



หนังสือคอมพิวเตอร์ที่มีผู้อ่านมากที่สุด



# Android App Development

## ฉบับสมบูรณ์

- เรียนรู้ตั้งแต่พื้นฐาน อย่างละเอียด เป็นขั้นตอน
- มีตัวอย่างการพัฒนาแอพพลิเคชัน สำหรับใช้งาน ทำตามได้จริง
- แนะนำสำหรับนักเรียน นักศึกษา และผู้ที่สนใจพัฒนาแอพพลิเคชัน Android



ดร.จักรชัย โซือนทร์  
พงษ์ศรี จันทร์ย้อย  
ณัฐณิชา วะระมงคลเลิศ  
บรรณาธิการ กิตตินันท์ พลสวัสดิ์

FREE CD Source Code, Android-SDK,  
Eclipse, Android Development Tool,  
Java Development Kit, AppServ



# 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1

“ กว่า 10 ล้านเล่ม กีเราเพลิต และยังมุ่งมั่นต่อไป ”

“ หนังสือคุณภาพ แก่นแท้ของ  
หนังสือคอมพิวเตอร์  
ที่เป็นผู้อ่านมากที่สุด ”

เคล็ดลับ  
ท่องเน็ต

คู่มือ  
พิมพ์  
(ภายใน 7 วัน)

Infopress  
Group

BEST COMPUTER BOOK

# Android App Development ฉบับสมบูรณ์

ผู้แต่ง	ดร.จักรชัย โลsinทร์ พงษ์ศรี จันทร์ย้อย <sup>*</sup> ณัฐนิชา วีระมงคลเลิศ	android.kku@gmail.com
บรรณาธิการ	กิตินันท์ พลสวัสดิ์ อนัน วาโชže	kitinan@infopress.co.th
ออกแบบปก		
ออกแบบและจัดรูปเล่ม	วุฒิพันธ์ สมพระเมฆ	
พิสูจน์อักษร	สุนทรี บรรลือศักดิ์, มนฤตี ครีอุทธโยกาส	
ประสานงานการผลิต	สุพัตรา อาจปруд, ณัตรชนก แก้วจันทร์	
E-BOOK PRODUCTION	สุรีย์รัตน์ จิญี่, จิราภรณ์ โลภา	
เนื้อหา	Android เป็นเครื่องหมายการค้าของบริษัท Google, Eclipse เป็นเครื่องหมายการค้าของบริษัท The Eclipse Foundation และเครื่องหมายการค้าอื่นๆ ที่อ้างถึงเป็นของบริษัทนั้นๆ ผลงานลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยบริษัท ไอเดียซี พรีเมียร์ จำกัด ห้ามลอกเลียน ไม่ว่าส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ ไม่ว่าในรูปแบบใดๆ นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษร จากผู้จัดพิมพ์เท่านั้น บริษัท ไอเดียซี พรีเมียร์ จำกัด จัดตั้งขึ้นเพื่อเผยแพร่ความรู้เทคโนโลยีสารสนเทศทั้งในระดับพื้นฐานและระดับสูง เรายินดีรับงานเขียนของนักวิชาการและนักเขียนทุกท่าน โดยเฉพาะงานที่เกี่ยวข้องกับสารสนเทศ ท่าน ผู้สนใจกรุณาติดต่อผ่านทางอีเมลที่ editor@infopress.co.th หรือโทรศัพท์หมายเลข 0-2962-1081 (อัตโนมัติ 10 คู่สาย) โทรสาร 0-2962-1084	
ข้อมูลการบรรณาธิการ	ดร.จักรชัย โลsinทร์ Android App Development ฉบับสมบูรณ์ ナンブリ : ไอเดียซี, 2555 360 หน้า	
เนื้อหา	1. แอนดรอยด์ 2. การเขียนโปรแกรม I พงษ์ศรี จันทร์ย้อย (ผู้แต่งร่วม) II ณัฐนิชา วีระมงคลเลิศ (ผู้แต่งร่วม) III ชื่อเรื่อง 006.76	
ISBN	978-616-200-361-5	
พิมพ์ครั้งที่ 1	กันยายน 2555	
จัดพิมพ์และจัดจำหน่ายโดย	บริษัท ไอเดียซี พรีเมียร์ จำกัด 200 หมู่ 4 ชั้น 19 ห้อง 1901 อาคารวัฒนธรรมอินเตอร์เนชั่นแนลทาวเวอร์ ถ.แจ้งวัฒนะ อ.ปากเกร็ด จ.นนทบุรี 11120 โทรศัพท์ 0-2962-1081 (อัตโนมัติ 10 คู่สาย) โทรสาร 0-2962-1084	
ราคา	275 บาท	



สำหรับร้านค้าและตัวแทนจำหน่าย

โทรศัพท์ 0-2962-1081-3 ต่อ 112-114

โทรสาร 0-2962-1084

ลูกค้าสัมพันธ์

โทรศัพท์ 0-2962-1081-3 ต่อ 601

โทรสาร 0-2962-1084

# ค่ำบ้ำ

ความริเริ่มในการจัดทำหนังสือ Android App Development ฉบับสมบูรณ์ เพื่อให้นักศึกษา ในรายวิชา 322261, 322262 และ 322749 ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่นได้เรียนรู้และมีความเข้าใจในหลักการทำงาน รวมไปถึงมีการเรียนการสอนแบบมุ่งเน้นไปยังการใช้งานได้จริง โดยที่เนื้อหาจะมุ่งเน้นไปทางด้านการพัฒนาแอปพลิเคชันแอนดรอยด์ (Android) บนโทรศัพท์เคลื่อนที่ อุปกรณ์ตามได้มีการปรับเนื้อหาให้ผู้อ่านทั่วไปศึกษาและพัฒนาได้ เช่นกัน โดยแบ่งการทำงานออกเป็น 2 ส่วน ทั้งในส่วนของการพัฒนาโปรแกรมบนเครื่องโทรศัพท์จำลอง (Emulator) และในส่วนของการนำไปใช้งานบนโทรศัพท์เคลื่อนที่จริง (Samsung Galaxy Nexus)

ถึงแม้ว่า Android จะเป็นซอฟต์แวร์แบบโอเพนซอร์ส (Open Source) แต่ในปัจจุบันจะสังเกตได้ว่า มีเอกสารหรือคู่มือประกอบการพัฒนาแอปพลิเคชัน Android น้อยมาก โดยส่วนใหญ่แล้วเอกสารเหล่านั้นจะ อธิบายถึงการสอนอแอปพลิเคชันและโทรศัพท์มือถือ Android ในรุ่นต่างๆ เพื่อทางการค้าเป็นสำคัญ โดยไม่มี การอธิบายถึงขั้นตอนและหลักการสำหรับการพัฒนาแอปพลิเคชัน โดยเฉพาะด้วยอย่างของแอปพลิเคชันแบบ สมบูรณ์ที่สามารถใช้งานได้จริง (นอกเหนือจากหนังสือ Basic Android APP Development ซึ่งแต่โดยผู้เขียนเอง ทั้งนี้ในการนี้ที่ผู้อ่านมีข้อสงสัยสามารถสอบถามได้จาก android.kku@gmail.com และกลุ่ม Facebook : <https://www.facebook.com/groups/114419038638590/>

หนังสือเล่มนี้เป็นส่วนขยายเพิ่มเติม โดยเฉพาะอย่างยิ่งได้เน้นไปทางด้านการพัฒนาแอปพลิเคชัน Android ในขั้นที่สูงขึ้น มีโค้ดที่กระชับขึ้น รวมไปถึงมีการปรับเปลี่ยนเวอร์ชันจากเนื้อหาเดิมให้ทันสมัย เช่น Android 4.0.3 (Ice Cream Sandwich) ทั้งในส่วนของโทรศัพท์และ Tablet โดยการพัฒนาแอปพลิเคชัน Android นั้นจะใช้ภาษาจาวา (Java) แต่พั้งก์ชันการใช้งานต่างๆ สำหรับการพัฒนาแอปพลิเคชัน Android นั้นยังมีข้อปลีกย่อยอีกมาก ดังนั้น หนังสือเล่มนี้จึงถูกจัดทำเพื่อให้ผู้อ่านมีความเข้าใจและมีหลักการ พื้นฐานในการพัฒนาแอปพลิเคชัน Android ได้อย่างถูกต้องและสามารถนำไปใช้งานได้จริงในทางปฏิบัติ

เนื้อหาของหนังสือเล่มนี้จะอธิบายเริ่มตั้งแต่การติดตั้ง Android Emulator รวมไปถึงเทคโนโลยีต่างๆ ที่ผู้อ่านจำเป็นต้องใช้ และนำเสนอแนวทางการแก้ไขปัญหา จนกระทั่งถึงการแสดงตัวอย่างของโปรแกรม ที่เป็นพื้นฐานสำคัญต่างๆ จากนั้นจะนำแอปพลิเคชันที่พัฒนาได้ไปติดตั้งบนโทรศัพท์จริง ซึ่งโดยภาพรวม สามารถแบ่งโครงสร้างของหนังสือเล่มนี้ออกเป็น 3 ส่วนคือ ส่วนแรกจะเป็นการแนะนำขั้นตอนวิธีการ ติดตั้งเครื่องมือสำหรับการพัฒนาแอปพลิเคชัน Android ส่วนที่ 2 จะเป็นการฝึกพัฒนาแอปพลิเคชันใน รูปแบบต่างๆ ซึ่งผู้เขียนได้ออกแบบเนื้อหาในแต่ละบทให้มีความครอบคลุมกับทักษะต่างๆ ที่จำเป็นสำหรับ การนำไปพัฒนาแอปพลิเคชันขั้นสูงได้ และส่วนที่ 3 จะเป็นการแนะนำเทคนิคในการแก้ไขปัญหาสำหรับ การพัฒนาแอปพลิเคชัน และการนำไปทดสอบที่ได้พัฒนาไว้บน Android Emulator เป็นต้นที่จะช่วยให้ผู้อ่านสามารถเรียนรู้และประยุกต์ใช้ได้จริง

สุดท้ายนี้สำหรับการเขียนหนังสือเล่มนี้ จะต้องขอบพระคุณคุณพ่อคุณแม่และครอบครัวของผู้เขียน ที่เคยเป็นกำลังใจและให้การสนับสนุนมาตลอด คณบุคคลอีกหลายฝ่ายทั้งในส่วนของภาควิชาเอง หัวหน้าภาควิชา รศ.สมจิตรา ออาจินทร์, คณาราชารย์ รศ.สุรศักดิ์ สงวนพงษ์ และรศ.ยืนภูรารอน ที่เป็นแรงบันดาลใจสำหรับการเขียนหนังสือต่างๆ และยังรวมไปถึงนักศึกษาช่วยสอน นายพงษ์ศธร จันทร์ย้อย และนางสาวณัฐณิชา วีระมงคลเลิศ ที่เป็นส่วนหลักในการผลัดดันและจัดทำข้อมูลส่วนของซอฟต์แวร์ (Source Code) ที่นำมาปรับแต่ง และสุดท้ายขอแสดงความยินดีกับนักศึกษานายธีรวัฒน์ นิมมคุกุวงค์รัตน์ นายคฑา โพธิสาร และนางสาวกิตติมา หนองหาร ที่คว้ารางวัลชนะเลิศ การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 14 (NSC 2012) หมวด Mobile Application โดยพัฒนาโปรแกรมดนตรีไทยวงเครื่องลาย ซึ่งเป็นการต่อยอดจากหนังสือเล่มนี้

ดร.จักรชัย โลอินทร์  
Chakchai So-In, Ph.D.  
กันยายน 2555

# Contents

<b>บทที่ 1 รู้จักแอนดรอยด์ (Android) .....</b>	<b>1</b>
แอนดรอยด์คืออะไร.....	1
ประเภทของระบบปฏิบัติการ Android.....	3
สามารถพัฒนาอะไรได้บ้างบน Android.....	3
สถาปัตยกรรมของแอนดรอยด์ (Android Architecture).....	5
ส่วนประกอบของแอพพลิเคชัน (Application Component) .....	9
วงรอบชีวิตของแอพพลิเคชัน (Application Life Cycle) .....	10
สรุปท้ายบท.....	12
แบบฝึกหัดท้ายบท.....	12
<b>บทที่ 2 การติดตั้งเครื่องมือสำหรับการพัฒนาแอพพลิเคชันบน Android .....</b>	<b>13</b>
การติดตั้ง JDK (Java Development Kit).....	14
การติดตั้งโปรแกรม Eclipse .....	18
เริ่มต้นเขียนโปรแกรมด้วย Eclipse.....	19
กำหนดให้ Text Editors และหมายเลขอาราทัด.....	24
การติดตั้ง Android SDK (Android Software Development Kit).....	25
การติดตั้ง ADT (Android Development Tool) .....	27
การปรับแต่งค่า ADT และ Android SDK.....	32
การติดตั้ง SDK Platform.....	36
การสร้างโทรศัพท์จำลอง (Emulator).....	38
สรุปท้ายบท.....	42
แบบฝึกหัดท้ายบท.....	42
<b>บทที่ 3 เริ่มต้นพัฒนาแอพพลิเคชัน .....</b>	<b>43</b>
เริ่มต้นสร้างโปรเจกต์.....	44
ออกแบบหน้าจอแอพพลิเคชัน.....	51
เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน.....	58
ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator) .....	60
การล็อกอินและนำเข้าโปรเจกต์ใน Eclipse .....	65
การล็อกอินโปรเจกต์เป็น Archive File.....	66
การนำเข้าโปรเจกต์จาก Archive File .....	69
เริ่มต้นพัฒนาแอพพลิเคชันบน Tablet .....	72
สรุปท้ายบท.....	74
แบบฝึกหัดท้ายบท.....	74

**บทที่ 4 พัฒนาแอพพลิเคชันอย่างง่าย.....75**

เริ่มต้นสร้างโปรเจกต์.....	76
ออกแบบหน้าจอแอพพลิเคชัน.....	78
การแก้ไขไฟล์ main.xml .....	78
การแก้ไขไฟล์ string.xml .....	80
เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน.....	82
ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator) .....	84
สรุปท้ายบท .....	90
แบบฝึกหัดท้ายบท.....	90

**บทที่ 5 การสร้างส่วนติดต่อกับผู้ใช้ (User Interface) .....91**

เริ่มต้นสร้างโปรเจกต์.....	92
TextView : แสดงข้อความ.....	95
การกำหนดคุณสมบัติ TextView .....	95
การเปลี่ยนแบบอักษรสำหรับ TextView.....	96
EditText : รับข้อความ.....	98
ImageView : แสดงรูปภาพ .....	100
Button : ปุ่ม.....	101
การเพิ่มอีเวนต์ (Event) ให้ Button .....	102
ImageButton : ปุ่มรูปภาพ .....	104
สรุปท้ายบท.....	106
แบบฝึกหัดท้ายบท.....	106

**บทที่ 6 รู้จักการใช้งานเลเยอร์ (Layout).....107**

Linear Layout.....	107
Relative Layout.....	111
Frame Layout.....	114
Absolute Layout.....	116
Table Layout .....	119
สรุปท้ายบท.....	121
แบบฝึกหัดท้ายบท.....	122

**บทที่ 7 รู้จักการใช้งานวิวกรุ๊ป (ViewGroup) ชนิดต่างๆ .....** 123

ListView .....	123
ScrollView .....	126
Image Switcher View .....	130
OptionsMenu.....	134
TabWidget.....	138
WebView .....	141

Preferences Screen .....	145
การจัดวางหน้าจอ (Screen Orientation) .....	150
สรุปท้ายบท .....	156
แบบฝึกหัดท้ายบท.....	156

## บทที่ 8 การพัฒนาแอพพลิเคชันรับส่งข้อความ (SMS) ..... 157

หลักการทำงานของแอพพลิเคชันรับส่งข้อความ.....	158
เริ่มต้นสร้างໂປຣເຈັກຕີ.....	159
กำหนดສື່ທີ່ການໃຫ້ແອພພລິເຄີ້ມ.....	161
ອອກແບບໜ້າຈອແອພພລິເຄີ້ມ.....	162
ເຂົ້າໂດດກຳນົດການໃຫ້ແອພພລິເຄີ້ມ.....	164
ການເຂົ້າໂດດສ່ວນຂອງການລົ່ງຂ້ອງຄວາມ .....	164
ການເຂົ້າໂດດສ່ວນຂອງການຮັບຂ້ອງຄວາມ .....	166
ທດສອນການໃຫ້ແອພພລິເຄີ້ມຜ່ານໂໂຮຄັ້ພທີ່ຈຳລອງ (Emulator) .....	168
ສຽງທ້າຍນິກ.....	171
ແບບຝຶກທັດທ້າຍນິກ.....	172

## บทที่ 9 การพัฒนาแอพพลิเคชันຕິດຕໍ່ກັບ GPS ແລະການອ້າງອີງກັບ R:ມູນແບນທີ່ (Map) ..... 173

ຄວາມຮູ້ພື້ນຖານເກີ່ມກັບ GPS .....	173
ການອ່ານຕຳແໜ່ງພິກັດຂອງຮບນ Gps .....	174
ເຮັ່ມຕັ້ນສ້າງໂປຣເຈັກຕີ .....	174
ກຳນົດສື່ທີ່ການໃຫ້ແອພພລິເຄີ້ມ.....	176
ເຂົ້າໂດດກຳນົດການໃຫ້ແອພພລິເຄີ້ມ .....	176
ທດສອນການໃຫ້ແອພພລິເຄີ້ມຜ່ານໂໂຮຄັ້ພທີ່ຈຳລອງ (Emulator) .....	179
ການແສດງຜົດຄ່າພິກັດບົນ Google Map .....	182
ສ້າງ Google API Key .....	182
ກຳນົດສື່ທີ່ການໃຫ້ແອພພລິເຄີ້ມ.....	184
ອອກແບບໜ້າຈອແອພພລິເຄີ້ມ .....	184
ເຂົ້າໂດດກຳນົດການໃຫ້ແອພພລິເຄີ້ມ .....	185
ທດສອນການໃຫ້ແອພພລິເຄີ້ມຜ່ານໂໂຮຄັ້ພທີ່ຈຳລອງ (Emulator) .....	187
ປັບແຕ່ງແອພພລິເຄີ້ມເພື່ອແສດງແຜນທີ່ໃນຮູ້ພື້ນຖານ .....	188
ການເພີ່ມ Marker ລົງບົນແຜນທີ່ .....	191
ສຽງທ້າຍນິກ .....	195
ແບບຝຶກທັດທ້າຍນິກ.....	195

## บทที่ 10 การพัฒนาแอพพลิเคชันຕິດຕໍ່ອຳນວຍໂຄຣຢ່າງຈ່າຍ ..... 197

ການສ້າງແອພພລິເຄີ້ມສ່ວນເຊີ່ວິ້ນໄວ່.....	198
ການສ້າງແອພພລິເຄີ້ມສ່ວນໄຄລເອນທີ .....	207

ทดสอบการทำงานของแอพพลิเคชัน.....	211
สรุปท้ายบท.....	213
แบบฝึกหัดท้ายบท.....	213
<b>บทที่ 11 การพัฒนาแอพพลิเคชันทำงานกับฐานข้อมูล ..... 215</b>	
ความรู้เบื้องต้นเกี่ยวกับการทำงานกับฐานข้อมูล .....	215
การสร้างล้วนซึ่อมต่อและใช้งานฐานข้อมูล .....	216
การใช้งานฐานข้อมูลผ่าน User Interface .....	224
ออกแบบหน้าจอแอพพลิเคชัน .....	224
เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน .....	229
ทดสอบการทำงานของแอพพลิเคชัน.....	238
สรุปท้ายบท.....	239
แบบฝึกหัดท้ายบท.....	240
<b>บทที่ 12 การพัฒนาเกมแบบ 2 มิติ ..... 241</b>	
วิธีการเล่นเกม .....	241
การสร้างคลาสหลักในการพัฒนาเกม XO .....	242
การสร้างหน้าจอส่วนต่างๆ ของเกม.....	248
การสร้างหน้าจومenuสำหรับเลือกรูปแบบการเล่น.....	248
การสร้างหน้าจอลำไրบ้อนซื่อผู้เล่น.....	251
การสร้างหน้าจอลำไรบ้อนเล่นเกม.....	254
การสร้างหน้าจอมือจับเกม .....	256
กำหนดรายละเอียดให้แอ็คทิวิตี้ต่างๆ.....	258
การสร้างกราฟิก (Graphic) เมื่อต้น.....	260
การจัดการเกี่ยวกับหน้าจอลัมป์ส์ (TouchScreen) .....	263
การปรับแต่งการแสดงผลเมื่อจบเกม .....	264
ทดสอบการทำงานของเกม.....	269
สรุปท้ายบท.....	270
แบบฝึกหัดท้ายบท.....	270
<b>บทที่ 13 การใช้งานเซนเซอร์ (Sensor) ใน Android ..... 271</b>	
การสร้างโปรเจกต์ (Project) .....	272
กำหนดลิทีธิ์การทำงานให้แอพพลิเคชัน.....	273
ออกแบบหน้าจอแอพพลิเคชัน.....	273
เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน .....	274
สร้างคลาสควบคุมการเปลี่ยนแปลงตำแหน่งของลูกบola .....	274
เขียนโค้ดกำหนดการทำงานในคลาสหลัก.....	277
ทดสอบการทำงานของแอพพลิเคชัน.....	280
สรุปท้ายบท.....	281
แบบฝึกหัดท้ายบท.....	281

## บทที่ 14 เทคนิคในการแก้ไขปัญหาสำหรับการพัฒนาแอพพลิเคชัน ..... 283

การใช้งานเครื่องมือ Dalvik Debug Monitor Server (DDMS) .....	285
Devices.....	287
Emulator Control.....	287
การใช้งาน LogCat.....	289
การตรวจสอบข้อผิดพลาดของแอพพลิเคชัน โดยการอ่านค่า Log.....	291
ค้นหาตำแหน่งของไฟล์ด้วย File Explorer .....	292
สรุปท้ายบท .....	293
แบบฝึกหัดท้ายบท.....	293

## บทที่ 15 การอัพโหลดแอพพลิเคชันเข้า Android Market ..... 295

การจัดเตรียมไฟล์ต่างๆ สำหรับอัพโหลดแอพพลิเคชัน.....	296
การจัดเตรียมไฟล์รูปภาพและวิดีโอลำไยท์สำหรับอัพโหลดแอพพลิเคชัน.....	296
การส่งออกไฟล์ .apk สำหรับอัพโหลดแอพพลิเคชัน.....	297
การอัพโหลดแอพพลิเคชันเข้า Android Market .....	303
การอัพเดตเวอร์ชันแอพพลิเคชัน.....	309
สรุปท้ายบท .....	313
แบบฝึกหัดท้ายบท.....	313

## บทที่ 16 การติดตั้งแอพพลิเคชันบนเครื่องโทรศัพท์จัง ..... 315

การติดตั้งไดรเวอร์ของเครื่องโทรศัพท์ (Driver Installation) .....	316
การส่งออกไฟล์ .apk .....	321
การเตรียมการส่งออกไฟล์ .apk .....	321
นำไฟล์ .apk ไปติดตั้งบนเครื่องโทรศัพท์จำลอง หรืออีเมลเตอร์เครื่องอื่น .....	324
นำไฟล์ .apk ไปติดตั้งบนโทรศัพท์เคลื่อนที่จริง .....	325
สรุปท้ายบท .....	326
แบบฝึกหัดท้ายบท.....	326

## Appendix ความรู้เบื้องต้นการเขียนโปรแกรมภาษาจาวา ..... 327

ชนิดของข้อมูล (Data Type) .....	327
การประกาศตัวแปร (Declaration).....	328
หลักการตั้งชื่อตัวแปร.....	328
ตัวแปรชนิดข้อความ .....	328
การประกาศค่าคงที่.....	329
ตัวดำเนินการ (Operators) .....	329
ลำดับความสำคัญของตัวดำเนินการ .....	331
การแสดงผลอักษรพิเศษ .....	331
การจัดรูปแบบแสดงผลตัวเลข .....	332
การแปลงชนิดข้อมูล .....	332

การแปลงชนิดข้อมูลแบบ Implicit type conversion .....	332
การแปลงชนิดข้อมูลแบบ Explicit type conversion .....	333
การเขียนคำอธิบายโปรแกรม.....	334
คำลั่งตัดสินใจ.....	334
คำลั่งวนลูป .....	336
ตัวแปรอาร์เรย์.....	337
ตัวแปรอาร์เรย์ 1 มิติ.....	337
ตัวแปรอาร์เรย์หลายมิติ.....	337
การเขียนโปรแกรมกับตัวแปรอาร์เรย์.....	338
การประกาศตัวแปรอาร์เรย์ .....	338
การกำหนดค่าข้อมูลให้ตัวแปรอาร์เรย์ .....	339
การอ้างถึงข้อมูลในตัวแปรอาร์เรย์ .....	339
พื้นฐานการเขียนโปรแกรมเชิงวัตถุ.....	340
<b>บรรณานุกรม .....</b>	<b>345</b>
<b>Index .....</b>	<b>347</b>

## รู้จักแอนดรอยด์ (Android)

บทนี้เป็นเนื้อหาที่อยู่ในส่วนแรกสำหรับหนังสือเล่มนี้ ซึ่งเป็นเนื้อหาที่อธิบายถึงแอนดรอยด์ในเบื้องต้น รวมไปถึงการอธิบายถึงสถาปัตยกรรมของแอนดรอยด์ที่สำคัญ ทั้งนี้เพื่อให้ผู้อ่านสามารถนำไปประยุกต์ใช้เพื่อ ทำความเข้าใจในการพัฒนาแอปพลิเคชัน โดยที่เนื้อหาในบทนี้จะแบ่งออกเป็น 4 ส่วนคือ แอนดรอยด์คืออะไร สถาปัตยกรรมของแอนดรอยด์ ส่วนประกอบของแอปพลิเคชัน และวิธีการเขียนแอปพลิเคชัน

### แอนดรอยด์คืออะไร

**แอนดรอยด์ (Android)** เป็นซอฟต์แวร์ที่มีโครงสร้างแบบเรียงทับชั้นหรือแบบสเต็ค (Stack) ซึ่ง รวมกระบวนการปฏิบัติการ มิดเดิลแวร์ และแอปพลิเคชันที่สำคัญเข้าไว้ด้วยกัน เพื่อใช้สำหรับทำงานบนอุปกรณ์ พกพาเคลื่อนที่โดยเฉพาะ เช่น โทรศัพท์มือถือ,แท็บเล็ต เป็นต้น



การทำงานของแอนดรอยด์มีพื้นฐานอยู่บนระบบลินกุซ์เคนเวล (Linux Kernel)

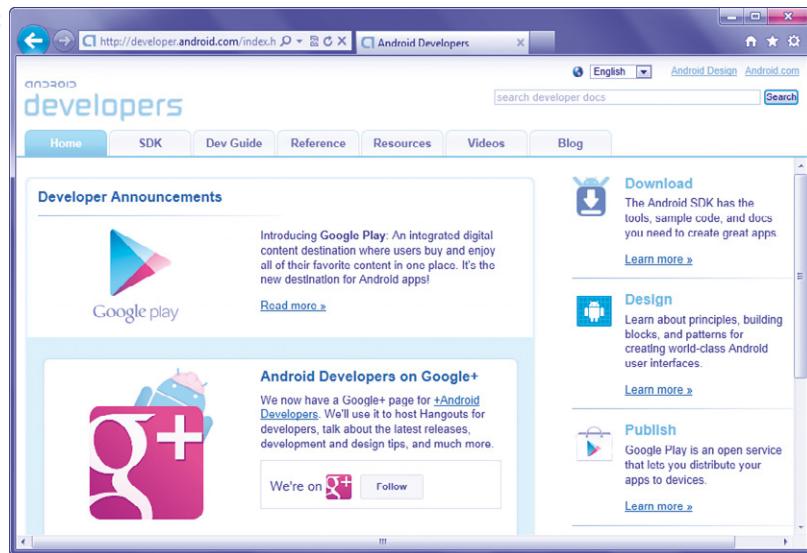
ซึ่งใช้ **Android SDK (Software Development Kit)** เป็นเครื่องมือสำหรับการพัฒนา แอปพลิเคชันบนระบบปฏิบัติการ Android โดยใช้ภาษา Java ในการพัฒนา

Android เริ่มพัฒนาโดยบริษัท แอนดรอยด์ และต่อมาได้ผนวกเข้ากับบริษัท Google ในเดือนพฤษภาคม ปี พ.ศ. 2550 ซึ่งมีการร่วมมือกันระหว่างบริษัทชั้นนำมากกว่า 33 บริษัทเพื่อพัฒนาระบบ Android ทั้งบริษัทที่ให้บริการโทรศัพท์เคลื่อนที่ บริษัทผู้ผลิตอุปกรณ์โทรศัพท์เคลื่อนที่ บริษัทซอฟต์แวร์ และบริษัทเอกชนต่างๆ เช่น HTC, LG, Motorola, Samsung, China Mobile Communications, KDDI, DoCoMo, Sprint/Nextel, T-Mobile, Telecom Italia, Telefonica, Audience, Broadcom, Intel, Marvel, NVidia, eBay, Packet Video, TAT และ Wind River เป็นต้น โดยใช้ชื่อกลุ่มว่า **OHA (Open Handset Alliances)**

ทั้งนี้ OHA ได้ร่วมมือกันพัฒนามาตรฐานสำหรับการพัฒนาซอฟต์แวร์ระบบเปิด (Open System) หรือ โอเพนซอร์ส (Open Source) โดยมีลิขสิทธิ์ตาม Apache Version 2 license

ซึ่งหลักลิขสิทธิ์ของ Apache จะอนุญาตให้ผู้พัฒนาสามารถนำไปโคด์ที่มีอยู่ไปพัฒนาต่อได้ ทั้งในส่วน ของแบบการค้า (Commercial) หรือซอฟต์แวร์กรรมลิขสิทธิ์ (Proprietary) และแบบเชิฟฟ์หรือฟรีแวร์ (Freeware) ก็ได้

ผู้อ่านสามารถศึกษาเพิ่มเติมหรือดาวน์โหลดซอฟต์แวร์ได้จากเว็บไซต์ <http://developer.android.com>



▲ เว็บไซต์ Android Developers

จนถึงปัจจุบันมีการพัฒนาเอนดรอยด์ออกมาหลายรุ่น ตั้งแต่รุ่นที่ 1.5 จนกระทั่งรุ่น 4.0.3 ดังตาราง

Platform	Codename	API Level	Distribution
Android 1.5	Cupcake	3	0.3%
Android 1.6	Donut	4	0.7%
Android 2.1	Eclair	7	5.5%
Android 2.2	Froyo	8	20.9%
Android 2.3-Android 2.3.2	Gingerbread	9	0.5%
Android 2.3.3-Android 2.3.7		10	63.9%
Android 3.0	Honeycomb	11	0.1%
Android 3.1		12	1.0%
Android 3.2		13	2.2%
Android 4.0-Android 4.0.2	Ice Cream Sandwich	14	0.5%
Android 4.0.3-Android 4.0.4		15	4.4%

## ประเภทของระบบปฏิบัติการ Android

เนื่องจากระบบปฏิบัติการ Android เป็นซอฟต์แวร์ระบบเปิด จึงอนุญาตให้นักพัฒนาหรือผู้ที่สนใจสามารถดาวน์โหลด Source Code ได้ ทำให้มีผู้พัฒนาจากหลาย ๆ ฝ่ายนำ Source Code มาปรับแต่งและพัฒนาระบบแอปพลิเคชันบนระบบ Android ในแบบฉบับของตนเองมากขึ้น โดยสามารถแบ่งประเภทของระบบ Android ออกเป็นกลุ่มฯ ได้ 3 ประเภทดังต่อไปนี้

- **Android Open Source Project (AOSP)** เป็นระบบ Android ประเภทแรกที่ทางบริษัท Google เปิดให้สามารถนำ Source Code ไปติดตั้งและใช้งานในอุปกรณ์ต่างๆ ได้โดยที่ไม่ต้องเสียค่าใช้จ่าย
- **Open Handset Mobile (OHM)** เป็นแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่ม Open Handset Alliances (OHA) ซึ่งบริษัทเหล่านี้จะพัฒนาระบบ Android ในแบบฉบับของตนเอง โดยมีรูปร่างหน้าตาการแสดงผล และฟังก์ชันการใช้งานที่แตกต่างกัน รวมไปถึงอาจจะมีความเป็นเอกลักษณ์ และรูปแบบการใช้งานเป็นของแต่ละบริษัท และโปรแกรมแอนดรอยด์ประเภทนี้ก็จะได้รับสิทธิ์บริการเสริมต่างๆ จากภูมิภาคที่เรียกว่า **GMS (Google Mobile Service)** ซึ่งเป็นบริการเสริมที่ทำให้ระบบ Android มีประสิทธิภาพมากขึ้นนั่นเอง
- **Cooking** หรือ **Customize** เป็นระบบ Android ที่นักพัฒนานำเอาชอร์สโค้ดจากแหล่งต่างๆ มาปรับแต่งให้อยู่ในแบบฉบับของตนเอง ซึ่งการพัฒนาจะต้องปลดล็อกสิทธิ์ในการใช้งานอุปกรณ์ (Unlock) เลยก่อนจะสามารถติดตั้งได้ ทั้งนี้ระบบ Android ประเภทนี้ถือได้ว่าเป็นประเภทที่มีความสามารถสูงที่สุด เนื่องจากจะได้รับการปรับแต่งขึดความสามารถต่างๆ ให้มีความเข้ากันได้กับอุปกรณ์นั้นๆ จากผู้ใช้งานจริง

อย่างไร้กีดขวาง Android ของแต่ละผู้ผลิตจะได้รับบริการ Google Mobile Service (GMS) นั้น ผู้ผลิตจะต้องทดสอบระบบและขออนุญาต กับทางภูมิภาค เลยก่อน จึงสามารถนำเครื่องออกสู่ตลาดได้

## สามารถพัฒนาอะไรได้บ้างบน Android

การพัฒนาแอปพลิเคชันบน Android มีไลบรารี (Library) การใช้งานมากมายที่อำนวยความสะดวกให้แก่นักพัฒนา ในทั้งข้อนี้จะยกตัวอย่างเฉพาะไลบรารีที่น่าสนใจ ตัวอย่างเช่น

### NOTE



**ไลบรารี (Library)** สำหรับใช้ในการพัฒนาแอปพลิเคชันจะมีโครงสร้างเป็นมาตรฐาน ซึ่งภายในไลบรารีนั้นจะมีโค้ดที่นำกลับมาใช้ได้ (Reusable) อยู่เป็นจำนวนมาก ซึ่งจะทำให้นักพัฒนาประหยัดเวลาในการพัฒนาแอปพลิเคชันต่างๆ ได้อย่างมาก เนื่องจากไม่มีความจำเป็นต้องเขียนโค้ดเดิมๆ ซ้ำๆ อีก เมื่อมีการพัฒนาแอปพลิเคชันใหม่ หรือแอปพลิเคชันที่มีฟังก์ชันการทำงานที่คล้ายคลึงกัน

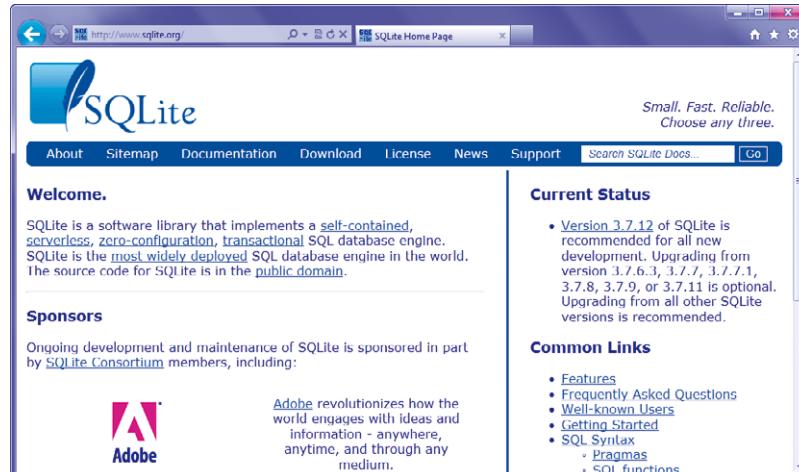
ดังนั้น ไลบรารีนี้จะทำหน้าที่หลักคือ การนำกลับมาใช้หรือพัฒนาต่อโดยการทำงานหลักให้ดีขึ้นนั่นเอง

- **Dalvik Virtual Machine (VM)** เป็นส่วนการสร้างเครื่องจำลองแบบเลมีอินที่มีการออกแบบให้เหมาะสมกับอุปกรณ์เคลื่อนที่หรืออุปกรณ์มือถือ
- **Integrated Browser** เป็นการผนวกกับ Web Browser เข้าไว้กันแน่นโดยยึด พั้นนี้มีพื้นฐานมาจากซอฟต์แวร์เว็บคิต (WebKit)



▲ เว็บไซต์ WebKit

- **Optimized Graphic** เป็นส่วนสนับสนุนการทำงานแบบกราฟิกทั้งในส่วน 2 มิติและ 3 มิติ โดยใช้เครื่องมือ OpenGL
- **SQLite** เป็นส่วนสนับสนุนการทำงานสำหรับการจัดเก็บฐานข้อมูล

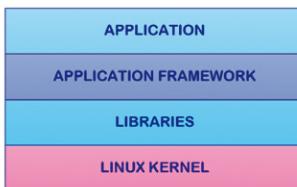


▲ เว็บไซต์ SQLite

- **Media Support** เป็นส่วนสนับสนุนการทำงานแบบลีโอประสม หรือมัลติมีเดีย เช่น ออดิโอ วิดีโอ และรูปภาพ
- **GSM Telephony** เป็นส่วนรองรับการทำงานบนระบบโทรศัพท์เคลื่อนที่แบบ GSM (Global System for Mobile Communications)
- **Bluetooth, EDGE, 3G, WiFi** เป็นส่วนรองรับการทำงานกับ Bluetooth, EDGE (Enhanced Data rates for GSM Evolution), 3G และ WiFi
- **Camera, GPS, Compass, Accelerometer** เป็นส่วนสนับสนุนการทำงานของระบบกล้องถ่ายรูป, ระบบกำหนดตำแหน่งบนโลก หรือ GPS (Global Positioning System), เซ็มทิค และการวัดอัตราความเร่ง
- **Rich Development Environment** เป็นส่วนสนับสนุนฟังก์ชันต่างๆ ที่ช่วยในการพัฒนาแอพพลิเคชัน เช่น Emulator, Debugging Tool, Memory and Performance Profiling และ Plug in สำหรับเครื่องมือ Eclipse

## สถาปัตยกรรมของแอนดรอยด์ (Android Architecture)

สถาปัตยกรรมการออกแบบแอนดรอยด์ (Android Architecture) นั้นถูกแบ่งออกเป็นลำดับชั้น หรือจะเรียกว่า **Layer** โดยที่แต่ละชั้นจะเรียกวิธารจากกระดับชั้นที่อยู่ด้านล่างของตัวเอง ซึ่งสามารถแบ่งออกเป็น 4 ชั้นหลักคือ ชั้นลีนักซ์เคอร์แนล (Linux Kernel), ชั้นไลบรารี (Libraries), ชั้นแอพพลิเคชันเฟรมเวิร์ค (Application Framework) และชั้นแอพพลิเคชัน (Application)



▲ สถาปัตยกรรม Android

- **ชั้นแอพพลิเคชัน (Application)** ชั้นนี้เป็นชั้นบนสุดของโครงสร้าง Android ซึ่งเป็นส่วนของแอพพลิเคชันที่พัฒนาขึ้นมาใช้งาน เช่น แอพพลิเคชันรับส่งอีเมล, แอพพลิเคชันโทรศัพท์ (Phone Dial), แอพพลิเคชันเบราว์เซอร์ (Web Browser) เป็นต้น ทั้งนี้โปรแกรมในชั้นแอพพลิเคชันนั้น จะอยู่ในรูปแบบของไฟล์ .apk ซึ่งโดยทั่วไปแล้วจะอยู่ในไดเรกทอรี data/app ของโทรศัพท์



▲ สถาปัตยกรรมชั้นแอพพลิเคชัน



▲ ตัวอย่างแอพพลิเคชัน

**TIPS**

แอพพลิเคชันและวิดเจ็ต (Widget) เมื่อถูกกันคือ เป็นโปรแกรมที่ถูกพัฒนาขึ้นเมื่อกัน แต่จะต่างกันที่แอพพลิเคชันนั้นจะทำงานโดยใช้พื้นที่เต็มจอ (Full Screen) แต่ในส่วนของวิดเจ็ต หรือบางครั้งเรียกว่า แก๊กเก็จ (Gadget) จะทำงานบนพื้นที่ขนาดเล็กไม่เต็มหน้าจอ

- **ชั้นแอพพลิเคชันเฟรมเวิร์ค (Application Framework)** โดยปกติแล้วนักพัฒนาสามารถเรียกใช้งาน Android ผ่าน API (Application Programming Interface) ได้ ซึ่ง Android ได้ออกแบบไว้เพื่อลดความซ้ำซ้อนในการใช้งานซ้ำของ Application Component ซึ่งมีตัวอย่างแอพพลิเคชันเฟรมเวิร์คดังนี้
  - **View System** เป็นส่วนควบคุมการทำงานสำหรับการสร้างแอพพลิเคชัน เช่น lists, grids, text boxes, buttons และ embeddable web browser
  - **Location Manager** เป็นส่วนจัดการค่าตำแหน่งของเครื่องอุปกรณ์พกพาเคลื่อนที่
  - **Content Provider** เป็นส่วนควบคุมการเข้าถึงของข้อมูลที่มีการใช้งานร่วมกัน (Share Data) ระหว่างแอพพลิเคชันที่แตกต่างกัน เช่น ข้อมูลผู้ใช้ (Contact)
  - **Resource Manager** เป็นส่วนจัดการการเข้าใช้ข้อมูลต่างๆ ที่ไม่ใช้โค้ด เช่น localized strings, graphics และ layout ซึ่งจะอยู่ในไฟล์ res/

ทั้งนี้ข้อมูลต่างๆ ในส่วนนี้จะถูกคอมไพล์ และพนวกเข้ากับโปรแกรมที่เขียนขึ้น ณ เวลาการ Build โดยแอนดรอยด์จะใช้เครื่องมือ adpt สำหรับการคอมไпал์ และหลังจากการคอมไпал์จะสร้างคลาสที่ชื่อว่า R ซึ่งเป็นส่วนที่จะบุกงข้อมูลต่างๆ ที่ใช้อ้างอิงสำหรับโปรแกรมที่ถูกพัฒนาขึ้น

- **Notification Manager** เป็นส่วนควบคุมอีเว้นต์ (Event) ต่างๆ ที่แสดงบนแถบสถานะ (Status bar) เช่น ในการแจ้งที่ได้รับข้อความและการแจ้งเตือนต่างๆ
- **Activity Manager** เป็นส่วนควบคุม Life Cycle ของแอพพลิเคชัน



▲ สถาปัตยกรรมชั้นแอพพลิเคชันเฟรมเวิร์ค

**NOTE**

API (Application Program Interface) คือ API หรือส่วนต่อประสานโปรแกรมประยุกต์คือวิธีการเฉพาะสำหรับการเรียกใช้งานระบบปฏิบัติการหรือแอพพลิเคชันอื่นๆ ซึ่งทำหน้าที่เชื่อมต่อการทำงานระหว่างแอพพลิเคชันกับระบบปฏิบัติการ ทั้งนี้การที่แอพพลิเคชันจะเชื่อมต่อการทำงานกับระบบปฏิบัติการได้นั้น จำเป็นต้องมี API เป็นตัวเชื่อม ซึ่งหากไม่มีการเปิดเผย API ของระบบปฏิบัติการออกมานั้นแล้ว ผู้ที่พัฒนาแอพพลิเคชันจะมีความลำบาก เมื่อต้องการพัฒนาแอพพลิเคชันให้มีความเข้ากันได้กับระบบปฏิบัติการได้อย่างมีประสิทธิภาพ

- **ชั้นไลบรารี (Library)** แอนดรอยด์ได้รวมรวมกลุ่มของไลบรารีต่างๆ ที่สำคัญและมีความจำเป็นต่อการพัฒนาโปรแกรมเอาไว้มาก many ซึ่งถูกเขียนไว้ด้วยภาษา C และ C++ โดยตัวอย่างของไลบรารีที่สำคัญที่ผู้อ่านควรรู้คือ
  - **System C library** เป็นกลุ่มของไลบรารีมาตรฐานที่อยู่บนพื้นฐานของภาษา C ไลบรารี (libc)
  - **Media Libraries** เป็นกลุ่มการทำงานมัลติมีเดีย เช่น ออดิโอ วิดีโอ รวมถึงรูปภาพต่างๆ เช่น ไฟล์สกุล MPEG4, H.264, MP3, AAC, AMR, JPG และ PNG
  - **Surface Manager** เป็นกลุ่มการจัดการรูปแบบของหน้าจอ การวาดหน้าจอ
  - **2D/3D library** เป็นกลุ่มของกราฟิกแบบ 2 มิติ หรือ SGL (Scalable Graphics Library) และแบบ 3 มิติ หรือ OpenGL
  - **FreeType** เป็นกลุ่มของบิตแมป (Bitmap) และเวคเตอร์ (Vector) สำหรับการрендเรอร์ (Render) ภาพ
  - **SQLite** เป็นกลุ่มของฐานข้อมูล ซึ่งเป็นระบบฐานข้อมูลที่ใช้เซ็นเดียวกันกับซอฟต์แวร์ Firefox และ Apple iPhone ทั้งนี้นักพัฒนาสามารถใช้ฐานข้อมูลนี้เก็บข้อมูลของแอพพลิเคชันต่างๆ ได้
  - **Browser Engine** เป็นกลุ่มของการแสดงผลบนเว็บบราวเซอร์โดยอยู่บนพื้นฐานของ WebKit ซึ่งมีลักษณะคล้ายคลึงกันกับ Google Chrome, Safari และ Nokia S60

