collection, also known as a sequence. Duplicates are generally permitted. Allows positional access. Extends the Collection interface.</p>

รูปที่ 4 ตัวอย่างหน้าคำอธิบายการใช้งานส่วนประกอบภายในเฟรมเวิร์ก

Home : Overview : Example : **Explanation** : Contact me : About me

► [Explanation Home](#) : ► **ArrayList**

Java Collection Framework

Collection Implementations

- [ArrayList](#)
- [LinkedList](#)
- [HashSet](#)
- [TreeSet](#)
- [HashMap](#)
- [TreeSet](#)

Abstract Implementations

- [AbstractCollection](#)
- [AbstractList](#)
- [AbstractSet](#)
- [AbstractSequentialList](#)
- [AbstractMap](#)

Collection Interface

- [Collection](#)
- [List](#)
- [Set](#)
- [SortedSet](#)
- [Map](#)
- [SortedMap](#)

## Class ArrayList

[ArrayList Interface Overview](#)

[Using Step and Example](#)

[ArrayList Field, Constructor and Method Summary](#)

[Inherited fields and methods](#)

**All Superinterfaces:**  
[List](#)

**All Abstraction Classes:**  
[AbstractList](#)

---

### ArrayList Class Overview

Primary qualities of ArrayList Class

1. This is a resizable-array implementation of the List interface.
2. ArrayList permits all elements, including null.
3. The size, isEmpty, get, set, iterator, and listIterator operations run in constant time.
4. The constant factor is low compared to that for the LinkedList implementation.
5. Each ArrayList instance has a capacity that is the size of the array used to store the elements in the list. As elements are added an ArrayList, its capacity grows automatically.
6. The ensureCapacity operation may reduce the amount of incremental reallocation.
7. This implementation is not synchronized. If multiple threads access an ArrayList instance concurrently, and at least one of the threads modifies the list structurally, it must be synchronized externally. If no such object exists, the list should be "wrapped" using the Collections.synchronizedList method.

```
List list = Collections.synchronizedList(new ArrayList(...));
```

Relation of ArrayList Interface

```

classDiagram
    class List {
        <<interface>>
    }
    class AbstractList {
        <abstract>
    }
    class AbstractSequentialList {
        <abstract>
    }
    class ArrayList
    class LinkedList

    List <|-- AbstractList
    AbstractList <|-- AbstractSequentialList
    AbstractList <|-- ArrayList
    AbstractSequentialList <|-- LinkedList
  
```

[Go to top](#)

---

### Using Step and Example

This is a step of how to use ArrayList.

1. import java.util.\*;
2. in the main method, make an instance of ArrayList Class for example,

```
ArrayList obj1 = new ArrayList();
```

รูปที่ ๖5 ตัวอย่างเนื้อหาในแต่ละเว็บไซต์

## Using Step and Example

This is a step of how to use ArrayList.

1. import java.util.\*;
2. in the main method, make an instance of ArrayList Class for example,

```
ArrayList obj1 = new ArrayList();
```

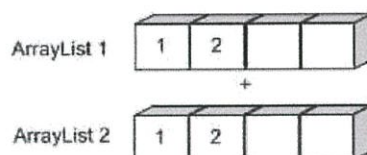
3. using method in ArrayList for example,

```
obj1.add("element1");
```

Inserted element1 into the end of obj1.

Example of using ArrayList [TestArray1.java]

There are two ArrayLists. Each of the lists features numbers 1 and 2, respectively. This example will add all the elements of similar positions in both lists together. The results will then be added to ArrayList 3. After that 9 and 5 are inserted respectively into position 1 in arraylist 3.



```
import java.util.*;

public class TestArray {

    public static void main(String[] args) {

        int sum;

        int temp;

        //create instance of ArrayList name: obj1
        ArrayList obj1 = new ArrayList();

        //create instance of ArrayList name: obj2
        ArrayList obj2 = new ArrayList();

        //create instance of ArrayList name: obj3
        ArrayList obj3 = new ArrayList();

        // add 1 and 2 repectively to obj1 and obj2

        obj1.add(new Integer (1));

        obj1.add(new Integer (2));

        obj2.add(new Integer (1));

        obj2.add(new Integer (2));

        // calculate sum of element in obj1 + obj and put in obj3
        for (int i = 0; i< obj1.size() ; i++){

            sum = ((Integer)obj1.get(i)).intValue() +
                ((Integer)obj2.get(i)).intValue();
        }
    }
}
```