สำหรับการเรียกใช้แอพพลิเคชันต่างๆ ในชั้นไลบรารี จะไม่สามารถเรียกใช้แอพพลิเคชันในตัวเองได้โดยจะต้องเรียกใช้แอพพลิเคชันในชั้นที่สูงกว่าเท่านั้นจึงจะสามารถเรียกใช้ได้ นอกจากนี้ในชั้นไลบรารีนี้ Android ยังแบ่งเป็นชั้นย่อยๆ ที่เรียกว่า **Android Runtime** ซึ่งประกอบไปด้วย 2 ส่วนหลักคือ Dalvik VM และ Core Java Library

- **Dalvik VM (Virtual Machine)** ส่วนนี้ถูกเขียนด้วยภาษา Java เพื่อใช้เฉพาะการใช้งานอุปกรณ์พกพาเคลื่อนที่ อย่างไรก็ตามสิ่งที่แตกต่างจาก Java VM (Virtual Machine) คือ Dalvik VM ซึ่งจะรันไฟล์ .dex ที่คอมไพล์มาจากการแปลงไฟล์ .class และ .jar โดยมี tool ที่ชื่อว่า dx ทำหน้าที่ในการบีบอัดคลาส Java ทั้งนี้ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์พกพามากกว่า .class โดยเฉพาะอย่างยิ่งประลิมิเตชันในการใช้พลังงานจากแบตเตอรี่
- **Core Java Library** ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็จะมีความแตกต่างจากไลบรารีของ Java SE (Java Standard Edition) และ Java ME (Java Mobile Edition)



▲ สถาปัตยกรรมชั้นไลบรารี

**TIPS**

ความแตกต่างระหว่าง Java VM และ Dalvik VM คือ การแปลงคำสั่ง หรือมีอินเตอร์พรีเตอร์ (Interpreter) ที่แตกต่างกัน โดยส่วนของ Java VM นั้นจะมีรูปแบบเป็นแบบสแต็ค (Stack) แต่ส่วนของ Dalvik VM นั้นจะเป็นแบบรีจิสเตอร์ (Register)

โดยที่ Java VM นั้นจะเป็นการอ่าน byte code (class) ของวัวจากนั้นจะแปลง byte code เหล่านั้นให้เป็นคำสั่งต่างๆ และโหลดเข้าไปเก็บไว้ในหน่วยความจำแบบสแต็ค

แต่ส่วน Dalvik VM นั้นจะมีการทำงานที่คล้ายกัน เพียงแต่จะเป็นการโหลดคำสั่งต่างๆ เข้าไปเก็บไว้ในหน่วยความจำแบบรีจิสเตอร์แทน

โดยภาพรวมแล้วทั้ง 2 ประเภทจะมีการทำงานที่คล้ายกัน แต่ Java VM นั้นจะมีการสร้างคำสั่งต่างๆ มากกว่า ซึ่งทำให้ไม่เหมาะสมกับการนำไปประยุกต์ใช้ในอุปกรณ์เคลื่อนที่ ซึ่งโดยปกติแล้วจะมีทรัพยากรที่จำกัด เช่น ชีดจำกัดของ CPU (Central Processing Unit) และ RAM (Random Access Memory)

- **ชั้นลิ้นกูซ์เครอเรนเดล (Linux Kernel)** ระบบ Android อยู่บนพื้นฐานของระบบปฏิบัติการ Linux โดยชั้น Linux Kernel นั้นมีฟังก์ชันการทำงานหลายๆ ส่วน ซึ่งแต่ละส่วนถูกพัฒนาขึ้นด้วยภาษา C เช่น การจัดการหน่วยความจำ (Memory Management), การจัดการโปรเซส (Process Management), การเชื่อมต่อเครือข่าย (Networking) และฟังก์ชันการทำงานส่วนอื่นที่เกี่ยวกับระบบปฏิบัติการ ทั้งนี้นักพัฒนาจะไม่มีสิทธิเข้าถึงส่วนนี้ได้โดยตรง อย่างไรก็ตามนักพัฒนาสามารถเข้าถึงระบบปฏิบัติการ Linux ได้จากชุดคำสั่ง Command Prompt เช่น adb shell ซึ่งจะสามารถใช้เครื่องมือต่างๆ ได้ เช่น การเข้าถึงระบบไฟล์ (File System), โพรเซสการคัดลอกไฟล์ (Copy File) เป็นต้น

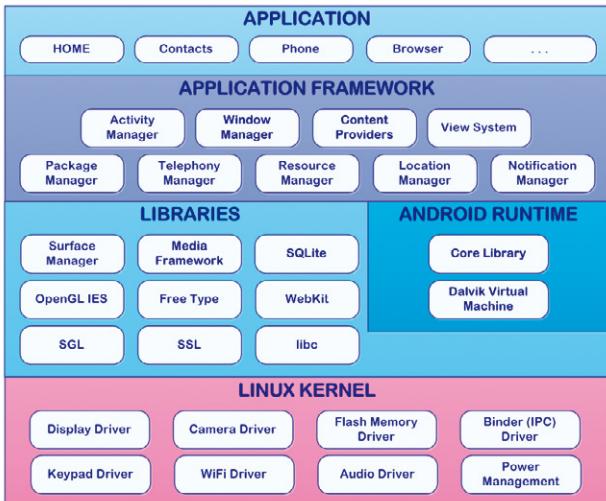


▲ สถาปัตยกรรมชั้น Linux Kernel

**NOTE**

Linux Kernel ถูกสร้างขึ้นโดย ลินุส เบเนดิกต์ โทรวัลเดลส์ (Linus Benedict Torvalds) ในปี ค.ศ.1991 ซึ่งขณะนั้นศึกษาอยู่ที่มหาวิทยาลัยเอลซิงกิ ประเทศฟินแลนด์ (The University of Helsinki)

จากที่กล่าวมาพอสรุปเป็นภาพของสถาปัตยกรรม Android ได้ดังนี้



▲ สถาปัตยกรรมของ Android

## ส่วนประกอบของแอพพลิเคชัน (Application Component)

คุณลักษณะอย่างหนึ่งของแอนดรอยด์คือ เป็นแอพพลิเคชันที่สามารถใช้เป็นส่วนประกอบของแอพพลิเคชันอื่นๆ ได้ ตัวอย่างเช่น ถ้าผู้อ่านต้องการสร้างแอพพลิเคชันให้แสดงการเลื่อนของรายการรูปภาพ โดยที่ผู้อ่านอาจมีแอพพลิเคชันส่วนอื่นที่ได้พัฒนาไว้แล้ว ผู้อ่านสามารถเรียกใช้แอพพลิเคชันในส่วนที่มีอยู่มาพัฒนาต่อได้ โดยที่ไม่จำเป็นต้องพัฒนาขึ้นมาเอง เพื่อความสะดวกและรวดเร็วในการพัฒนาแอพพลิเคชัน ซึ่งเรียกแอพพลิเคชันเหล่านั้นว่า **Application Component**

ชั้น Application Component ของแอนดรอยด์สามารถแบ่งออกเป็น 4 ส่วนคือ Activity, Service, Content Provider และ Broadcast and Intent Receiver

- **Activity** คือ หน้าจอที่ติดต่อกับผู้ใช้ ทั้งนี้ในแต่ละแอพพลิเคชันอาจจะมีได้มากกว่า 1 หน้าจอ หรือ 1 Activity ซึ่งแต่ละ Activity จะทำหน้าที่เก็บสถานะการใช้งานในส่วนต่างๆ ตัวอย่างเช่น
  - ในการแสดงรายการเมนู นักพัฒนาสามารถเลือกให้รายการเมนูที่แสดงออกมามีภาพและคำบรรยายได้ภาพได้
  - สำหรับแอพพลิเคชันส่วนข้อความอาจจะมี Activity หนึ่งที่แสดงรายการของส่วนติดต่อในการส่งข้อความ อีก Activity หนึ่งจะเป็นส่วนของการเลือกการติดต่อ และ Activity อื่นๆ จะทำหน้าที่ถือข้อความเก่าที่ถูกส่งมาแล้ว เป็นต้น
- **Service** คือ งานหรือบริการต่างๆ ที่ทำงานอยู่เบื้องหลัง เช่น Service ที่เปิดคนตีอู่ ขณะที่ผู้ใช้งานทำงานอื่นๆ หรือใช้แอพพลิเคชันอื่นๆ ไปด้วย

- **Broadcast and Intent Receiver** คือ การตอบสนอง ซึ่งโดยปกติแล้ว Broadcast Receiver จะเป็นการตอบสนองต่อการเกิดอีเวนต์ของระบบในวงกว้าง เช่น การประกาศเตือนว่าแบตเตอรี่ใกล้จะหมดแล้ว เป็นต้น นอกจากนี้ Intent Receiver เป็นส่วนทำให้แอพพลิเคชันอื่นๆ เข้าถึงการทำงานของ Activity และ Service ซึ่งในการปฏิบัติงานแต่ละอย่างเป็นการตอบสนองการร้องขอจากข้อมูลหรือบริการของ Activity อื่นๆ
- **Content Provider** คือ ส่วนของการให้บริการข้อมูลสำหรับแต่ละแอพพลิเคชัน ทั้งนี้ข้อมูลสามารถเก็บอยู่ในรูปแบบของระบบไฟล์ หรือฐานข้อมูลก็ได้ เช่น Google สามารถเข้าใช้งานข้อมูลร่วมกันกับผู้ใช้งานได้ในแอพพลิเคชันที่ต้องการข้อมูลของผู้ใช้งาน

**NOTE**

การอนุญาตให้การเข้าใช้งานในแต่ละส่วนประกอบของแอพพลิเคชัน (Activating Component) หรือ Intent คือ วิธีที่อธิบาย Action ต่างๆ เช่น การเลือกรูปภาพ การเปิดโปรแกรมฟังเพลง โดยที่ไปแล้ว Intent จะถูกใช้สำหรับการสร้างแอพพลิเคชันใหม่จากแอพพลิเคชันเดิม โดยที่นักพัฒนาไม่จำเป็นต้องสร้างใหม่ทั้งหมด (Mobile Mashup) ซึ่ง Intent อนุญาตให้นักพัฒนาสามารถติดต่อระหว่างแอพพลิเคชันและบริการที่แตกต่างกันได้

**TIPS**

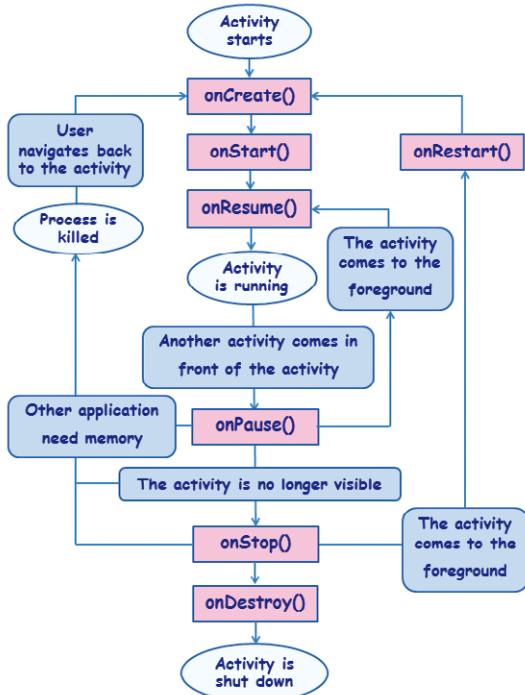
โดยปกติแล้วในแต่ละแอพพลิเคชันจะทำงานแยกกันในแต่ละprocress (Process) ซึ่งมีการทำงานที่แยกจากกันโดยชัดเจน ในกรณีที่นักพัฒนาต้องการเข้าถึงการทำงานหรือการขอสิทธิ์เข้าใช้นั้น นักพัฒนาจะต้องกำหนดการขอเข้าใช้ที่ไฟล์ AndroidManifest.xml เช่น

- INTERNET เป็นการอนุญาตให้เข้าใช้อินเทอร์เน็ต
- READ\_CONTACTS เป็นการอนุญาตให้อ่านข้อมูลผู้ใช้
- WRITE\_CONTACTS เป็นการอนุญาตให้เขียนข้อมูลผู้ใช้
- RECEIVE\_SMS เป็นการอนุญาตให้ตรวจสอบการรับ SMS (Short Message Service)
- ACCESS\_COARSE\_LOCATION เป็นการอนุญาตให้ใช้การอ้างอิงตำแหน่งแบบกว้างๆ (Coarse)
- ACCESS\_FINE\_LOCATION เป็นการอนุญาตให้ใช้การอ้างอิงตำแหน่งที่มีความถูกต้องสูง (Fine)
- GPS

## วงรอบชีวิตของแอพพลิเคชัน (Application Life Cycle)

โดยปกติแล้วแอพพลิเคชันจะทำงานแยกกันในแต่ละprocress และในแต่ละprocressอาจจะมี Activity/Service ที่ทำงานอยู่มากกว่า 1 Activity/Service ดังนั้น ในแต่ละแอพพลิเคชันอาจจะมีมากกว่า 1 Activity ซึ่งในการเริ่มทำงานของ Activity จะเริ่มด้วย **startActivity()** สำหรับแบบซิงโครนัส (Synchronous) และจะเริ่มด้วย **startSubActivity()** สำหรับแบบอะซิงโครนัส (Asynchronous) โดยในแต่ละ Activity จะมีวงรอบชีวิต (Life Cycle) ที่แยกจากกันโดยชัดเจน ซึ่งมีสถานะการทำงานหลักดังนี้

- **onCreate (Bundle savedInstanceState)** ส่วนนี้จะถูกเรียกใช้งานเมื่อเริ่มทำงาน ในกรณีที่ มีการเรียกใช้งานเมธอด (Method) นี้ Android Framework จะนำ Bundle object ไปบันทึกไว้ใน Activity ก่อนที่ Activity จะทำงาน ซึ่งจากนั้นจะตามด้วยฟังก์ชัน onStart()
- **onStart()** ส่วนนี้เป็นการระบุว่า Activity นั้นๆ จะถูกแสดงขึ้นมา จากนั้นสถานะจะถูกย้ายไป เป็นสถานะ onResume แต่ถ้า Activity นั้นไม่สามารถทำงานได้ด้วยเหตุผลบางประการ สถานะ จะถูกย้ายไปเป็นสถานะ onStop
- **onRestart()** ส่วนนี้เป็นการระบุว่า Activity นั้นจะถูกแสดงขึ้นมาอีกครั้งหนึ่ง ซึ่งจะตามด้วย สถานะ onStart()
- **onResume()** ส่วนนี้จะถูกเรียกเมื่อ Activity นั้นๆ มีการติดต่อบวญิพิมพันธ์กับผู้ใช้งาน เช่น นัก พัฒนาต้องการเรียก Activity นั้นให้ขึ้นมาทำงานอีกรอบหนึ่ง หลังจากที่ Activity นั้นอยู่ในสถานะ onPause
- **onPause()** ส่วนนี้จะถูกเรียกใช้เมื่อผู้ใช้ไม่ต้องใช้งาน Activity นั้นๆ ในช่วงระยะเวลาหนึ่งๆ ซึ่งจะ ตามด้วยสถานะ onRestart() เมื่อต้องการกลับมาทำ Activity นั้นอีกครั้งหนึ่ง หรือตามด้วยสถานะ onDestroy() เมื่อต้องการปิด Activity นั้นๆ
- **onStop()** ส่วนนี้จะถูกเรียกเมื่อผู้ใช้ไม่ต้องใช้งาน Activity นั้นๆ ในช่วงระยะเวลาหนึ่งๆ ซึ่งจะ ตามด้วยสถานะ onRestart() เมื่อต้องการกลับมาทำ Activity นั้นอีกครั้งหนึ่ง หรือตามด้วยสถานะ onDestroy() เมื่อต้องการปิด Activity นั้นๆ
- **onDestroy()** ส่วนนี้จะถูกเรียกเมื่อมีการปิดการทำงานของแต่ละ Activity



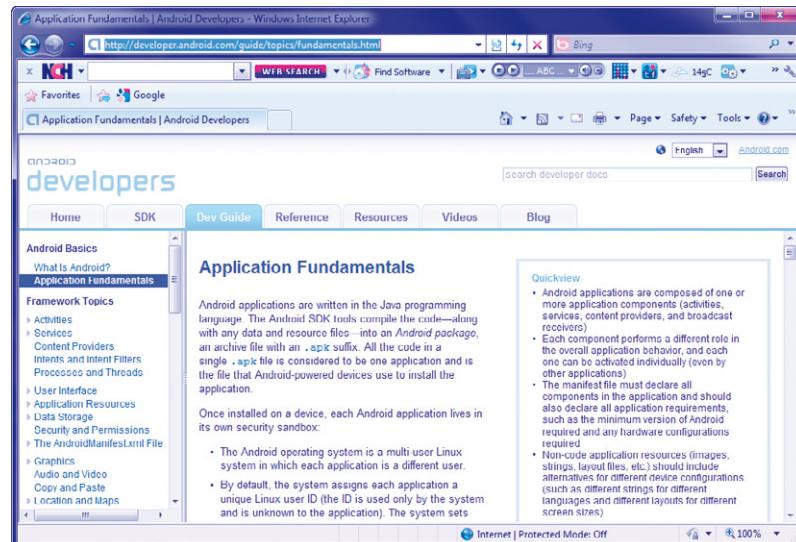
▲ Application Life Cycle

## NOTE



สำหรับ Android นั้น แอพพลิเคชันที่อยู่เบื้องหน้าจะมีได้เพียง 1 หน้าจอเท่านั้น ซึ่งโดยปกติแล้ว จะใช้พื้นที่ทั้งหน้าจอของโทรศัพท์เคลื่อนที่ ยกเว้นส่วนของแถบสถานภาพ หรือแถบสถานะ (Status bar) แต่อย่างไรก็ตาม ผู้ใช้สามารถเลือกหรือเปลี่ยนแอพพลิเคชันขึ้นมาใช้งานได้ ทั้งนี้แอพพลิเคชันทั้งหมด จะถูกเก็บค่าสถานะไว้ที่แอพพลิเคชันสแต็ค (Application Stack) ซึ่งควบคุมโดย Activity Manager

ดังนั้น เมื่อผู้ใช้คลิกปุ่มกลับ หรือ Back ในส่วนของ Activity Manager ก็จะดึงเอาแอพพลิเคชันบน สแต็คขึ้นมาทำงาน ทั้งนี้ผู้อ่านสามารถศึกษาเพิ่มเติมได้จากเว็บไซต์ <http://developer.android.com/guide/topics/fundamentals.html>



## สรุปก้ายบท

ในบทนี้เน้นการอธิบายถึงพื้นฐานและทฤษฎีเบื้องต้นของแอนดรอยด์ รวมไปถึงสถาปัตยกรรม ส่วนประกอบของแอพพลิเคชัน และวิธีการเขียนแอพพลิเคชัน

## แบบฝึกหัดก้ายบท

- จงอธิบายความแตกต่างระหว่าง Content Provider และ Resource Manager
- จงอธิบายความสัมพันธ์ระหว่างแอพพลิเคชัน, โฟร์เชล และแอ็คทิวิตี้

## การติดตั้งเครื่องมือสำหรับการพัฒนา แอปพลิเคชันบน Android

เนื้อหาในบทนี้ เพื่อให้ผู้อ่านได้เรียนรู้และสามารถติดตั้งเครื่องมือต่างๆ ที่จำเป็นสำหรับใช้ในการพัฒนาแอปพลิเคชันบน Android ได้ ซึ่งถือได้ว่าเป็นก้าวแรกสำหรับการเริ่มต้นพัฒนาแอปพลิเคชันบน Android โดยทั่วไปแล้วแอปพลิเคชันใน Android จะพัฒนาด้วยภาษา Java ดังนั้น ผู้อ่านจะต้องมีความคุ้นเคยกับภาษา รวมไปถึงการใช้งานฟังก์ชันต่างๆ ที่จำเป็น เนื่องจากในหนังสือเล่มนี้จะไม่ได้อธิบายถึงพื้นฐานของ การเขียนโปรแกรมภาษา Java (ภาคผนวกจะกล่าวถึงรายละเอียดเชิงโครงสร้างและหลักการที่สำคัญเบื้องต้น ของภาษาJavaเพิ่มเติม)

ในหนังสือเล่มนี้ผู้เขียนจะเริ่มอธิบายตั้งแต่การติดตั้งการใช้งานเครื่องมือต่างๆ โดยมีสมมติฐานว่า ผู้อ่านไม่มีความรู้สำหรับการพัฒนาแอปพลิเคชันบน Android มา ก่อน ดังนั้น เนื้อหาในบทที่ 2 จะประกอบด้วย เนื้อหาดังนี้

1. การติดตั้ง JDK (Java Development Kit)
2. การติดตั้งโปรแกรม Eclipse
3. การติดตั้ง Android SDK (Android Software Development Kit)
4. การติดตั้ง ADT (Android Development Tool)
5. การสร้าง AVD (Android Visual Device)

สำหรับขั้นตอนการติดตั้งเครื่องมือต่างๆ นั้น ผู้เขียนมีสมมติฐานว่าผู้อ่านมีการเชื่อมต่อ กับอินเทอร์เน็ต (Internet) ตลอดระยะเวลาในการติดตั้ง

## NOTE



ในหนังสือเล่มนี้มีการอ้างอิงเครื่องมือต่างๆ มากมาย ไม่ว่าจะเป็นโปรแกรม Eclipse หรือ Android SDK ซึ่งมีเวอร์ชันต่างๆ ให้ใช้งานมากมาย

ดังนั้น การพัฒนาแอปพลิเคชันภายในหนังสือเล่มนี้ ผู้เขียนพัฒนาโดยใช้โปรแกรมดังนี้

- จาวา JDK เวอร์ชัน 7 (jdk-7u2-windows-i586.exe)
- โปรแกรม Eclipse เวอร์ชัน Eclipse IDE for Java Developer (eclipse-jee-indigo-SR1-win32.zip) ที่ทำงานอยู่บนระบบปฏิบัติการวินโดว์ Windows 7 32 Bit
- Android SDK เวอร์ชัน 16 (android-sdk\_r16-windows.zip)
- ADT เวอร์ชัน 16 (ADT-16.0.0.zip)

โดยทดสอบการทำงานของแอปพลิเคชันบนเครื่องโทรศัพท์จำลองเวอร์ชัน Android 4.0.3 (หรือในบางตัวอย่างพัฒนาโดยใช้ Android SDK และทดสอบการทำงานของแอปพลิเคชันบนเครื่องโทรศัพท์ Samsung Galaxy Nexus ที่มี Android 4.0 ติดตั้งอยู่)

ซึ่งเมื่อผู้อ่านได้ติดตั้งหรือใช้งานโปรแกรมต่างๆ ก็อาจจะมีการเปลี่ยนแปลงตามเทคโนโลยี อย่างไรก็ตามโดยทั่วไปแล้วการทำงานก็จะมีความคล้ายคลึงกัน อาจจะมีการปรับแต่งพารามิเตอร์บางอย่างเพื่อให้มีความเข้ากันได้

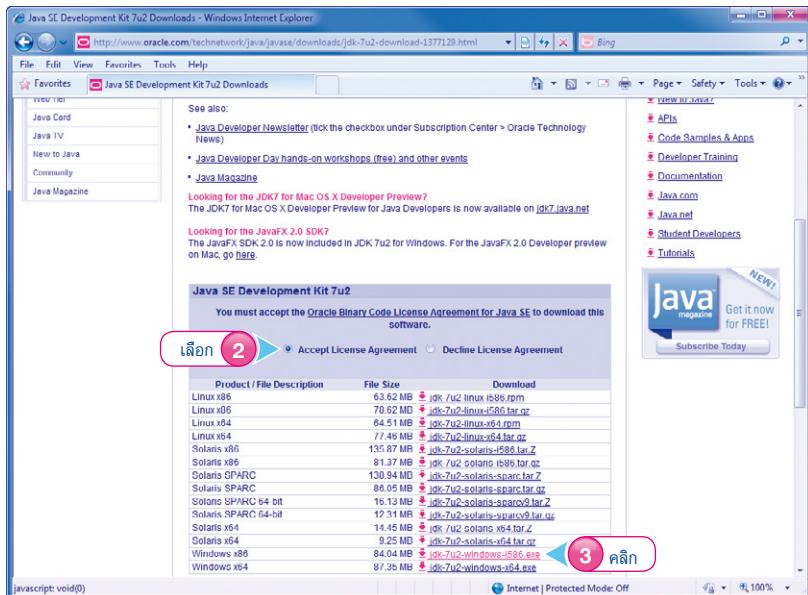
ทั้งนี้แอปพลิเคชันทุกแอปพลิเคชันที่นำเสนอภายในหนังสือเล่มนี้ ผู้อ่านสามารถนำไปพัฒนาบน Android เวอร์ชันใดก็ได้ ซึ่งมีขั้นตอนหรือวิธีการไม่แตกต่างกันมากนัก

ในที่นี้ผู้เขียนจะแรมโปรแกรมทุกตัวมาให้ใน CD ทั้งหมด เพื่อให้ผู้อ่านพร้อมศึกษาการพัฒนาแอปพลิเคชันได้ในทันที และมีโค้ดตัวอย่างแอปพลิเคชันในบางบทมาใน CD ด้วย เพื่อให้ผู้อ่านใช้ในการตรวจสอบความถูกต้อง แต่ผู้เขียนขอละโค้ดในบางตัวอย่างไว้ เนื่องจากต้องการให้ผู้อ่านได้ลองเขียนโค้ดและพัฒนาแอปพลิเคชันนั้นๆ ด้วยตัวเอง

## การติดตั้ง JDK (Java Development Kit)

เนื่องจากแอปพลิเคชันบน Android ถูกพัฒนาด้วยภาษา Java จึงมีความจำเป็นที่ผู้อ่านจะต้องติดตั้ง JDK (Java Development Kit) ทั้งนี้ผู้อ่านสามารถใช้ JDK ตั้งแต่เวอร์ชัน 5 ขึ้นไปได้ เนื่องจาก Android SDK ที่ใช้ประกอบกับ JDK นั้นรองรับ JDK ตั้งแต่เวอร์ชัน 5 ขึ้นไปเท่านั้น ซึ่งผู้อ่านสามารถตรวจสอบได้จากเว็บไซต์ <http://developer.android.com/sdk/requirements.html> โดยการติดตั้ง JDK นั้นผู้อ่านสามารถดาวน์โหลดตัวติดตั้งได้จากเว็บไซต์ <http://www.oracle.com/technetwork/java/javase/downloads> ซึ่งมีขั้นตอนดังนี้





ในหนังสือเล่มนี้ผู้เขียนได้ใช้ JDK เวอร์ชัน 7 (jdk-7u2-windows-i586.exe) ในการพัฒนาบนระบบปฏิบัติการ Windows 7 และเมื่อผู้อ่านดาวน์โหลดชุดติดตั้งมาเรียบร้อยแล้ว ก็สามารถติดตั้งได้ตามลำดับขั้นตอนต่อไปนี้

### NOTE



ผู้อ่านสามารถตรวจสอบเครื่องคอมพิวเตอร์ของผู้อ่านก่อนได้ว่า มีการติดตั้ง JDK อยู่ก่อนแล้วหรือไม่ ซึ่งผู้อ่านไม่มีความจำเป็นที่จะต้องติดตั้งช้าอีกครั้ง หรือในการนี้ที่ JDK ที่ติดตั้งอยู่เป็นเวอร์ชันก่อนหน้านี้ ก็สามารถติดตั้งช้าได้

ผู้อ่านสามารถตรวจสอบการติดตั้งได้โดยพิมพ์คำสั่ง "java -version" ที่ Command Prompt โดยคำสั่งนี้จะแสดงถึงเวอร์ชันของ JDK ที่ได้ดำเนินการติดตั้งไว้

```

Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Bannok>java -version
java version "1.7.0_01"
Java(TM) SE Runtime Environment (build 1.7.0_01-b08)
Java HotSpot(TM) Client VM (build 21.1-b02, mixed mode, sharing)

C:\Users\Bannok>

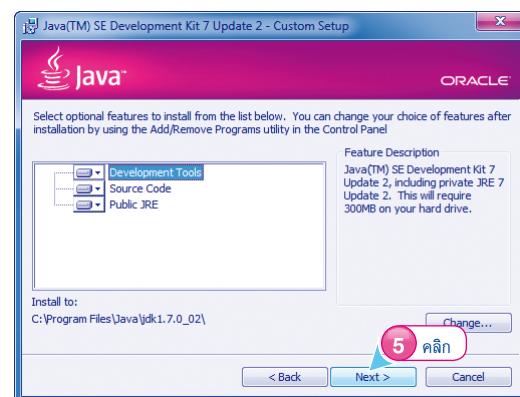
```



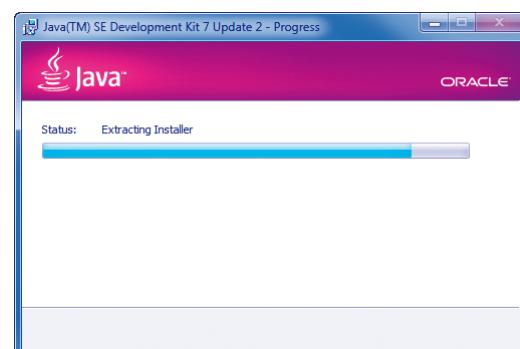
1. เริ่มต้นให้ผู้อ่านดับเบิลคลิกไฟล์ติดตั้ง JDK
2. จะปรากฏหน้าต่าง Java SE Development Kit ขึ้นมา
3. คลิกปุ่ม **Next >**



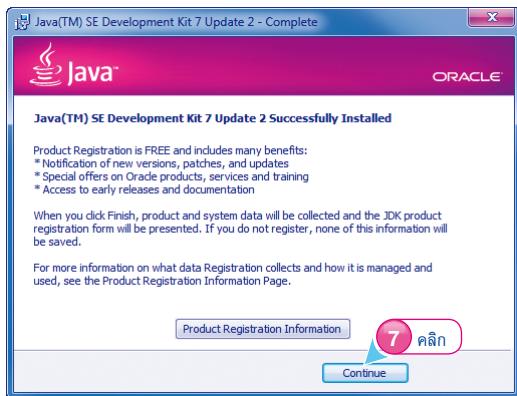
4. เลือกตำแหน่งติดตั้งโปรแกรม ในที่นี่แนะนำให้เลือกค่าเดิมที่โปรแกรมกำหนดไว้
5. คลิกปุ่ม **Next >**



6. ดำเนินการติดตั้งโปรแกรม



## 7. เมื่อติดตั้งสมบูรณ์ ให้คลิกปุ่ม Continue



## 8. เมื่อติดตั้งโปรแกรมเสร็จสมบูรณ์ ผู้อ่านควรจะตรวจสอบการติดตั้งอีกครั้งโดยพิมพ์คำสั่ง "java -version" ที่ Command Prompt โดยคำสั่งนี้จะแสดงถึงเวอร์ชันของ JDK ที่ได้ดำเนินการติดตั้งเรียบร้อยแล้ว



### NOTE

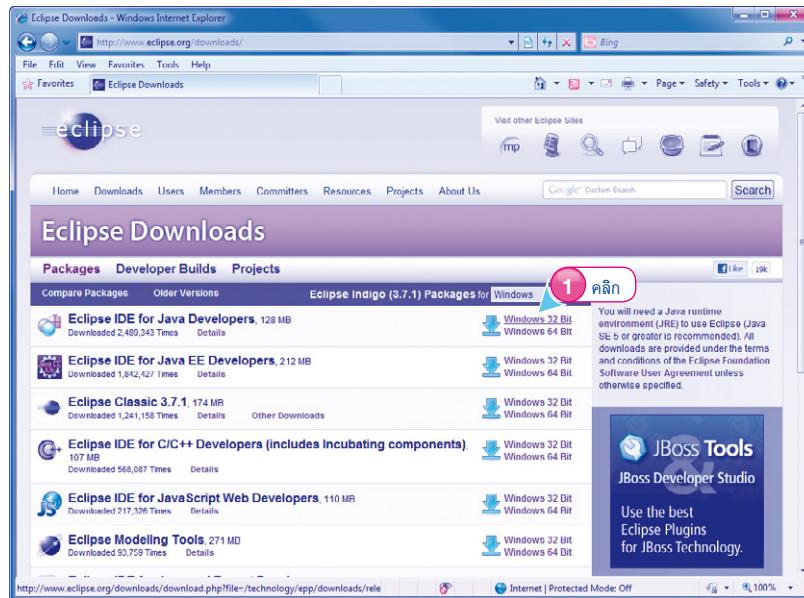


ผู้อ่านที่เริ่มพัฒนาแอพพลิเคชันบน Android จะจะมีความลับสนกันคำว่า ระบบปฏิบัติการ เป็นจากในที่นี้ระบบปฏิบัติการ 2 ระบบดังนี้

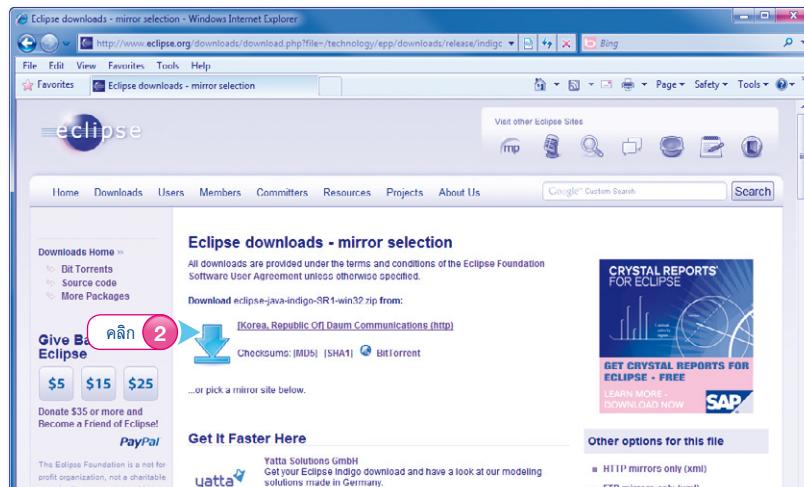
- ระบบปฏิบัติการ Windows คือ ระบบปฏิบัติการที่ใช้ในการพัฒนาแอพพลิเคชัน โดยมี JDK และเครื่องมือต่างๆ ทำงานอยู่บนระบบปฏิบัติการนี้
- ระบบปฏิบัติการ Android คือ ระบบปฏิบัติการที่ใช้งานสำหรับแอพพลิเคชันที่สร้างขึ้นดังนั้น ในการพัฒนาแอพพลิเคชันบนระบบปฏิบัติการ Windows เราจะใช้โทรศัพท์จำลอง (Emulator) จำลองโทรศัพท์ Android จริงขึ้นมา เพื่อทดสอบการทำงานของแอพพลิเคชัน และเมื่อผู้อ่านได้นำแอพพลิเคชันที่พัฒนาเสร็จลิ้นแล้วไปติดตั้งบนเครื่องโทรศัพท์เคลื่อนที่จริง แอพพลิเคชันนั้นก็จะทำงานอยู่บนระบบปฏิบัติการ Android

# การติดตั้งโปรแกรม Eclipse

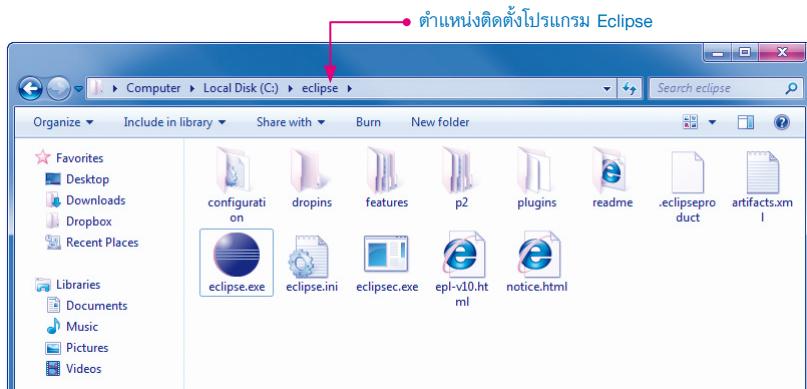
สำหรับการพัฒนาแอปพลิเคชันบน Android นั้นจะใช้ภาษา Java ในการพัฒนา ซึ่งในที่นี้ผู้เขียนจะใช้โปรแกรม Eclipse เป็นเครื่องมือในการพัฒนา โดยผู้อ่านสามารถดาวน์โหลด Eclipse IDE ได้จากเว็บไซต์ <http://www.eclipse.org/downloads/> ดังรูป



เมื่อผู้อ่านเข้าไปยังเว็บไซต์ของ Eclipse และให้ผู้อ่านดาวน์โหลด Eclipse IDE for Java Developer (`eclipse-jee-indigo-SR1-win32.zip`) ที่ทำงานอยู่บนระบบปฏิบัติการวินโดว์ ในที่นี้ผู้เขียนใช้ระบบปฏิบัติการ Windows 7 32 Bit



สำหรับการติดตั้ง Eclipse นั้นจะไม่มีความซับซ้อนเหมือนกับการติดตั้ง JDK โดยการติดตั้งจะเป็นเพียงการคลายไฟล์ (Unzip) เท่านั้น ในส่วนนี้ผู้เขียนจะติดตั้งโปรแกรม Eclipse ไว้ที่ไดเร็คทอรี C:\eclipse\

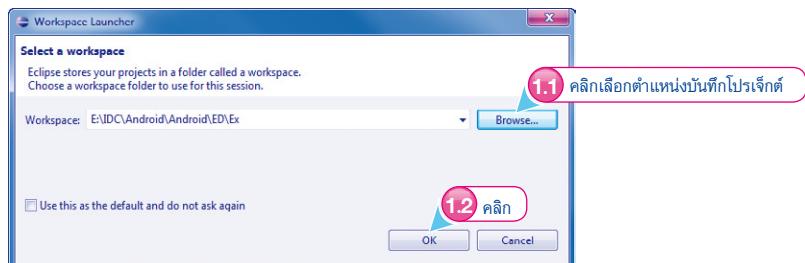


## เริ่มต้นเขียนโปรแกรมด้วย Eclipse

หัวข้อนี้จะแนะนำการใช้เครื่องมือ Eclipse สำหรับเขียนโปรแกรม และถือเป็นการทดสอบการติดตั้ง JDK และ Eclipse ว่าสามารถใช้งานได้จริง ในกรณีที่ไม่สามารถเขียนโปรแกรมด้วย Eclipse ได้ให้ผู้อ่านกลับไปตรวจสอบวิธีการติดตั้ง JDK และ Eclipse ใหม่อีกรอบ ก่อนที่จะดำเนินการติดตั้ง Android SDK

ในที่นี้จะทดสอบเขียนโปรแกรมแสดงข้อความ โดยมีขั้นตอนการเขียนโปรแกรมดังนี้

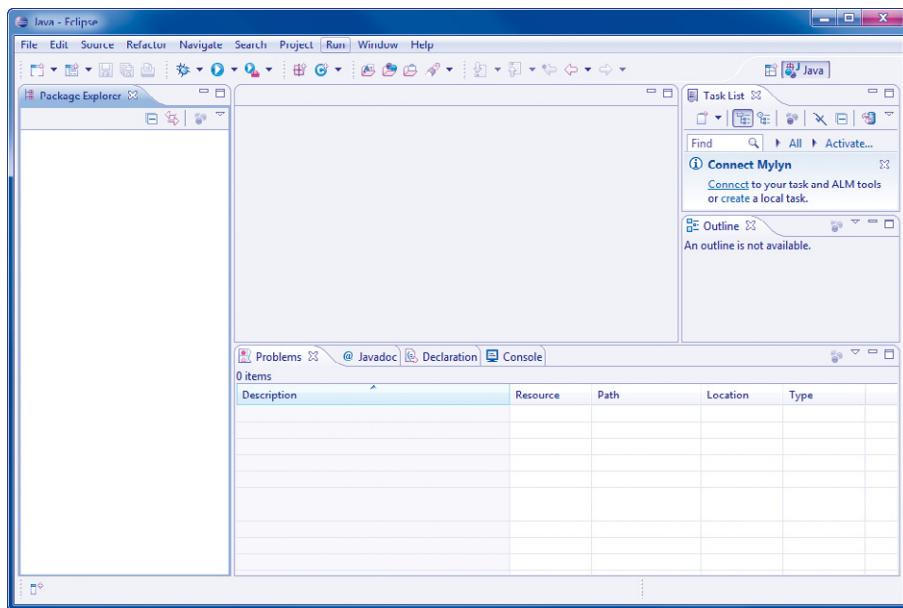
1. เปิดโปรแกรม Eclipse ขึ้นมา จะปรากฏ dialogue ของล็อกบันทึกผู้ใช้ สอบถามความต้องการติดตั้ง



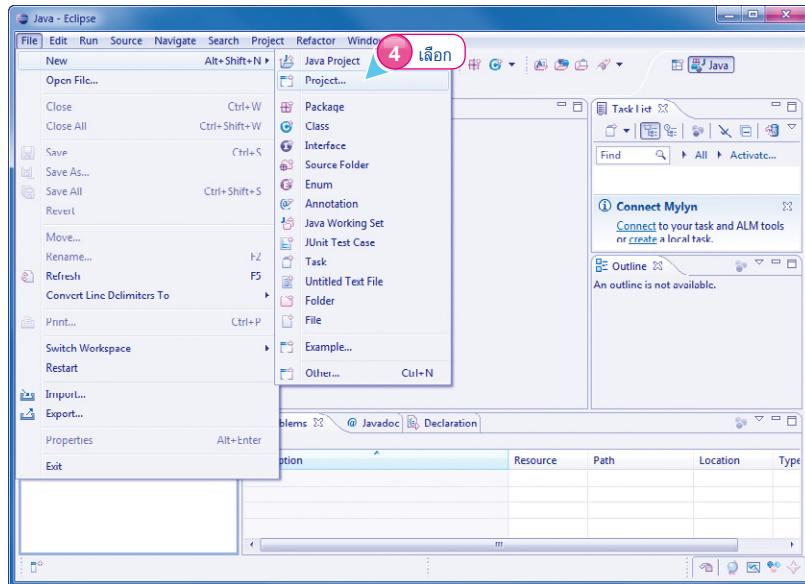
2. จะปรากฏหน้าต่างโปรแกรม Eclipse ขึ้นมา ให้คลิกเลือก Workbench



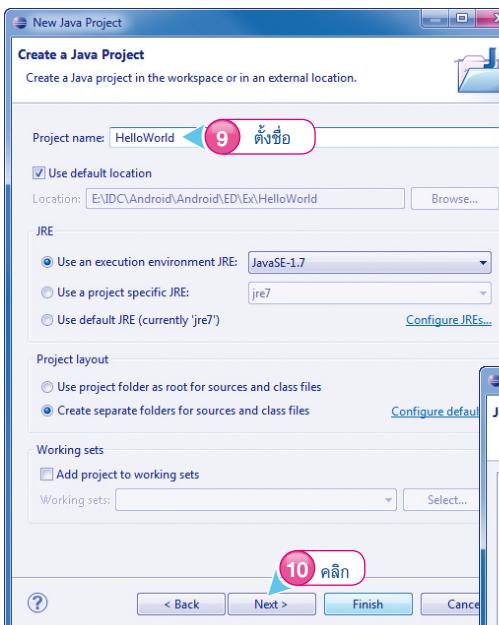
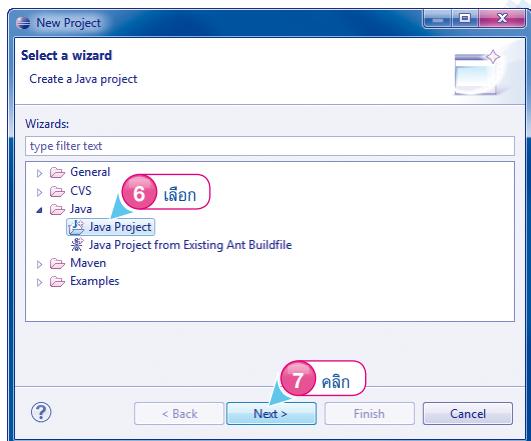
### 3. Eclipse จะแสดงหน้าต่างหลักของโปรแกรมดังรูป



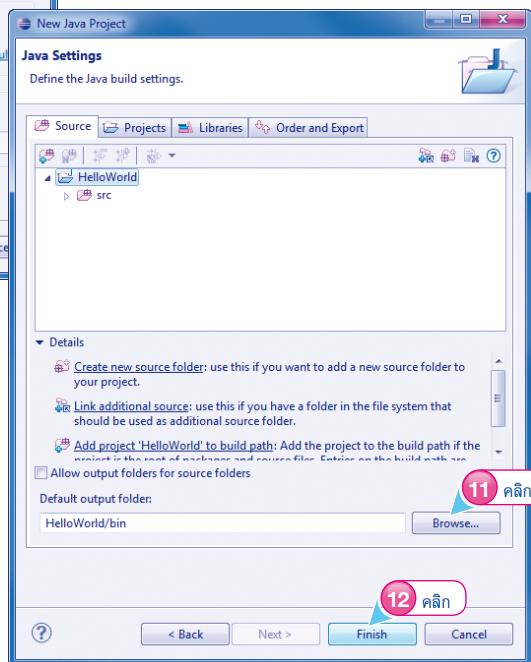
### 4. คลิกเมนู File > New > Project...



5. จะปรากฏ dialogue ตั้งชื่อ New Project ขึ้นมา
6. คลิกเลือก Java/Java Project
7. คลิกปุ่ม **Next >**

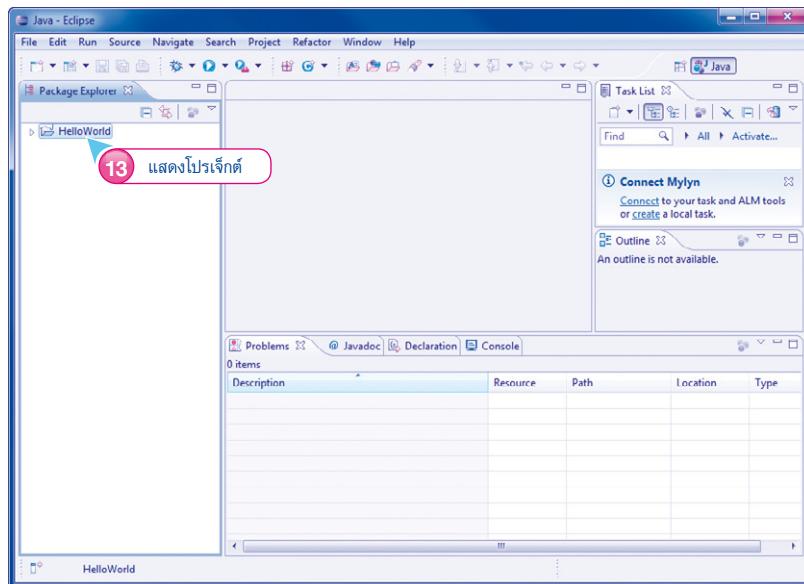


8. จะปรากฏ dialogue ตั้งชื่อ New Java Project ขึ้นมา
9. ตั้งชื่อโปรเจกต์
10. คลิกปุ่ม **Next >**
11. คลิกเลือกตำแหน่งบันทึกไฟล์โปรแกรม
12. คลิกปุ่ม **Finish**

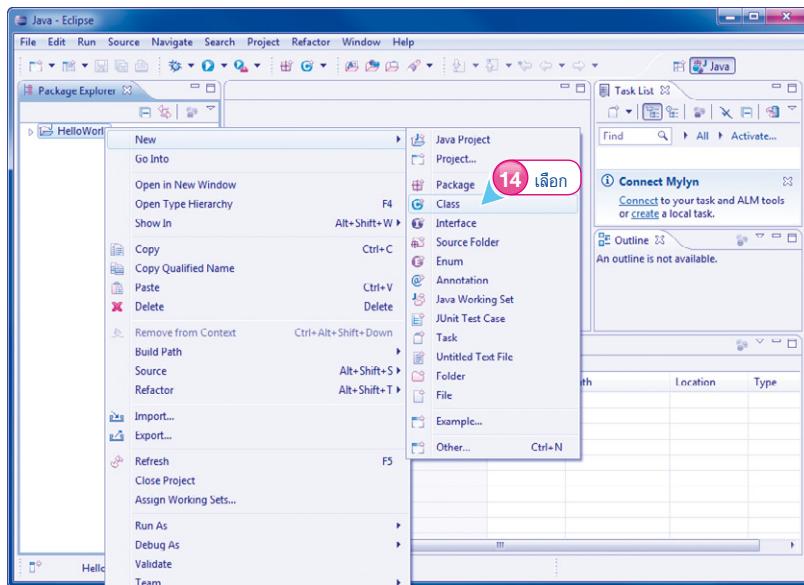




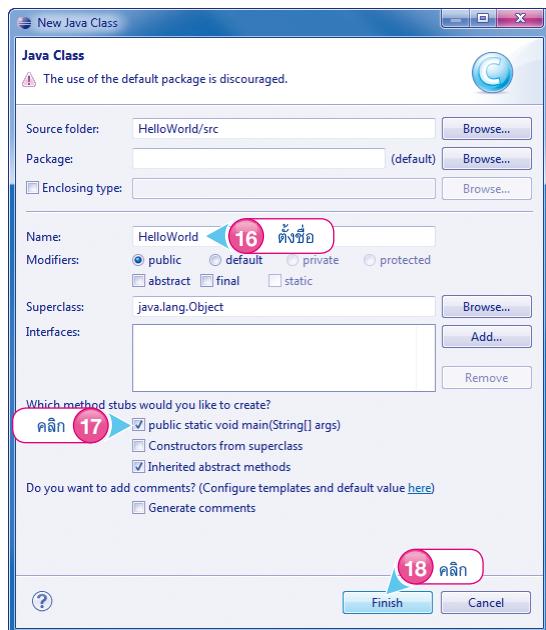
### 13. จะปรากฏโปรเจกต์ใน Package Explorer



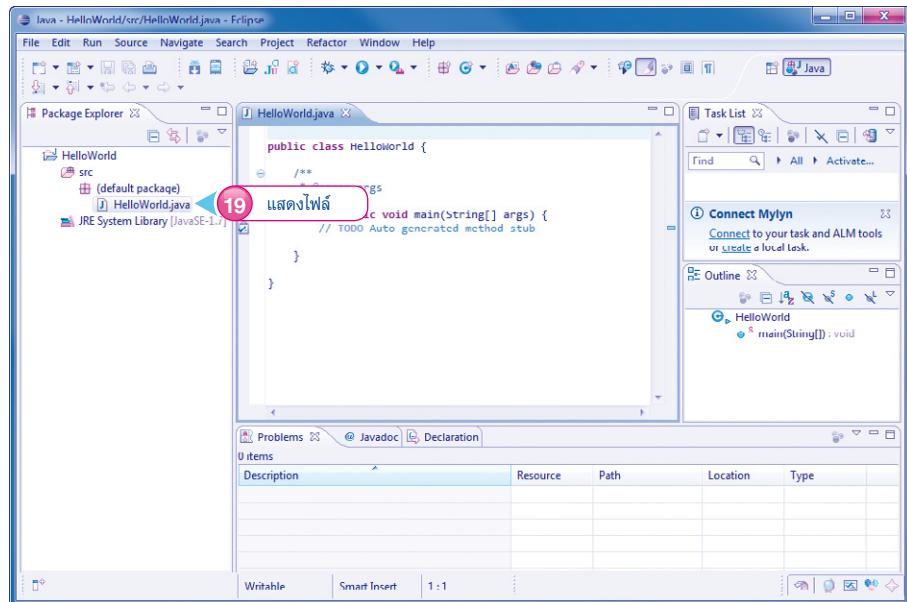
### 14. คลิกขวาที่ชื่อโปรเจกต์ใน Package Explorer แล้วเลือกคำสั่ง New > Class



15. จะปรากฏได้อัตโนมัติ New Java Class ขึ้นมา
16. ตั้งชื่อคลาส
17. คลิกเครื่องหมายถูกหน้า public static void main(String[] args)
18. คลิกปุ่ม **Finish**



19. จะปรากฏไฟล์สกุล java ในโปรเจกต์ พร้อมโค้ดเริ่มต้นดังรูป



## 20. เขียนโค้ดเพิ่มเติมดังนี้

```

public class HelloWorld {
    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello World");
    }
}

```

## 21. คลิกปุ่ม เพื่อทดสอบการทำงาน

## 22. ได้ผลลัพธ์ดังนี้

Java - HelloWorld/src/HelloWorld.java - Eclipse

File Edit Run Source Refactor Navigate Project Window Help

Package Explorer

HelloWorld  
src  
(default package)  
HelloWorld.java

Java System Library [JavaSE-1.7]

HelloWorld.java

```

1 public class HelloWorld {
2     /**
3      * @param args
4      */
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Hello World");
8     }
9 }
10
11
12
13

```

Problems Declaration Console

<terminated> HelloWorld [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Jan 7, 2012 10:53:07 PM)

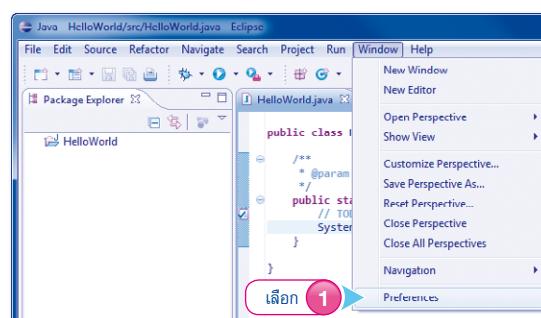
Hello World

เนื่องจากผู้อ่านสามารถเขียนโปรแกรมภาษา Java ด้วย Eclipse ได้ ดังนั้น ถือได้ว่าการติดตั้ง JDK และ Eclipse เสร็จสมบูรณ์

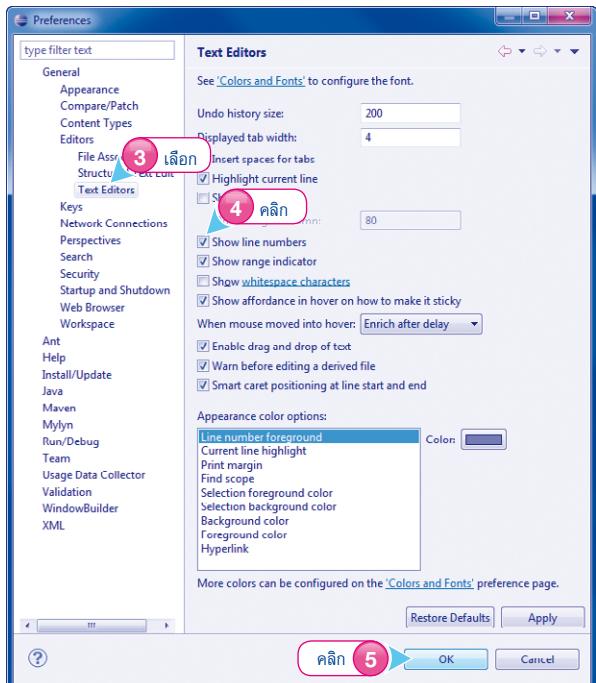
## กำหนดให้ Text Editors แสดงหน้ายเลบบรรกัด

ในการพัฒนาโปรแกรมที่มีขนาดใหญ่และมีความซับซ้อนสูง หาก Text Editors มีการแสดงหน้ายเลบบรรกัดของโค้ด ก็จะช่วยให้ผู้พัฒนาโปรแกรมสามารถปรับปรุงหรือแก้ไขโปรแกรมได้ง่ายขึ้น โดยมีขั้นตอนกำหนดให้ Text Editors แสดงหน้ายเลบบรรกัดดังนี้

### 1. คลิกเมนู Window > Preferences



2. จะปรากฏ dialogue ต่อไปนี้
- Preferences ขึ้นมา
- คลิกเมนู General > Editors > Text Editors
- คลิกเครื่องหมายถูกหน้า Show line numbers
- คลิกปุ่ม OK



6. Text Editors แสดงหมายเลขบรรทัด

The screenshot shows the Eclipse Java editor window with the file 'HelloWorld.java' open. On the left side of the editor, line numbers from 1 to 13 are displayed next to each line of code. The code itself is a simple 'Hello World' application.

```

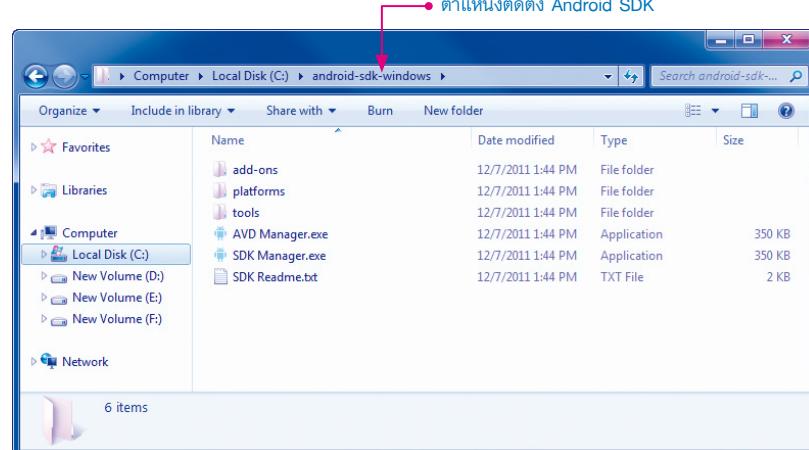
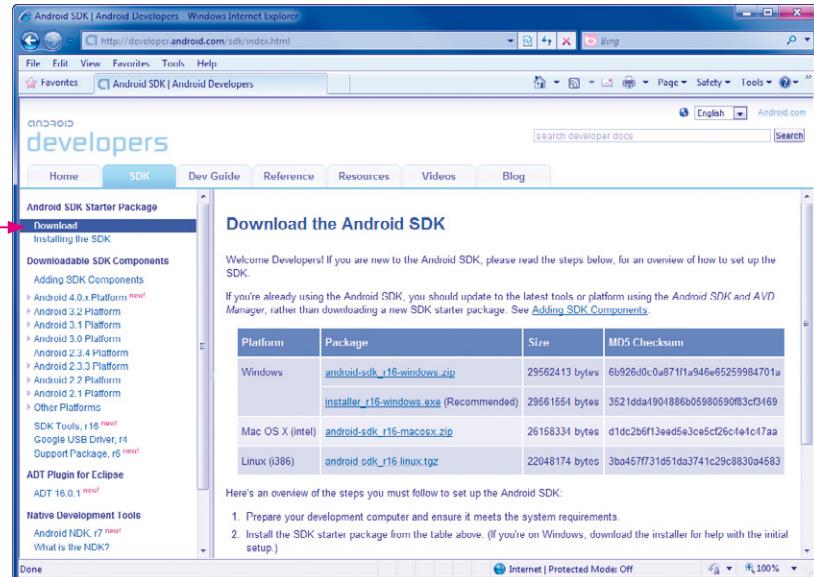
1 public class HelloWorld {
2
3     /**
4      * @param args
5      */
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         System.out.println("Hello World");
9     }
10
11
12 }
13

```

## การติดตั้ง Android SDK (Android Software Development Kit)

เมื่อผู้อ่านติดตั้ง JDK และ Eclipse เรียบร้อยแล้ว ผู้อ่านก็สามารถพัฒนาแอพพลิเคชันด้วยภาษา Java ได้แล้ว ต่อไปเป็นการติดตั้งการเชื่อมต่อกับ Android SDK (Android Software Development Kit) ซึ่งการติดตั้งนั้นไม่มีความซับซ้อนมากนัก เพราะผู้เขียนจะใช้ Plug-in ของ Eclipse เข้ามาร่วมด้วย โดยใช้ ADT (Android Development Tool) ซึ่งจะอธิบายถึงรายละเอียดในหัวข้อดังไป

ผู้อ่านสามารถดาวน์โหลด Android SDK ได้จากเว็บไซต์ <http://developer.android.com/sdk/index.html> และให้ผู้อ่านคลายไฟล์ (Unzip) android-sdk\_r16-windows.zip นี้เพื่อติดตั้ง Android SDK (เมื่ອันกับการติดตั้ง Eclipse) ในที่นี้ผู้เขียนจะติดตั้ง Android SDK ไว้ที่ไดเร็คทอรี C:\android-sdk-windows



### NOTE



ในที่นี้ยกตัวอย่างการใช้งาน Android SDK เวอร์ชัน 16 ซึ่งผู้อ่านสามารถใช้เวอร์ชันใดๆ ก็ได้

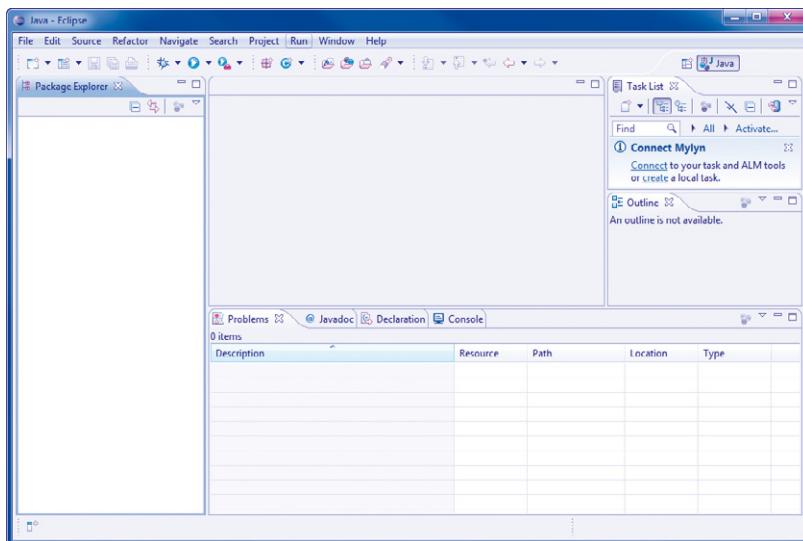
# การติดตั้ง ADT (Android Development Tool)

หลังจากที่ผู้อ่านได้ติดตั้ง Android SDK เสร็จลิ้นแล้ว ส่วนเพิ่มเติมที่สำคัญที่จะช่วยให้ผู้อ่านสามารถพัฒนาแอปพลิเคชันได้ง่ายและสะดวกยิ่งขึ้นก็คือ เครื่องมือ ADT (Android Development Tool) ทั้งนี้ ADT เป็น Plug-in ของ Eclipse ที่จะช่วยอำนวยความสะดวกในการเตรียมเครื่องมือต่างๆ ที่จำเป็นสำหรับการพัฒนาแอปพลิเคชันไว้ให้ โดยที่ผู้อ่านไม่มีความจำเป็นต้องปรับแต่งค่าต่างๆ ด้วยตัวเอง เช่น ในการสร้างโปรเจกต์ใหม่ ADT จะเป็นตัวช่วยสร้างโปรเจกต์ต้นแบบโดยมีโค้ดเริ่มต้นที่จำเป็นไว้ให้ด้วย

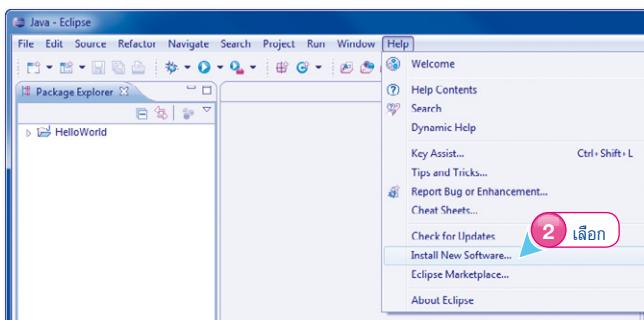
นอกจากนี้ ADT จะเป็นตัวช่วยในการสร้างหน้าจอ แก้ไขโปรแกรม รวมไปถึงการส่งโปรแกรมออกเป็นไฟล์นามสกุล .apk เพื่อใช้สำหรับการติดตั้งแอปพลิเคชันบนเครื่องโทรศัพท์อีกด้วย

ทั้งนี้ไฟล์นามสกุล .apk เป็นเสมือนไฟล์ที่ใช้งานจริงที่จะได้หลังจาก Compile โปรแกรมที่พัฒนาขึ้นทำให้สามารถนำไปรันบนระบบ Android ได้ หรืออีกหนึ่งก็เปลี่ยนเที่ยบได้กับไฟล์นามสกุล .exe ที่ใช้บน Windows นั่นเอง ซึ่งการติดตั้ง ADT มีขั้นตอนดังต่อไปนี้

## 1. เปิดโปรแกรม Eclipse



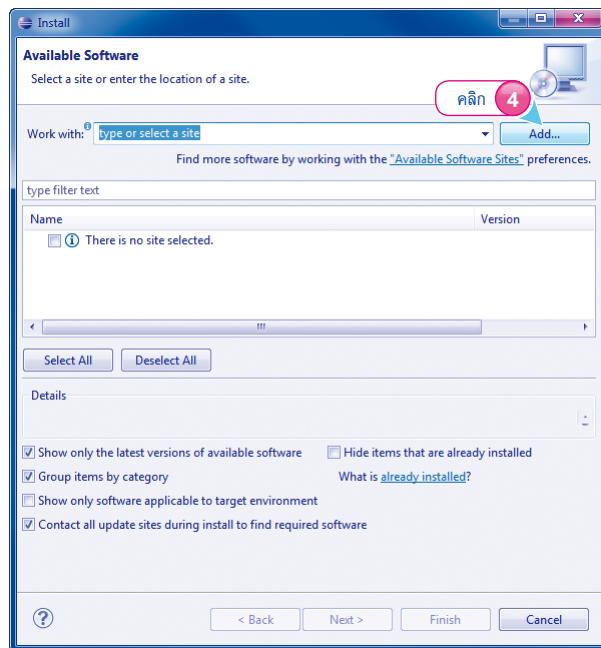
## 2. คลิกเมนู Help > Install New Software...





3. จะปรากฏ dialogue ต่อ คลิกปุ่ม **Add...**

4. คลิกปุ่ม **Add...**



5. จะปรากฏ dialogue ต่อ คลิกปุ่ม **Add Repository** ขึ้นมา

6. กรอกชื่อ Repository ที่ต้องการ ในที่นี้กรอกชื่อ **Android**

7. ในส่วนของ Location: ให้ผู้อ่านกรอก <https://dl-ssl.google.com/android/eclipse/>

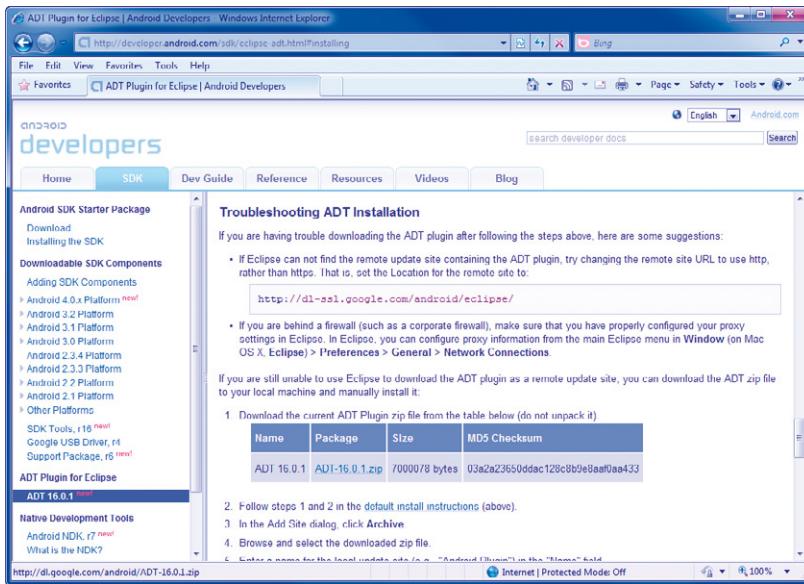
8. คลิกปุ่ม **OK**



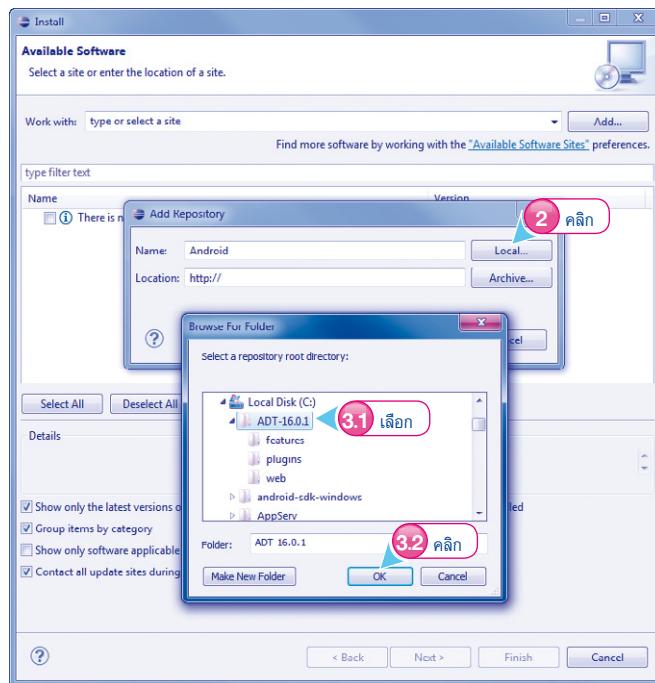
### NOTE



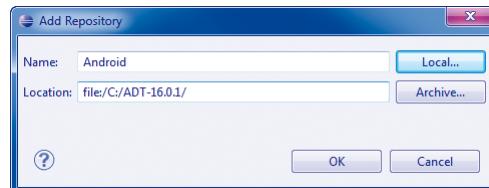
ในกรณีที่ผู้อ่านไม่สามารถติดตั้ง ADT ตามขั้นตอนดังกล่าวได้ อาจจะมีสาเหตุหลายประการ ซึ่งส่วนใหญ่เกิดจากปัญหาของการเชื่อมต่ออินเทอร์เน็ต ดังนั้น ผู้อ่านสามารถดำเนินการติดตั้งได้อีกวิธีหนึ่งคือ ให้ดาวน์โหลดไฟล์ติดตั้งจากเว็บไซต์ <http://developer.android.com/sdk/eclipse-adt.html#installing> ซึ่งมีขั้นตอนการติดตั้งดังนี้ (ADT-16.0.0.zip)



1. ให้คลายไฟล์ (Unzip) ที่ดาวน์โหลดมาได้
2. คลิกปุ่ม **Local...**
3. จะปรากฏหน้าจอ **Browse For Folder** ให้ผู้อ่านเลือกไปยังไดร์เรกทอรีที่คลายไฟล์ไว้ ในที่นี้คือ C:\

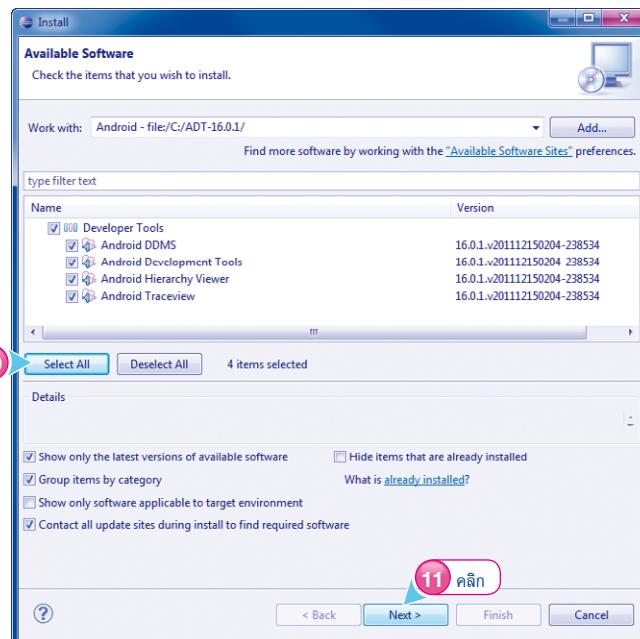


#### 4. ได้ผลลัพธ์ดังนี้

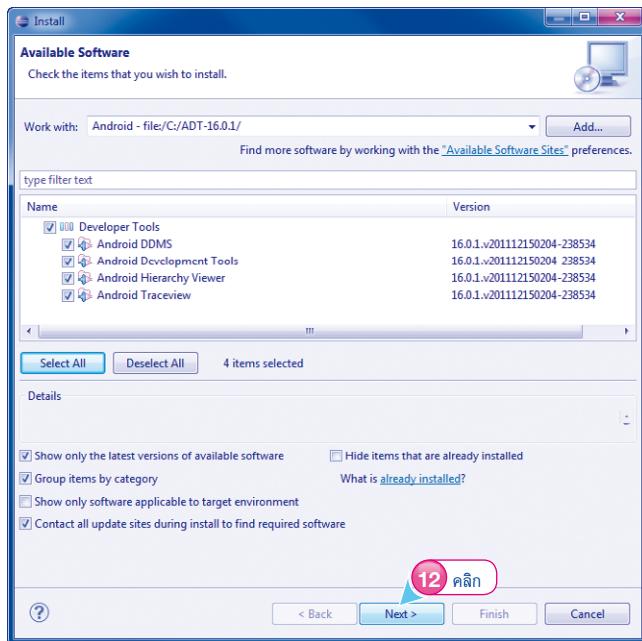


ผู้อ่านจะสังเกตได้ว่า ไดอะล็อกนี้ออกชื่อแตกต่างที่ล้วนของ Location: เท่านั้น ซึ่งหมายถึง แหล่งที่อยู่ของตัวติดตั้ง ADT เท่านั้น

9. รอสักครู่ ระบบจะแสดงรายการให้เลือกดังรูป
10. คลิกปุ่ม **Select All** เพื่อเลือกตัวเลือกทั้งหมด
11. คลิกปุ่ม **Next >**

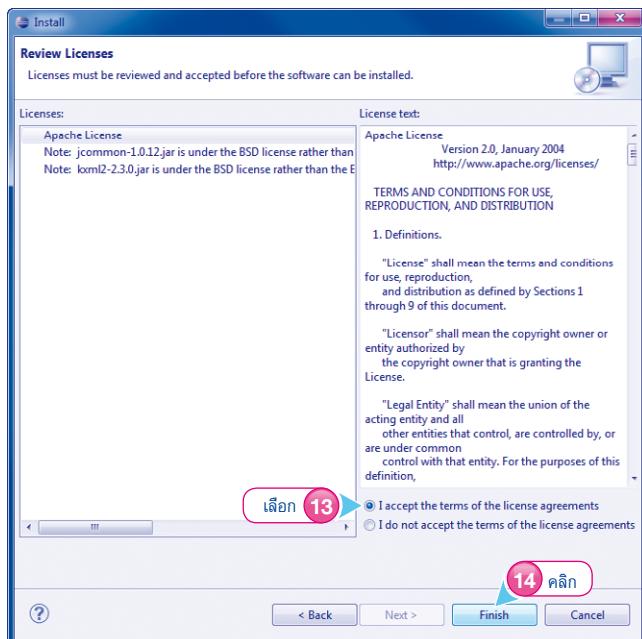


12. รอลักษณะ ระบบจะแสดงรายละเอียดของโปรแกรมต่างๆ ที่จะติดตั้ง ให้คลิกปุ่ม Next >

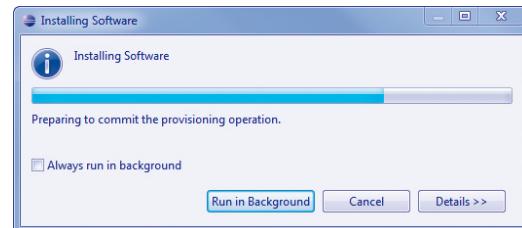


13. ให้คlikเลือก I accept the terms of the license agreements

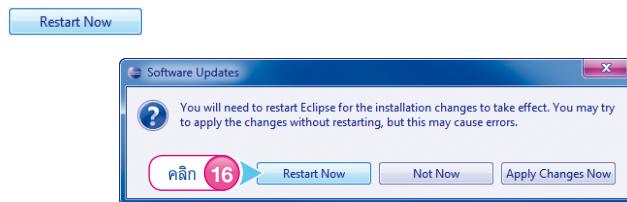
14. คลิกปุ่ม Finish



## 15. รอลักษณะระบบจะติดตั้งโปรแกรม



## 16. เมื่อติดตั้งเสร็จสมบูรณ์ ระบบจะแนะนำให้รีสตาร์ทโปรแกรม Eclipse ใหม่ ให้คลิกปุ่ม



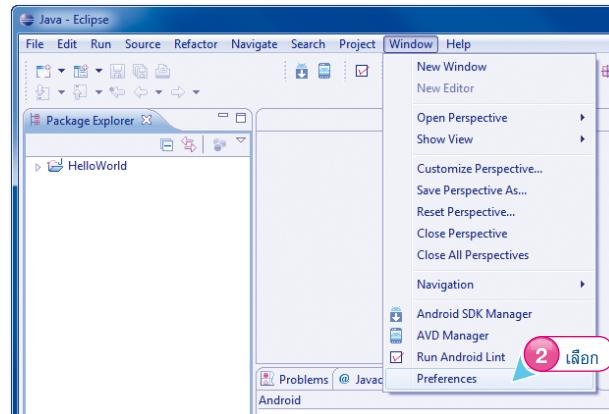
### TIPS

Android DDMS (Android Dalvik Debug Monitor Server) เป็นเครื่องมือสำหรับการตรวจสอบและแก้จุดบกพร่องของโปรแกรมที่สร้างขึ้น (Debug) โดยที่ Android DDMS สามารถอ่านรายละเอียดและสถานะต่างๆ ของโทรศัพท์จำลองได้ ซึ่งถือเป็นเครื่องมือที่อำนวยความสะดวกให้นักพัฒนาได้มาก เนื่องจากเมื่อติดตั้งเสร็จแล้ว สามารถใช้ DDMS ในการตรวจสอบการทำงานของแอปพลิเคชันได้

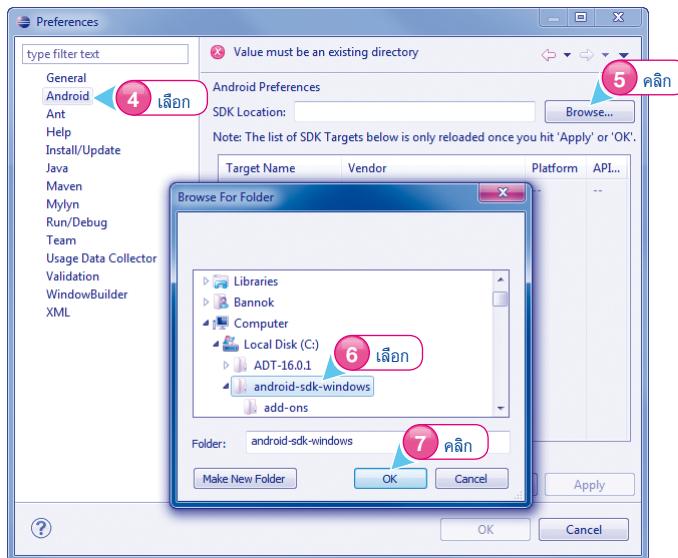
## การปรับแต่งค่า ADT และ Android SDK

หลังจากที่ผู้อ่านติดตั้ง ADT เสร็จลิ้นแล้ว ขั้นตอนต่อไปคือ การปรับแต่งค่า ADT และ Android SDK เพื่อให้สามารถทำงานร่วมกันได้ ซึ่งมีขั้นตอนดังต่อไปนี้

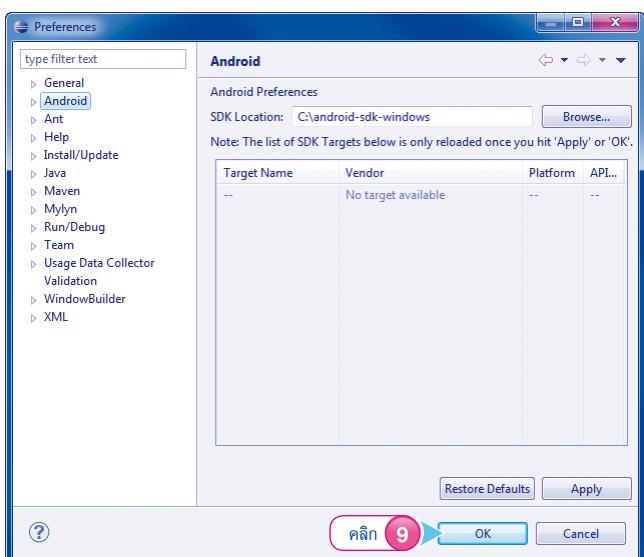
1. เปิดโปรแกรม Eclipse
2. คลิกเมนู Window > Preferences



3. จะปรากฏ dialogue ออกมือก็อปซ์ Preferences ขึ้นมา
4. คลิกเลือก Android
5. จะลังเกตได้ว่า ทางด้านขวา มีจะปรากฏช่องความแสดงค่าพิเศษ Value must be an existing directory ในกรณีนี้ให้คลิกปุ่ม **Browse...**
6. จะปรากฏ dialogue ออกมือก็อปซ์ Browse For Folder ในขั้นตอนนี้ให้ผู้อ่านเลือกไดเร็คทอรี่ที่ได้ติดตั้ง Android SDK ไว้ ซึ่งในที่นี้คือ C:\android-sdk-windows
7. คลิกปุ่ม **OK**

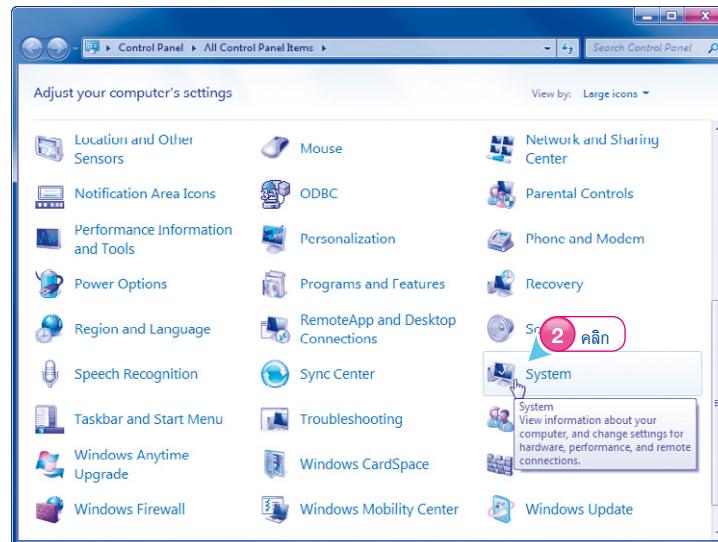


8. ลังเกตได้ว่าข้อผิดพลาดที่แสดงขึ้นมาก็จะหายไป
9. คลิกปุ่ม **OK**

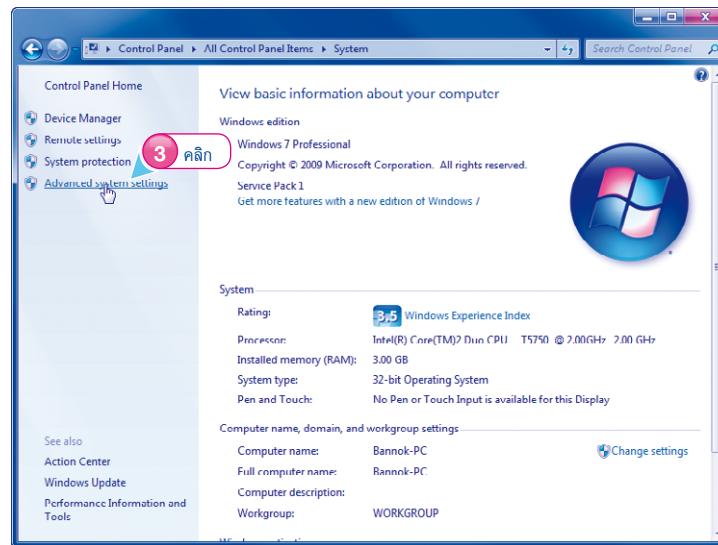


ทั้งนี้ผู้อ่านสามารถเพิ่มพาธของ android-sdk-windows ในล่วงของเครื่องมือต่างๆ ให้สามารถเข้าถึงโดยง่าย ซึ่งผู้อ่านสามารถทำได้ดังนี้

1. คลิกปุ่ม  > Control Panel
2. คลิกไอคอน System



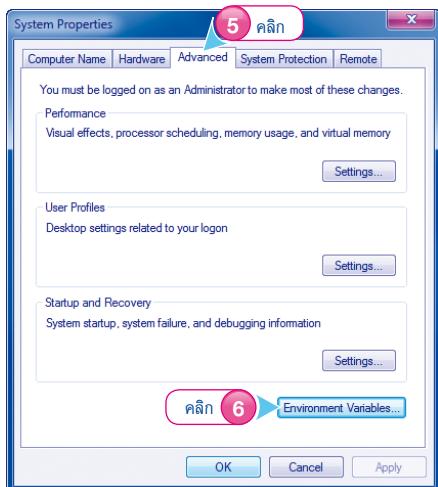
3. คลิกที่ Advanced system settings



4. จะปรากฏ dialogue ออกบอช์ System Properties ขึ้นมา

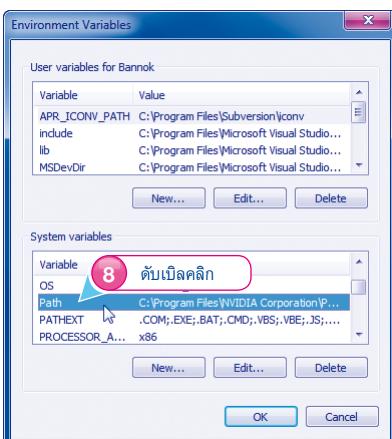
5. คลิกแท็บ Advanced

6. คลิกปุ่ม Environment Variables...



7. จะปรากฏ dialogue ออกบอช์ Environment Variables ขึ้นมา

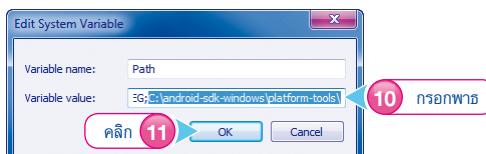
8. ในส่วนของ System variables ให้ดับเบิลคลิกตัวแปร Path



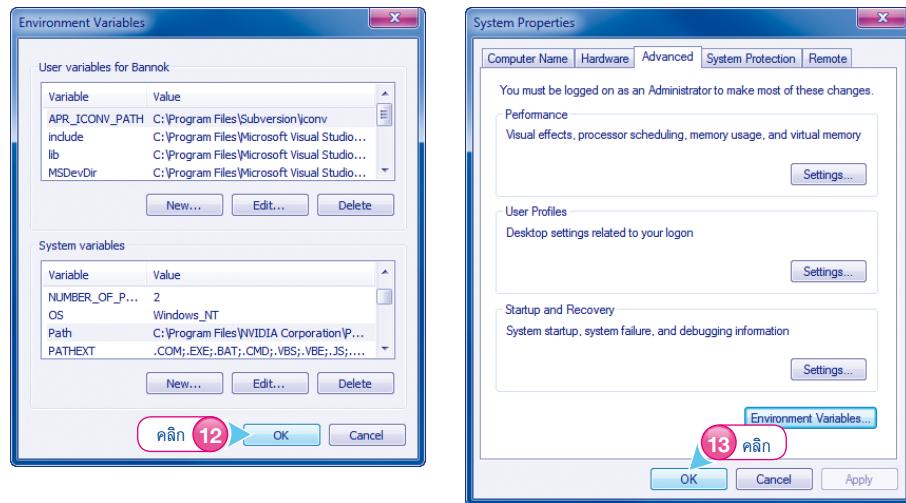
9. จะปรากฏ dialogue ออกบอช์ Edit System Variable ขึ้นมา

10. กรอกพาธของ tools เพิ่มเข้าไปดังนี้ “;C:\android-sdk-windows\platform-tools\;”

11. คลิกปุ่ม OK



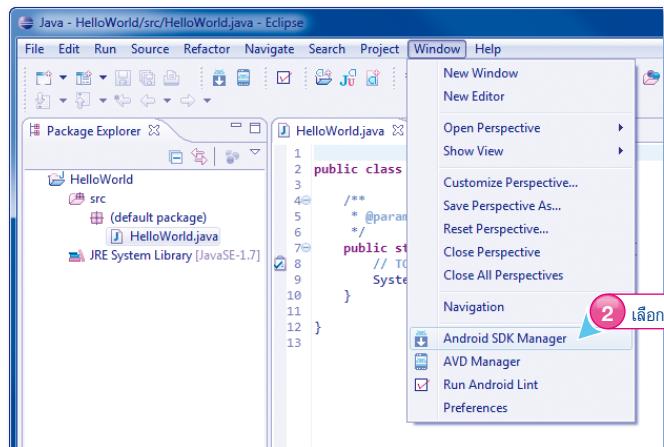
12. คลิกปุ่ม **OK**
13. คลิกปุ่ม **OK**



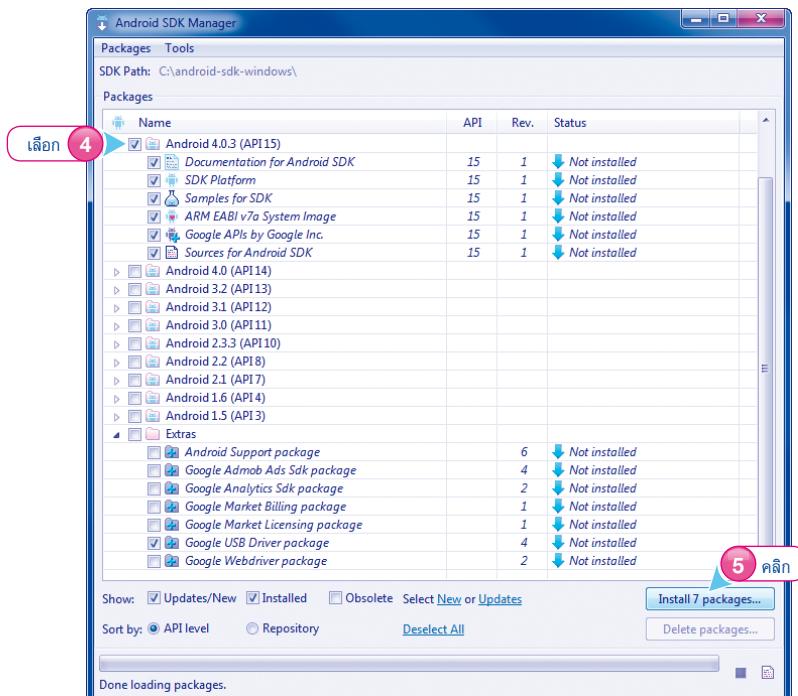
## การติดตั้ง SDK Platform

เมื่อติดตั้งและปรับแต่งค่าต่างๆ ระหว่าง ADT กับ Android SDK เสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการติดตั้งแพ็คเกจต่างๆ เพื่อใช้สำหรับพัฒนาแอพพลิเคชัน ทั้งนี้ผู้อ่านสามารถเลือกได้ตามต้องการ แต่ในที่นี้ผู้เขียนจะพัฒนาแอพพลิเคชันบนพื้นฐานของ Android 4.0.3 ซึ่งมีขั้นตอนดังนี้

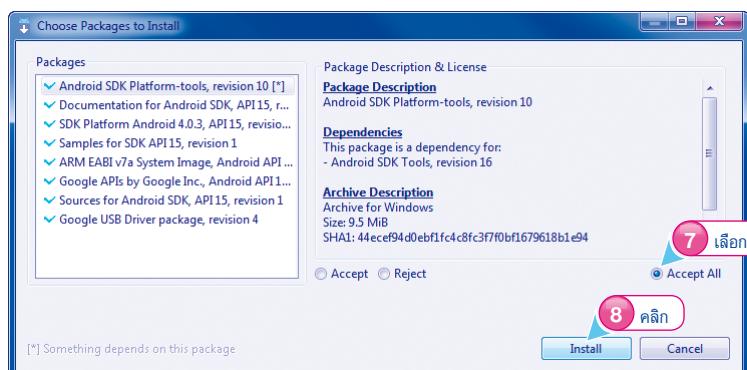
1. เปิดโปรแกรม Eclipse
2. คลิกเมนู Window > Android SDK Manager