**รูปที่ ๖6** คำอธิบายการใช้งานและตัวอย่างโปรแกรมประยุกต์ในอาเรย์ลิสต์

Field Summary	
<b>Fields inherited from class</b> java.util.AbstractList	
	modCount

Constructor Summary	
<b>ArrayList()</b>	Constructs an empty list with an initial capacity of ten.
<b>ArrayList(Collection c)</b>	Constructs a list containing the elements of the specified collection, in the order they are returned by the collection's iterator.
<b>ArrayList(int initialCapacity)</b>	Constructs an empty list with the specified initial capacity.

ArrayList Method Summary	
Insert Method	
boolean	<b>add</b> (Object o) Appends the specified element to the end of this list.
void	<b>add</b> (int index, Object element) Inserts the specified element at the specified position in this list.
boolean	<b>addAll</b> (Collection c) Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator.
boolean	<b>addAll</b> (int index, Collection c) Inserts all of the elements in the specified collection into this list at the specified position.
Delete Method	
void	<b>clear</b> () Removes all of the elements from this list
Object	<b>remove</b> (int index) Removes the element at the specified position in this list.
boolean	<b>remove</b> (Object o) Removes the first occurrence in this list of the specified element.
boolean	<b>removeAll</b> (Collection c) Removes from this list all the elements that are contained in the specified collection.
boolean	<b>retainAll</b> (Collection c) Retains only the elements in this list that are contained in the specified collection
Search Method	
boolean	<b>contains</b> (Object o) Returns true if this list contains the specified element.
boolean	<b>containsAll</b> (Collection c) Returns true if this list contains all of the elements of the specified collection.
boolean	<b>equals</b> (Object o) Compares the specified object with this list for equality.
Object	<b>get</b> (int index) Returns the element at the specified position in this list.
List	<b>subList</b> (int fromIndex, int toIndex) Returns a view of the portion of this list between the specified fromIndex, inclusive, and toIndex, exclusive.

รูปที่ ๗7 แสดงตัวอย่างการสรุปรายการคำสั่งอย่างย่อของอาเรย์ลิสต์

### ArrayList Method Detail

Prepared data example: oldAnimals that contains of: `dogs, cats, ducks, pigs`

#### Insert Method

- **add(Object o)**

Postcondition: o has been inserted at the back of this ArrayList object and true has been returned. The averageTime(n) is O(1). The worstTime(n) is still O(n)

```
boolean add(Object o);
```

Example We can create a four-element ArrayList object as follows:

```
ArrayList oldAnimals = new ArrayList();
```

```
oldAnimals.add("dogs");
oldAnimals.add("cats");
oldAnimals.add("ducks");
oldAnimals.add("pigs");
```

**\*\*Note1\*\*** In all subsequent examples,

oldAnimals consists of: `dogs, cats, ducks, pigs`

**\*\*Note2\*\*** This method always returns **true** because the corresponding method in the Collection interface returns a boolean value: true if the class of the calling object allow duplicates or the element to be added is not already in the calling object, and false otherwise. There are some implementations of the Collections interface in which duplicates are not allowed such as Set SortedSet.

- **add(int index, Object element)**

Precondition:  $0 \leq \text{index} \leq \text{size}()$ . Otherwise, IndexOutOfBoundsException will be thrown.

Postcondition: element has been inserted at position index in this ArrayList object, and all elements that were at positions  $\geq$  index before this call have been moved to the next higher position. The worstTime(n) is O(n).

```
void add(int index, Object element);
```

Example Given the following.

```
ArrayList animals = new ArrayList(oldAnimals);
animals.add(1, "tigers");
```

**\*\*Note\*\*** The ArrayList object animals now consists of :

`dogs, tigers, cats, ducks, pigs`

- **addAll(Collection c)**

Postcondition: The contents of c have been appended to this ArrayList object. If this ArrayList object was changed (That is, if c was not empty), true as been returned. Otherwise, false has been returned.

รูปที่ ๖8 แสดงตัวอย่างการสรุปรายการคำสั่งอย่างละเอียดของอาเรย์ลิสต์

ภาคผนวก จ.

ตัวอย่างคู่มือการใช้งานเฟรมเวิร์กของบริษัทชั้นไมโครซิสเต็มส์  
(จาวาคอเล็กชันเฟรมเวิร์ก)



# The Collections Framework

[Documentation Contents](#)

The collections framework is a unified architecture for representing and manipulating collections, allowing them to be manipulated independently of the details of their representation. It reduces programming effort while increasing performance. It allows for interoperability among unrelated APIs, reduces effort in designing and learning new APIs, and fosters software reuse. The framework is based on six collection interfaces. It includes implementations of these interfaces, and algorithms to manipulate them.

## Overview

- [Overview](#) - An overview of the Collections framework.

## API Specification

- [API Reference](#) - An annotated outline of the classes and interfaces comprising the collections framework, with links into the JavaDoc.

## Tutorial

- [Tutorial](#) - A tutorial introduction to the collections framework with plenty of programming examples.

## API Enhancements

- [API Enhancements in J2SDK 1.4](#) - An annotated list of API changes between the 1.3 and 1.4 releases.
- [API Enhancements in the Previous Release](#) - An annotated list of previous API changes in the 1.3 release.

## Design FAQ

- [Design FAQ](#)- Answers to frequently asked questions concerning the design of the collections framework.

Copyright © 2002 Sun Microsystems, Inc. All Rights Reserved.

Please send comments to: [collections-comments@java.sun.com](mailto:collections-comments@java.sun.com)



Java Software

## รูปที่ ๑1 หน้าเมนูหลัก

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Java™ 2 Platform  
Std. Ed. v1.4.1

[Submit a bug or feature](#)

For further API reference and developer documentation, see [Java 2 SDK SE Developer Documentation](#). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2002 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

## รูปที่ ๑2 การเขียนส่วนท้ายของคู่มือ



# Collections Framework Overview

## Introduction

The Java 2 platform includes a new *collections framework*. A *collection* is an object that represents a group of objects (such as the familiar [Vector](#) class). A collections framework is a unified architecture for representing and manipulating collections, allowing them to be manipulated independently of the details of their representation.

The primary advantages of a collections framework are that it:

- **Reduces programming effort** by providing useful data structures and algorithms so you don't have to write them yourself.
- **Increases performance** by providing high-performance implementations of useful data structures and algorithms. Because the various implementations of each interface are interchangeable, programs can be easily tuned by switching implementations.
- **Provides interoperability between unrelated APIs** by establishing a common language to pass collections back and forth.
- **Reduces the effort required to learn APIs** by eliminating the need to learn multiple ad hoc collection APIs.
- **Reduces the effort required to design and implement APIs** by eliminating the need to produce ad hoc collections APIs.
- **Fosters software reuse** by providing a standard interface for collections and algorithms to manipulate them.

The collections framework consists of:

- **Collection Interfaces** - Represent different types of collections, such as sets, lists and maps. These interfaces form the basis of the framework.
- **General-purpose Implementations** - Primary implementations of the collection interfaces.
- **Legacy Implementations** - The collection classes from earlier releases, `Vector` and `Hashtable`, have been retrofitted to implement the collection interfaces.
- **Wrapper Implementations** - Add functionality, such as synchronization, to other implementations.
- **Convenience Implementations** - High-performance "mini-implementations" of the collection interfaces.
- **Abstract Implementations** - Partial implementations of the collection interfaces to facilitate custom implementations.
- **Algorithms** - Static methods that perform useful functions on collections, such as sorting a list.
- **Infrastructure** - Interfaces that provide essential support for the collection interfaces.
- **Array Utilities** - Utility functions for arrays of primitives and reference objects. Not, strictly speaking, a part of the Collections Framework, this functionality is being added to the Java platform at the same time and relies on some of the same infrastructure.

---

## Collection Interfaces

There are six *collection interfaces*. The most basic interface is `Collection`. Three interfaces extend `Collection`: `Set`, `List`, and `SortedSet`. The other two collection interfaces, `Map` and `SortedMap`, do not extend `Collection`, as they represent mappings rather than true collections. However, these interfaces contain *collection-view* operations, which allow them to be manipulated as collections.