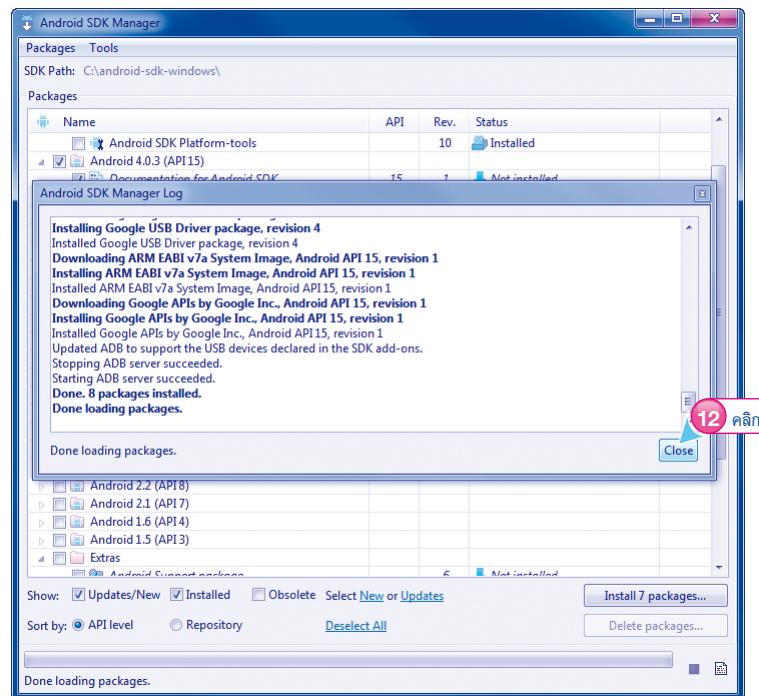
3. จะปรากฏ dialogue ออกบอชช์ Android SDK Manager ขึ้นมา
4. เลือกรายการแพ็คเกจ Android รุ่นต่างๆ ซึ่งผู้อ่านสามารถเลือกได้ตามต้องการ (ในที่นี่เลือกเฉพาะ Android 4.0.3)
5. หลังจากผู้อ่านเลือกแพ็คเกจต่างๆ เรียบร้อยแล้ว ให้คลิกปุ่ม **Install 7 packages...**



6. จะปรากฏ dialogue ออกบอชช์ Choose Packages to Install เพื่อให้ผู้อ่านยอมรับแพ็คเกจที่จะดำเนินการติดตั้ง
7. ให้คลิกเลือก Accept All
8. คลิกปุ่ม **Install**



9. รอลักษณะ เพื่อให้ระบบติดตั้งแพ็คเกจต่างๆ ให้เสร็จเรียบร้อย ซึ่งในขั้นตอนนี้อาจจะใช้เวลานาน  
พอสมควร โดยจะขึ้นอยู่กับความเร็วของการเชื่อมต่ออินเทอร์เน็ต ดังนั้น คำแนะนำคือ ให้เลือก  
เฉพาะแพ็คเกจหรือรุ่นที่จะใช้งานสำหรับการพัฒนาจริงเท่านั้น
10. เมื่อ Install เสร็จเรียบร้อยแล้ว จะปรากฏ dialogue ของ ADB Restart ขึ้นมา
11. คลิกปุ่ม **Yes** เพื่อรีสตาร์ท ADB
12. ให้คลิกปุ่ม **Close** เพื่อปิด dialogue นี้



## การสร้างโทรศัพท์จำลอง (Emulator)

เมื่อผู้อ่านได้ติดตั้งเครื่องมือต่างๆ เสร็จลืนแล้ว ในหัวข้อนี้คือ การจำลองอุปกรณ์โทรศัพท์เคลื่อนที่บนระบบ Android ขึ้นมา ซึ่งจะเรียกว่า **AVD (Android Virtual Device)** จุดประสงค์เพื่อนำมาใช้ทดสอบ  
การทำงานของแอปพลิเคชันที่พัฒนาขึ้นมาบ้าง

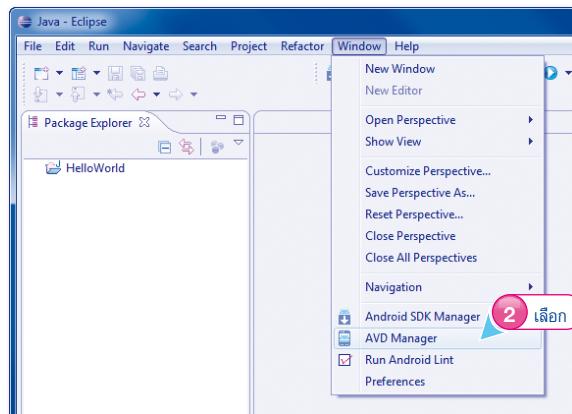
โดยปกติแล้วแอปพลิเคชันที่พัฒนาขึ้นมาและรันบนเครื่องโทรศัพท์จำลองนี้ เมื่อนำไปติดตั้งบน  
โทรศัพท์จริงก็สามารถนำไปใช้งานได้จริง เช่นกัน ซึ่งจะกล่าวถึงอีกครั้งในบทสุดท้าย

ในการทดสอบแอพพลิเคชันที่ผู้อ่านพัฒนาขึ้นมาแล้ว จะต้องมีการสร้าง AVD ขึ้นมา ก่อน แล้วรันให้ AVD ทำงาน เป็นโทรศัพท์จำลอง ที่มีสภาพแวดล้อมเหมือนโทรศัพท์จริง ซึ่งจะเรียกว่า **Emulator** เพื่อใช้ทดสอบการทำงานของแอพพลิเคชัน ทั้งนี้หลักการทำงานของ Emulator ก็จะเหมือนกับการจำลองโทรศัพท์ที่ใช้ระบบปฏิบัติการ Android จริงๆ ขึ้นมาบนเครื่องคอมพิวเตอร์นั่นเอง โดยที่คุณสมบัติต่างๆ ของ Emulator จะมีความคล้ายคลึงกับโทรศัพท์ที่ใช้จริง อย่างไรก็ตาม ก็อาจจะมีข้อแตกต่างกันบ้างตามบริษัทผู้จัดจำหน่ายโทรศัพท์

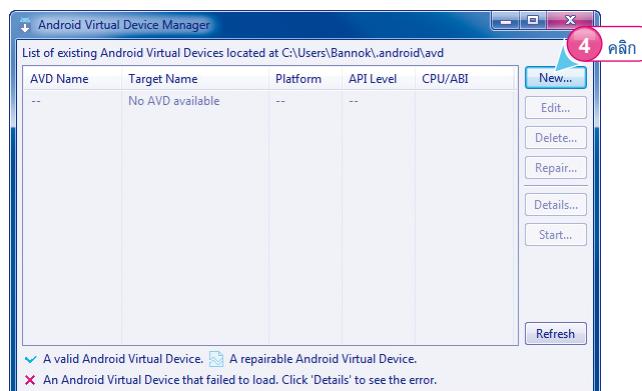
ข้อดีของการใช้งาน Emulator ก็คือ ในกรณีที่ผู้อ่านไม่สามารถซื้อโทรศัพท์จริงได้ ก็สามารถใช้งานตาม Feature ที่ระบบ Android มีให้ได้เช่นกัน เช่น Google Map เป็นต้น

ทั้งนี้ข้อดีอีกข้อหนึ่งที่สำคัญสำหรับนักพัฒนา ก็คือ ความเร็ว เนื่องจากในการพัฒนาแอพพลิเคชันนั้นอาจจะมีข้อผิดพลาดต่างๆ มากมาย ดังนั้น จึงเป็นการดีกว่าที่จะพัฒนานบน Emulator ก่อน เพื่อทดสอบว่า แอพพลิเคชันสามารถทำงานได้อย่างถูกต้องตามที่ต้องการหรือไม่ จากนั้นจึงนำแอพพลิเคชันที่มีการตรวจสอบข้อผิดพลาดแล้วไปติดตั้งบนโทรศัพท์จริงอีกรอบหนึ่ง ซึ่งขั้นตอนการสร้าง AVD มีดังต่อไปนี้

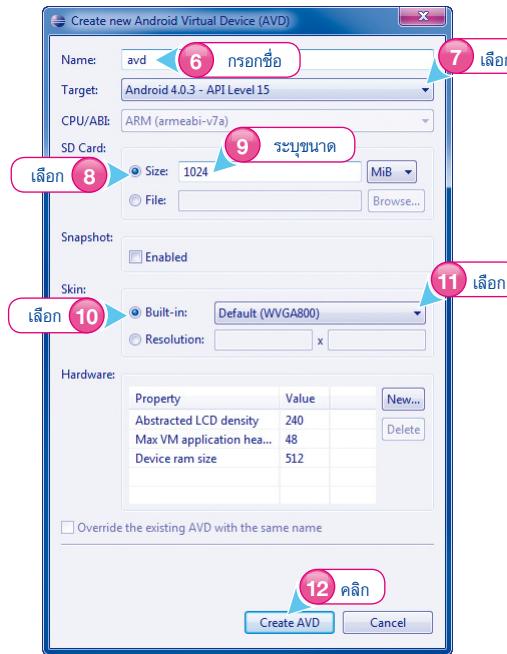
1. เปิดโปรแกรม Eclipse
2. คลิกเมนู Window > AVD Manager



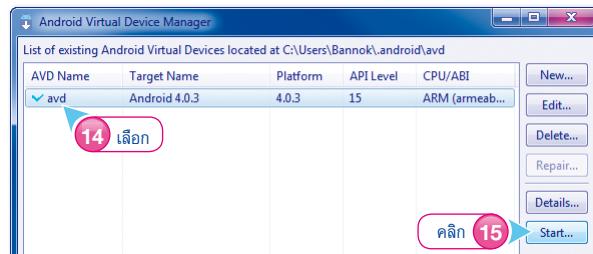
3. จะปรากฏ dialogue ออกบอช์ Android Virtual Device Manager ขึ้นมา
4. คลิกปุ่ม **New...** เพื่อสร้าง AVD



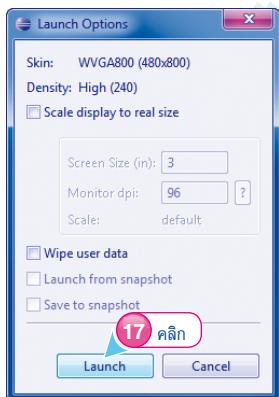
5. จะปรากฏ dialogue ตั้งชื่อ设备 Create new Android Virtual Device (AVD) เพื่อให้กำหนดคุณสมบัติ ต่างๆ ของ AVD
6. ตั้งชื่อ AVD ในที่นี้ตั้งชื่อเป็น avd
7. ในส่วนของ Target: คลิกเลือก Android 4.0.3 ทั้งนี้ผู้อ่านสามารถเลือกรุ่นอื่นๆ ได้
8. ในส่วนของ SD Card: คลิกเลือก Size
9. กรอกความจุของหน่วยความจำ ในที่นี้กรอกขนาด 1024 MiB (เมกะไบต์)
10. ในส่วนของ Skin: คลิกเลือก Built-in:
11. คลิกเลือก WVGA800 ย่อมาจาก Wide Video Graphics Array ซึ่งเป็นการเลือกขนาดหน้าจอแสดงผลของแอพพลิเคชัน ซึ่งผู้อ่านควรเลือกตามขนาดของหน้าจอโทรศัพท์ที่ใช้งาน
12. คลิกปุ่ม **Create AVD** เพื่อสร้าง AVD



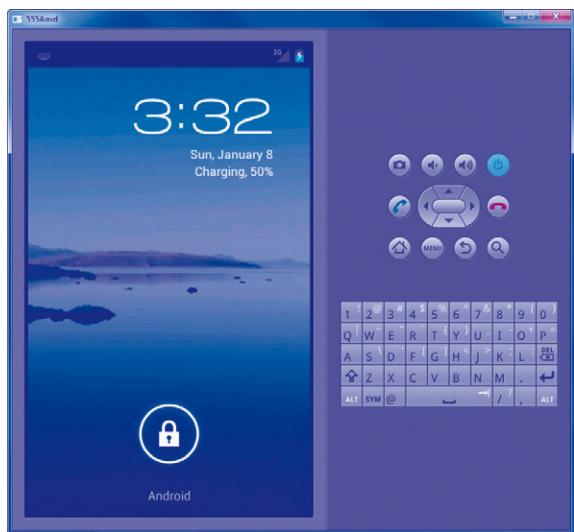
13. จากนั้นจะมีชื่อ AVD ที่เราสร้างไว้แสดงขึ้นมาในรายการ
14. คลิกเลือก AVD ที่สร้างไว้
15. คลิกปุ่ม **Start...** เพื่อเปิดเครื่อง AVD



16. จะปรากฏໄດ້ລົງທຶນອັນຫຼາຍໆ Launch Options ຂຶ້ນມາ  
 17. ຄລືກປຸ່ມ **Launch** ເພື່ອອຳນຸມາດໃຫ້ AVD ສາມາດກຳທຳການໄດ້



18. ຈະປາກຢູ່ທີ່ຕ້ອງກ່າວ Command Prompt ຂຶ້ນມາ ເພື່ອຕິດຕັ້ງ  
 AVD ໃຫ້ຮອບສັກຄູ່ Emulator ຈະສາມາດໃຊ້ງານໄດ້



**NOTE**

สำหรับการรัน AVD ในครั้งแรกจะใช้เวลาค่อนข้างนาน เช่น 2-3 นาที อย่างไรก็ตามระยะเวลาจะน้อยลงในครั้งต่อๆ ไป ซึ่งในการพัฒนาแอพพลิเคชันทุกครั้งไม่ควรเปิดปิด Emulator บ่อยๆ ซึ่งจะทำให้เสียเวลาในการรันค่อนข้างมาก

## สรุปหัวข้อบท

ในบทนี้เป็นการอธิบายถึงขั้นตอนต่างๆ ในการติดตั้งเครื่องมือสำหรับพัฒนาแอพพลิเคชันบน Android เพื่อเตรียมความพร้อมก่อนพัฒนาแอพพลิเคชันต่อไป เช่น การติดตั้ง JDK (Java Development Kit), การติดตั้งโปรแกรม Eclipse, การติดตั้ง Android SDK, การติดตั้ง ADT และการสร้าง AVD

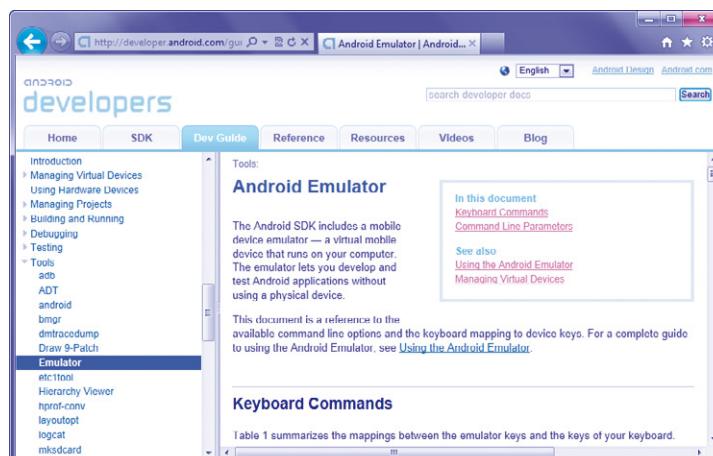
## แบบฝึกหัดหัวข้อบท

- หลังจากที่ผู้อ่านติดตั้งเครื่องมือต่างๆ ที่ใช้ในการพัฒนาแอพพลิเคชันเรียบร้อยแล้ว ให้ผู้อ่านศึกษาเมนูการใช้งานต่างๆ ทั้งหมด เช่น บุ๊ม Menu, Home, Back และ Zoom รวมไปถึงการตั้งค่าต่างๆ ว่ามีการทำงานอย่างไร
- ให้ผู้อ่านทดลองล้าง AVD ที่มีคุณสมบัติต่างกัน 2 เครื่อง

## เริ่มต้นพัฒนาแอพพลิเคชัน

เนื้อหาในบทนี้ มีวัตถุประสงค์เพื่อให้ผู้อ่านเริ่มฝึกพัฒนาแอพพลิเคชันบนระบบ Android ได้ และเป็นการเริ่มต้นเรียนรู้การใช้งานเครื่องมือต่างๆ ไปพร้อมๆ กัน ซึ่งในที่นี้จะเริ่มสอนการพัฒนาแอพพลิเคชันแสดงข้อความ Hello World ซึ่งเป็นแอพพลิเคชันแรกๆ ที่ถูกใช้เป็นตัวอย่างในการพัฒนาเบื้องต้นของการเขียนโปรแกรม นอกจากนี้แอพพลิเคชัน Hello World ยังถูกใช้เป็นตัวตรวจสอบการทำงานของคอมไพล์เตอร์ (Compiler) และส่วนสำคัญต่างๆ ของเครื่องมือที่ได้ติดตั้งในบทที่ผ่านมา อีกทั้งยังเป็นตัวช่วยในการตรวจสอบข้อผิดพลาดระหว่างการติดตั้ง IDE (Integrated Development Environment) อีกด้วย ดังนั้น การฝึกพัฒนาแอพพลิเคชันในบทนี้ จึงเป็นพื้นฐานสำหรับการพัฒนาแอพพลิเคชันประยุกต์ในขั้นสูงต่อไป

ทั้งนี้การฝึกพัฒนาแอพพลิเคชันจะเป็นการเรียนรู้ด้วยตัวอย่าง ซึ่งผู้อ่านสามารถพัฒนาแอพพลิเคชันตามขั้นตอนและสามารถนำไปใช้งานได้จริง โดยไม่มุ่งเน้นถึงการอธิบายในรายละเอียดเชิงลึกของการใช้งานแต่ละฟังก์ชัน เนื่องจากผู้อ่านสามารถหาข้อมูลเพิ่มเติมได้จากเว็บไซต์ต่างๆ เช่น <http://developer.android.com/guide/developing/tools/emulator.html> เป็นต้น



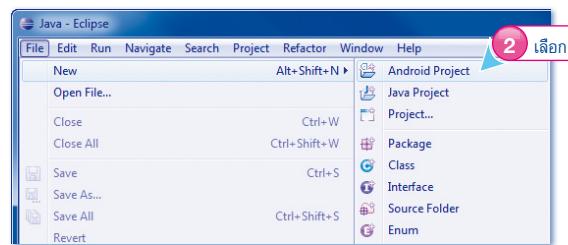
ชี้งเนื้อหาในบทนี้ ผู้เขียนจะแบ่งออกเป็นขั้นตอนดังนี้

1. เริ่มต้นสร้างโปรเจกต์
2. ออกแบบหน้าจอแอพพลิเคชัน
3. เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน
4. ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator)
5. การส่งออกและนำเข้าโปรเจกต์ใน Eclipse

## เริ่มต้นสร้างโปรเจกต์

ขั้นตอนแรกในการพัฒนาแอพพลิเคชันทุกครั้งคือ การสร้างโปรเจกต์ ซึ่งผู้อ่านสามารถทำได้ตามขั้นตอนดังต่อไปนี้

1. เปิดโปรแกรม Eclipse
2. คลิกเมนู File > New > Android Project

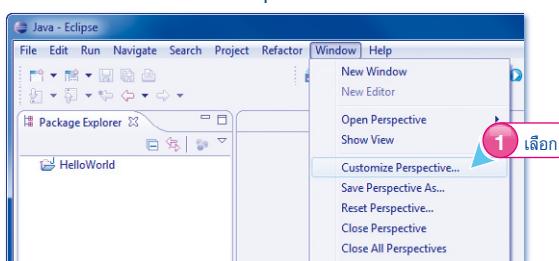


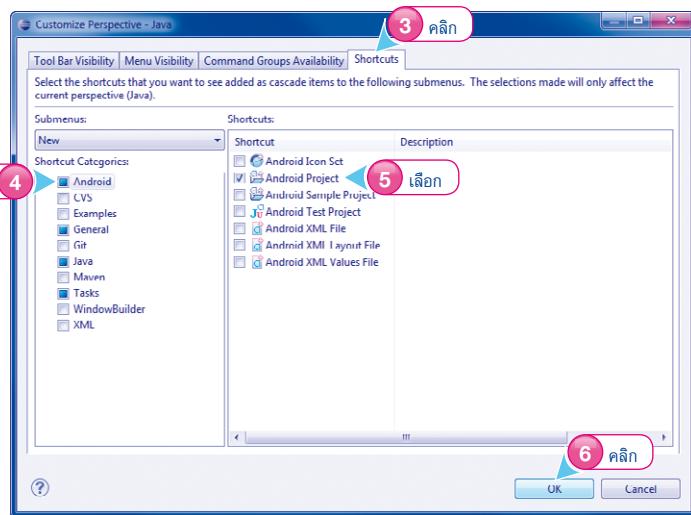
### NOTE



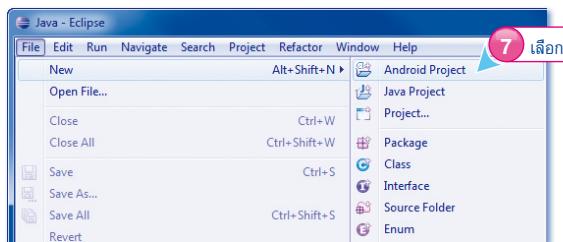
ในกรณีที่ผู้อ่านไม่พบ Android Project หลังจากคลิกเมนู File > New ให้ผู้อ่านทำตามขั้นตอนดังนี้

1. คลิกเมนู Window > Customize Perspective...
2. จะปรากฏ dialogue ล็อกบ็อกซ์ Customize Perspective ขึ้นมา
3. คลิกแท็บ Shortcuts
4. คลิกเลือก Android ที่กล่องเมนู Shortcut Categories: ที่อยู่ทางด้านซ้าย
5. จะปรากฏรายการการ Shortcut ที่เมนูทางด้านขวา ให้คลิกเลือก Android Project
6. คลิกปุ่ม **OK**

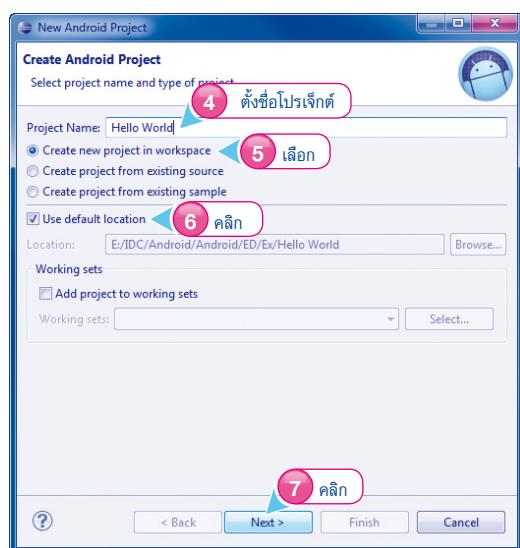




7. เมื่อคลิกเมนู File > New อีกครั้ง จะปรากฏเมนู Android Project ให้เลือกดังรูป

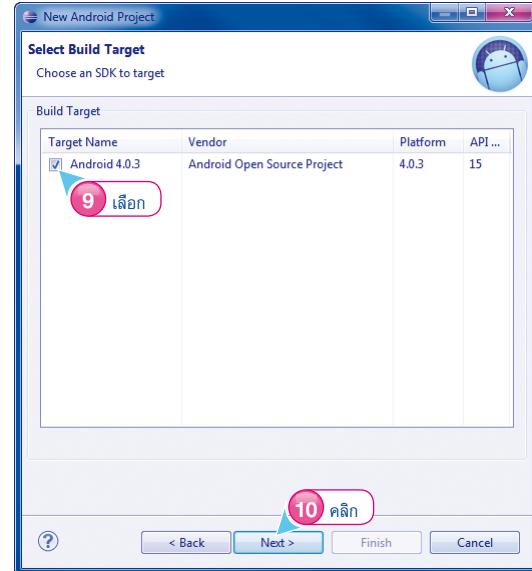


3. จะปรากฏได้จะล็อกบีกซ์ New Android Project ในส่วนแรกสำหรับกำหนดชื่อและรูปแบบของโปรเจกต์
4. ตั้งชื่อโปรเจกต์ ในนี่ตั้งชื่อว่า Hello World
5. ในส่วนของ Contents คลิกเลือก Create new project in workspace เพื่อสร้างโปรเจกต์ใหม่
6. คลิกเครื่องหมายถูกหน้า Use default location เพื่อพัฒนาแอพพลิเคชันในตำแหน่งที่อยู่เดียวกันกับโปรเจกต์ Eclipse
7. คลิกปุ่ม **Next >**

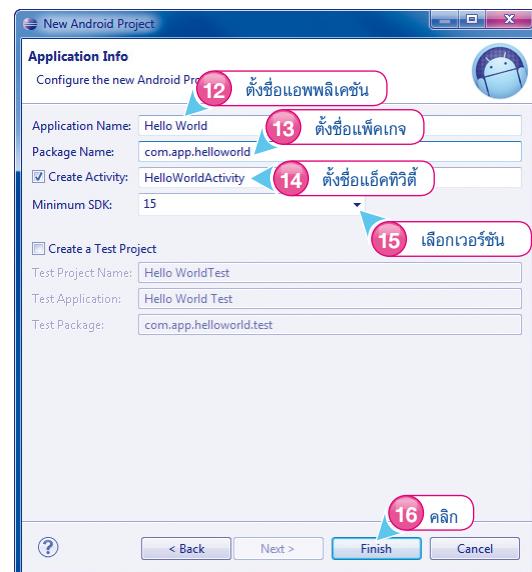




8. จะปรากฏได้คลื่อน้ำหน้า New Android Project ในส่วนที่สอง สำหรับเลือกเวอร์ชันของ Android
9. คลิกเลือกเวอร์ชันของ Android ที่จะพัฒนา ในที่นี้เลือก 4.0.3
10. คลิกปุ่ม **Next >**



11. จะปรากฏได้คลื่อน้ำหน้า New Android Project ในส่วนที่สาม สำหรับกำหนดรายละเอียดของแอปพลิเคชัน
12. ตั้งชื่อของแอปพลิเคชันคือ Hello World
13. ชื่อแพ็คเกจจะอยู่ในรูปแบบของ xxx.xxx... ซึ่งในที่นี้ตั้งชื่อแพ็คเกจเป็น com.app.helloworld
14. ตั้งชื่อ Activity ในที่นี้ตั้งชื่อเป็น HelloWorldActivity
15. ในส่วนของ Minimum SDK: คลิกเลือกเวอร์ชัน 15 ซึ่งเป็นเวอร์ชันเดียวกันกับ API Level ที่เลือกในข้อ 9
16. คลิกปุ่ม **Finish** ซึ่งถือเป็นอันเสร็จสิ้นการสร้างโปรเจกต์



## TIPS

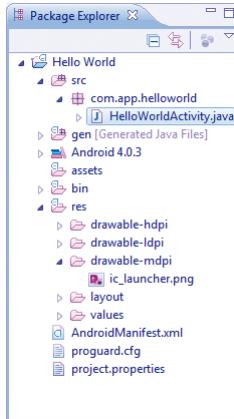


กฎการตั้งชื่อ Activity มีหลักดังนี้

1. ชื่อต้องไม่มีช่องว่าง
2. ชื่อต้องไม่มีเครื่องหมายพิเศษ
3. ชื่อต้องไม่มีเครื่องหมายทางคณิตศาสตร์ต่างๆ
4. ชื่อไม่สามารถเขียนต้นด้วยตัวเลขได้

ตัวอย่างเช่น HelloWorld และ SendSMS เป็นต้น

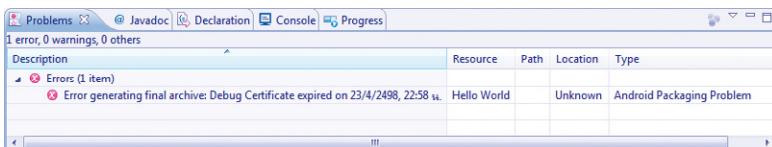
**17.** จะปรากฏโปรเจกต์ที่ Package Explorer ซึ่งแสดงถึงโครงสร้างของโปรเจกต์ที่ได้สร้างขึ้นมาดังรูป



## WARN

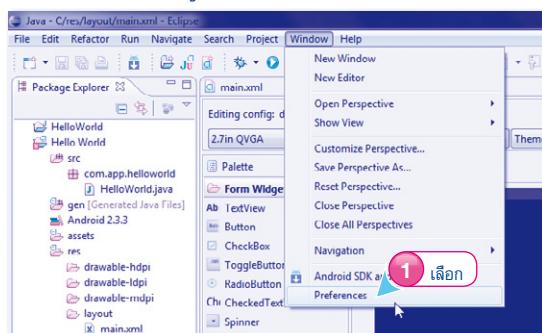


ในกรณีที่สร้างโปรเจกต์แล้วมีสัญลักษณ์ Error ที่ชื่อโปรเจกต์ ซึ่งมีรายละเอียดดังนี้

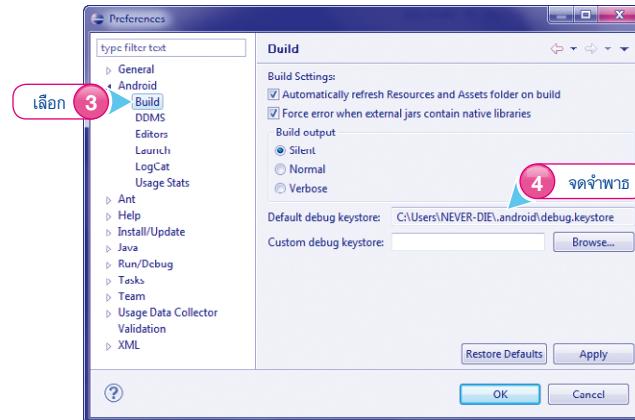


ผู้อ่านสามารถแก้ไขได้ดังนี้

**1.** เปิดโปรแกรม Eclipse คลิกเมนู Window > Preferences



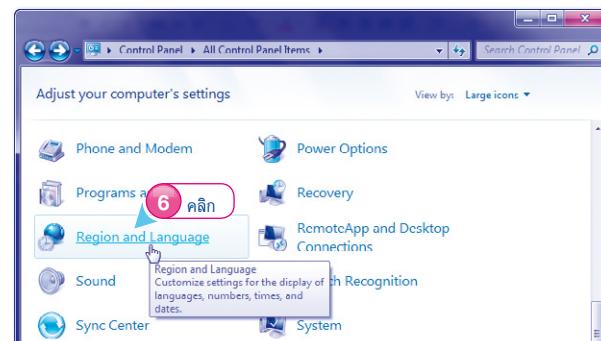
2. จะปรากฏได้จะล็อกบ็อกซ์ Preferences ขึ้นมา
3. คลิกเลือก Android > Build
4. จดจำพาร์ทในส่วนของ Default debug keystore:



5. ลบไฟล์ debug.keystore จากพาร์ทที่อยู่ในข้อ 4

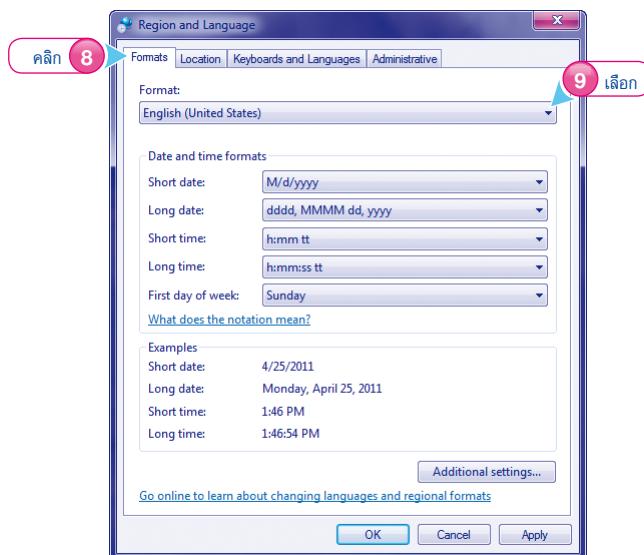


6. ปิดโปรแกรม Eclipse
7. เปิด Control Panel คลิกไอคอน Region and Language



8. คลิกแท็บ Formats

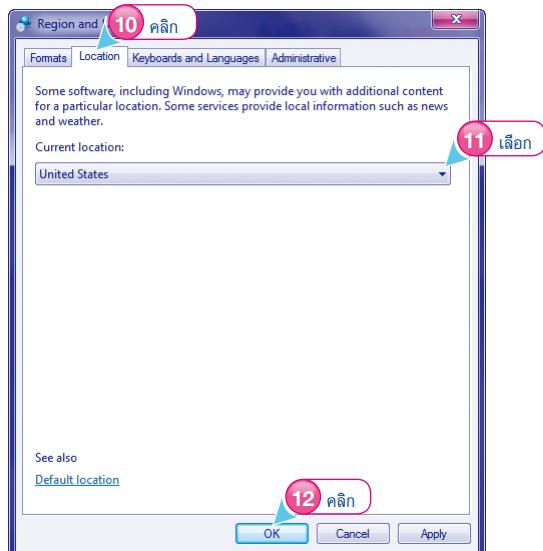
9. ในส่วนของ Format: คลิกเลือก English (United States)



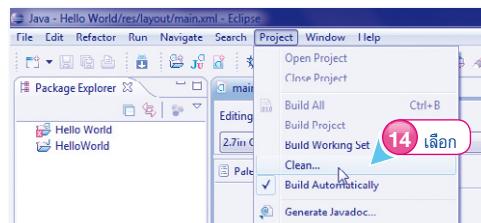
10. คลิกแท็บ Location

11. ในส่วนของ Current location: ให้คลิกเลือก United States

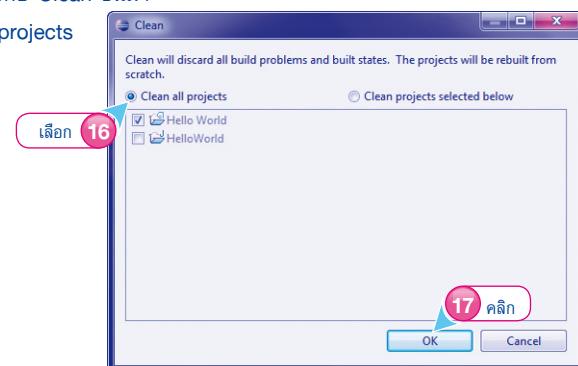
12. คลิกปุ่ม OK



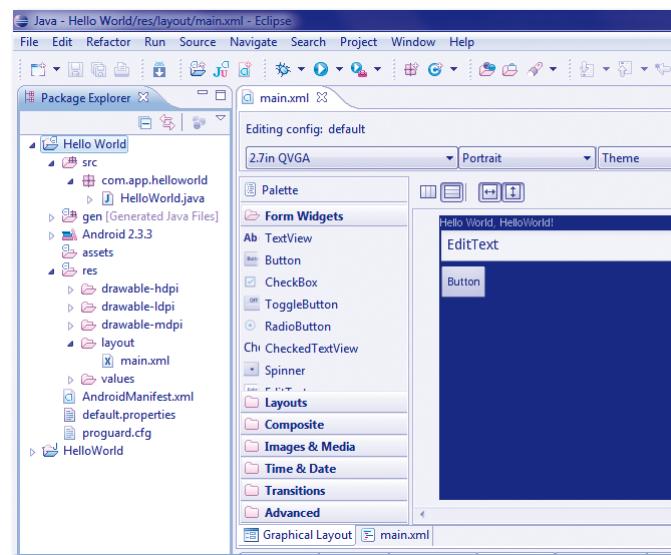
13. จากนั้นเปิดโปรแกรม Eclipse
14. คลิกเมนู Project > Clean...



15. จะปรากฏ dialogue ต่อไปนี้
16. คลิกเลือก Clean all projects
17. คลิกปุ่ม OK



18. ลังเกตได้ว่าลัญลักษณ์ Error ที่ชื่อโปรเจ็คหายไปแล้ว ซึ่งแสดงว่าสามารถรันโปรเจ็คต์ได้แล้วนั่นเอง



ในขั้นตอนต่อไปจะเป็นการเริ่มพัฒนาแอพพลิเคชัน โดยที่ไฟล์หลักๆ ที่สำคัญที่ต้องแก้ไข 2 ไฟล์คือ main.xml และ HelloWorld.java

- ไฟล์ main.xml ซึ่งเป็นไฟล์สำหรับการออกแบบหน้าจอแอพพลิเคชัน โดยกำหนดให้มีรูปแบบตามความต้องการของผู้ใช้งาน เช่น มีปุ่มที่ใช้ประเภทไดบां มีกล่องข้อความประเภทได โดยปกติแล้วไฟล์ main.xml จะถูกสร้างขึ้นเองโดยอัตโนมัติหลังจากที่ผู้อ่านได้สร้างโปรเจกต์ และผู้อ่านสามารถแก้ไขเพิ่มเติมได้ในภายหลัง
- ไฟล์ HelloWorld.java ซึ่งเป็นไฟล์หลักที่ใช้สำหรับกำหนดการทำงานของแอพพลิเคชัน ให้แอพพลิเคชันทำงานได้ตามความต้องการของนักพัฒนา

#### TIPS

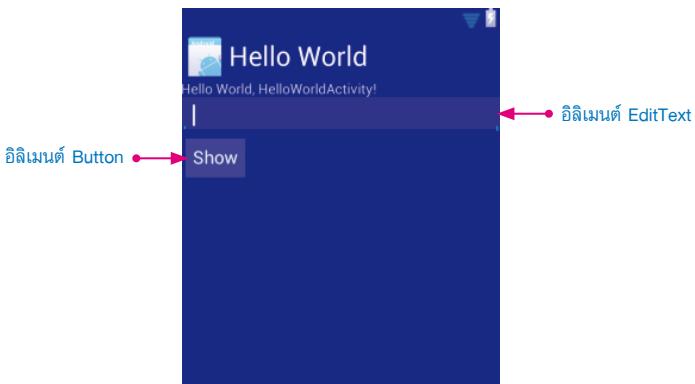


แอ็คทิวิตี้ (Activity) ถูกใช้สำหรับการสร้างส่วนต่อประสานกับผู้ใช้ (User Interface) เช่น เมนู (Menu) ปุ่มกด (Button) และกล่องข้อความ (Text Box) เป็นต้น ซึ่งในแอพพลิเคชันที่นี่ฯ จะสามารถสร้างแอ็คทิวิตี้ได้มากกว่า 1 แอ็คทิวิตี้ อย่างไรก็ตามจะต้องมีเพียง 1 แอ็คทิวิตี้ที่ใช้เป็นแอ็คทิวิตี้หลัก

ทั้งนี้จากตัวอย่างแอพพลิเคชัน Hello World การสร้างโปรเจกต์มีการสร้างแอ็คทิวิตี้ชื่อ HelloWorld ซึ่งหมายความว่าผู้อ่านได้สร้างคลาส (Class) HelloWorld ที่สืบทอดคุณสมบัติต่างๆ ทั้งหมดของคลาส แอ็คทิวิตี้และกำหนดให้คลาส HelloWorld เป็นแอ็คทิวิตี้หลักของโปรแกรมนั้นเอง

## ออกแบบหน้าจอแอพพลิเคชัน

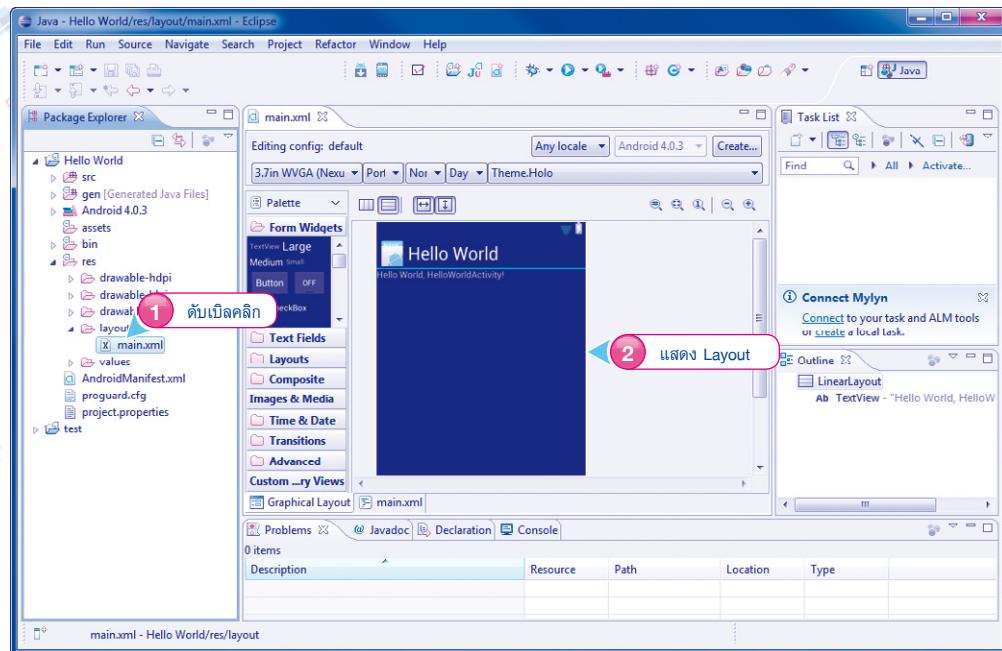
แอพพลิเคชันในบทนี้เป็นแอพพลิเคชันแสดงข้อความแจ้งเตือนทางหน้าจอ เมื่อผู้อ่านคลิกปุ่มกดโดยแอพพลิเคชันจะดึงข้อมูลที่พิมพ์ลงใน EditText มาแสดง เช่น “Hello World : ตามด้วยข้อความ” ซึ่งมีการออกแบบหน้าจอดังนี้



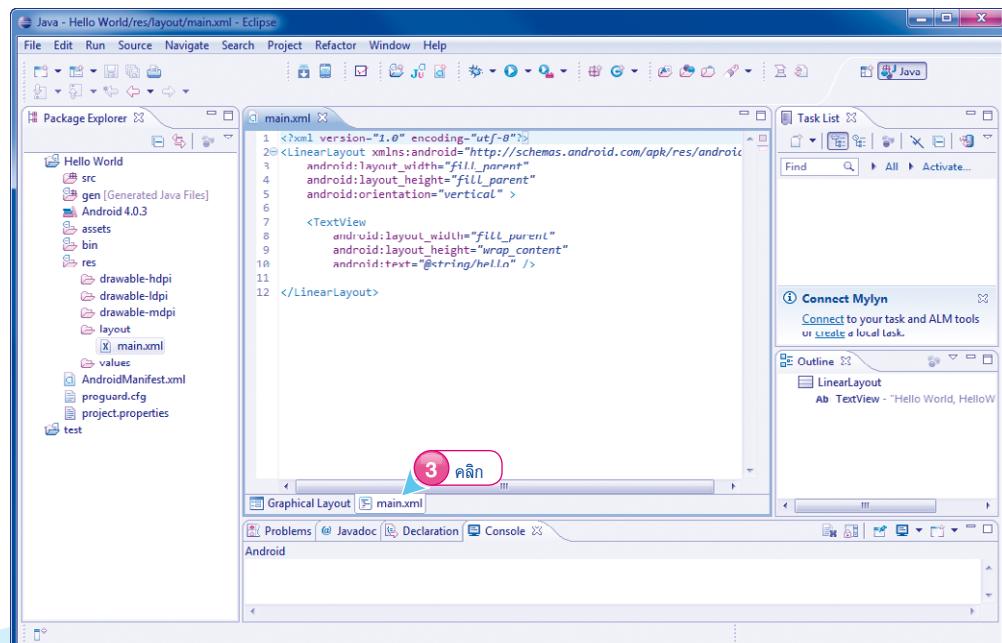
▲ หน้าจอแอพพลิเคชัน Hello World

การออกแบบหน้าจอแอพพลิเคชันเป็นการแก้ไขโค้ดในไฟล์ XML (Extensible Markup Language) ซึ่งส่วนของแอพพลิเคชัน Hello World นี้จะแก้ไขที่ไฟล์ main.xml ซึ่งเป็นไฟล์หลักสำหรับการออกแบบหน้าจอของแอพพลิเคชัน ซึ่งมีขั้นตอนการแก้ไขดังนี้

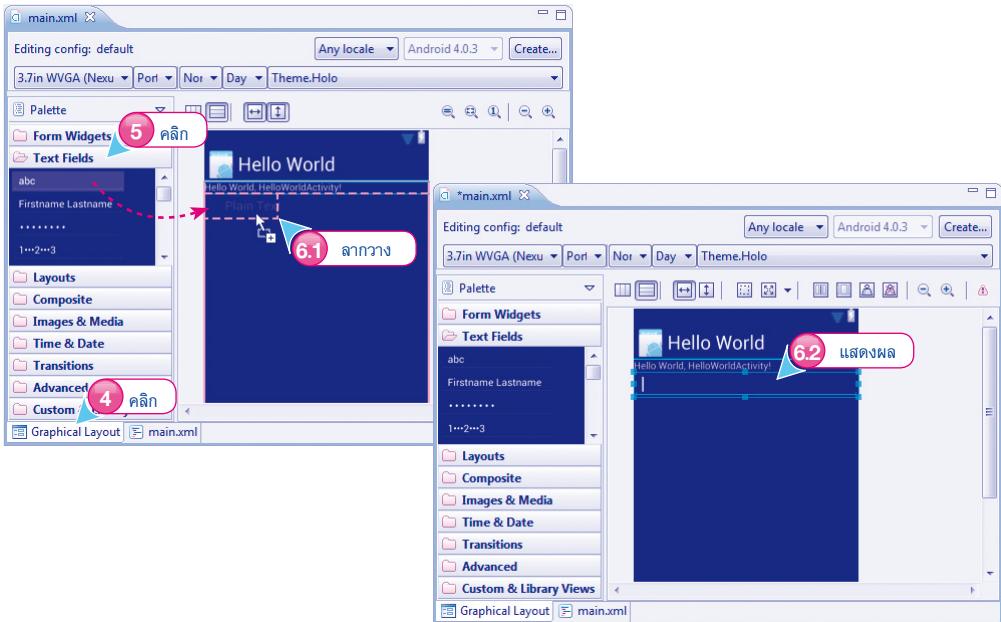
1. ดับเบิลคลิกไฟล์ main.xml ที่อยู่ในล文ของ res/layout
2. จะปรากฏ Layout ทางด้านขวาเมื่อ