All of the modification methods in the collection interfaces are labeled *optional*. Some implementations may not perform one or more of these operations, throwing a runtime exception (`UnsupportedOperationException`) if they are attempted. Implementations must specify in their documentation which optional operations they support. Several terms are introduced to aid in this specification:

- Collections that do not support any modification operations (such as `add`, `remove` and `clear`) are referred to as *unmodifiable*. Collections that are not unmodifiable are referred to *modifiable*.
- Collections that additionally guarantee that no change in the `Collection` object will ever be visible are referred to as *immutable*. Collections that are not immutable are referred to as *mutable*.
- Lists that guarantee that their size remains constant even though the elements may change are referred to as *fixed-size*. Lists that are not fixed-size are referred to as *variable-size*.
- Lists that support fast (generally constant time) indexed element access are known as *random access* lists. Lists that do not support fast indexed element access are known as *sequential access* lists. The `RandomAccess` marker interface is provided to allow lists to advertise the fact that they support random access. This allows generic algorithms to alter their behavior to provide good performance when applied to either random or sequential access lists.



## Annotated Outline of Collections Framework

The collections framework consists of:

- **Collection Interfaces** - The primary means by which collections are manipulated.
  - **Collection** - A group of objects. No assumptions are made about the order of the collection (if any), or whether it may contain duplicate elements.
  - **Set** - The familiar set abstraction. No duplicate elements permitted. May or may not be ordered. Extends the Collection interface.
  - **List** - Ordered collection, also known as a *sequence*. Duplicates are generally permitted. Allows positional access. Extends the Collection interface.
  - **Map** - A mapping from keys to values. Each key can map to at most one value.
  - **SortedSet** - A set whose elements are automatically sorted, either in their *natural ordering* (see the [Comparable](#) interface), or by a [Comparator](#) object provided when a SortedSet instance is created. Extends the Set interface.
  - **SortedMap** - A map whose mappings are automatically sorted by *key*, either in the keys' *natural ordering* or by a comparator provided when a SortedMap instance is created. Extends the Map interface.
- **General-Purpose Implementations** - The primary implementations of the collection interfaces.
  - **HashSet** - Hash table implementation of the Set interface. The best all-around implementation of the Set interface.
  - **TreeSet** - Red-black tree implementation of the SortedSet interface.
  - **LinkedHashSet** - Hash table and linked list implementation of the Set interface. An insertion-ordered Set implementation that runs nearly as fast as HashSet.
  - **ArrayList** - Resizable-array implementation of the List interface. (Essentially an unsynchronized Vector.) The best all-around implementation of the List interface.
  - **LinkedList** - Doubly-linked list implementation of the List interface. May provide better performance than the ArrayList implementation if elements are frequently inserted or deleted within the list. Useful for queues and double-ended queues (dequeues).
  - **HashMap** - Hash table implementation of the Map interface. (Essentially an unsynchronized Hashtable that supports null keys and values.) The best all-around implementation of the Map interface.
  - **TreeMap** - Red-black tree implementation of the SortedMap interface.
  - **LinkedHashMap** - Hash table and linked list implementation of the Map interface. An insertion-ordered Map implementation that runs nearly as fast as HashMap. Also useful for building caches (see [removeEldestEntry](#) ([Map.Entry](#))).
- **Wrapper Implementations** - Functionality-enhancing implementations for use with other implementations. Accessed solely through static factory methods.
  - **Collections.unmodifiableInterface** - Return an unmodifiable view of a specified collection that throws an [UnsupportedOperationException](#) if the user attempts to modify it.
  - **Collections.synchronizedInterface** - Return a synchronized collection that is backed by the specified (typically unsynchronized) collection. As long as all accesses to the backing collection are through the returned collection, thread-safety is guaranteed.
- **Convenience Implementations** - High-performance "mini-implementations" of the collection interfaces.
  - **Arrays.asList** - Allows an array to be viewed as a list.
  - **EMPTY\_SET**, **EMPTY\_LIST** and **EMPTY\_MAP** - Constants representing the empty set and list (immutable).
  - **singleton**, **singletonList**, and **singletonMap** - Returns an immutable "singleton" set, list, or map, containing only the specified object (or key-value mapping).
  - **nCopies** - Returns an immutable list consisting of n copies of a specified object.
- **Legacy Implementations** - Older collection classes have been retrofitted to implement the collection interfaces.
  - **Vector** - Synchronized resizable-array implementation of the List interface with additional "legacy methods."
  - **Hashtable** - Synchronized hash table implementation of the Map interface that does not allow null keys or values, with additional "legacy methods."
- **Special Purpose Implementations**
  - **WeakHashMap** - an implementation of the Map interface that stores only *weak references* to its keys. Storing only weak references allows key-value pairs to be garbage-collected when the key is no longer referenced outside of the WeakHashMap. This class provides the easiest way to harness the power of weak references. It is useful for implementing "registry-like" data structures, where the utility of an entry vanishes when its key is no longer reachable by any thread.