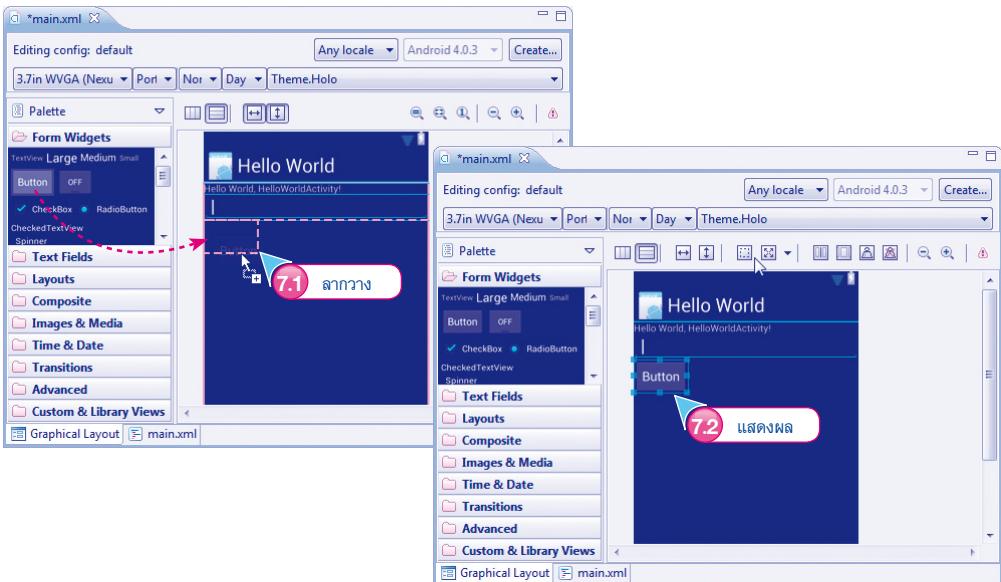
3. ผู้อ่านสามารถลับจากรูปแบบ Layout เป็นแบบโค้ด XML ได้ โดยคลิกแท็บ main.xml ซึ่งมีโค้ด xml ดังนี้



4. ในส่วนของการออกแบบแอพพลิเคชัน ให้ผู้อ่านคลิกแท็บ Graphical Layout
5. คลิกที่ Text Fields
6. คลิกเลือกอิลิเมนต์ (Element) EditText โดยคลิกค้างไว้ แล้วลากมาระหว่างที่หน้าจอออกแบบทางด้านขวา เพื่อสร้างกล่องข้อความบนหน้าจอ

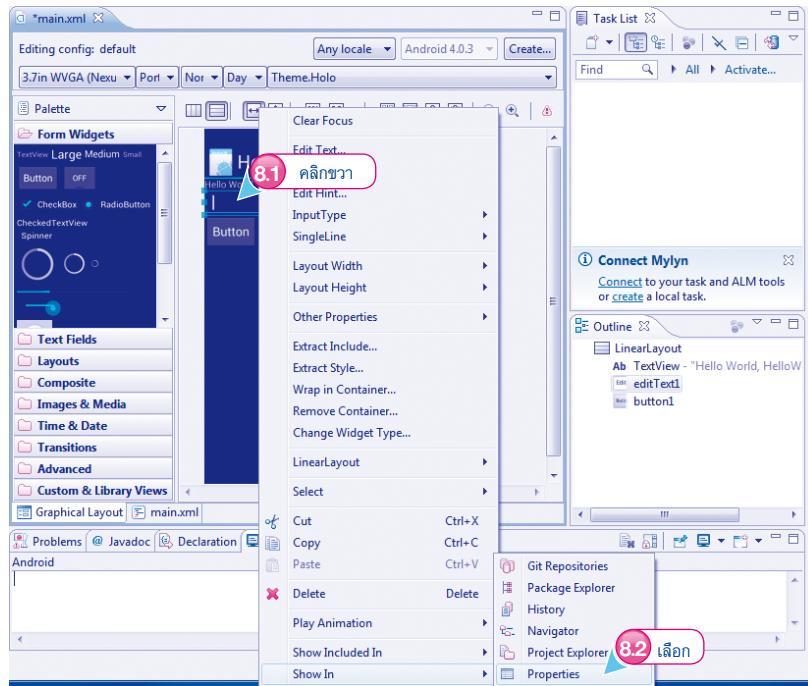


7. คลิกเลือกอิลิเมนต์ Button แล้วลากมาระหว่างที่หน้าจอออกแบบทางด้านขวา เพื่อสร้างปุ่มกดบนหน้าจอ

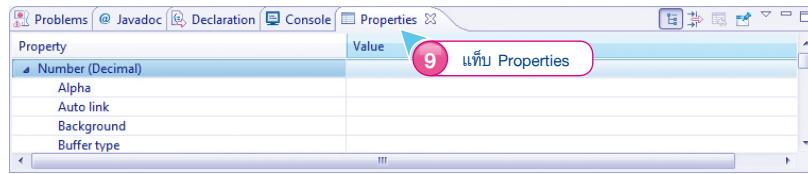




8. ในกรณีที่ผู้อ่านลรรังหน้าจอเลื่อนลิ้นแล้ว ผู้อ่านสามารถปรับแต่งหรือแก้ไขหน้าจอให้ดูเรียบร้อย สวยงามตามที่ออกแบบไว้ได้ เช่น แก้ไขข้อความ หรือปรับขนาดความกว้างและความยาว โดยให้คลิกขวาที่อิลิเม้นต์ และเลือกคำสั่ง Properties



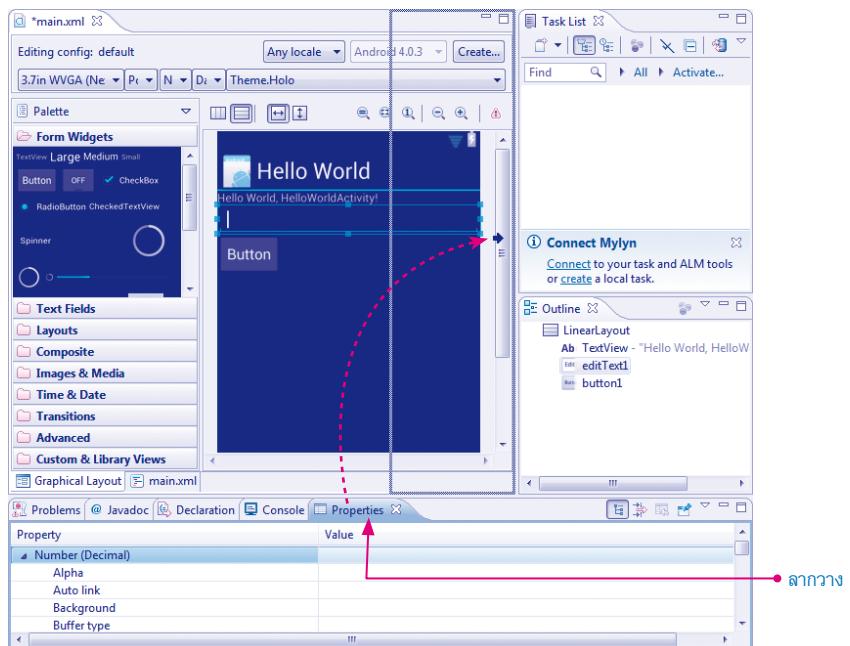
### 9. จะปรากฏแท็บ Properties ด้านล่างดังนี้



## TIPS

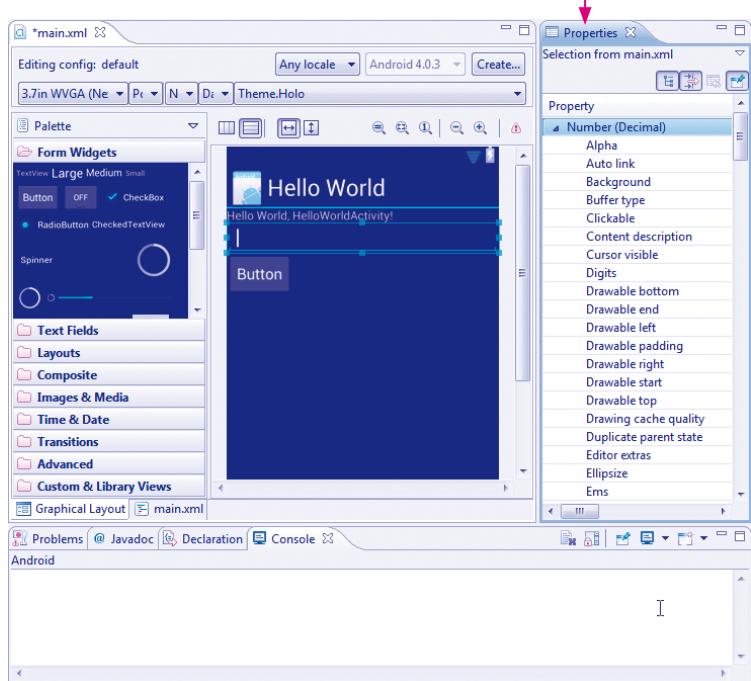


ผู้อ่านสามารถปรับเปลี่ยนตำแหน่งของหน้าต่าง Properties ได้โดยคลิกแท็บ Properties ค้างไว้ แล้วลากไปยังตำแหน่งที่ต้องการดังนี้



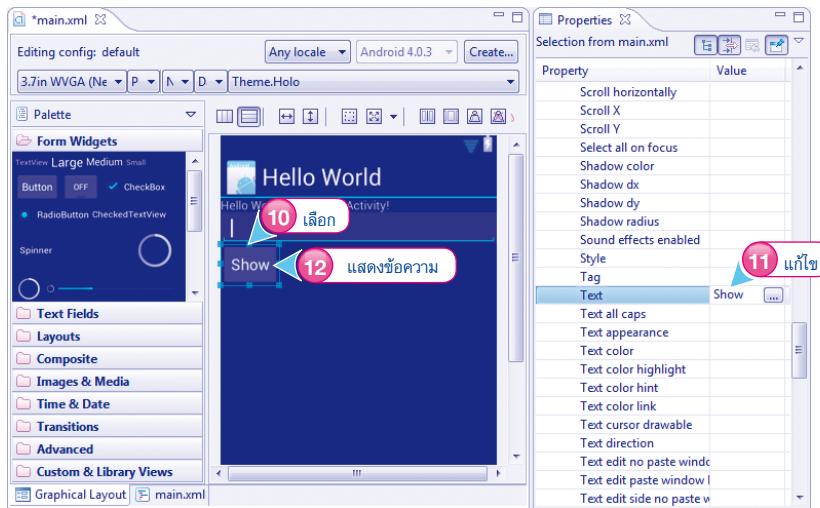
ลากว่าง

แสดงผล

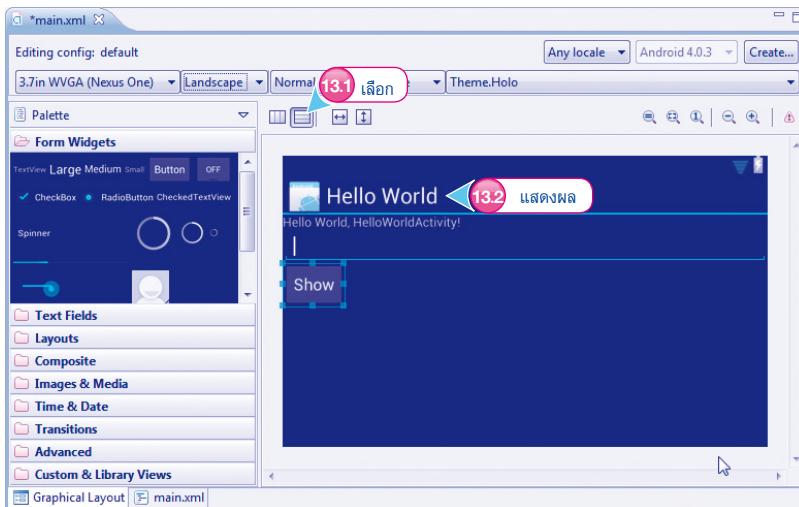




10. แก้ไขข้อความของอิลิเม้นต์ Button โดยคลิกเลือกอิลิเม้นต์ Button
11. ในส่วนของพร็อพเพอร์ตี้ Text แก้ไขข้อความจาก Button เป็น Show
12. ข้อความของอิลิเม้นต์ Button จะเปลี่ยนไป



13. ผู้อ่านยังสามารถปรับเปลี่ยนขนาดของจอและการหมุนได้ จากกลุ่มเมนูที่อยู่ด้านบนของส่วน Layouts เช่น รูปแบบแนวตั้งหรือแนวนอน ทั้งนี้หน้าจอของ Emulator ก็จะเปลี่ยนแปลงไปตาม การกำหนดค่าของผู้อ่าน



## จากการออกแบบหน้าจอที่ฝ่ายน้ำ โปรแกรม Eclipse จะสร้างโค้ด XML ให้โดยอัตโนมัติดังนี้

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical" >
6
7     <TextView
8         android:layout_width="fill_parent"
9         android:layout_height="wrap_content"
10        android:text="@string/hello" />
11
12    <EditText
13        android:id="@+id/editText1"
14        android:layout_width="match_parent"
15        android:layout_height="wrap_content" >
16
17        <requestFocus />
18    </EditText>
19
20    <Button
21        android:id="@+id/button1"
22        android:layout_width="wrap_content"
23        android:layout_height="wrap_content"
24        android:text="Show" />
25
26 </LinearLayout>

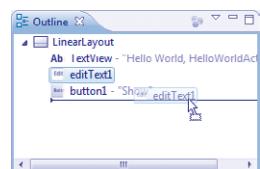
```

- กำหนดคุณสมบัติของ TextView
- กำหนดคุณสมบัติของ EditText
- ชื่ออิลิเม้นต์ EditText
- กำหนดคุณสมบัติของ Button
- ชื่ออิลิเม้นต์ Button

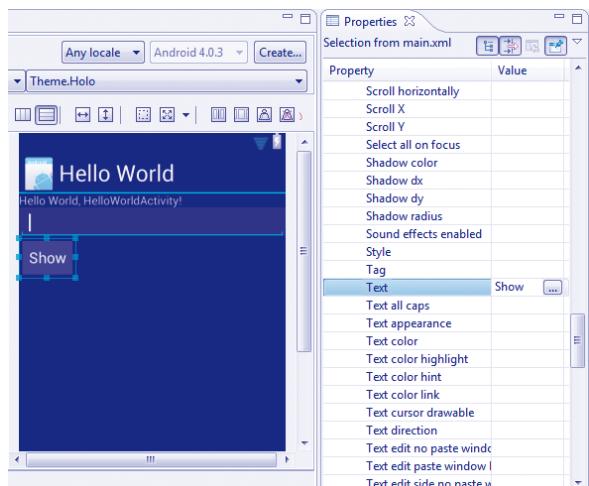
นอกจากนี้ผู้อ่านสามารถเขียนโค้ด XML เพื่อออกแบบหน้าจอด้วยตนเองจากไฟล์ main.xml ก็ได้ แต่ในความคิดเห็นส่วนตัวของผู้เขียนแล้ว การเริ่มต้นพัฒนาแอพพลิเคชันบน Android นั้น การแก้ไขหน้าจอ แอพพลิเคชันโดยการคลิกแล้วลากตามวิธีพื้นที่บนหน้าจอ ถือว่าเป็นวิธีที่ง่ายที่สุด ซึ่งไม่มีความจำเป็นต้องแก้ไขส่วนโค้ดของ XML แต่อย่างใด และลดโอกาสที่จะเกิดข้อผิดพลาดให้น้อยลงด้วยเช่นกัน

ขั้นตอนต่อไปจะอธิบายถึงเครื่องมือช่วยเหลือในการสร้างหน้าจอให้มีความง่ายยิ่งขึ้น ดังนี้

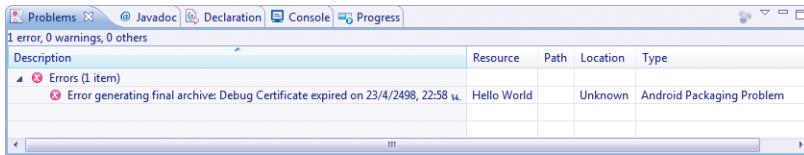
- Outline** เป็นกล่องเครื่องมือทางขวา มีอุปกรณ์ที่ช่วยในการสร้าง ลบ จัดลำดับก่อนหลังของอิลิเม้นต์ต่างๆ ภายในหน้าจอ เช่น การปรับให้อิลิเม้นต์ EditText ไว้ด้านล่างอิลิเม้นต์ Button ทำได้โดยลากอิลิเม้นต์ EditText ลงในตำแหน่งที่ต้องการดังรูป



- Properties** เป็นกล่องเครื่องมือสำหรับการกำหนดคุณสมบัติต่างๆ ของอิลิเม้นต์ เช่น แก้ไขข้อความ สีตัวอักษร ความกว้าง ความสูง โดยเครื่องมือนี้จะช่วยลดความยุ่งยากในการเขียนโค้ด XML ได้มากเลยทีเดียว



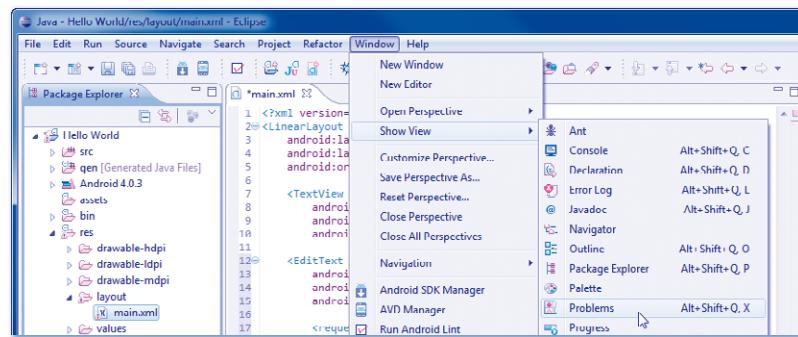
นอกจากกล่องเครื่องมือ Properties แล้ว ส่วนด้านล่างสุดได้รวมเครื่องมือช่วยในการพัฒนาแอ��泲ิเคชันอีกหนึ่งชั้น Problems, Javadoc, Declaration และ Console เอาไว้ด้วย ซึ่งจะถูกใช้สำหรับการทำงานในส่วนอื่นๆ ต่อไป เช่น ส่วน Problems จะรายงานปัญหาต่างๆ ในการทำงานของแอฟ泲ิเคชัน เป็นต้น



### NOTE



สำหรับการพัฒนาโปรแกรมบนเครื่องมือ Eclipse นั้น บ่อยครั้งที่เกิดข้อผิดพลาดของเครื่องมือต่างๆ จึงหายไปผู้อ่านสามารถเรียกใช้กล่องเครื่องมือต่างๆ ได้จากการคลิกเมนู Window > Show View จากนั้นเลือกเครื่องมือที่ต้องการดังรูป



## เขียนโค้ดกำหนดการทำงานของแอฟ泲ิเคชัน

เมื่อผู้อ่านได้ออกแบบหน้าจอโทรศัพท์จำลองหรือ Emulator เรียบร้อยแล้ว ขั้นตอนต่อไปคือ การเขียนโค้ดกำหนดการทำงานของแอฟ泲ิเคชัน โดยเขียนโค้ดในไฟล์ HelloWorld.java เพื่อกำหนดให้แอฟ泲ิเคชันแสดงผลคำว่า Hello World ออกทางหน้าจอ ซึ่งมีขั้นตอนดังนี้

1. ดับเบิลคลิกไฟล์ HelloWorldActivity.java ในไดเร็คทอรี src/com.app.helloworld/HelloWorldActivity.java เพื่อเปิดไฟล์ขึ้นมา ซึ่งมีโค้ดเริ่มต้นแสดงดังรูป

```

1 package com.app.helloworld;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5
6 public class HelloWorldActivity extends Activity {
7     /** Called when the activity is first created. */
8     @Override
9     public void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.main);
12     }
13 }

```

## 2. เขียนโค้ดกำหนดการทำงานของแอพพลิเคชันเพิ่มเติมต่อไปนี้

```

1 package com.app.helloworld;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.view.View.OnClickListener;
7 import android.widget.Button;
8 import android.widget.EditText;
9 import android.widget.Toast;
10
11 public class HelloWorldActivity extends Activity {
12     /** Called when the activity is first created. */
13     @Override
14     public void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.main);
17
18         Button button = (Button) findViewById(R.id.button1); ← การสร้างปุ่มกด
19
20         button.setOnClickListener(new OnClickListener() ← เขียนโค้ดเพิ่มเติม
21
22             @Override
23             public void onClick(View v) {
24                 EditText name = (EditText) findViewById(R.id.editText1);
25                 Toast.makeText(HelloWorldActivity.this,
26                             "Hello World :" + name.getText(), Toast.LENGTH_SHORT).show();
27             }
28         });
29     }
30 }

```

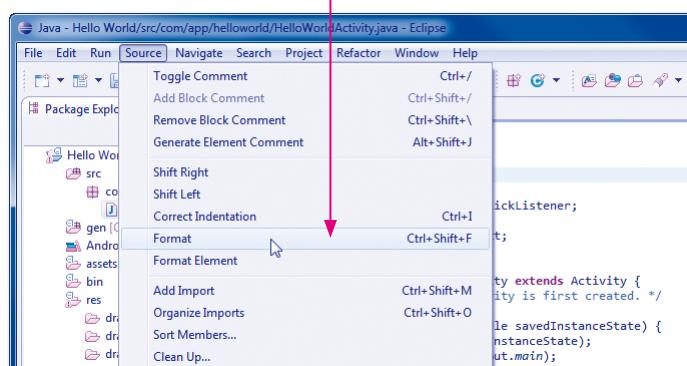
### อธิบายโค้ดการทำงานของแอพพลิเคชัน

- ในบรรทัดที่ 16 พิจารณาคำสั่ง setContentView(R.layout.main); ซึ่งเป็นพังก์ชันวาดหน้าจอ โดยอ้างอิงจากไฟล์ main.xml ที่ผู้อ่านได้ออกแบบไว้ให้ข้อที่ผ่านมาแล้ว
- ในบรรทัดที่ 20 และ 23 ทั้งสองส่วนเป็นการอ้างอิงไปยังอิลิเม้นต์ต่างๆ บน Emulator เพื่อให้โปรแกรมสามารถทำงานได้อย่างถูกต้อง เช่น การตัดจับเมื่อคลิกปุ่ม (button.setOnClickListener) และการดึงค่าข้อมูลมาใช้งาน (Toast.makeText) เป็นต้น ซึ่งการทำงานในส่วนนี้จะเป็นการเพิ่มเหตุการณ์หรืออีเวนต์ (Event) ให้กับปุ่มกด (Button)

### TIPS



หากผู้อ่านลังเลจะเห็นได้ว่า โค้ดที่เขียนขึ้นนี้ในบางครั้งจะไม่มีการจัดรูปแบบที่เหมาะสม เช่น การใช้เครื่องหมายแท็บหรือการเริ่มบรรทัด ดังนั้น ผู้อ่านสามารถจัดรูปแบบของโค้ดได้โดยคลิกเมนู Source > Format หรือกดปุ่ม **Ctrl+Shift+F** ซึ่งผู้อ่านสามารถเลือกส่วนที่จะจัดรูปแบบหรือเลือกทั้งหมดในไฟล์นั้นๆ ก็ได้



```

1 package com.app.helloworld;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.view.View.OnClickListener;
7 import android.widget.Button;
8 import android.widget.EditText;
9 import android.widget.Toast;
10
11 public class HelloWorldActivity extends Activity {
12     /* Called when the activity is first created. */
13     @Override
14     public void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.main);
17
18         Button button = (Button) findViewById(R.id.button1);
19
20         button.setOnClickListener(new OnClickListener() {
21
22             public void onClick(View v) {
23                 EditText name = (EditText) findViewById(R.id.editText1);
24                 Toast.makeText(HelloWorldActivity.this,
25                             "Hello World :" + name.getText(), Toast.LENGTH_SHORT)
26                             .show();
27             }
28         });
29     }
30 }