รูปที่ ๑4 หน้าคำอธิบายการใช้งานส่วนประกอบภายในเฟรมเวิร์ก

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)  
PREV CLASS NEXT CLASS  
SUMMARY RESTED | FIELD | CONSTR | METHOD

FRAMES NO FRAMES [All Classes](#)  
DETAIL: FIELD | CONSTR | METHOD

**Java™ 2 Platform**  
**Std. Ed. v1.4.1**

---

**java.util**  
**Class ArrayList**

```

java.lang.Object
|
+-- java.util.AbstractCollection
|   |
|   +-- java.util.AbstractList
|       |
|       +-- java.util.ArrayList
    
```

**All Implemented Interfaces:**  
[Cloneable](#), [Collection](#), [List](#), [RandomAccess](#), [Serializable](#)

---

public class **ArrayList**  
 extends [AbstractList](#)  
 implements [List](#), [RandomAccess](#), [Cloneable](#), [Serializable](#)

Resizable-array implementation of the [List](#) interface. Implements all optional list operations, and permits all elements, including null. In addition to implementing the [List](#) interface, this class provides methods to manipulate the size of the array that is used internally to store the list. (This class is roughly equivalent to [vector](#), except that it is unsynchronized.)

The size, isEmpty, get, set, iterator, and listIterator operations run in constant time. The add operation runs in *amortized constant time*, that is, adding n elements requires O(n) time. All of the other operations run in linear time (roughly speaking). The constant factor is low compared to that for the [LinkedList](#) implementation.

Each [ArrayList](#) instance has a *capacity*. The capacity is the size of the array used to store the elements in the list. It is always at least as large as the list size. As elements are added an [ArrayList](#), its capacity grows automatically. The details of the growth policy are not specified beyond the fact that adding an element has constant amortized time cost.

An application can increase the capacity of an [ArrayList](#) instance before adding a large number of elements using the [ensureCapacity](#) operation. This may reduce the amount of incremental reallocation.

**Note that this implementation is not synchronized.** If multiple threads access an [ArrayList](#) instance concurrently, and at least one of the threads modifies the list structurally, it *must* be synchronized externally. (A structural modification is any operation that adds or deletes one or more elements, or explicitly resizes the backing array; merely setting the value of an element is not a structural modification.) This is typically accomplished by synchronizing on some object that naturally encapsulates the list. If no such object exists, the list should be "wrapped" using the [Collections.synchronizedList](#) method. This is best done at creation time, to prevent accidental unsynchronized access to the list:

```
List list = Collections.synchronizedList(new ArrayList(...));
```

The iterators returned by this class's [iterator](#) and [listIterator](#) methods are *fail-fast*: If the list is structurally modified at any time after the iterator is created, in any way except through the iterator's own [remove](#) or [add](#) methods, the iterator will throw a [ConcurrentModificationException](#). Thus, in the face of concurrent modification, the iterator fails quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future.

Note that the fail-fast behavior of an iterator cannot be guaranteed as it is, generally speaking, impossible to make any hard guarantees in the presence of unsynchronized concurrent modification. Fail-fast iterators throw [ConcurrentModificationException](#) on a best-effort basis. Therefore, it would be wrong to write a program that depended on this exception for its correctness: *the fail-fast behavior of iterators should be used only to detect bugs.*

**Since:**  
 1.2

**See Also:**  
[Collection](#), [List](#), [LinkedList](#), [Vector](#), [Collections.synchronizedList\(list\)](#), [Serialized Form](#)

### Field Summary

Fields inherited from class [java.util.AbstractList](#)

[modCount](#)

### Constructor Summary

[ArrayList\(\)](#)  
 Constructs an empty list with an initial capacity of ten.