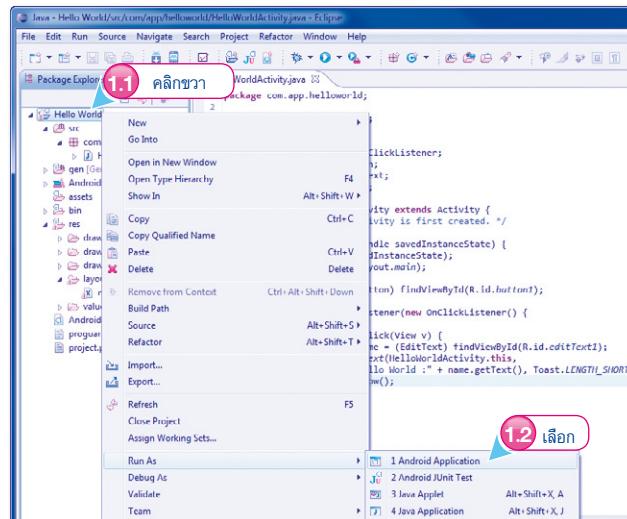
```

ผลลัพธ์เมื่อจัดรูปแบบของโค้ดด้วยคำสั่ง Format

## กดสอบการทำงานของแอพพลิเคชันผ่านໂຕຣສັບກໍາຈາລອງ (Emulator)

หลังจากการพัฒนาโปรแกรมเสร็จลື້ນ ขັ້ນຕອນຕ່ອໄປຄື່ອ ກາຣຄອມໄພລ໌ແລະຮັນແອພພລິເຄີຊັ້ນ ທັ້ນນີ້ ເນື່ອຈາກແອພພລິເຄີຊັ້ນຈະທດສອບການທຳການຝ່າຍ Emulator ດັ່ງນັ້ນ ເມື່ອຜູ້ອ່ານພັດທະນາແອພພລິເຄີຊັ້ນເລື່ອງຈຳລັກສິນ ສາມາດທຳລອບແອພພລິເຄີຊັ້ນໄດ້ຫລາຍວິທີ ຂຶ້ງວິທີທີ່ຜູ້ເຂົ້າໃຈແນະນຳມີຂັ້ນຕອນດັ່ງນີ້

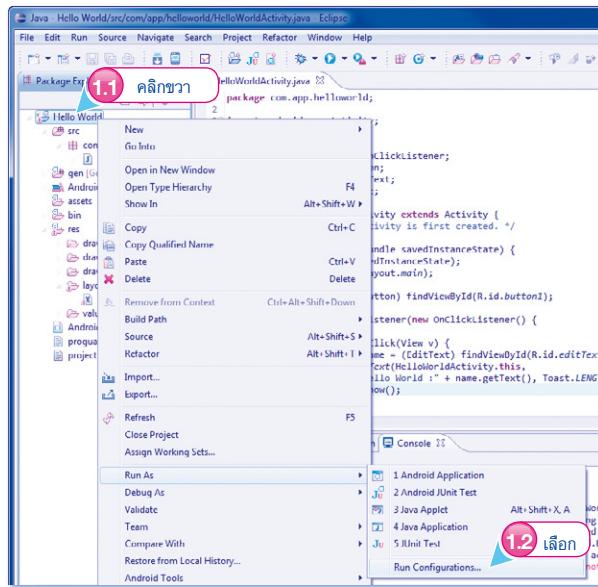
1. ຄລິກຂວາທີ່ໂປຣເຈັກ ແລ້ວເລືອກຄຳສັ່ງ Run As > Android Application





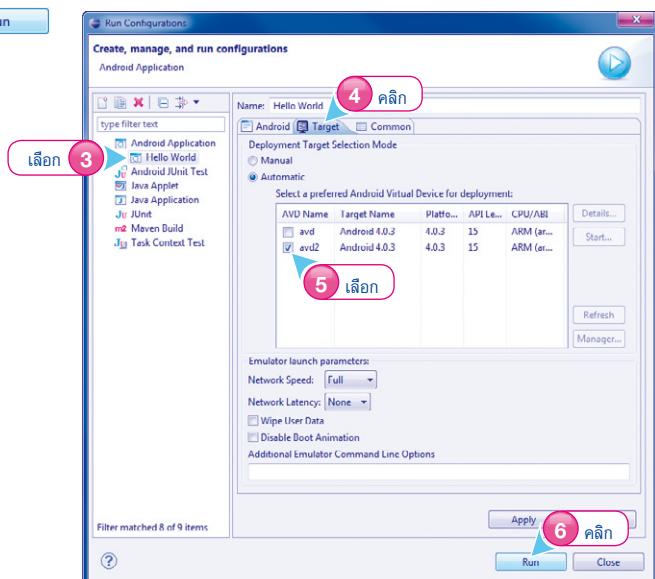
ในกรณีที่ผู้อ่านสร้าง AVD ไว้หลายเครื่อง ผู้อ่านสามารถเลือกรันเครื่องที่ต้องการตามขั้นตอนดังนี้

- คลิกขวาที่โปรเจกต์ แล้วเลือกคำสั่ง Run As > Run Configurations...

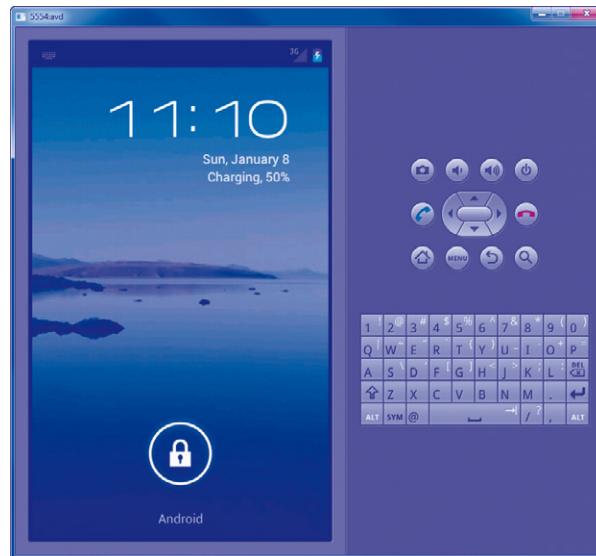


- จะปรากฏ dialogue ออกชื่อ Run Configurations เพื่อให้กำหนดค่าต่างๆ สำหรับการรันแอพพลิเคชัน
- ในส่วนของ Android Application คลิกเลือกโปรแกรม Hello World
- คลิกแท็บ Target
- คลิกเลือก AVD ที่จะใช้สำหรับรัน ในกรณีที่ผู้อ่านสร้าง AVD ไว้หลายเครื่อง ระบบก็จะแสดงเครื่องต่างๆ ให้เลือกใช้ในการรัน หรือผู้อ่านอาจจะเลือกแบบอัตโนมัติ (Automatic) ก็ได้เช่นกัน

- คลิกปุ่ม Run

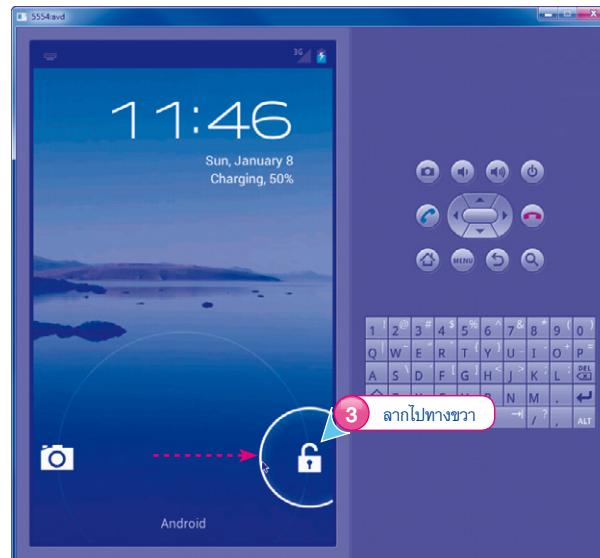


2. รอลักษณะ ก็จะปรากฏหน้าจอของ Emulator และแอพพลิเคชัน Hello World ที่ถูกโหลดเข้าไป

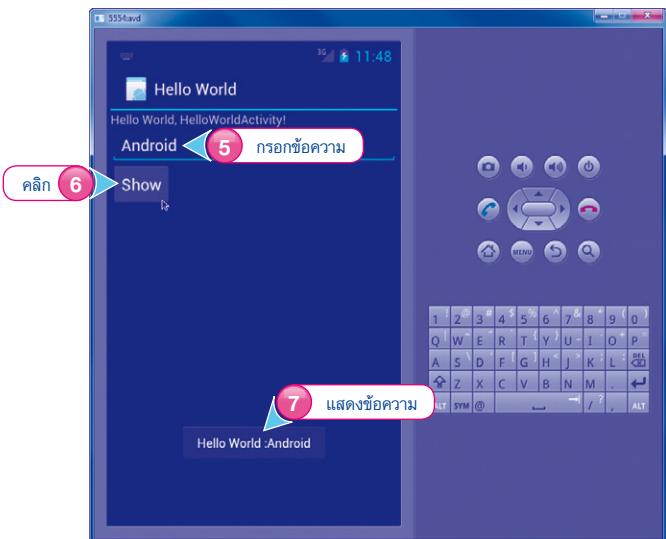


ทั้งนี้ให้สังเกตเมนูในส่วนบนของโทรศัพท์ว่ามีลัญลักษณ์ 3G และค่าลัญญาณแสดงความแรง (Signal Strength) ของเครือข่ายไร้สายหรือไม่ ในกรณีที่ไม่มีลัญญาณให้ผู้อ่านปิดและเปิด AVD ใหม่อีกครั้งหนึ่ง

3. ในกรณีที่โทรศัพท์ถูกล็อกไว้ ผู้อ่านสามารถเปิด Emulator โดยลากปุ่มล็อก Emulator จากทางซ้ายมือไปยังทางขวาเมื่อ ชิ้นก็จะคลายกับการใช้งานโทรศัพท์จริงนั้นเอง



4. เมื่อผู้อ่านเปิด Emulator รวมไปถึงการปลดล็อกเรียบร้อยแล้ว แอพพลิเคชัน Hello World จะถูกโหลดขึ้นมาโดยอัตโนมัติ
5. กรอกข้อความลงในกล่องข้อความ
6. คลิกปุ่ม Show
7. จะมีข้อความปรากฏขึ้นมาทางด้านล่างว่า Hello World : และตามด้วยข้อความที่ผู้อ่านกรอกลงในกล่องข้อความนั่นเอง ในกรณีนี้ผู้เขียนกรอกคำว่า Android ก็จะปรากฏคำว่า Hello World : Android ดังรูป



ทั้งนี้การใช้งาน Emulator นั้นจะเหมือนกับการใช้งานโทรศัพท์จริง แต่จะแตกต่างกันที่การใช้เมาส์ ในการเลือกการทำงานหรือแอพพลิเคชันต่างๆ แทนที่จะใช้มือในการลัมพัสหน้าจอ เช่น ผู้อ่านสามารถคลิกปุ่ม เพื่อกลับไปยังหน้าจอแรก





ผู้อ่านยังสามารถคลิกปุ่ม เพื่อเปิดเมนูของแอพพลิเคชันอยู่ๆ ของ Android ทั้งหมดขึ้นมาแสดง ซึ่งก็จะรวมไปถึงแอพพลิเคชัน Hello World ที่ผู้อ่านพัฒนาขึ้นมาด้วย และผู้อ่านสามารถตัดบีบคลิกที่แอพพลิเคชันต่างๆ เพื่อเข้าใช้งานได้เหมือนกับกรณีที่กล่าวมาแล้วในขั้นตอน



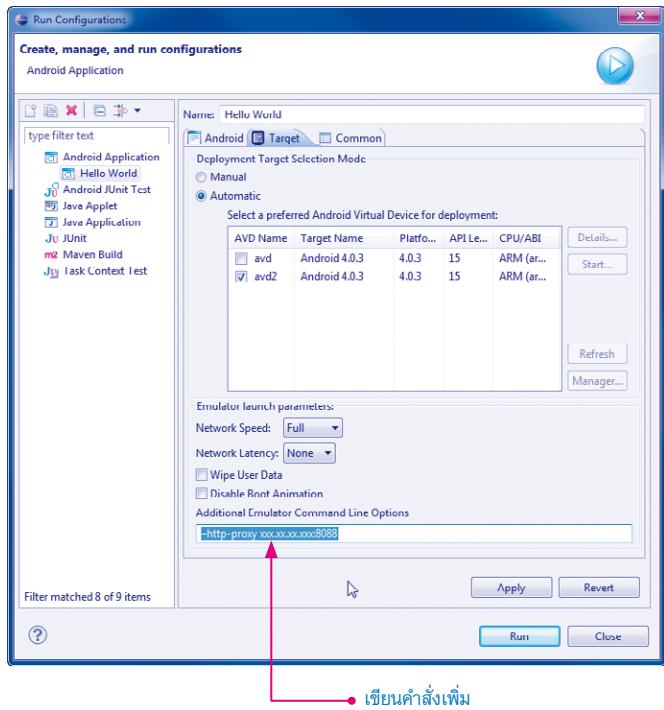
หรือจะเรียกใช้งานอินเทอร์เน็ต โดยคลิกปุ่ม



## WARN



ผู้อ่านสำคัญอีกส่วนหนึ่งที่ผู้อ่านอาจจะประสบปัญหาเมื่อเปิด Emulator คือ ไม่สามารถใช้งานอินเทอร์เน็ตได้ ในส่วนนี้เป็นปัญหาหลักของ Android โดยเฉพาะเมื่อผู้อ่านใช้งานเครือข่ายผ่านเครื่องบริการแทนหรือเครื่องพร็อกซี่ (Proxy) ซึ่งในกรณีนี้จะต้องสอบถามกับผู้ให้บริการเครือข่ายของผู้อ่าน ในกรณีของผู้เขียนจะเชื่อมต่ออินเทอร์เน็ตผ่านเครื่องพร็อกซี่ที่มีค่าไอพี (IP) = xxx.xx.xx.xxx ที่พอร์ต (Port) 8088 ดังนั้น ในส่วนของ Additional Emulator Command Line Options จะต้องเขียนเพิ่มเติมดังนี้ -http-proxy xxx.xx.xx.xxx:8088 ซึ่งผู้อ่านก็สามารถปรับเปลี่ยนค่าได้ตามความเหมาะสม



ทั้งนี้ผู้อ่านไม่สามารถโทรศัพท์เข้าและโทรออก Emulator นี้ได้จริง

## การส่งออกและนำเข้าโปรเจกต์ใน Eclipse

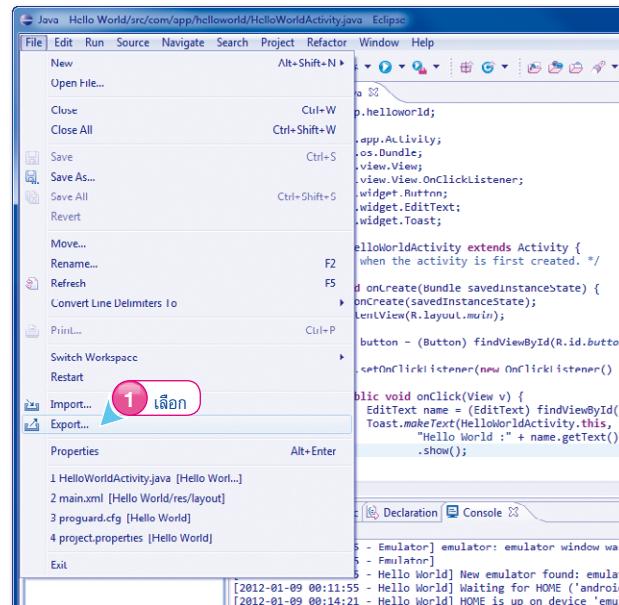
สำหรับผู้อ่านที่มีความคุ้นเคยกับโปรแกรม Eclipse และ ก็สามารถข้ามหัวข้อนี้ไปได้ อย่างไรก็ตาม สำหรับผู้ที่เริ่มพัฒนาแอพพลิเคชันนั้น ฟังก์ชันการนำเข้าและส่งออก (Import/Export) โปรเจกต์จะมีประโยชน์มากเลยทีเดียว

ทั้งนี้การนำเข้าและส่งออกนั้นสามารถทำได้หลายรูปแบบ เช่น การเก็บแบบการ (Archive), การเลือกเก็บในแต่ละโปรเจกต์หรือเก็บทั้งหมด แต่วิธีที่ผู้เขียนลังเกตเห็นได้ว่ามีประโยชน์ก็คือ การส่งออกแบบ Archive File และนำเข้ากลับมาในรูปแบบเดิม เช่น ในไฟล์ลีบลีฟайлตัวอย่าง .zip แนบมาด้วยกับ CD ซึ่งผู้อ่านสามารถคลายไฟล์ออก และสามารถนำเข้ามาใช้งานได้ทันที เพื่อตรวจสอบความถูกต้องในการพัฒนาโปรแกรม อย่างไรก็ตามผู้เขียนแนะนำให้ผู้อ่านทดลองพัฒนาโปรแกรมด้วยตนเองก่อน

## การส่งออกโปรเจกต์เป็น Archive File

จากโปรเจกต์ที่ได้พัฒนาไว้สมบูรณ์แล้ว ผู้เขียนจะแนะนำการส่งออกโปรเจกต์เป็น Archive File ก่อน ซึ่งมีขั้นตอนการส่งออก (Export) ไปยัง Archive File ชื่อ Backup\_Hello\_World.zip ดังนี้

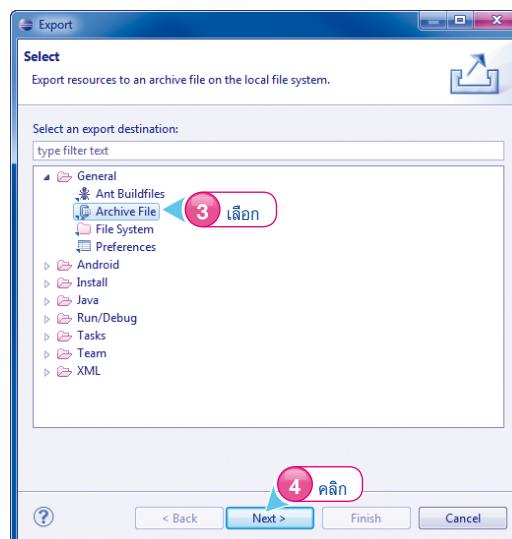
- คลิกเมนู File > Export...



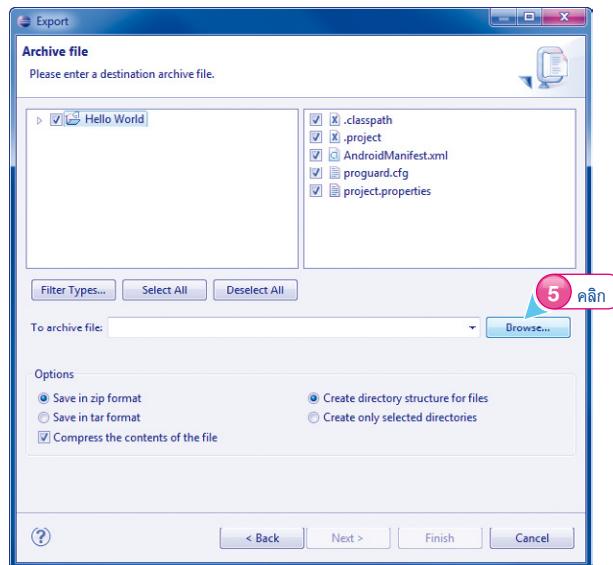
- จะปรากฏ dialogue ออกมือกซ์ Export ขึ้นมา

- คลิกเลือก General/Archive File

- คลิกปุ่ม **Next >**



5. คลิกปุ่ม **Browse...** เพื่อเลือกตำแหน่งจัดเก็บ Archive File

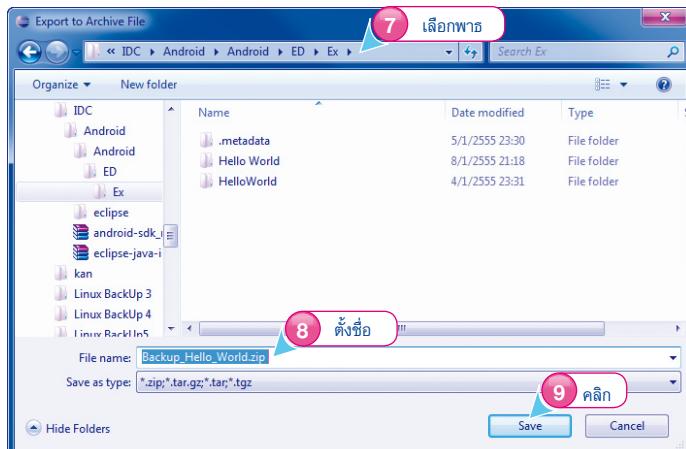


6. จะปรากฏ dialogue ออกบันทึก Export to Archive File ขึ้นมา

7. ในที่นี่จะเก็บไว้ที่ E:\IDC\Android\Android\ED\Ex

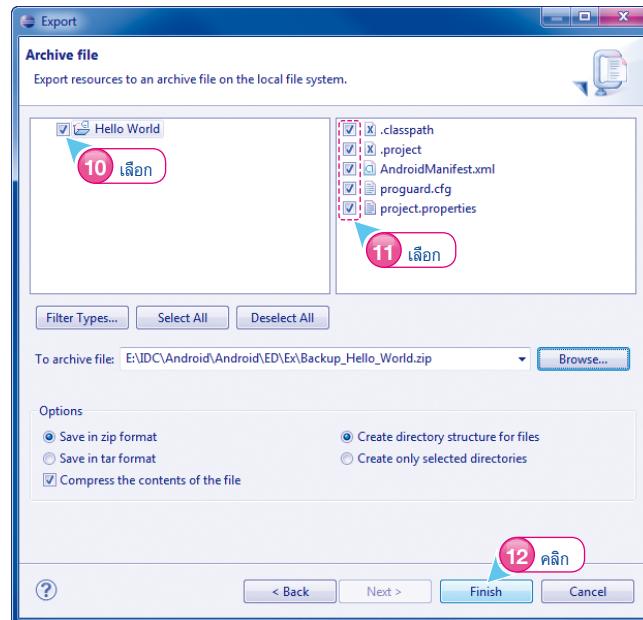
8. ตั้งชื่อไฟล์ ในที่นี้ผู้เขียนตั้งชื่อไฟล์คือ Backup\_Hello\_World.zip

9. คลิกปุ่ม **Save**

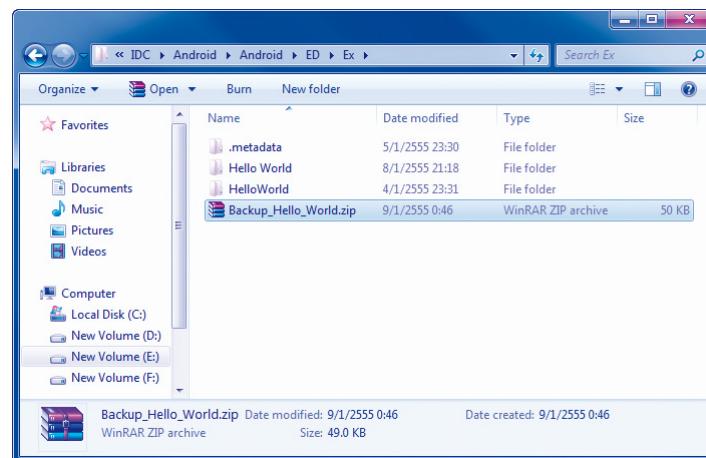




10. เลือกโปรเจกต์ที่ต้องการจัดเก็บ ซึ่งในตัวอย่างนี้ต้องการจัดเก็บโปรเจกต์ Hello World ดังนั้น ให้คลิกเลือกทางช้ายมือ
11. คลิกเลือกไฟล์ที่เกี่ยวข้องทางขวาเมื่อด้วย เช่น .classpath และ AndroidManifest.xml เป็นต้น
12. คลิกปุ่ม **Finish**



13. ได้ Archive File ในตำแหน่งที่กำหนดไว้



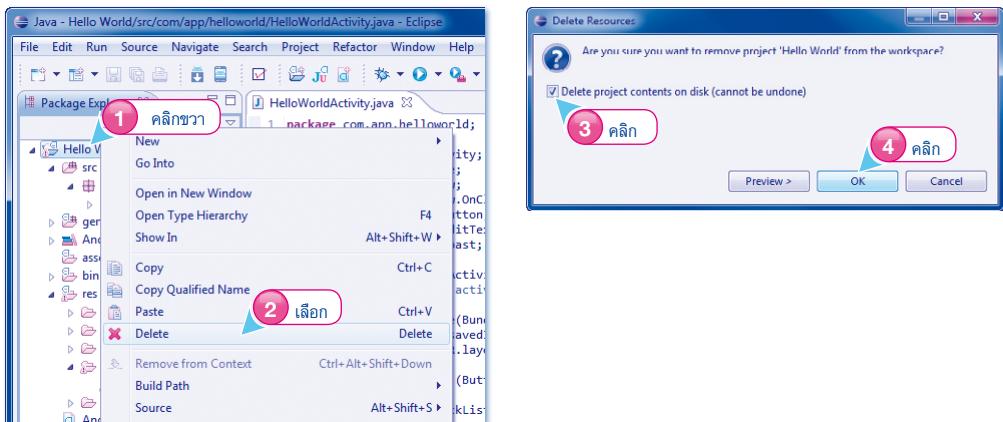
## การนำเข้าโปรเจกต์จาก Archive File

ขั้นตอนต่อไปเป็นขั้นตอนการนำเข้า (Import) Archive File ที่ชื่อ Backup\_Hello\_World.zip ที่ได้จากหัวข้อที่ผ่านมา ทั้งนี้เนื่องจากเดิมมีโปรเจกต์ Hello World อยู่ในโปรแกรม Eclipse ให้ผู้อ่านลบโปรเจกต์เดิมออกก่อน ซึ่งสามารถทำได้โดยคลิกขวาที่โปรเจกต์ แล้วเลือกคำสั่ง Delete

**WARN**

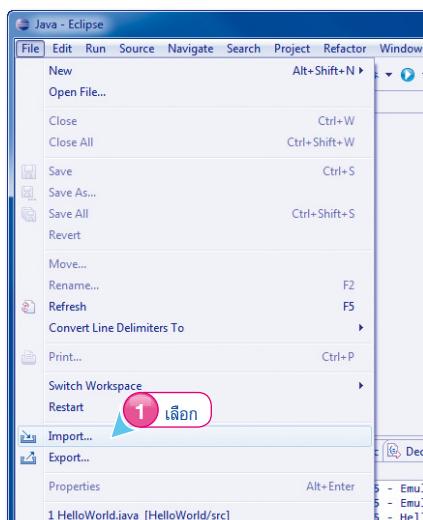


ข้อควรระวังคือ ผู้อ่านควรตรวจสอบให้แน่ใจว่า การส่งออกโปรเจกต์นั้นสมบูรณ์หรือไม่ เพราะเมื่อลบโปรเจกต์แล้ว จะไม่สามารถคืนข้อมูลเดิมมาได้อีก

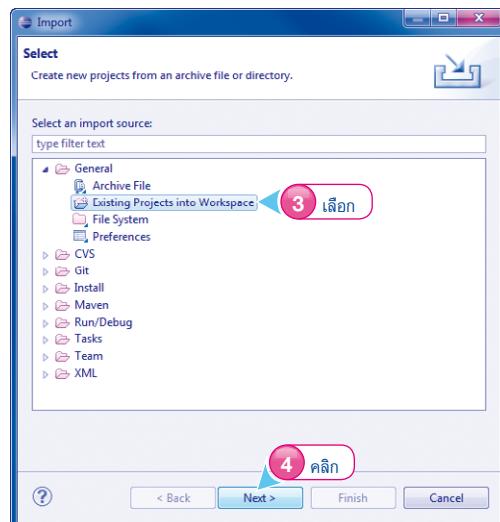


เมื่อลบโปรเจกต์ Hello World ออกแล้ว ขั้นตอนต่อไปจะเป็นการนำเข้าโปรเจกต์ ซึ่งมีขั้นตอนดังนี้

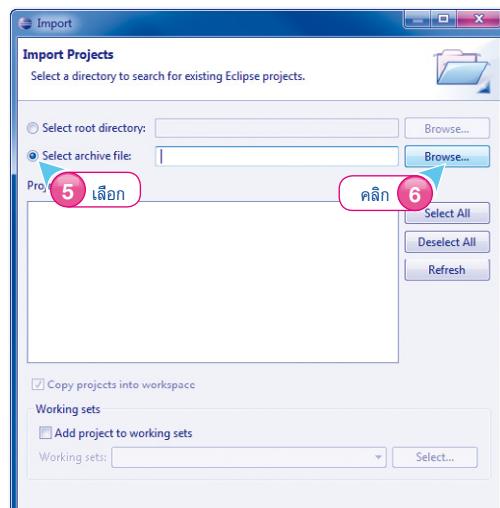
### 1. คลิกเมนู File > Import...



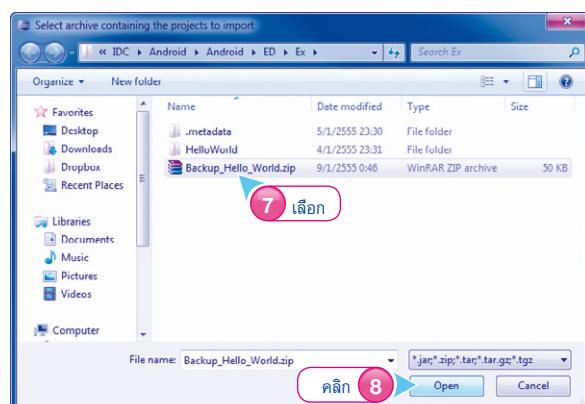
2. จะปรากฏ dialogue ให้เลือกมีอักษร Import ขึ้นมา
3. คลิกเลือก General/Existing Projects into Workspace
4. คลิกปุ่ม **Next >**



5. คลิกเลือก Select archive file:
6. คลิกปุ่ม **Browse...** เพื่อเลือกไฟล์ Archive File ที่ต้องการนำเข้ามา ซึ่งในตัวอย่างนี้อยู่ที่ E:\IDC\Android\Android\ED\Ex

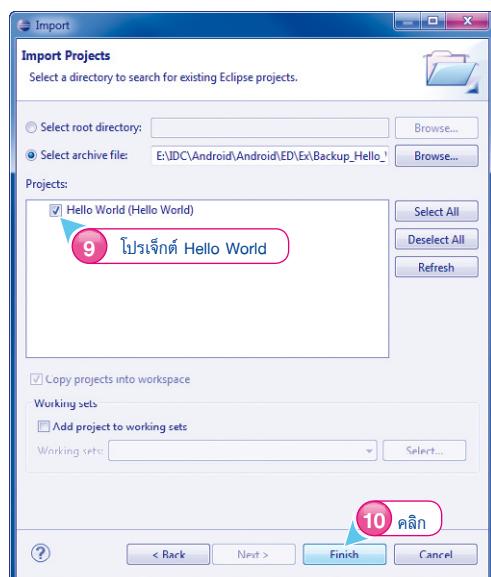


7. คลิกเลือกไฟล์ชื่อ Backup\_Hello\_World.zip ซึ่งเป็นไฟล์ที่ได้จากหัวข้อที่ผ่านมา
8. คลิกปุ่ม **Open**

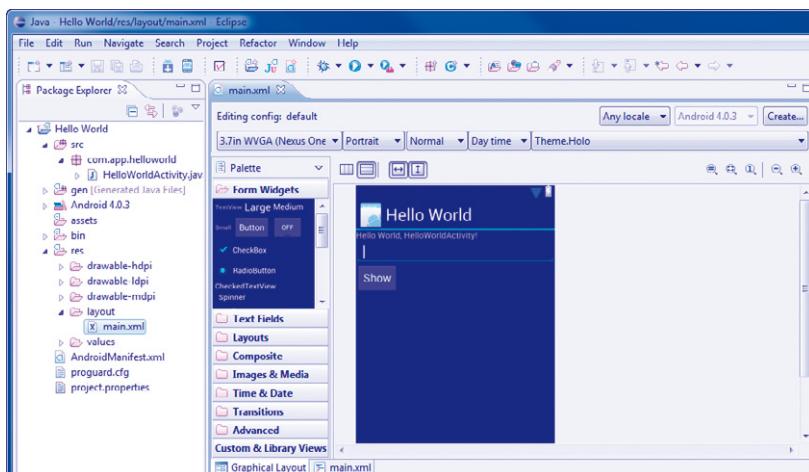


9. จะปรากฏโปรเจกต์ Hello World ขึ้นมา

10. คลิกปุ่ม **Finish**



11. จะได้โปรเจกต์ Hello World เข้ามา



### NOTE



ผู้อ่านที่สนใจการพัฒนาแอพพลิเคชันจากตัวอย่าง สามารถศึกษาโดยเพิ่มเติมได้จากเว็บไซต์ดังนี้

- <http://developer.android.com/resources/samples/index.html>
- <http://code.google.com/p/apps-for-android/>
- <http://code.google.com/hosting/search?q=label%3aAndroid>

ทั้งนี้เว็บไซต์ดังกล่าวจะมีตัวอย่างโค้ดประเภทต่างๆ ให้ผู้อ่านสามารถดาวน์โหลดได้ฟรีจากทาง Google เช่น Bluetooth Chat, BusinessCard, Contact Manager, Live Wallpaper, Searchable Dictionary, Snake Game และ Wiktionary เป็นต้น นอกจากนี้ทางบริษัทบังมีการสอนให้ฝึกพัฒนาแอพพลิเคชันได้อย่างง่าย (Tutorial) ด้วย เช่น Hello World และ NotePad ซึ่งผู้อ่านสามารถศึกษาเพิ่มเติมได้จากเว็บไซต์

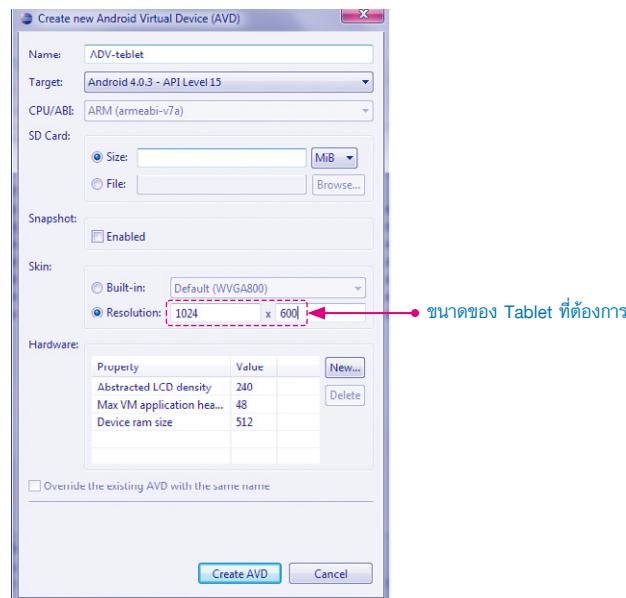
- <http://developer.android.com/resources/tutorials/hello-world.html>
- <http://developer.android.com/resources/tutorials/notepad/index.html>

# เริ่มต้นพัฒนาแอพพลิเคชันบน Tablet

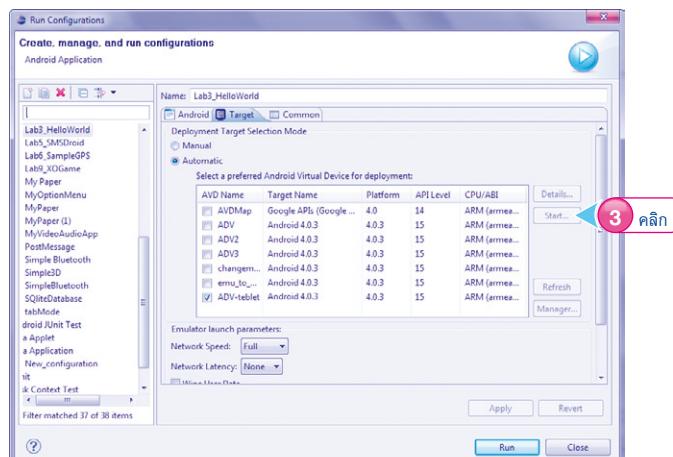
ในหัวข้อที่ผ่านมา เป็นการอธิบายขั้นตอนการพัฒนาแอพพลิเคชัน Android บนโทรศัพท์เคลื่อนที่ ในหัวข้อนี้ผู้เขียนจะแนะนำขั้นตอนการพัฒนาแอพพลิเคชัน Android บน Tablet บ้าง ซึ่งมีขั้นตอนแตกต่างกัน เนื่องจากขั้นตอนการสร้างโทรศัพท์จำลอง (Emulator) เท่านั้น ส่วนขั้นตอนอื่นๆ จะไม่แตกต่างกันเลย ไม่ว่าจะเป็น ขั้นตอนการออกแบบหน้าจอหรือการเขียนโปรแกรมก็ตาม

ในที่นี้ขออธิบายขั้นตอนการพัฒนาแอพพลิเคชันบน Tablet โดยสรุปดังนี้

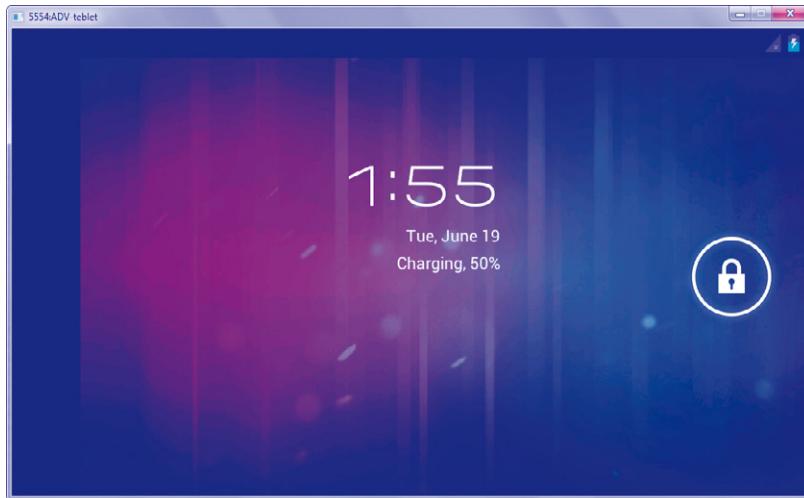
- สร้าง Emulator ใหม่ โดยกำหนดคุณสมบัติตั้งนี้



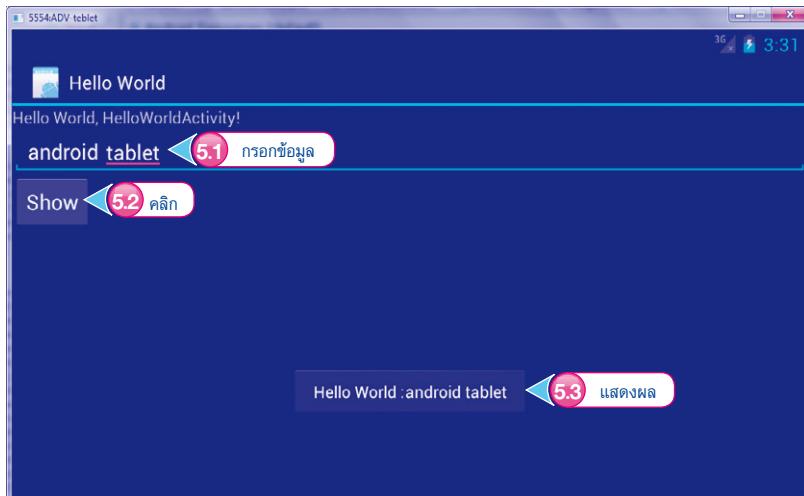
- จะมีชื่อ AVD ที่สร้างไว้ด้วย
- คลิกปุ่ม **Start...** เพื่อเปิดเครื่อง AVD



#### 4. รอลักษณะ Emulator จะสามารถใช้งานได้ดังรูป



#### 5. รันໂປຣັງເຈັດ Hello World ໃ້ມອືກຄັ້ງ ໂດຍເລືອກ AVD ເປັນ AVD-tablet ທີ່ສ້າງຂຶ້ນ ຈະໄດ້ໜ້າຈອແອພພລິເຄັ້ນດັ່ງນີ້



ຈາກຂັ້ນຕອນທີ່ກ່າວມາ ຈະລັ້ງເກີດໄດ້ວ່າການພັດນາແອພພລິເຄັ້ນ Android ບນໂທຣຕັພທີ່ເຄີ່ອນທີ່ກັບການພັດນາແອພພລິເຄັ້ນ Android Tablet ແຕກຕ່າງກັນເພີ່ມການລົງທຶນ AVD ລໍາຫຽນ Tablet ແລະການເລືອກ AVD ທີ່ໃຊ້ຮັນແອພພລິເຄັ້ນເທົ່ານັ້ນ



# สรุปท้ายบท

ในบทนี้เป็นการอธิบายถึงการเริ่มต้นพัฒนาแอพพลิเคชัน โดยอธิบายผ่านแอพพลิเคชันชื่อ Hello World ซึ่งเป็นการแสดงข้อความออกทางหน้าจอโทรศัพท์ รวมไปถึงเทคนิคต่างๆ ในการพัฒนาโปรแกรม เช่น การล็อกอินและนำเข้าโปรเจกต์ใน Eclipse เป็นต้น

## แบบฝึกหัดท้ายบท

แบบฝึกหัดในส่วนนี้ให้ผู้อ่านพัฒนาแอพพลิเคชันโดยมีกล่องข้อความจำนวน 2 กล่อง และปุ่มกด 1 ปุ่ม ซึ่งมีการทำงานดังนี้

เมื่อผู้อ่านพิมพ์ข้อความลงในกล่องทั้งสองแล้ว และคลิกปุ่ม Click Me! ให้แอพพลิเคชันแสดงข้อความ โดยการนำข้อความทั้งสองข้อความมาต่อกันและเว้นวรรคด้วย 1 ช่องว่าง (Space) เช่น พิมพ์ Hello และ Chakchai จากนั้นแสดงผล Hello Chakchai เป็นต้น

## พัฒนาแอพพลิเคชันอย่างจ่าย

เนื้อหาในบทที่ 4 นี้จะเป็นส่วนเพิ่มเติมจากบทที่ 3 จุดประสงค์เพื่อให้ผู้อ่านฝึกพัฒนาแอพพลิเคชันที่มีความซับซ้อนเพิ่มขึ้น เพื่อให้เข้าใจในรายละเอียดพื้นฐานสำหรับการพัฒนาแอพพลิเคชัน โดยที่จะเป็นการพัฒนาแอพพลิเคชันแปลงค่าระยะทางจากหน่วยกิโลเมตร (Kilometer) เป็นหน่วยไมล์ (Mile) และทำงานในทางกลับกันได้ นอกจากนี้เนื้อหาในบทนี้จะอธิบายถึงส่วนขยาย สำหรับการกำหนดรายละเอียดของแอตทริบิวต์ (Attribute) เพิ่มเติม เพื่อนำมาใช้ในการอ้างอิงในไฟล์ main.xml

สำหรับตัวอย่างในบทนี้ จะเป็นการประยุกต์ใช้พื้นฐานความรู้เรื่องระบบเมตริกซ์ (Metric System) ซึ่งเป็นระบบการวัดที่ใช้โดยนานาชาติ ซึ่งต่อมาหน่วยงาน SI (Système International d'Unités) ได้ยอมรับเป็นมาตรฐานและได้มีการนำไปใช้ในหลายประเทศ ยกเว้นในประเทศไทยมา ໄหลบเรีย และสหรัฐอเมริกา ซึ่งหน่วยวัดแบบเมตริกซ์นั้นโดยทั่วไปแล้วจะถูกใช้ในงานทางด้านวิทยาศาสตร์ (Science) และงานทั่วไป รวมไปถึงด้านธุรกิจต่างๆ เช่น หน่วยเมตร กิโลกรัม และวินาที (meter-kilogram-second) อีกด้วย

อย่างไรก็ตามก็ยังมีมาตรฐานการวัดอีกประเภทหนึ่งที่เรียกว่า Imperial Units หรือ Imperial Systems โดยมีต้นกำเนิดและใช้กันทั่วไปที่ประเทศอังกฤษ และกลุ่มที่เคยถูกปกครองโดยประเทศอังกฤษ เช่น ไมล์ (Mile) หรือหลา (Yard) และมีการนำมาใช้งานในชีวิตประจำวันเช่นเดียวกัน

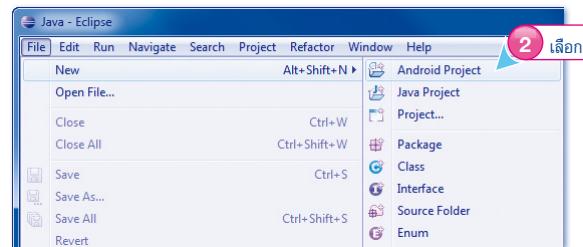
โดยเนื้อหาในบทนี้จะแบ่งออกเป็นขั้นตอนหลักๆ ดังนี้

1. เริ่มต้นสร้างโปรเจกต์
2. ออกแบบหน้าจอแอพพลิเคชัน
3. เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน
4. ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator)

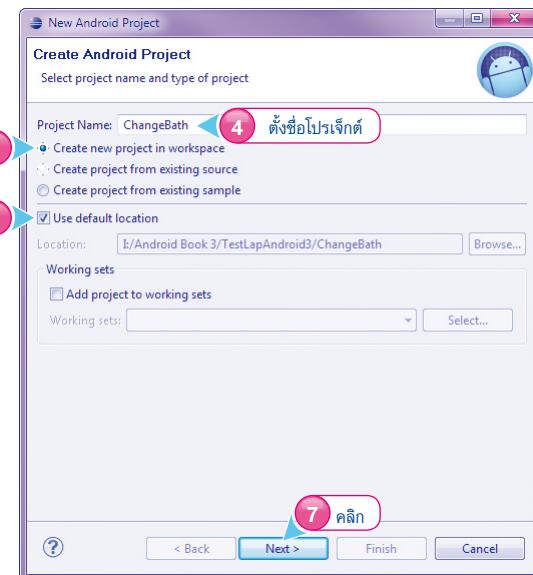
# เริ่มต้นสร้างโปรเจกต์

ขั้นตอนแรกในการพัฒนาแอพพลิเคชันทุกรูปแบบ การสร้างโปรเจกต์ ชึ่งผู้อ่านสามารถทำได้ตามขั้นตอนดังต่อไปนี้

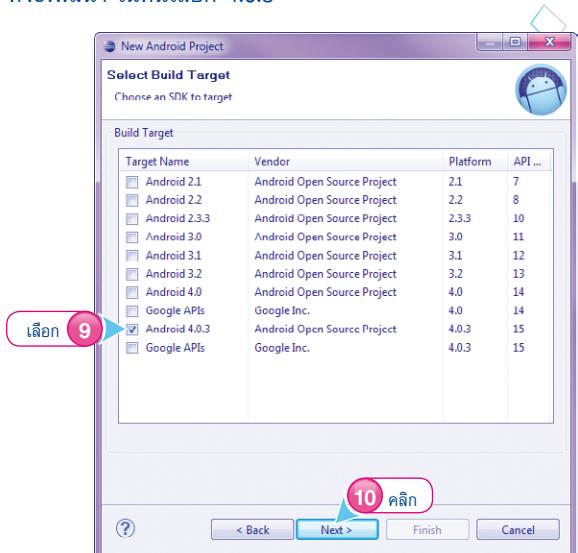
1. เปิดโปรแกรม Eclipse
2. คลิกเมนู File > New > Android Project



3. จะปรากฏ dialogue ขึ้นมา คลิก 'New Android Project' เพื่อกำหนดรายละเอียดต่างๆ ของโปรเจกต์
4. ตั้งชื่อโปรเจกต์ ในที่นี่ตั้งชื่อว่า ChangeBath
5. ในส่วนของ Contents คลิกเลือก 'Create new project in workspace' เพื่อสร้างโปรเจกต์ใหม่
6. คลิกเครื่องหมายถูกหน้า 'Use default location' เพื่อพัฒนาแอพพลิเคชันในตำแหน่งที่อยู่เดิมกับโปรเจกต์ Eclipse
7. คลิกปุ่ม **Next >**



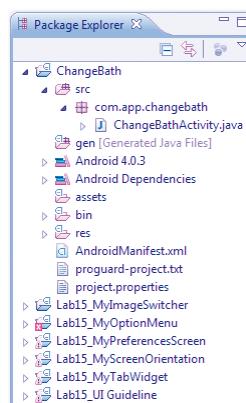
8. จะปรากฏ dialogue ให้เลือกบันทึกชื่อ New Android Project ในส่วนที่ล่อง ลำหัวบันทึกชื่อ Android
9. คลิกเลือกเวอร์ชันของ Android ที่จะพัฒนา ในที่นี่เลือก 4.0.3
10. คลิกปุ่ม **Next >**



11. จะปรากฏ dialogue ให้เลือกบันทึกชื่อ New Android Project ในส่วนที่ลาม ลำหัวบันทึกชื่อรายละเอียดของแอพพลิเคชัน
12. ตั้งชื่อของแอพพลิเคชันคือ ChangeBath
13. ชื่อแพ็คเกจจะอยู่ในรูปแบบของ xxx.xxx... ซึ่งในที่นี่ตั้งชื่อแพ็คเกจเป็น com.app.changebath
14. ตั้งชื่อ Activity ในที่นี่ตั้งชื่อเป็น ChangeBathActivity
15. ในส่วนของ Minimum SDK: คลิกเลือกเวอร์ชัน 15 ซึ่งเป็นเวอร์ชันเดียวกันกับ API Level ที่เลือกในข้อ 9
16. คลิกปุ่ม **Finish** ซึ่งถือเป็นอันเสร็จสิ้นการสร้างโปรเจกต์



17. จะปรากฏโปรเจกต์ที่ Package Explorer ซึ่งแสดงถึงโครงสร้างของโปรเจกต์ที่ได้สร้างขึ้นมา



ขั้นตอนต่อไปจะเป็นการแก้ไขไฟล์หลักๆ ซึ่งในบทนี้จะมี 3 ไฟล์ดังนี้

- ไฟล์ main.xml ซึ่งจะใช้เป็นไฟล์สำหรับการออกแบบหน้าจอที่อยู่บนแอพพลิเคชัน
- ไฟล์ string.xml ซึ่งเป็นไฟล์ที่สามารถกำหนดรายละเอียดของแอ็ตทริบิวต์ (Attribute) ซึ่งอยู่ที่ไดเรกทอรี “res/values/string.xml” โดยไฟล์นี้จะถูกใช้สำหรับอ้างอิงรายละเอียดต่างๆ จากไฟล์อื่นๆ ด้วย เช่น สามารถเรียกใช้งานค่าต่างๆ ผ่านไฟล์ main.xml ได้
- ไฟล์ ChangeBath.java ซึ่งเป็นไฟล์หลักที่ใช้สำหรับกำหนดการทำงานของแอพพลิเคชัน

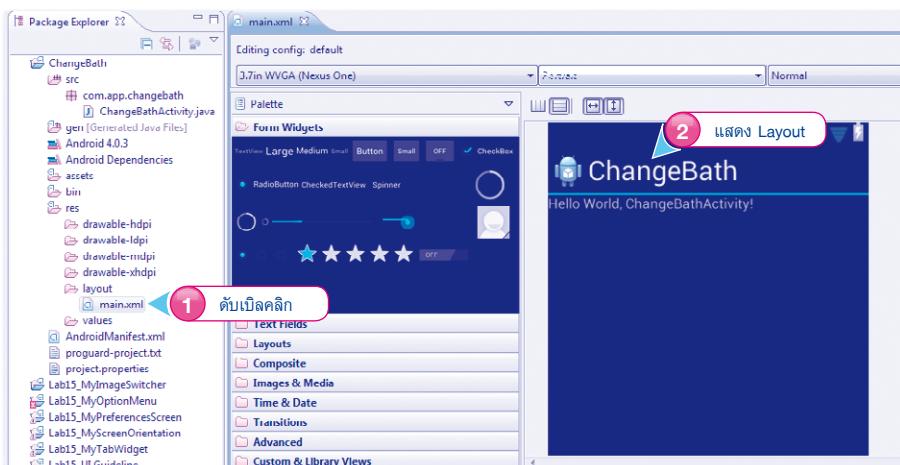
## ออกแบบหน้าจอแอพพลิเคชัน

การออกแบบหน้าจอแอพพลิเคชันเป็นการแก้ไขไฟล์ XML สำหรับแอพพลิเคชันการแปลงค่าระยะทางนั้น จะต้องแก้ไขไฟล์ 2 ไฟล์ย่อยๆ คือ ไฟล์ main.xml และไฟล์ string.xml

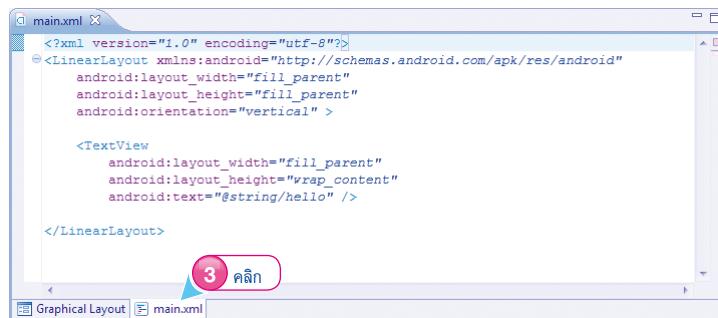
### การแก้ไขไฟล์ main.xml

การแก้ไขไฟล์ XML (Extensible Markup Language) ซึ่งส่วนของแอพพลิเคชัน ChangeBath นี้ไฟล์ main.xml จะเป็นไฟล์หลักสำหรับการออกแบบหน้าจอของแอพพลิเคชัน ซึ่งมีขั้นตอนการแก้ไขดังนี้

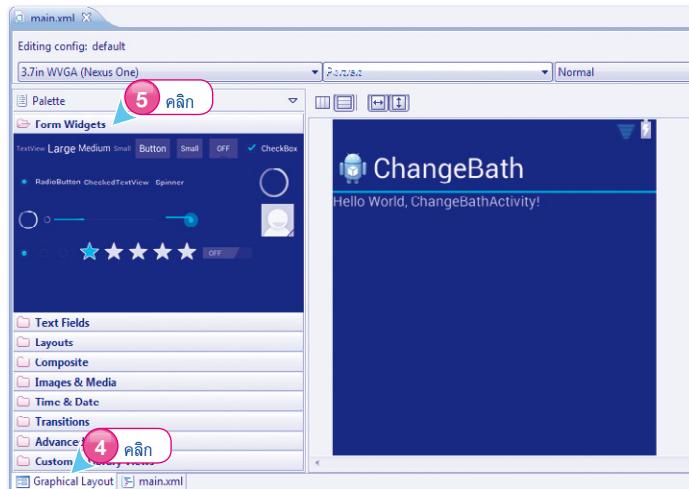
- ดับเบิลคลิกไฟล์ main.xml ที่อยู่ในส่วนของ res/layout
- จะปรากฏ Layout ทางด้านขวาเมื่อ



3. ผู้อ่านสามารถลับจาก群แบบ Layout เป็นแบบ XML โดยคลิกแท็บ main.xml ซึ่งมีโค้ด xml ดังนี้



4. ในส่วนของการออกแบบแอพพลิเคชัน ให้ผู้อ่านคลิกแท็บ Graphical Layout  
5. คลิกที่ Form Widgets  
6. คลิกเลือกอิลิเมนต์ต่างๆ เพื่อออกแบบแอพพลิเคชันดังรูป



## 7. คลิกแท็บ main.xml จะได้โค้ด XML ดังนี้

```

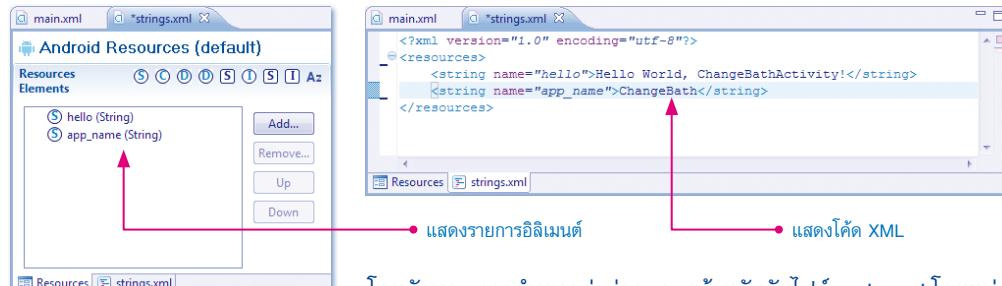
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="@string/hello" />
    <EditText
        android:id="@+id/editText1" android:layout_width="match_parent"
        android:layout_height="wrap_content" android:ems="10" />
    </EditText>
    <RadioButton
        android:id="@+id/radioButton1" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="RadioButton" />
    <RadioButton
        android:id="@+id/radioButton2" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="RadioButton" />
    <Button
        android:id="@+id/button1" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Button" />
</LinearLayout>

```

- กำหนดคุณสมบัติ EditText
- กำหนดคุณสมบัติ RadioButton ตัวที่ 1
- กำหนดคุณสมบัติ RadioButton ตัวที่ 2
- กำหนดคุณสมบัติ Button

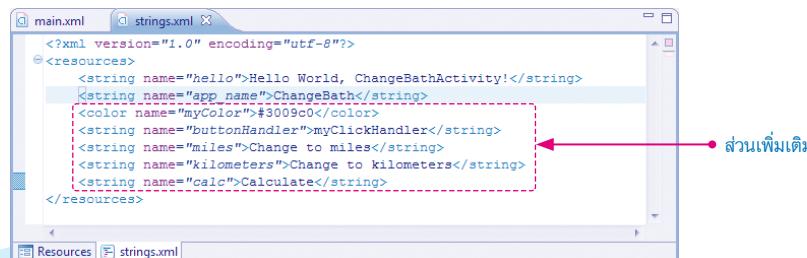
## การแก้ไขไฟล์ string.xml

ในส่วนนี้เป็นส่วนเพิ่มเติมสำหรับช่วยในการอ้างอิงถึงรายละเอียดต่างๆ ของแอ็ตทริบิวต์ (Attribute) ให้มีความสะดวกในการเรียกใช้งาน โดยเฉพาะเมื่อมีการเรียกใช้จากไฟล์ main.xml ซึ่งไฟล์ string.xml จะอยู่ที่ไดเร็คทอรี “res/values/string.xml”

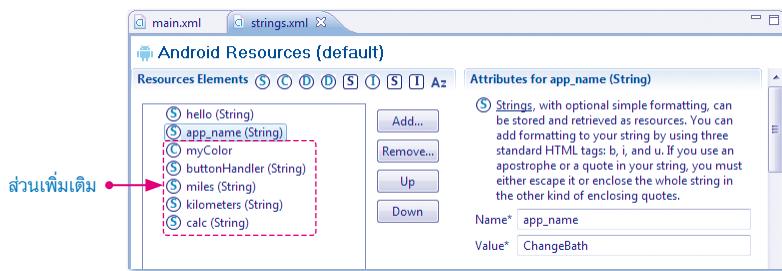


โดยลักษณะการกำหนดค่าต่างๆ จะคล้ายกันกับไฟล์ main.xml โดยแบ่งออกเป็น 2 แบบคือ แบบ XML และแบบ Layout

ซึ่งตัวอย่างในบทนี้จะเป็นการเพิ่มเติมค่า myColor, Change to miles, Change to kilometers และ Calculate ซึ่งระบุเป็นค่าสตริง (String) ไว้ดังนี้



## จากการเพิ่มโค้ด XML ดังกล่าวได้ผลลัพธ์ดังนี้



จากนั้นเมื่อกำหนณค่าต่างๆ ในไฟล์ string.xml เรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการแก้ไขไฟล์ main.xml ที่จะเรียกใช้งานแอ็ตทริบิวต์ต่างๆ โดยแก้ไขไฟล์ main.xml ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/myColor">
```

```
    <EditText
        android:id="@+id/EditText01"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:inputType="numberSigned|numberDecimal">
    </EditText>
```

1 กำหนดคุณสมบัติ EditText

```
    <RadioGroup
        android:id="@+id/RadioGroup01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
```

```
        <RadioButton
            android:id="@+id/RadioButton01"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/kilometers"
            android:checked="true">
        </RadioButton>
```

2 กำหนดคุณสมบัติ RadioButton

```
        <RadioButton
            android:id="@+id/RadioButton02"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/miles">
        </RadioButton>
```

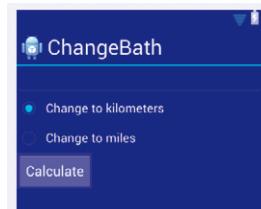
```
    </RadioGroup>
```

```
    <Button
        android:id="@+id/Button01"
        android:layout_height="wrap_content"
        android:onClick="@string/buttonHandler"
        android:layout_width="wrap_content"
        android:text="@string/calc">
    </Button>
```

3 กำหนดคุณสมบัติ Button

```
</LinearLayout>
```

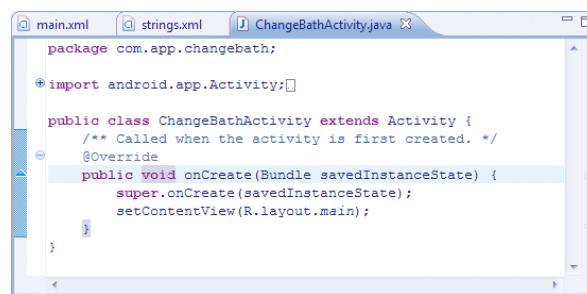
จากการเพิ่มโค้ด XML ดังกล่าวได้ผลลัพธ์ดังนี้



## เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน

เมื่อผู้อ่านสร้างหน้าจอแอพพลิเคชันเรียบร้อยแล้ว ขั้นตอนต่อไปคือ การเขียนโค้ดกำหนดการทำงานของแอพพลิเคชันในไฟล์ ChangeBathActivity.java เพื่อใช้ในการแปลงค่าระยะทาง โดยมีขั้นตอนดังต่อไปนี้

1. เปิดไฟล์ ChangeBathActivity.java ซึ่งมีโค้ดดังนี้



2. แก้ไขโค้ดในไฟล์ ChangeBathActivity.java ดังนี้

```
package com.app.changebath;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Toast;

public class ChangeBath extends Activity {
    private EditText text;

    /** Called when the activity is first created. */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        text = (EditText) findViewById(R.id.EditText01);
    }

    public void myClickHandler(View view) { ← 1
        switch (view.getId()) { ← 2
            case R.id.Button01: ← 3 กรณีกดปุ่ม
                RadioButton milesButton = (RadioButton)
                    findViewById(R.id.RadioButton01);
```

```

        RadioButton kilometersButton = (RadioButton)
            findViewById(R.id.RadioButton02);
        if (text.getText().length() == 0) {
            Toast.makeText(this,
                "Invalid Number",
                Toast.LENGTH_LONG).show();
        }
        return;
    }

    float inputValue = Float.parseFloat
        (text.getText().toString());
    if (milesButton.isChecked()) {
        text.setText(String.valueOf
            (convertToKilometers(inputValue)));
    } else {
        text.setText(String.valueOf
            (convertToMiles(inputValue)));
    }
    // Switch to the other button
    if (milesButton.isChecked()) {
        milesButton.setChecked(false);
        kilometersButton.setChecked(true);
    } else {
        milesButton.setChecked(true);
        kilometersButton.setChecked(false);
    }
    break;
}
}

private float convertToKilometers(float miles) { ← 2
    return (float) (miles * 1.609344);
}

private float convertToMiles(float kilometers) {
    return (float) (kilometers * 0.621371192);
}
}

```

## อธิบายโค้ดการทำงานของแอพพลิเคชัน

- ฟังก์ชัน myClickHandler เป็นฟังก์ชันกำหนดคุณลักษณะ รวมไปถึงขั้นตอนการทำงานต่างๆ ที่จะเกิดขึ้นเมื่อมีการคลิกปุ่ม Button ซึ่งมาเด็ดตั้งนี้

```

public void myClickHandler(View view) {
    switch (view.getId()) {
        case R.id.button1:
            RadioButton milesButton = (RadioButton)
                findViewById(R.id.radioButton1);
            RadioButton kilometersButton = (RadioButton)
                findViewById(R.id.radioButton2);
            if (text.getText().length() == 0) {
                Toast.makeText(this,
                    "Invalid Number",
                    Toast.LENGTH_LONG).show();
            }
            return;
    }
}

```

```

        float inputValue = Float.parseFloat
            (text.getText().toString());
        if (milesButton.isChecked()) {
            text.setText(String.valueOf
                (convertToKilometers(inputValue)));
        } else {
            text.setText(String.valueOf
                (convertToMiles(inputValue)));
        }
        // Switch to the other button
        if (milesButton.isChecked()) {
            milesButton.setChecked(false);
            kilometersButton.setChecked(true);
        } else {
            milesButton.setChecked(true);
            kilometersButton.setChecked(false);
        }
    }
}

```

จะลังเก็ตได้ว่าแอพพลิเคชันมีการตรวจสอบค่า RadioButton เพื่อเปลี่ยนแปลงค่าจากไมล์ (Miles) เป็นกิโลเมตร (Kilometers) และในทางกลับกันคือ เปลี่ยนแปลงค่าจากกิโลเมตร (Kilometers) เป็นไมล์ (Miles) จากนั้นเงื่อนไข milesButton.isChecked() จะถูกใช้งาน เพื่อเรียกฟังก์ชันในการแปลงค่าระหว่างทั้งให้ทำงาน

- ฟังก์ชันแปลงค่าจะทำงาน เมื่อมีการคลิกปุ่ม โดยที่การแปลงค่านั้นจะอ้างอิงจากการคำนวนค่า 1 ไมล์ = 1.609344 กิโลเมตร และ 1 กิโลเมตร = 0.621371192 ไมล์ ซึ่งมี 2 ฟังก์ชันดังนี้

```

private float convertToKilometers(float miles) {
    return (float) (miles * 1.609344);
}

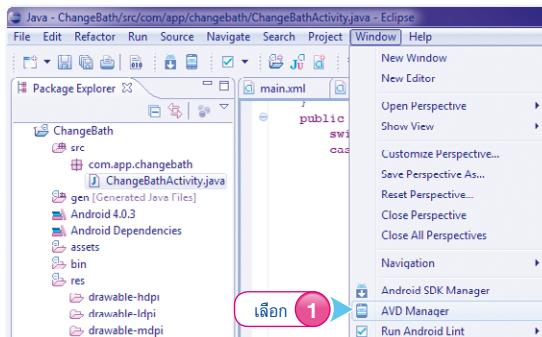
private float convertToMiles(float kilometers) {
    return (float) (kilometers * 0.621371192);
}

```

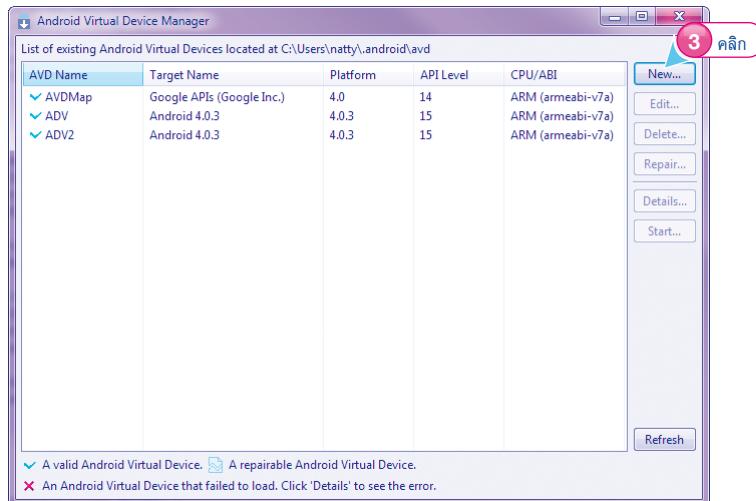
## ทดสอบการทำงานของแอพพลิเคชันผ่านໂຕຣສັບກຈໍາລວງ (Emulator)

หลังจากการพัฒนาแอพพลิเคชันเสร็จสิ้นแล้ว ขั้นตอนต่อไปคือ การคอมไพล์และรันแอพพลิเคชัน ทั้งนี้เนื่องจากแอพพลิเคชันจะรันโปรแกรมผ่าน Emulator ดังนั้น จะมีขั้นตอนต่อๆ ดังต่อไปนี้

- สร้าง AVD ใหม่ขึ้นมา โดยคลิกเมนู Window > AVD Manager



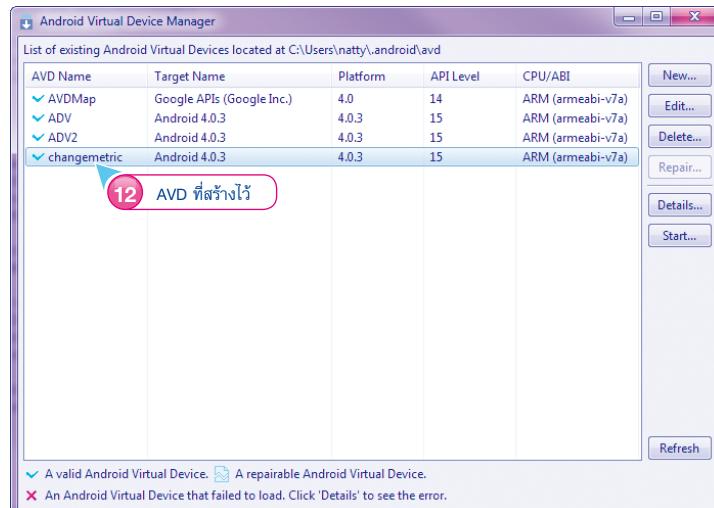
- จะปรากฏ dialogue ออกบอกรหัส Android Virtual Device Manager ขึ้นมา
- คลิกปุ่ม **New...** เพื่อสร้าง AVD



- จะปรากฏ dialogue ออกบอกรหัส Create new Android Virtual Device (AVD) เพื่อให้กำหนดคุณสมบัติ ต่างๆ ของ AVD
- ตั้งชื่อ AVD ในที่นี้ชื่อว่า changemetric
- ในส่วนของ Target: คลิกเลือก Android 4.0.3 ทั้งนี้ผู้อ่านสามารถเลือกรุ่นอื่นๆ ได้
- ในส่วนของ SD Card: คลิกเลือก Size:
- กรอกความจุของหน่วยความจำ ในที่นี้กรอกขนาด 1024 MiB (เมกะไบต์)
- ในส่วนของ Skin: คลิกเลือก Built-in:
- คลิกเลือก HVGA ย่อมาจาก Half-size Video Graphics Array ซึ่งเป็นการเลือกขนาดหน้าจอแสดงผลของแอพพลิเคชัน ซึ่งผู้อ่านควรเลือกตามขนาดของหน้าจอโทรศัพท์ที่ใช้งาน
- คลิกปุ่ม **Create AVD** เพื่อสร้าง AVD



12. จากนั้นจะมีชื่อ AVD ที่สร้างไว้แสดงขึ้นมาในรายการ



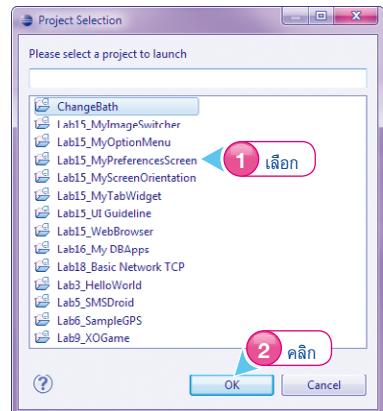
13. คลิกขวาที่โปรเจกต์ แล้วเลือกคำสั่ง Run As > Run Configurations...



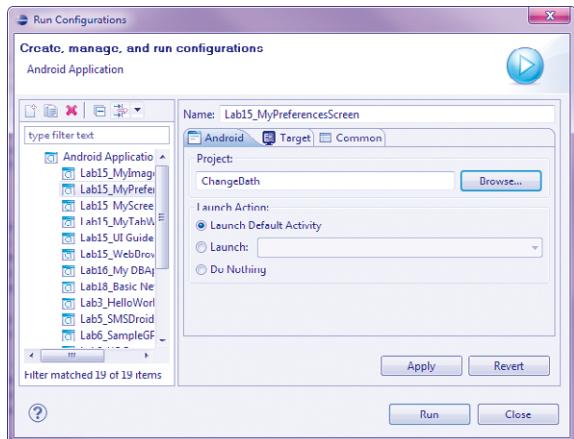
## NOTE



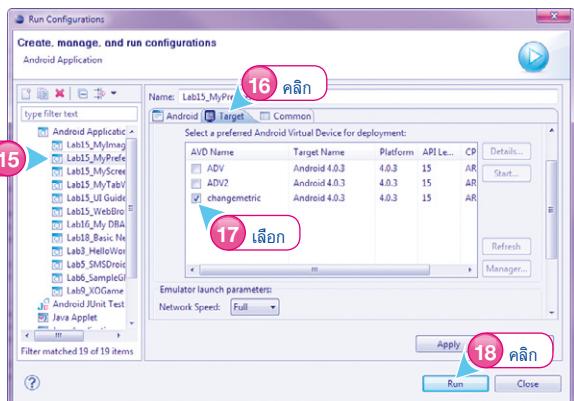
ในกรณีที่ไม่มีชื่อโปรเจกต์ ให้คลิกปุ่ม **Browse...**  
เพื่อเลือกโปรเจกต์ ChangeBath และคลิกปุ่ม **OK**



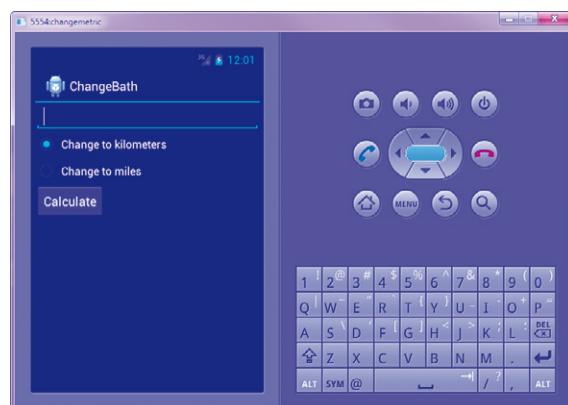
## จะปรากฏโปรเจกต์ดังนี้



14. จะปรากฏได้อะล็อกบัญชี Run Configurations เพื่อให้กำหนดค่าต่างๆ สำหรับการรันแอพพลิเคชัน
15. ในส่วนของ Android Application คลิกเลือกแอพพลิเคชันที่ต้องการรัน
16. คลิกแท็บ Target
17. คลิกเลือก AVD ที่จะใช้ สำหรับรัน
18. คลิกปุ่ม **Run**



19. รอลักคู่รู้ ก็จะปรากฏหน้าจอของ Emulator และแอพพลิเคชัน ChangeBath ก็จะถูกโหลดขึ้นมาโดยอัตโนมัติ

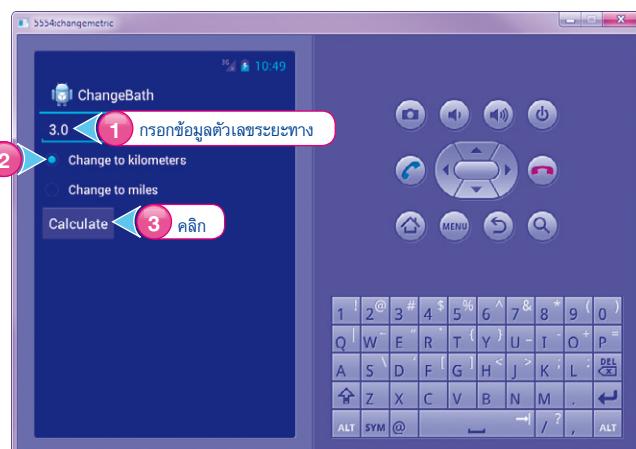


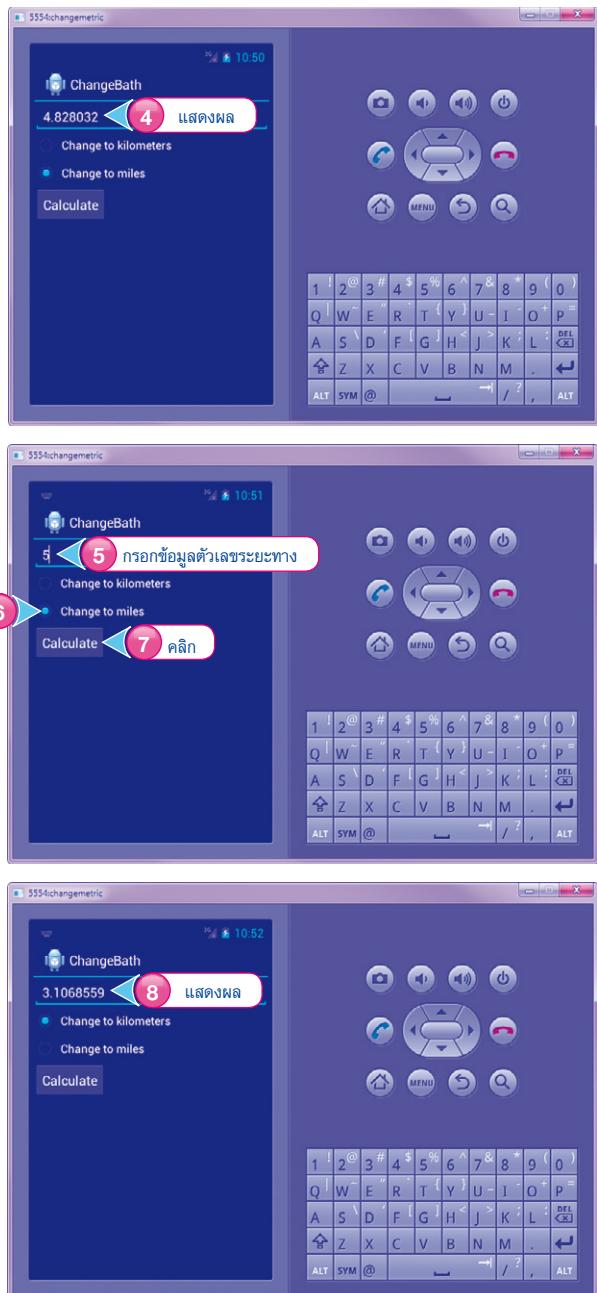
#### NOTE



ในกรณีที่โทรศัพท์ถูกล็อคไว้ ผู้อ่านสามารถเปิด Emulator โดยหากบุ่มล็อค Emulator จากทางซ้ายมือไปยังทางขวาเมื่อ และเรียกใช้แอพพลิเคชัน Change miles to kilometers ได้ ซึ่งก็จะคลายกับการใช้งานโทรศัพท์จริงนั้นเอง

20. ทดสอบการทำงานของแอพพลิเคชัน ได้ผลลัพธ์ดังนี้





### NOTE



- 1 ไมล์ = 1.609344 กิโลเมตรร
- 1 กิโลเมตร = 0.621371192 ไมล์



## สรุปก้ายบต

ในบทนี้เป็นการทบทวนให้ผู้อ่านได้ศึกษาขั้นตอนการพัฒนาแอพพลิเคชัน และได้ลองเขียนโปรแกรม กำหนดการทำงานของแอพพลิเคชันจริงๆ ซึ่งถือเป็นพื้นฐานที่สำคัญในการพัฒนาแอพพลิเคชันในบทต่อไป

## แบบฝึกหัดก้ายบต

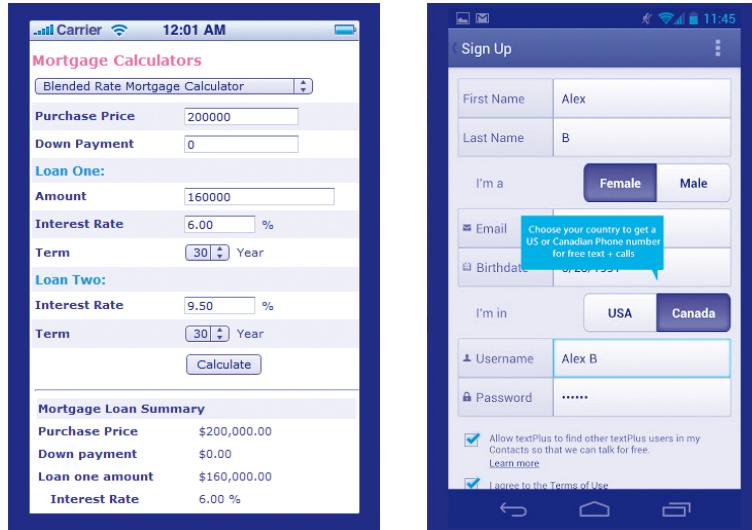
หลังจากผู้อ่านทดลองพัฒนาแอพพลิเคชัน เปลี่ยนค่าระยะทางจากไมล์เป็นกิโลเมตร และในทำนองกลับกันเรียบร้อยแล้ว แบบฝึกหัดในส่วนนี้ให้ผู้อ่านพัฒนาแอพพลิเคชันดังนี้

- พัฒนาแอพพลิเคชันสำหรับเปลี่ยนค่าเงินจากค่าเงินสกุลบาท (฿) เป็นค่าเงินสกุลдолลาร์ (\$) และให้สามารถแปลงกลับกันได้
- พัฒนาแอพพลิเคชันสำหรับเปลี่ยนค่าอุณหภูมิจากเซลเซียส (Celsius) เป็นฟาราเนไฮต์ (Fahrenheit) และให้สามารถแปลงกลับกันได้
- พัฒนาแอพพลิเคชันเครื่องคิดเลขขนาดเล็กอย่างง่าย ที่สามารถบวก ลบ คูณ และหารได้ โดยมีกีล่องข้อความ 2 กล่องสำหรับตัวเลข และปุ่มให้เลือกในการประมวลผล จากนั้นให้ผลลัพธ์แสดงออกทางหน้าจอตามลำดับ

# การสร้างส่วนติดต่อกับผู้ใช้ (User Interface)

ในบทนี้มีจุดประสงค์เพื่อให้ผู้อ่านมีความคุ้นเคยกับล้วนต่อประสานกับผู้ใช้ หรือ UI (User Interface) สำหรับใช้ในการพัฒนาแอ�플ิเคชัน

ยูไอ (UI—User Interface) หรือส่วนต่อประสานกับผู้ใช้คือ ลิ้งที่มีไว้ให้ผู้ใช้งานทำหรือปฏิสัมพันธ์กับระบบหรือลิ้งของต่างๆ ของระบบ ซึ่งอาจจะเป็นคอมพิวเตอร์ เครื่องจักรกล อุปกรณ์ໃเชิฟฟ้า เป็นต้น ทั้งนี้มีจุดประสงค์เพื่อให้ลิ้งฯ นั้น สามารถทำงานได้ตามความต้องการของผู้ใช้งานนั่นเอง



▲ ตัวอย่างแอ�플ิเคชันที่ใช้งาน UI—User Interface

ในหัวข้อนี้ผู้เขียนจะออกแบบแอพพลิเคชันอย่างง่าย เพื่อใช้ในการอธิบายการใช้งานวิวัชnidต่างๆ ที่มีความจำเป็นในการพัฒนาแอพพลิเคชันเบื้องต้น ได้แก่ TextView, EditText, ImageView, Button และ ImageButton ซึ่งผู้เขียนจะยังไม่ขอกล่าวถึงการเขียนโปรแกรมที่ซับซ้อนแต่อย่างใด โดยมีหน้าตาของโปรแกรมดังนี้



หน้าจอของแอพพลิเคชันที่จะพัฒนาในบทนี้ ▶

### NOTE

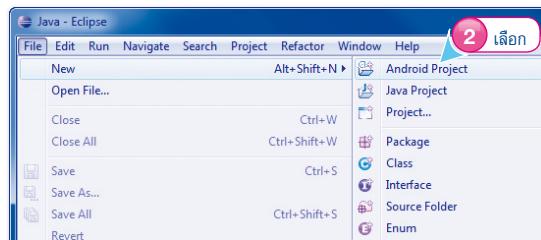


**วิว (View)** คือ ส่วนประกอบต่างๆ ของหน้าจอที่ใช้สำหรับสร้างส่วนเชื่อมต่อกับผู้ใช้งาน เช่น ส่วนที่ผู้อ่านใช้งาน Button และ EditText ในบทที่ผ่านมา เป็นต้น

## เริ่มต้นสร้างโปรเจกต์

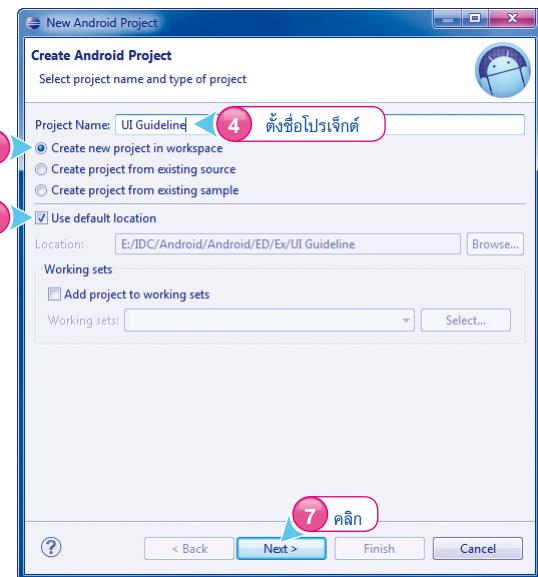
ขั้นตอนแรกในการพัฒนาแอพพลิเคชันทุกครั้งก็คือ การสร้างโปรเจกต์ ซึ่งผู้อ่านสามารถทำได้ตามขั้นตอนดังต่อไปนี้

1. เปิดโปรแกรม Eclipse
2. คลิกเมนู File > New > Android Project



3. จะปรากฏได้ล็อกมือกซ์ New Android Project เพื่อกำหนดรายละเอียดต่างๆ ของโปรเจกต์
4. ตั้งชื่อโปรเจกต์ ในที่นี้ตั้งชื่อว่า UI Guideline
5. ในส่วนของ Contents คลิกเลือก Create new project in workspace เพื่อสร้างโปรเจกต์ใหม่
6. คลิกเครื่องหมายถูกหน้า Use default location เพื่อพัฒนาแอพพลิเคชันในตำแหน่งที่อยู่เดียวกัน กับโปรเจกต์ Eclipse

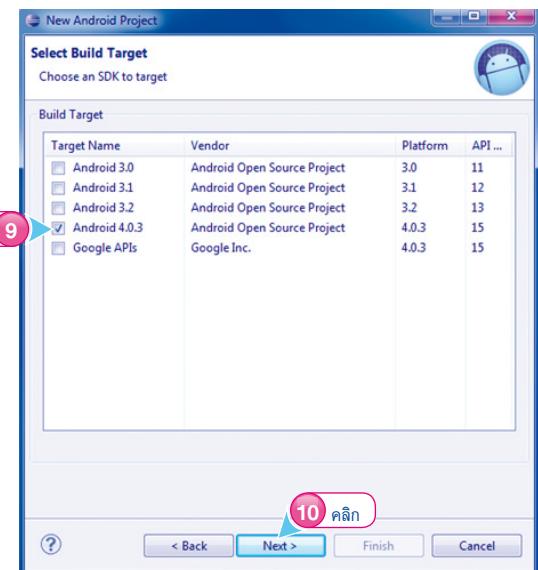
7. คลิกปุ่ม **Next >**



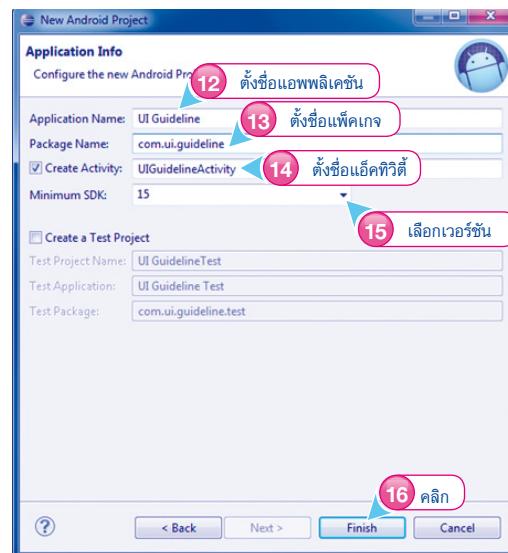
8. จะปรากฏ dialogue ให้อະລັກນີ້ອັກສ່ New Android Project ในສ່ວນທີ່ 2 ລຳຮັບເລືອກເວຼັອຮັບຂັ້ນຂອງ Android

9. คลິກເລືອກເວຼັອຮັບຂັ້ນຂອງ Android ທີ່ຈະພົມນາ ໃນທີ່ນີ້ເລືອກ 4.0.3

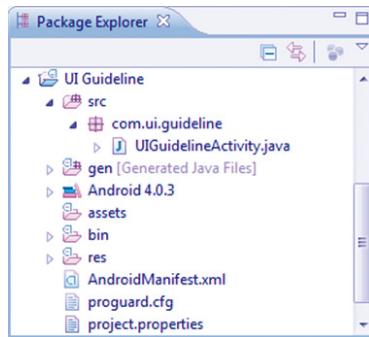
10. คลິກປຸ່ມ **Next >**



11. จะปรากฏ dialogue ตั้งชื่อ New Android Project ในส่วนที่ 3 สำหรับกำหนดรายละเอียดของแอพพลิเคชัน
12. ตั้งชื่อของแอพพลิเคชันคือ UI Guideline
13. ชื่อแพ็คเกจจะอยู่ในรูปแบบของ xxx.xxxx... ซึ่งในที่นี้ตั้งชื่อแพ็คเกจเป็น com.ui.guideline
14. ตั้งชื่อ Activity ในที่นี้ตั้งชื่อเป็น UIGuidelineActivity
15. ในส่วนของ Minimum SDK: คลิกเลือกเวอร์ชัน 15 ซึ่งเป็นเวอร์ชันเดียวกันกับ API Level ที่เลือกในข้อ 9
16. คลิกปุ่ม **Finish** ซึ่งถือเป็นอันเสร็จสิ้นการสร้างโปรเจกต์



17. จะปรากฏโปรเจกต์ที่ Package Explorer ซึ่งแสดงถึงโครงสร้างของโปรเจกต์ที่ได้สร้างขึ้นมา



หลังจากที่สร้างโปรเจกต์เสร็จลื้นแล้ว ไฟล์ UIGuidelineActivity.java จะเป็นไฟล์หลักที่ใช้สำหรับการเขียนโค้ดในแอพพลิเคชันนี้ และไฟล์ main.xml นั้นมีไว้เพื่อใช้ในการออกแบบ UI ซึ่งใช้ในการกำหนดโครงสร้างของหน้าจอบนโทรศัพท์เคลื่อนที่และแท็บเล็ต

# TextView : แสดงข้อความ

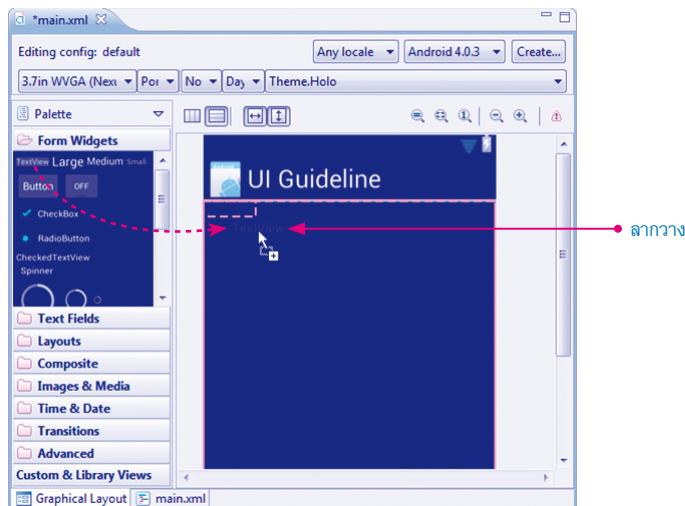
TextView เป็นอิลิเม้นต์ที่ทำหน้าที่แสดงข้อความ และผู้อ่านสามารถสับเปลี่ยนไปที่โหมด XML เพื่อกำหนดคุณสมบัติของ TextView ได้ นอกจากนี้ผู้อ่านยังสามารถที่จะเปลี่ยนแบบอักษร (Font) ได้ตามต้องการ อีกด้วย

ตัวอย่างแอพพลิเคชันที่ใช้งาน TextView ▶



## การกำหนดคุณสมบัติ TextView

ให้ผู้อ่านเพิ่มอิลิเม้นต์ TextView ลงในหน้าจอแอพพลิเคชันด้วยวิธีลากวางดังนี้



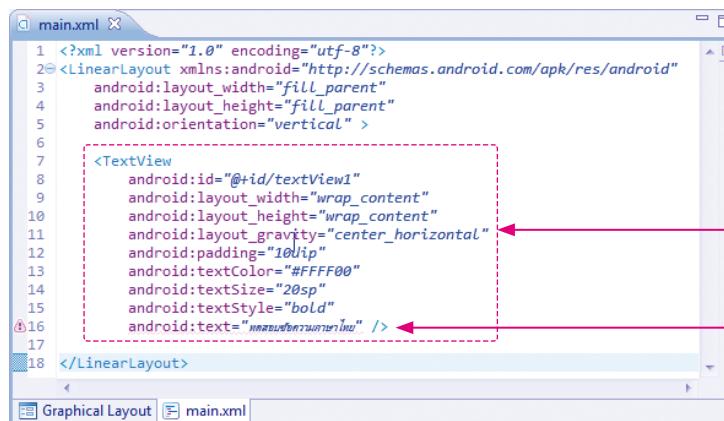
จะได้โค้ด XML ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView" />
</LinearLayout>
```

ผู้อ่านสามารถกำหนดคุณสมบัติอื่นๆ เพิ่มเติมให้อลิเม้นต์ TextView ได้ โดยเพิ่มแอ็ตทริบิวต์เข้าไปด้วยโค้ด XML ยกตัวอย่างเช่น

- android:textColor กำหนดสีของข้อความ เช่น "#FFFFFF"
- android:textSize กำหนดขนาดตัวอักษร เช่น "20sp"
- android:gravity กำหนดการจัดวางแนวของข้อความ เช่น "Top"
- android:layout\_width กำหนดความกว้างของกล่องข้อความ เช่น "100dp"
- android:layout\_height="wrap\_content" กำหนดความสูงของกล่องข้อความ เช่น "120dp"

จะได้โค้ด XML ดังนี้



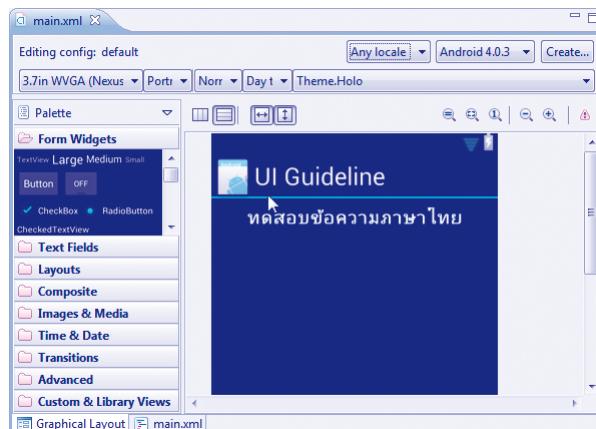
```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical" >
6
7     <TextView
8         android:id="@+id/textView1"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:layout_gravity="center_horizontal"
12        android:padding="10dp"
13        android:textColor="#FFFF00"
14        android:textSize="20sp"
15        android:textStyle="bold"
16        android:text="ทดสอบความภาษาไทย" />
17
18 </LinearLayout>

```

กำหนดคุณสมบัติของ TextView  
สามารถใช้ภาษาไทยได้

เมื่อผู้อ่านกลับมาอ่านหน้าแสดงผล จะได้หน้าจอแอพพลิเคชันดังนี้

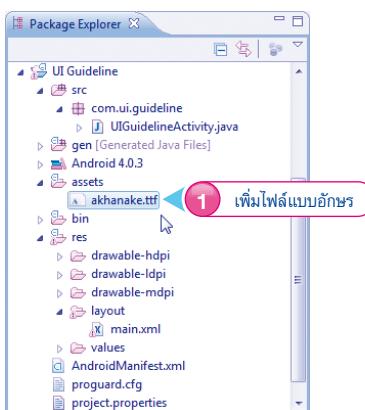


## การเปลี่ยนแบบอักษรสำหรับ TextView

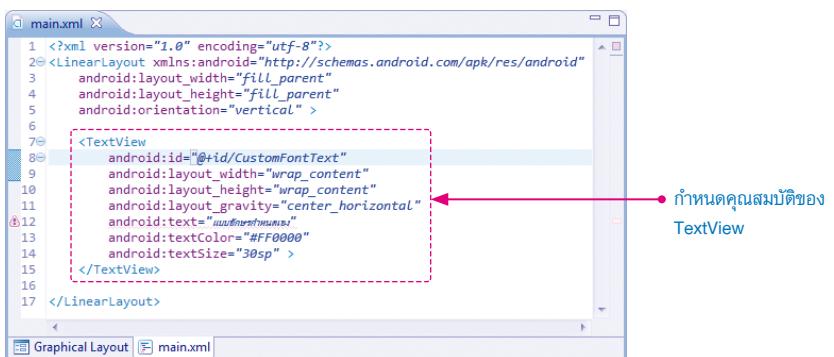
โดยปกติแล้วแบบอักษร (Font) มาตรฐานของแอนดรอยด์ไม่ค่อยจะมีความสวยงามมากนัก ดังนั้น เพื่อให้แอพพลิเคชันมีความน่าสนใจ จึงจำเป็นต้องใช้แบบอักษรที่สวยงาม ทั้งนี้ผู้อ่านไม่สามารถเปลี่ยนแบบอักษรได้ด้วยโค้ดในไฟล์ XML โดยตรง

สำหรับขั้นตอนการเปลี่ยนแบบอักษรนี้มีดังต่อไปนี้

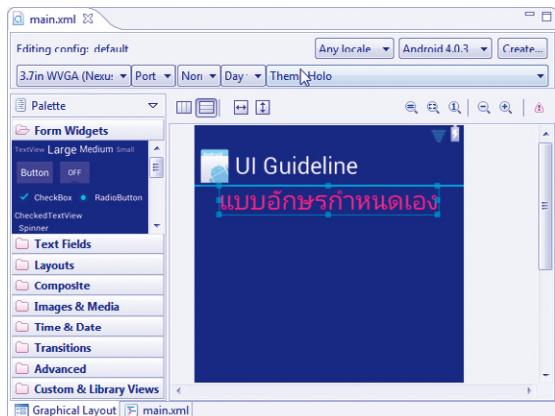
1. คัดลอกแบบอักษรที่ต้องการไปเก็บไว้ในส่วนของ assets (ในที่นี่ใช้ไฟล์แบบอักษรชื่อ akhanake.ttf)



2. เพิ่มอิลิเม้นต์ TextView ในหน้าจอ และกำหนดคุณสมบัติดังนี้ (ในที่นี่กำหนด id เป็น CustomFontText)



ได้ผลลัพธ์ดังนี้



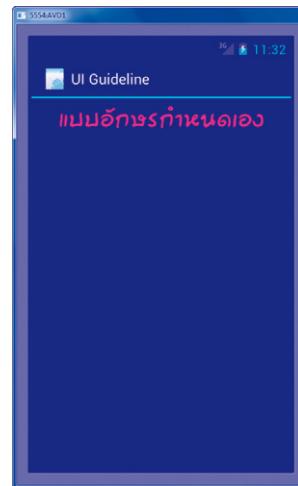
3. แก้ไขโค้ดโปรแกรมในไฟล์ Java ในนี่คือไฟล์ `UIGuidelineActivity.java` เพื่อกำหนดแบบอักษรให้กับอิลิเม้นต์ `TextView` ดังนี้

```

1 package com.ui.guideline;
2
3 import android.app.Activity;
4 import android.graphics.Typeface;
5 import android.os.Bundle;
6 import android.widget.TextView;
7
8 public class UIGuidelineActivity extends Activity {
9     /** Called when the activity is first created. */
10    @Override
11    public void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.main);
14
15        Typeface tf = Typeface.createFromAsset(getAssets(),
16            "akhanake.ttf");
17        TextView tv = (TextView) findViewById(R.id.CustomFontText);
18        tv.setTypeface(tf);
19    }
20 }

```

4. รันแอพพลิเคชันเพื่อทดสอบการทำงาน จะได้ผลลัพธ์ดังรูป



นอกจากอิลิเม้นต์ `TextView` แล้ว ผู้อ่านยังสามารถเรียกใช้งาน เมธอด `setTypeface()` เพื่อกำหนดแบบอักษรของอิลิเม้นต์ `EditText` และ `Button` ได้เช่นเดียวกัน

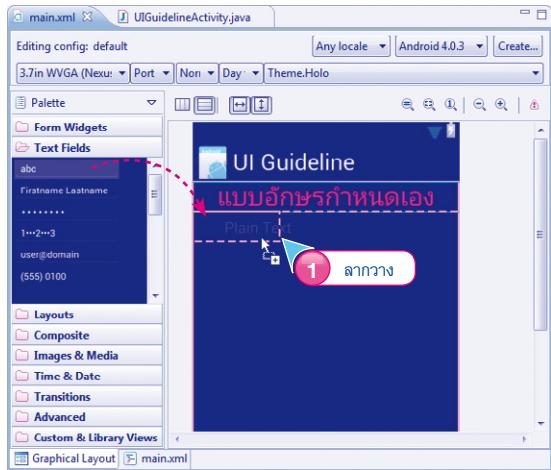
## EditText : รับข้อความ

`EditText` เป็นอิลิเม้นต์ที่ทำหน้าที่รับข้อความจากผู้ใช้เพื่อนำไปใช้งาน ซึ่งการใช้งาน `EditText` มีขั้นตอนดังต่อไปนี้



ตัวอย่างแอพพลิเคชันที่ใช้งาน `EditText` ▶

## 1. เพิ่มอัลิเมนต์ EditText จาก Text Fields ลงในหน้าจอด้วยวิธีการลากวาง



2. ผู้อ่านสามารถกำหนดคุณสมบัติของอัลิเมนต์ EditText โดยการเพิ่มแอ็ตทริบิวต์ต่างๆ ได้ เช่นเดียวกับการกำหนดคุณสมบัติของอัลิเมนต์ TextView เช่น กำหนดขนาดตัวอักษร กำหนดสีตัวอักษร เป็นต้น ซึ่งมีโค้ด XML ดังนี้

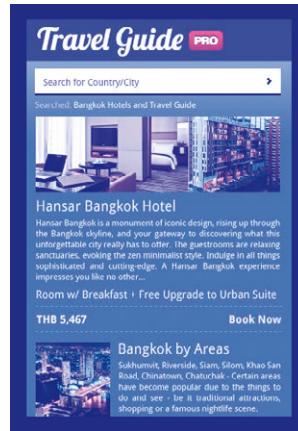
```
14     android:textSize="30sp" >
15   </TextView>
16
17   <EditText
18     android:id="@+id/editText1"
19     android:layout_width="match_parent"
20     android:layout_height="wrap_content" >
21
22     <requestFocus />
23
24   </EditText>
25
26 </LinearLayout>
```

3. แก้ไขโค้ดในไฟล์ Java เพื่อใช้งานข้อมูลจากอัลิเมนต์ EditText โดยอ้างอิงไปยังอัลิเมนต์ EditText โดยใช้ id และเรียกใช้เมธอด getText() ดังโค้ดต่อไปนี้

```
EditText edittext1 = (EditText) findViewById(R.id.editText1);
String text = edittext1.getText().toString();
```

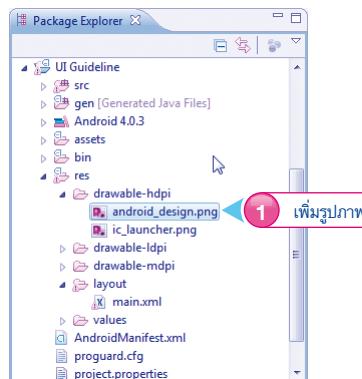
# ImageView : ॥ส่งรูปภาพ

ImageView เป็นวิวที่ทำหน้าที่แสดงรูปภาพบนหน้าจอ ซึ่งมีขั้นตอนการใช้งานดังต่อไปนี้

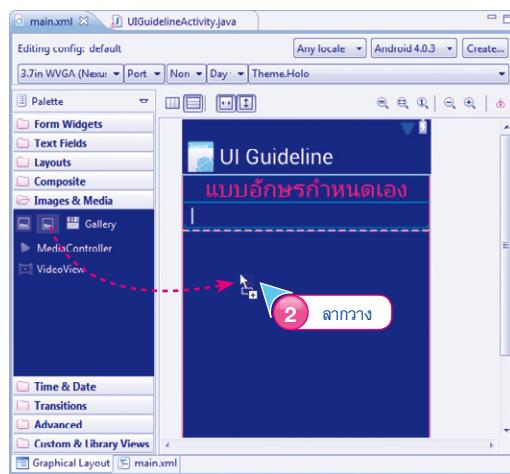


▲ ตัวอย่างแอพพลิเคชันที่ใช้งาน ImageView

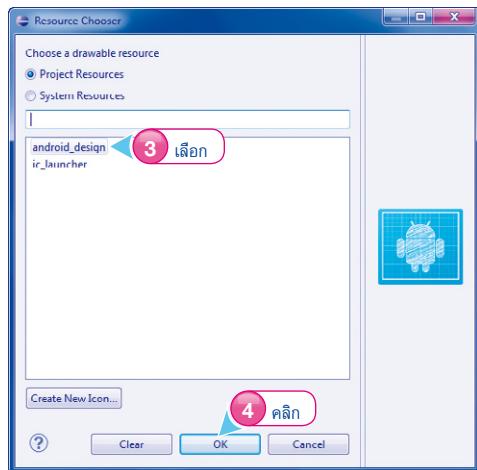
- คัดลอกรูปภาพที่ต้องการเข้ามาไว้ในส่วนของ res/drawable-hdpi ในที่นี่สามารถลากรูปที่ต้องการมาวางในโฟลเดอร์ได้เลย



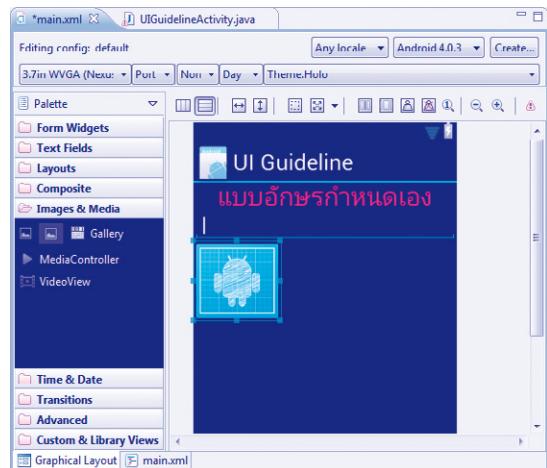
- เพิ่ม ImageView ด้วยวิธีการลากวางจาก Images & Media ลงบนหน้าจอ



3. จะปรากฏหน้าต่างสำหรับเลือกรูปภาพ ให้ผู้อ่านเลือกรูปภาพที่ต้องการ
4. คลิกปุ่ม **OK**



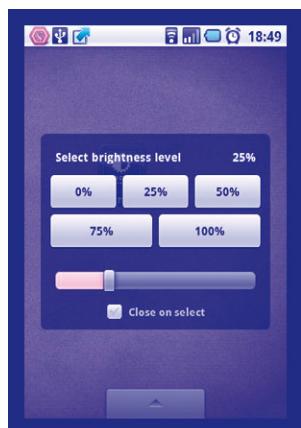
5. จะปรากฏหน้าจอแอพพลิเคชันตั้งรูป



ทั้งนี้ผู้อ่านสามารถกำหนดคุณลักษณะของ ImageView ด้วยโค้ด XML ได้เช่นกัน

## Button : ปุ่ม

Button เป็นอิลิเม้นต์ที่ทำหน้าที่แสดงปุ่มบนหน้าจอ ซึ่งผู้อ่านสามารถเพิ่มอิลิเม้นต์ Button ในหน้าจอได้โดยใช้วิธีการลากวางได้ เช่นเดียวกับ TextView, EditText และ ImageView ในที่นี้ผู้เขียนจะไม่อธิบาย การกำหนดคุณลักษณะต่างๆ ของอิลิเม้นต์ Button แต่จะอธิบายการเพิ่ม อีเวนต์ (Event) ให้กับอิลิเม้นต์ Button ซึ่งมีความสำคัญอย่างมากในการ พัฒนาแอพพลิเคชัน

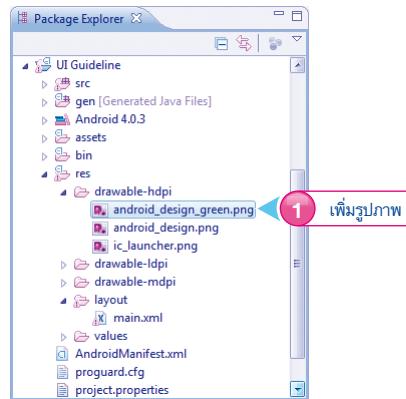


ตัวอย่างแอพพลิเคชันที่ใช้งาน Button ▶

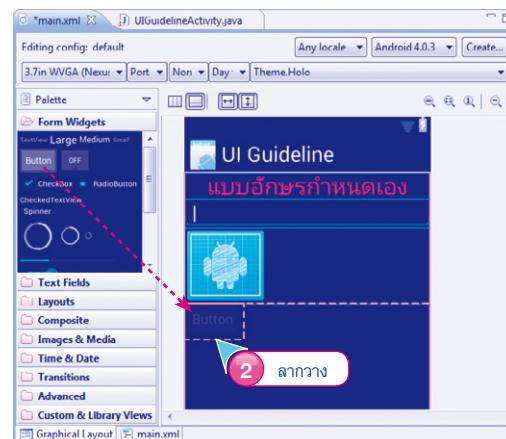
## การเพิ่มอีเวนต์ (Event) ให้ Button

ในที่นี้จะสร้างปุ่มหนึ่งปุ่มเพื่อใช้ในการเปลี่ยนรูปภาพของ ImageView ที่เพิ่มเข้าไปในหัวข้อที่ผ่านมา และเขียนโปรแกรมต่อจากหัวข้อดังกล่าว โดยมีขั้นตอนดังต่อไปนี้

- คัดลอกรูปภาพที่ต้องการเข้ามาไว้ในล้วนของ res/drawable-hdpi อีกหนึ่งรูปภาพ



- เพิ่มอิลิเม้นต์ Button ลงในหน้าจอโดยวิธีลากวาง และกำหนด id ในที่นี้กำหนดเป็น button1



ได้โค้ด XML ดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/android_design" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Switch Image" />
</LinearLayout>

```

3. เขียนโค้ดเพิ่มในไฟล์ Java ในนี่คือไฟล์ `UIGuidelineActivity.java` เพื่อเพิ่ม Event ให้กับปุ่มกด (`button1`) โดยกำหนดให้เปลี่ยนรูปภาพของ `ImageView` เมื่อคลิกปุ่ม ซึ่งมีโค้ดดังนี้

```
package com.ui.guideline;

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

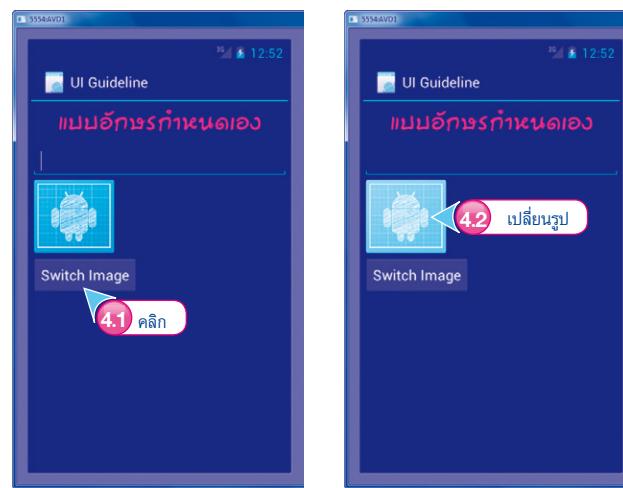
public class UIGuidelineActivity extends Activity {
    /** Called when the activity is first created. */
    int current_img;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Typeface tf = Typeface.createFromAsset(getAssets(), "akhanake.ttf");
        TextView tv = (TextView) findViewById(R.id.CustomFontText);
        tv.setTypeface(tf);

        Button button = (Button) findViewById(R.id.button1);
        current_img = R.drawable.android_design; ← กำหนดรูปภาพ
        button.setOnClickListener(new OnClickListener() { ← สร้าง Event Click
            public void onClick(View v) {
                ImageView my_android = (ImageView) findViewById(R.id.imageView1);

                if (current_img == R.drawable.android_design){ ← เปลี่ยนรูปภาพ
                    my_android.setImageDrawable(getResources().getDrawable(
                        R.drawable.android_design_green));
                    current_img = R.drawable.android_design_green;
                }else{
                    my_android.setImageDrawable(getResources().getDrawable(
                        R.drawable.android_design));
                    current_img = R.drawable.android_design;
                }
            }
        });
    }
}
```

- รันแอพพลิเคชันแล้วลองคลิกปุ่ม จะสังเกตเห็นได้ว่ารูปภาพเปลี่ยนจากรูปแอนดรอยด์สีน้ำเงิน เป็นสีเขียวดังรูป

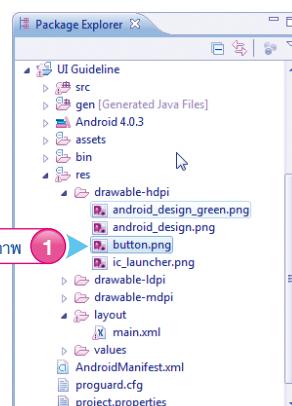
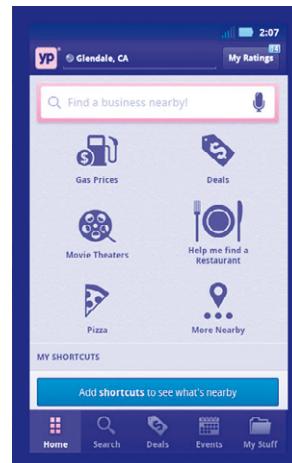


## ImageButton : บุ่มรูปภาพ

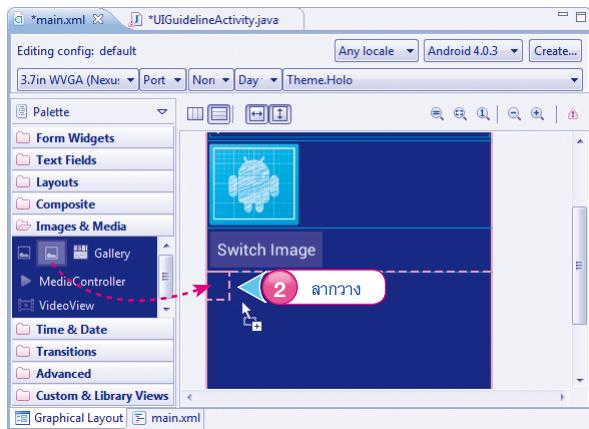
อิลิเมนต์ ImageButton จะแตกต่างจากอิลิเมนต์ Button คือ อิลิเมนต์ Button เป็นปุ่มข้อความ ส่วน อิลิเมนต์ ImageButton จะเป็นรูปภาพ ซึ่งการใช้งาน อิลิเมนต์ ImageButton สามารถทำได้เช่นเดียวกับ อิลิเมนต์อื่นๆ ดังนี้