[ArrayList\(Collection c\)](#)  
 Constructs a list containing the elements of the specified collection, in the order they are returned by the collection's iterator.

[ArrayList\(int initialCapacity\)](#)  
 Constructs an empty list with the specified initial capacity.

รูปที่ ๑5 เนื้อหาในแต่ละเว็บไซต์

Method Summary	
boolean	<b>add</b> (Object o) Ensures that this collection contains the specified element (optional operation).
boolean	<b>addAll</b> (Collection c) Adds all of the elements in the specified collection to this collection (optional operation).
void	<b>clear</b> () Removes all of the elements from this collection (optional operation).
boolean	<b>contains</b> (Object o) Returns true if this collection contains the specified element.
boolean	<b>containsAll</b> (Collection c) Returns true if this collection contains all of the elements in the specified collection.
boolean	<b>equals</b> (Object o) Compares the specified object with this collection for equality.
int	<b>hashCode</b> () Returns the hash code value for this collection.
boolean	<b>isEmpty</b> () Returns true if this collection contains no elements.
Iterator	<b>iterator</b> () Returns an iterator over the elements in this collection.
boolean	<b>remove</b> (Object o) Removes a single instance of the specified element from this collection, if it is present (optional operation).
boolean	<b>removeAll</b> (Collection c) Removes all this collection's elements that are also contained in the specified collection (optional operation).
boolean	<b>retainAll</b> (Collection c) Retains only the elements in this collection that are contained in the specified collection (optional operation).
int	<b>size</b> () Returns the number of elements in this collection.
Object []	<b>toArray</b> () Returns an array containing all of the elements in this collection.
Object []	<b>toArray</b> (Object[] a) Returns an array containing all of the elements in this collection; the runtime type of the returned array is that of the specified array.

รูปที่ ๑๖ แสดงการสรุปรายการคำสั่งอย่างย่อของอาเรย์ลิสต์

## Method Detail

### trimToSize

public void `trimToSize()`

Trims the capacity of this `ArrayList` instance to be the list's current size. An application can use this operation to minimize the storage of an `ArrayList` instance.

### ensureCapacity

public void `ensureCapacity(int minCapacity)`

Increases the capacity of this `ArrayList` instance, if necessary, to ensure that it can hold at least the number of elements specified by the minimum capacity argument.

**Parameters:**

`minCapacity` - the desired minimum capacity.

### size

public int `size()`

Returns the number of elements in this list.

**Specified by:**

`size` in interface `List`

**Specified by:**

`size` in class `AbstractCollection`

**Returns:**

the number of elements in this list.

### isEmpty

public boolean `isEmpty()`

Tests if this list has no elements.

**Specified by:**

`isEmpty` in interface `List`

**Overrides:**

`isEmpty` in class `AbstractCollection`

**Returns:**

`true` if this list has no elements; `false` otherwise.

### contains

public boolean `contains(Object elem)`

Returns `true` if this list contains the specified element.

**Specified by:**

`contains` in interface `List`

**Overrides:**

`contains` in class `AbstractCollection`

**Parameters:**

`elem` - element whose presence in this List is to be tested.

**Returns:**

`true` if the specified element is present; `false` otherwise.

### indexOf

public int `indexOf(Object elem)`

Searches for the first occurrence of the given argument, testing for equality using the `equals` method.

**Specified by:**

`indexOf` in interface `List`

**Overrides:**

`indexOf` in class `AbstractList`

**Parameters:**

`elem` - an object.

**Returns:**

the index of the first occurrence of the argument in this list; returns `-1` if the object is not found.

**See Also:**

`Object.equals(Object)`

รูปที่ ๑7 แสดงการสรุปรายการคำสั่งอย่างละเอียดของอาเรย์ลิสต์

## ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวสุธิดา สัจจรัตนันท์
วัน เดือน ปีเกิด	27 กุมภาพันธ์ 2524
ที่อยู่	105 ซ.พัฒนาการ 46 ถ.พัฒนาการ แขวงสวนหลวง เขตสวนหลวง กรุงเทพฯ 10250
ประวัติการศึกษา	2544 ครุศาสตรบัณฑิต สาขาวิชาวิทยาศาสตร์ทั่วไปและคอมพิวเตอร์ (เกียรตินิยมอันดับ 2) จุฬาลงกรณ์มหาวิทยาลัย