ด้าอย่างแอพพลิเคชันที่ใช้งาน ImageButton ▶

- คัดลอกรูปภาพที่ต้องการใช้สร้างปุ่มเข้ามาไว้ในส่วนของ res/drawable-hdpi

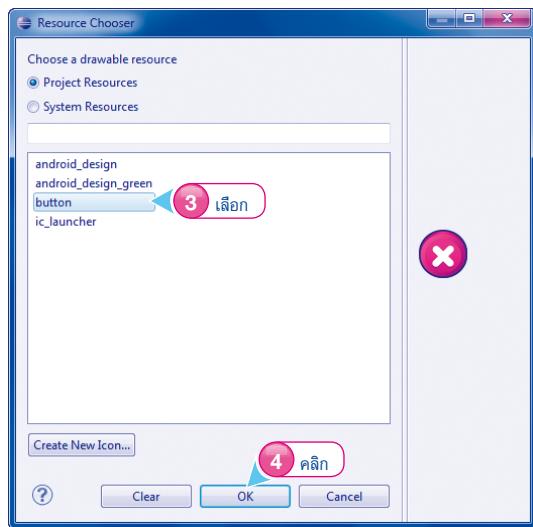


## 2. เพิ่มอัลิเมนต์ ImageButton ด้วยวิธีลากวาง

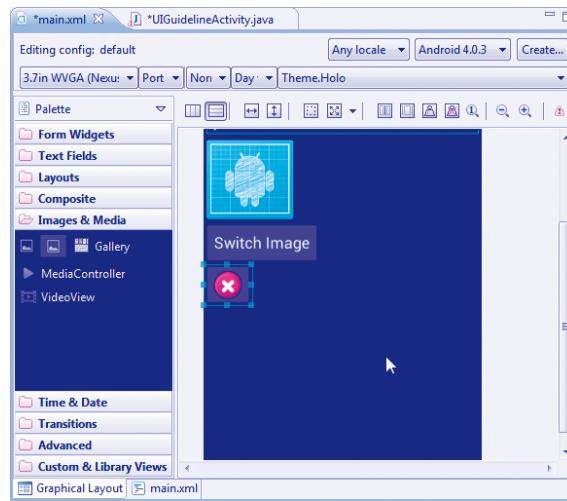


3. จะปรากฏหน้าต่างให้เลือกรูปสำหรับปุ่มกด ให้ผู้อ่านเลือกรูปภาพที่ต้องการ

4. คลิกปุ่ม **OK**



## 5. จะได้ผลลัพธ์ดังรูป



สำหรับการสร้างอีเวนต์ให้กับ ImageButton ก็สามารถทำได้เช่นเดียวกับ Button ในที่นี้ผู้เขียนจึงขอข้ามรายละเอียดในส่วนนี้ไป

## สรุปหัวยบท

ในบทนี้เป็นการอธิบายถึงการพัฒนาแอพพลิเคชัน เพื่อเชื่อมต่อกับส่วนติดต่อกับผู้ใช้งาน (UI–User Interface) ซึ่งถือเป็นส่วนสำคัญสำหรับให้ผู้ใช้งานสามารถใช้แอพพลิเคชันได้อย่างสะดวกนั่นเอง ซึ่งในบทนี้จะแนะนำอิลิเม้นต์เป็นแนวทางดังนี้

- TextView สำหรับแสดงข้อความ
- EditText สำหรับรับข้อความ
- ImageView สำหรับแสดงรูปภาพ
- Button สำหรับแสดงปุ่ม
- ImageButton สำหรับแสดงปุ่มรูปภาพ

## แบบฝึกหัดหัวยบท

หลังจากผู้อ่านทดลองใช้งานอิลิเม้นต์ต่างๆ เรียบร้อยแล้ว ให้ผู้อ่านทำแบบฝึกหัดดังนี้

- ศึกษาการใช้งานอิลิเม้นต์ไดๆ เพิ่มอีก 3 ตัว เช่น RadioButton เป็นต้น
- ให้ผู้อ่านพัฒนาแอพพลิเคชันสำหรับแสดงข้อมูลชื่อ รหัสประจำตัวประชาชน รูปถ่าย เมื่อคลิกปุ่ม ที่กำหนด
- ให้ผู้อ่านพัฒนาแอพพลิเคชัน โดยนำอิลิเม้นต์ที่ศึกษาเพิ่มเติมในหัวข้อที่ 1 มาประกอบด้วย

## รู้จักรากฐานการใช้งานเลเยอร์เอาร์ (Layout)

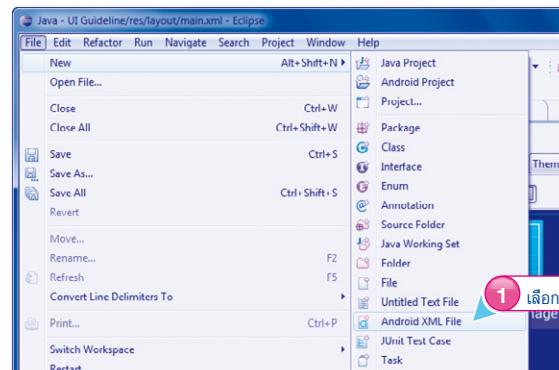
ในบทนี้ผู้อ่านจะได้ศึกษาถึงการใช้งานเลเยอร์เอาร์แบบต่างๆ เพื่อใช้ในการออกแบบหน้าจอการใช้งานของแอพพลิเคชัน ซึ่งในที่นี้ผู้เขียนขอเสนอเลเยอร์เอาร์ที่สำคัญจำนวน 5 รูปแบบคือ

- ลีเนียร์ เลเยอร์เอาร์ (Linear Layout)
- รีลেทีฟ เลเยอร์เอาร์ (Relative Layout)
- เฟรม เลเยอร์เอาร์ (Frame Layout)
- แคมโฉลุต เลเยอร์เอาร์ (Absolute Layout)
- เทเบิล เลเยอร์เอาร์ (Table Layout)

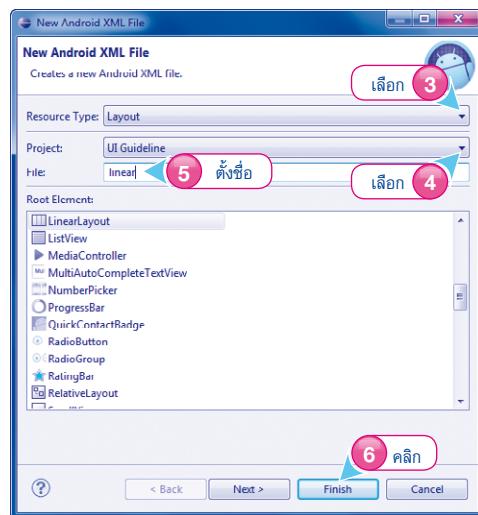
### Linear Layout

**ลีเนียร์ เลเยอร์เอาร์ (Linear Layout)** คือ เลเยอร์เอาร์ที่มีการจัดเรียงอิลิเม้นต์ที่อยู่ภายใต้เป็นแบบเล้นตรง เช่น เรียงในแบบแนวตั้ง (Vertical) หรือเรียงในแบบแนวนอน (Horizontal) โดยที่มีการทำหนดคุณสมบัติ (Property) android:orientation ซึ่งมีขั้นตอนต่างๆ ดังต่อไปนี้

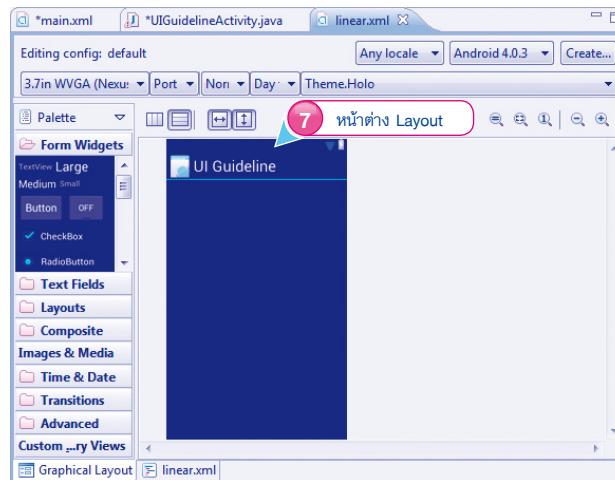
1. สร้างไฟล์ XML ขึ้นมาใหม่ โดยคลิกเมนู File > New > Android XML File



2. จะปรากฏ dialogue ตั้งชื่อ New Android XML File ขึ้นมา
3. ในส่วนของ Resource Type: ให้คลิกเลือก Layout
4. ในส่วนของ Project: ให้คลิกเลือก UI Guideline
5. ตั้งชื่อไฟล์ XML ในที่นี่ตั้งชื่อว่า linear
6. คลิกปุ่ม **Finish**



7. จะปรากฏหน้าต่าง Layout ทางด้านขวาเมื่อ



8. ผู้อ่านสามารถลับจากรูปแบบ Graphical Layout เพื่อแก้ไขโค้ด XML ได้ โดยคลิกแท็บ linear.xml ซึ่งมีโค้ด xml ดังนี้

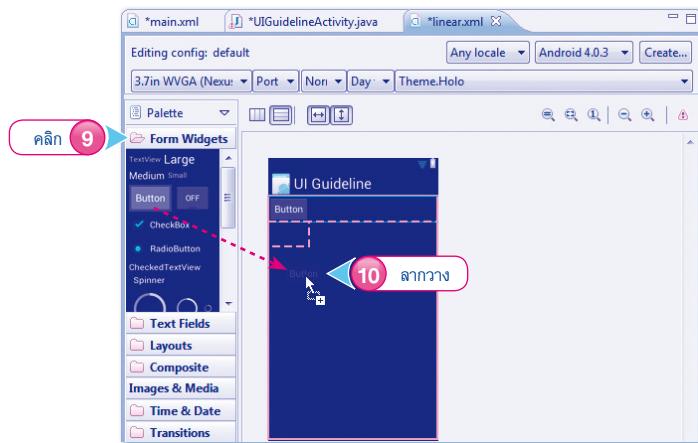
```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical" >
6
7
8 </LinearLayout>

```

9. คลิกที่ Form Widgets

10. เลือกอัลิเม้นต์ต่างๆ มาวางในหน้าจอแอพพลิเคชัน ในที่นี่จะสร้างปุ่มกด (Button) 2 ปุ่ม



11. ให้ผู้อ่านคลิกแท็บ linear.xml

12. จะปรากฏโค้ด android:orientation="vertical" ซึ่งเป็นการจัดวางปุ่มในแนวตั้ง

The screenshot shows the Android Studio code editor with the XML file "linear.xml" open. The code defines a Linear Layout with two buttons, both set to "wrap\_content" width and height, and "Button" text. A red circle with the number "11" is pointing to the "Graphical Layout" tab at the bottom. A red circle with the number "12" is pointing to the line "android:orientation="vertical"" with the text "แสดงโค้ด" (Show code).

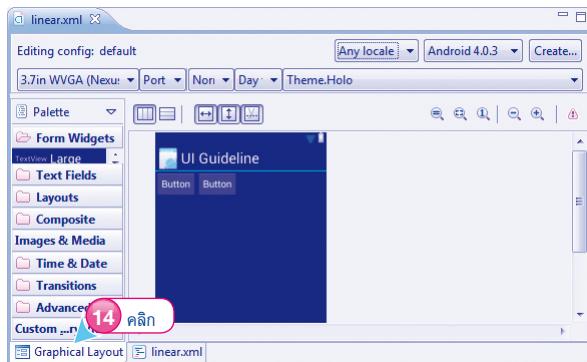
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" />
</LinearLayout>
```

13. ผู้อ่านสามารถเปลี่ยนให้มีวางเรียงกันในแนวนอน โดยแก้ไขโค้ดจาก android:orientation="vertical"  
เป็น android:orientation="horizontal"

The screenshot shows the Android Studio code editor with the XML file "linear.xml" open. The code is identical to the previous one, but the "android:orientation" attribute for the LinearLayout is now set to "horizontal". A red circle with the number "13" is pointing to the line "android:orientation="horizontal"" with the text "แก้ไขโค้ด" (Edit code).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" />
</LinearLayout>
```

#### 14. คลิกแท็บ Graphical Layout เพื่อดูหน้าจอแอพพลิเคชัน



#### 15. ผู้อ่านสามารถสร้าง Linear Layout หลายๆ อิลิเม้นต์ซ้อนกันได้ โดยตัวอย่างนี้จะเพิ่ม EditText เข้าไปแล้วจัดวางให้สวยงาม ซึ่งมีโค้ด XML ดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/editText1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >
        <requestFocus />
    </EditText>

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:orientation="horizontal" >

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="OK" />
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Cancel" />
    </LinearLayout>
</LinearLayout>

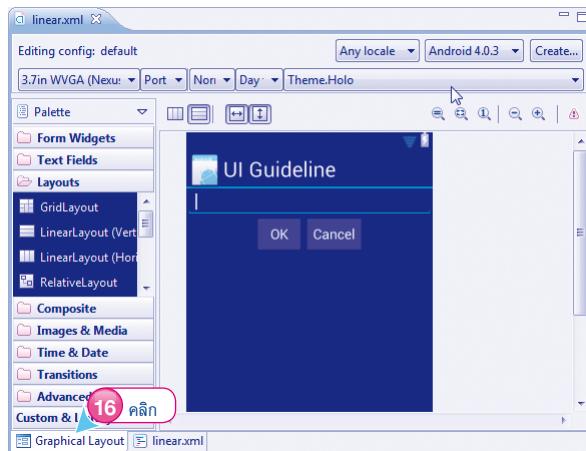
```

กำหนดคุณสมบัติของ EditText

กำหนดคุณสมบัติของ Button ตัวที่ 1

กำหนดคุณสมบัติของ Button ตัวที่ 2

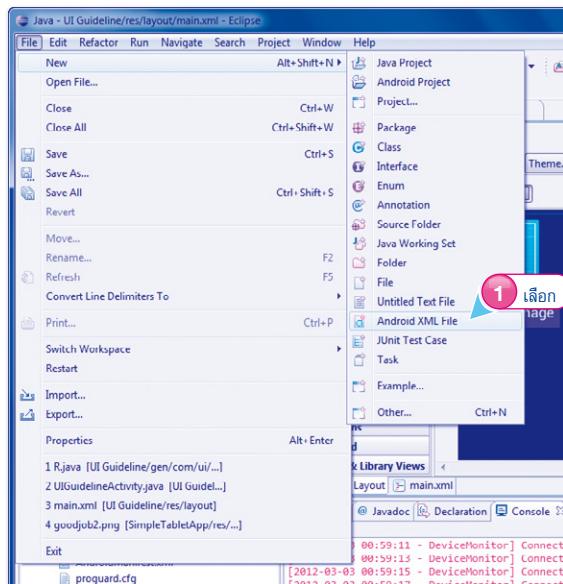
## 16. คลิกแท็บ Graphical Layout เพื่อดูหน้าจอที่ได้จากการเพิ่มโค้ด XML ดังรูป



## Relative Layout

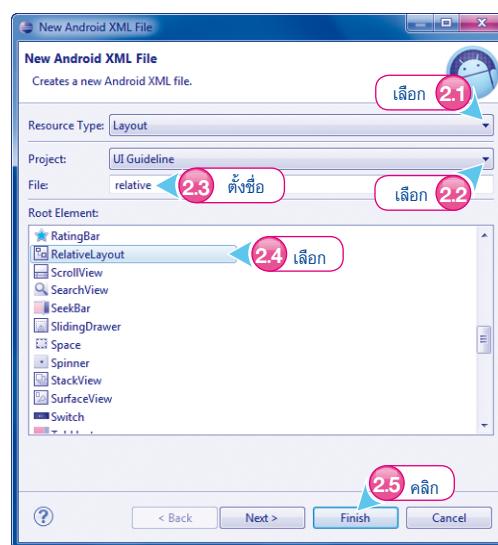
รีลেทีฟ เลเยอร์เอาต์ (**Relative Layout**) คือ เลเยอร์เอาต์ที่มีการจัดเรียงอิลิเม้นต์ที่อยู่ภายนอก โดยอ้างอิงตำแหน่งของอิลิเม้นต์นั้นๆ กับอิลิเม้นต์อื่นๆ ในเลเยอร์เอาต์ หรืออาจจะอ้างอิงกับตัวเลเยอร์เอาต์ที่อิลิเม้นต์นั้นอยู่ภายนอกได้ ซึ่งสามารถอ้างอิงผ่าน id ของอิลิเม้นต์หรือเลเยอร์เอาต์ ทั้งนี้การสร้างรีลেทีฟ เลเยอร์เอาต์มีขั้นตอนดังต่อไปนี้

1. สร้างไฟล์ XML ขึ้นมาใหม่ โดยคลิกเมนู File > New > Android XML File



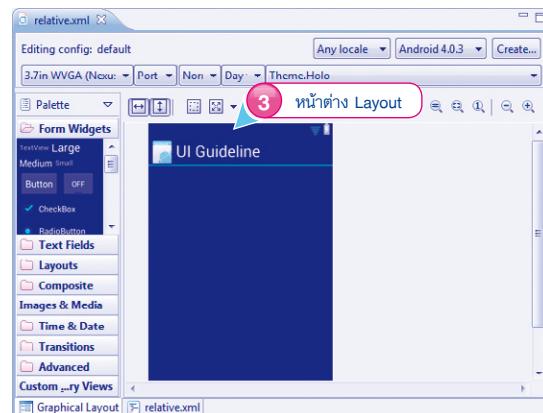
## 2. จะ pragmایไดอะล็อกนือกซ์

New Android XML File และ  
กำหนดคุณสมบัติดังนี้

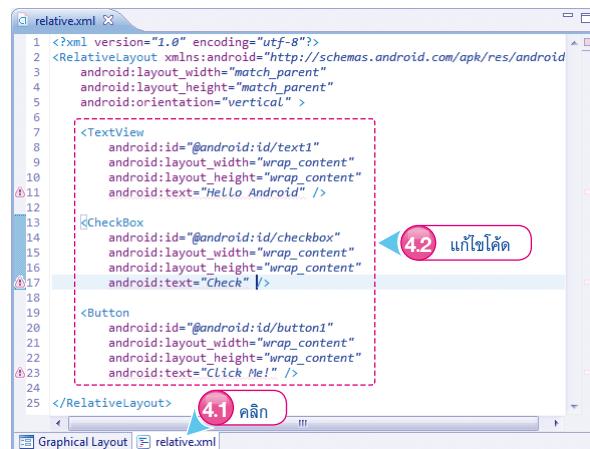


## 3. จะ pragmایหน้าต่าง Layout

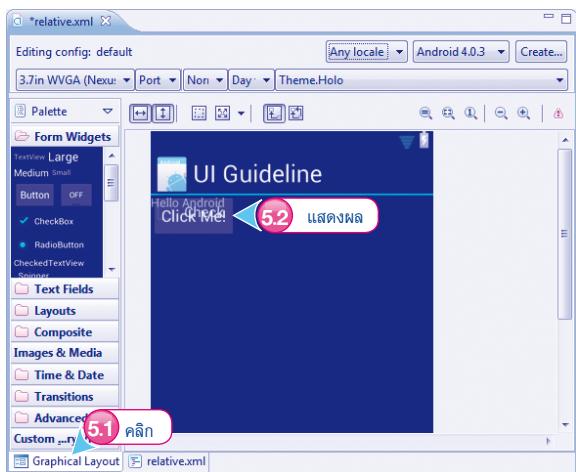
ทางด้านขวาเมื่อ



## 4. คลิกแท็บ relative.xml เพื่อแก้ไขโค้ด XML ดังนี้



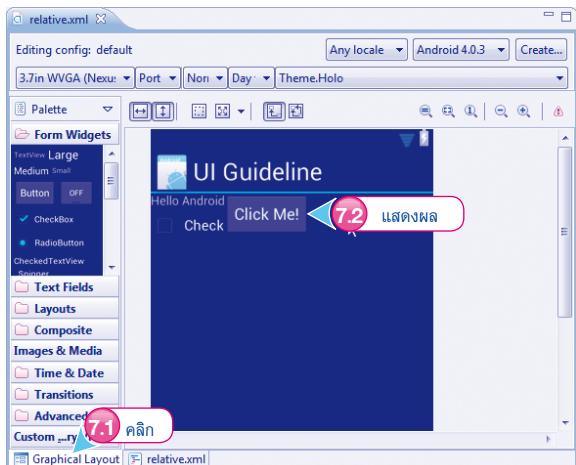
## 5. คลิกเทป Graphical Layout จะปรากฏอิลิเมนต์ต่างๆ ที่ซ้อนกันอยู่ดังรูป



6. แก้ไขโค้ดในไฟล์ XML  
อีกครั้ง เพื่อจัดตำแหน่ง  
อิลิเมนต์ต่างๆ ในที่นี่จะให้  
TextView อยู่ด้านบนสุด  
ถัดลงมาเป็น CheckBox  
และ Button ซึ่ง Button  
จะอยู่ด้านขวาของ CheckBox

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical" >
6
7     <TextView
8         android:id="@+id/text1"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Hello Android" />
12
13     <CheckBox
14         android:id="@+id/checkbox"
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content"
17         android:layout_below="@+id/text1"
18         android:text="Check" />
19
20     <Button
21         android:id="@+id/button1"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:layout_toRightOf="@+id/checkbox"
25         android:text="Click Me!" />
26
27 </RelativeLayout>
```

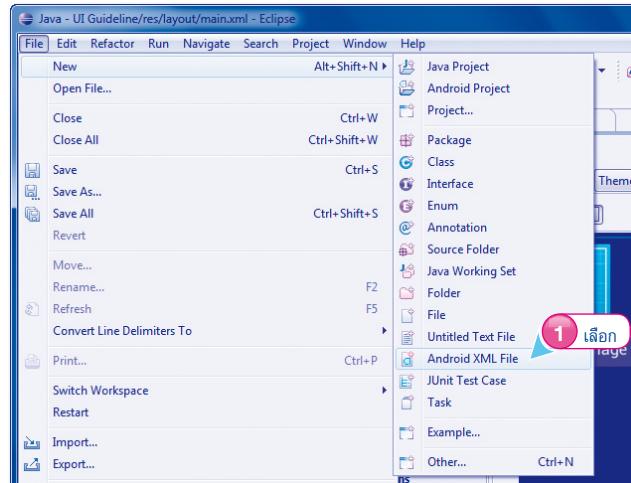
7. คลิกเทป Graphical Layout  
ก็จะปรากฏอิลิเมนต์ต่างๆ  
จัดเรียงกันดังรูป



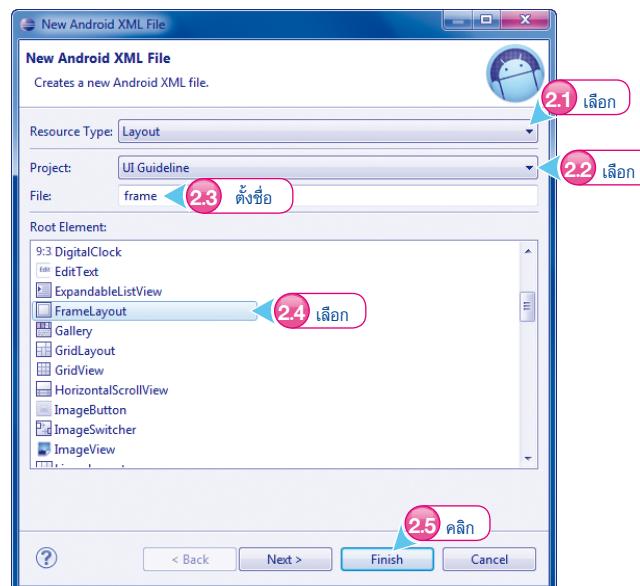
# Frame Layout

เฟรม เลย์เออร์ (Frame Layout) คือ เลย์เออร์ที่มีการจัดวางอิลิเมนต์ต่างๆ เป็นชั้นๆ โดยจะเริ่มวางอิลิเมนต์ที่มุ่งช้ายด้านบนของเลย์เออร์เสมอ จากนั้นจะนำอิลิเมนต์ที่ลรังมาวางไว้บนลุ่ด ซึ่งการวางแบบนี้จะเรียกอีกอย่างหนึ่งว่า **การวางแบบสเต็ค** ทั้งนี้การสร้างเฟรม เลย์เออร์มีขั้นตอนต่างๆ ดังต่อไปนี้

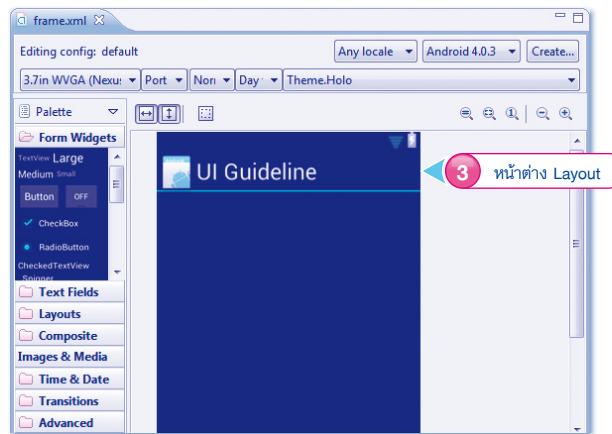
- สร้างไฟล์ XML ขึ้นมาใหม่ โดยคลิกเมนู File > New > Android XML File



- จะปรากฏได้คลิกบันทึกชื่อ New Android XML File และกำหนดคุณสมบัติดังนี้



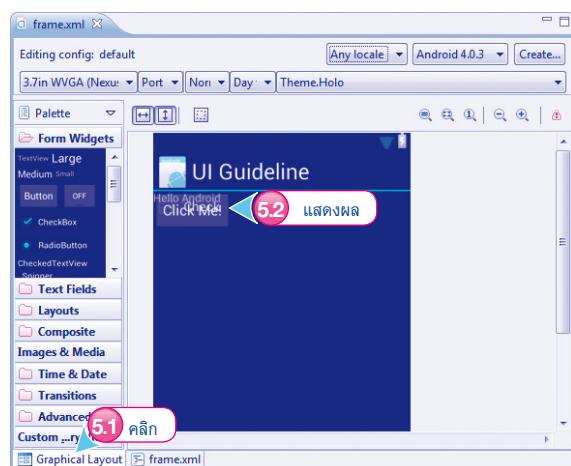
### 3. จะปรากฏหน้าต่าง Layout ทางด้านขวาเมื่อ



### 4. คลิกแท็บ frame.xml เพื่อแก้ไขโค้ด XML ดังนี้



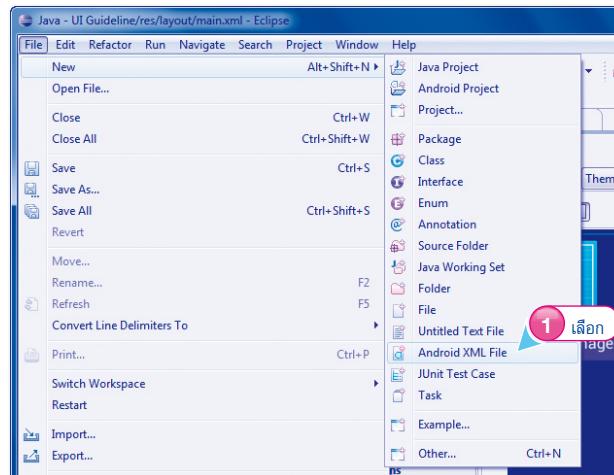
### 5. คลิกแท็บ Graphical Layout จะเห็นอิเลิมентаต่างๆ ข้อนกันอยู่เป็นชั้นๆ ซึ่งเป็นลักษณะของ Frame Layout



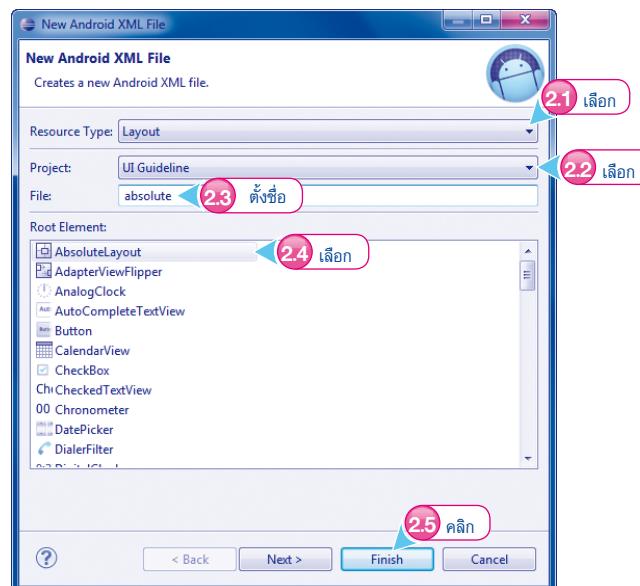
# Absolute Layout

แอนโซลูต เลย์เออร์ (**Absolute Layout**) คือ เลย์เออร์ที่มีการจัดวางอิลิเม้นต์ต่างๆ ที่อยู่ภายในตามตำแหน่งจริงบนหน้าจอ โดยกำหนดตำแหน่งของอิลิเม้นต์ผ่านพาร์เพอวาร์ต์ `android:layout_x` และ `android:layout_y` สำหรับการสร้างแอนโซลูต เลย์เออร์มีขั้นตอนดังๆ ดังต่อไปนี้

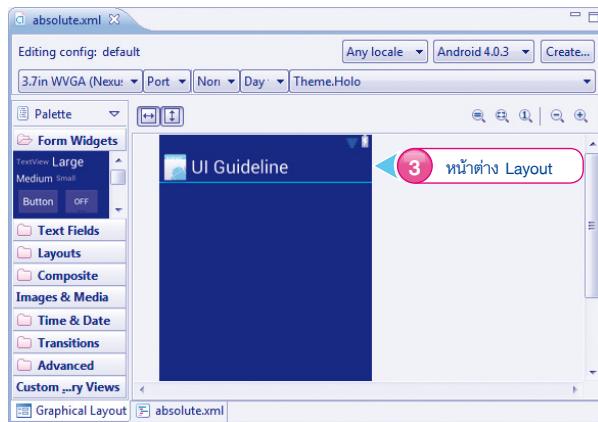
- สร้างไฟล์ XML ใหม่ โดยคลิกเมนู File > New > Android XML File



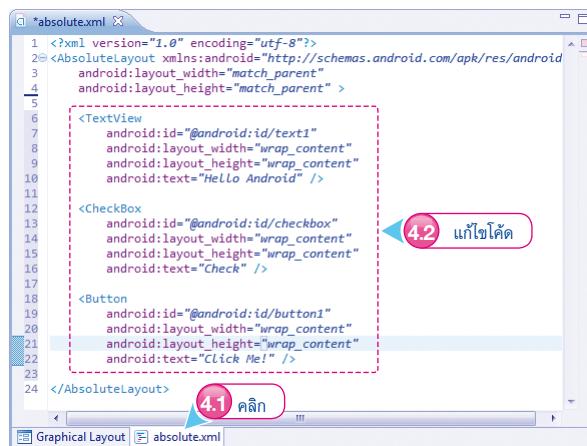
- จะปรากฏ dialogue ให้เลือกบันทึกชื่อ New Android XML File และกำหนดคุณสมบัติดังนี้



### 3. จะปรากฏหน้าต่าง Layout ทางด้านขวาเมื่อ

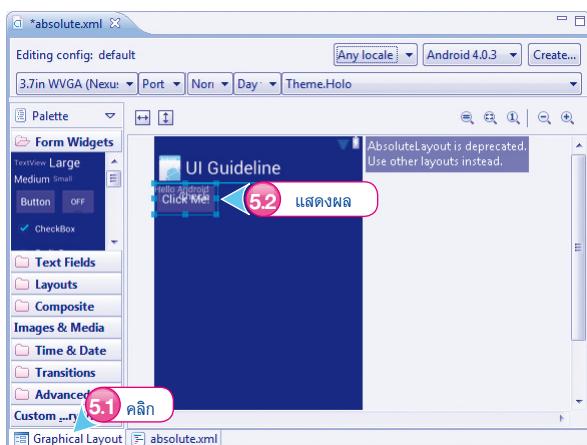


### 4. คลิกแท็บ absolute.xml เพื่อแก้ไขโค้ด XML ดังนี้



### 5. คลิกแท็บ Graphical Layout

จะเห็นอิลิเม้นต์ต่างๆ ซ้อนกันอยู่ ซึ่งเกิดจากการที่ยังไม่ได้กำหนดตำแหน่งให้กับแต่ละอิลิเม้นต์



6. แก้ไขโค้ดในไฟล์ XML อีกครั้ง เพื่อจัดตำแหน่งของแต่ละอิลิเม้นต์ โดยเพิ่มพร็อพเพอร์ตี้ android:layout\_x และ android:layout\_y

```

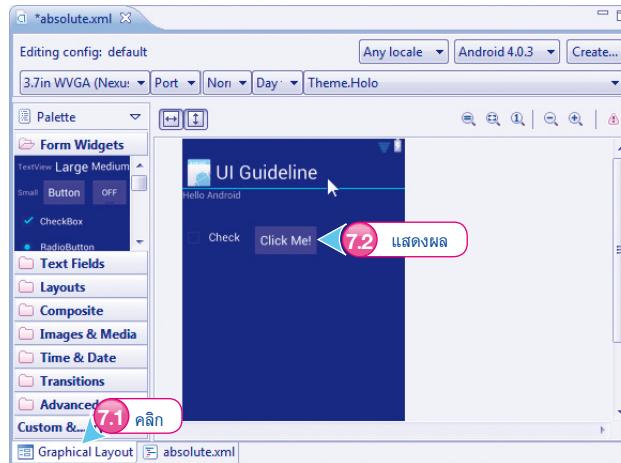
1 <?xml version="1.0" encoding="utf-8"?>
2 <AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent" 
5
6     <TextView
7         android:id="@+id/text1"
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="Hello Android" />
11
12     <CheckBox
13         android:id="@+id/checkbox"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:layout_x="80dp"
17         android:layout_y="50dp"
18         android:text="Check" />
19
20     <Button
21         android:id="@+id/button1"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:layout_x="100dp"
25         android:layout_y="50dp"
26         android:text="Click Me!" />
27
28 </AbsoluteLayout>

```

6.1 คลิก

6.2 แก้ไขโค้ด

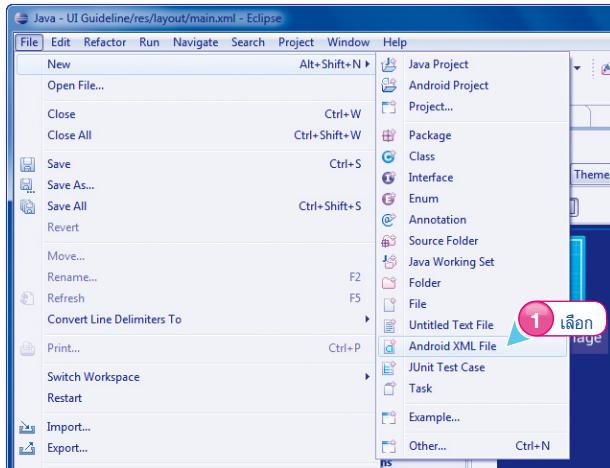
7. คลิกแท็บ Graphical Layout จะเห็นอิลิเม้นต์ที่วาง จัดเรียงกันตั้งรูป



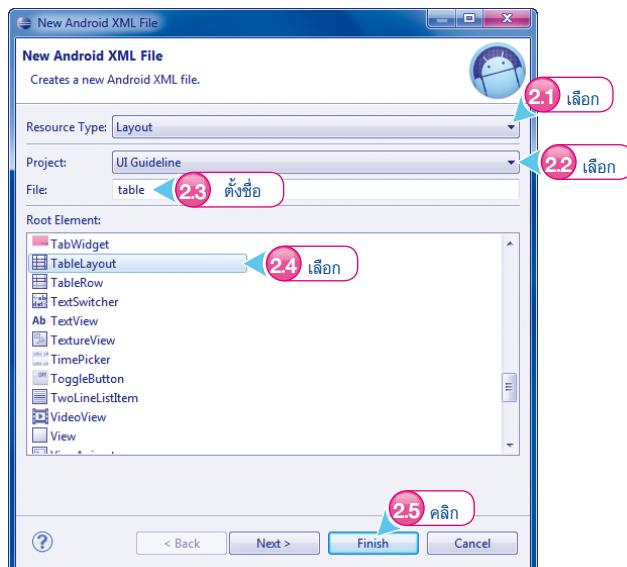
# Table Layout

ເທັນລີ ເລີຍເອົາຕີ (Table Layout) ອີ່ວ້າ ເລີຍເອົາຕີທີ່ມີການຈັດວາງອືລິແມນດີຕ່າງໆ ເປັນແບບຕາຈາງໂດຍທີ່ຄອນໂກຣແຕ່ລະສ່ວນຈະຄືວ້າເປັນ 1 ຄອລັນນີ້ ນອກຈາກນີ້ຢັງສາມາດພື່ມແກວໄດ້ໂດຍການເພີ່ມແກ້ກໍ TableRow ສໍາຮັບການສ້າງເທັນລີ ເລີຍເອົາຕີມີຂັ້ນຕອນດັ່ງຕ່ອງໄປນີ້

- ສ້າງໄຟ XML ຫຼື້ມາໃໝ່ ໂດຍຄຶກເມຸນ File > New > Android XML File

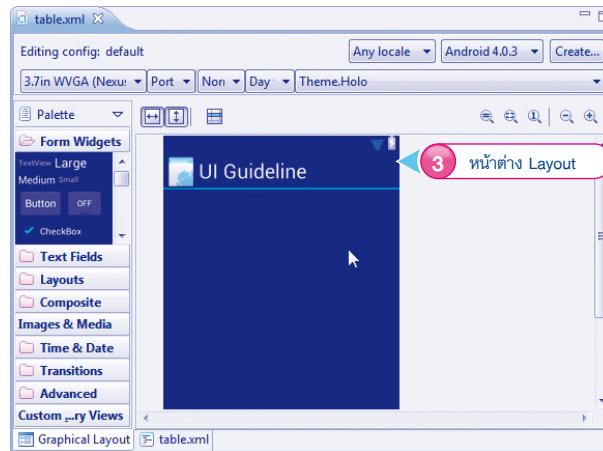


- ຈະປາກ້າໄດ້ລືອກນີ້ອັກສ້າງ New Android XML File ແລະ ກໍາທັນດຄູນລົມບໍດິດັ່ງນີ້





### 3. จะปรับกฎหน้าต่าง Layout ทางด้านขวาเมื่อ



### 4. แก้ไขโค้ดในไฟล์ table.xml ดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TableRow android:id="@+id/tableRow1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <TextView android:text="Username :"
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
        </TextView>
        <EditText android:id="@+id/editText1"
            android:layout_width="200dip"
            android:layout_height="wrap_content">
        </EditText>
    </TableRow>
    <TableRow android:id="@+id/tableRow2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <TextView android:text="Password :"
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
        </TextView>
        <EditText android:id="@+id/editText2"
            android:layout_width="200dip"
            android:layout_height="wrap_content">
        </EditText>
    </TableRow>
</TableLayout>

```

กำหนดคุณสมบัติและอิลิเม้นท์ในແກ່ວ່າ 1

กำหนดคุณสมบัติและອิลิเมນໍໃນແກ່ວ່າ 2

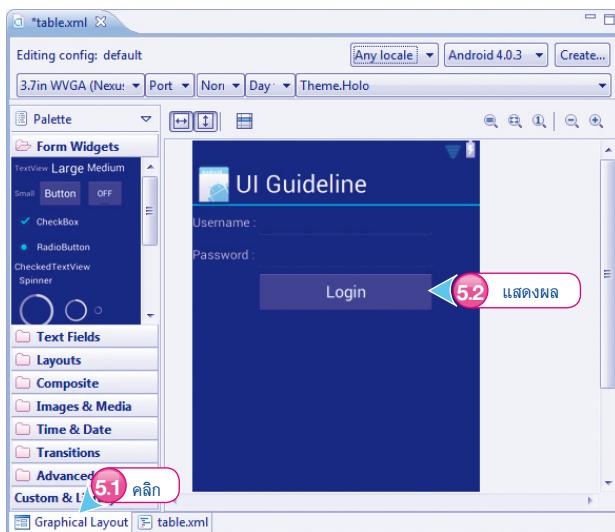
```

<TableRow android:layout_height="wrap_content"
          android:layout_width="wrap_content"
          android:id="@+id/tableRow3">
    <TextView android:text=""
              android:id="@+id/textView3"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content">
    </TextView>
    <Button android:layout_width="wrap_content"
            android:text="Login"
            android:id="@+id/button1"
            android:layout_height="wrap_content">
    </Button>
</TableRow>
</TableLayout>

```

← กำหนดคุณสมบัติและอิลิเมนต์ในແນວທີ 3

### 5. ຄລືກແຫັບ Graphical Layout ຈະເຫັນອີລິມັນຕໍ່ຕ່າງໆ ຈັດເຮັງກັນດັ່ງຽຸ່ງ



## ສຽງກ້າຍບານ

ໃນບັນທຶນນີ້ເປັນການອືບຍາຍຄືການໃຊ້ຈານເລີຍເຄົາຕໍ່ລໍາຫວັບອອກແບບແອພພລິເຄຂັນ ເພື່ອຈັດວາງສ່ວນຕິດຕ່ອງກັນຜູ້ໃຊ້ຈານ (UI–User Interface) ໃຫ້ສ້າງຈານແລະໃຊ້ຈານໄດ້ອ່າຍສະວັກດັ່ງນີ້

- ລືເນີຍ໌ ເລີຍເຄົາ (Linear Layout) ອີ່ ເລີຍເຄົາທີ່ມີການຈັດເຮັງອີລິມັນທີ່ອ່ຽງ່າຍໃນເນື່ອແບບເລັ້ນຕຽງ
- ວິເລທີ່ພ ເລີຍເຄົາ (Relative Layout) ອີ່ ເລີຍເຄົາທີ່ມີການຈັດເຮັງອີລິມັນໂດຍອ້າງອີງຕໍ່ແໜ່ງຂອງອີລິມັນທີ່ນີ້ ກັບອີລິມັນທີ່ອື່ນໆ
- ເຟຣມ ເລີຍເຄົາ (Frame Layout) ອີ່ ເລີຍເຄົາທີ່ມີການຈັດວາງອີລິມັນຕໍ່ຕ່າງໆ ເປັນຫັ້ນໆ
- ແອບໂໂລຸຕ ເລີຍເຄົາ (Absolute Layout) ອີ່ ເລີຍເຄົາທີ່ມີການຈັດວາງອີລິມັນຕໍ່ຕ່າງໆ ທີ່ອ່ຽງ່າຍໃນຕາມຕໍ່ແໜ່ງຈົງບນ້າຈອ
- ເທົເມີລ ເລີຍເຄົາ (Table Layout) ອີ່ ເລີຍເຄົາທີ່ມີການຈັດວາງອີລິມັນຕໍ່ຕ່າງໆ ເປັນແບບຕາරັງ

## แบบฝึกหัดก้ายบต

- อธิบายความเหมือนและความแตกต่างกันระหว่างเฟรม เลย์เอาต์ (Frame Layout) และแอปโซลูต เลย์เอาต์ (Absolute Layout)
- จงอธิบายว่าหากต้องการพัฒนาแอพพลิเคชันเครื่องคิดเลข จะต้องใช้งานเลย์เอาต์ (Layout) อะไร เพราะอะไร
- ให้ผู้อ่านอ่านออกแบบแอพพลิเคชันโดยใช้งานเลย์เอาต์ (Layout) ที่แตกต่างกันจำนวน 3 แอพพลิเคชัน

# รู้จักรากฐานการใช้งานวิวกรุ๊ป (ViewGroup) ชนิดต่างๆ

ในบทที่ 5 และบทที่ 6 ผู้อ่านได้ศึกษาการใช้งานอิลิเมนต์และเลเยอร์เอกสารพื้นฐานกันไปแล้ว ในบทนี้ ผู้อ่านจะได้ศึกษาถึงการใช้งานวิวกรุ๊ป (ViewGroup) ชนิดต่างๆ ซึ่งเป็นพื้นฐานที่สำคัญอีกเรื่องหนึ่งในการออกแบบ และพัฒนาแอ��เพลิเคชันบนระบบ Android

**วิวกรุ๊ป (ViewGroup)** เป็นวิวที่รวมอิลิเมนต์ต่างๆ เช่น TextView, EditText และ CheckBox ไว้ภายใน ซึ่งในบทนี้จะขอยกตัวอย่างอิลิเมนต์บางตัวที่มีการใช้งานบ่อยครั้ง รวมถึงอธิบายการประยุกต์ใช้วิวกรุ๊ปชนิดต่างๆ ซึ่งประกอบด้วย ListView, ScrollView, Image Switcher View, Options Menu และ TabWidget

## NOTE



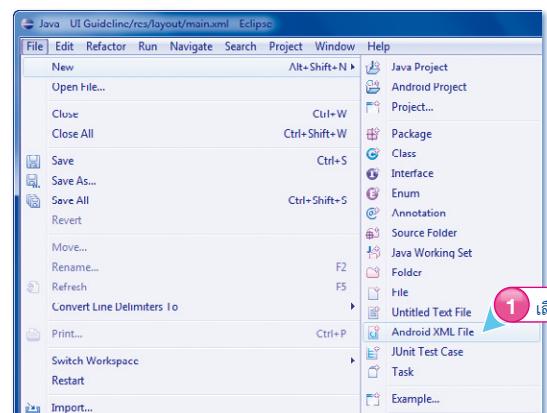
ในที่นี้ขั้นตอนบางขั้นตอนที่เคยอธิบายไว้ในบทที่ผ่านมา ผู้เขียนจะไม่ขออธิบายซ้ำ ซึ่งผู้อ่านสามารถศึกษาได้จากบทที่ผ่านมา เช่น ขั้นตอนการสร้างโปรเจกต์, ขั้นตอนการเพิ่มไฟล์ XML เป็นต้น

## List View

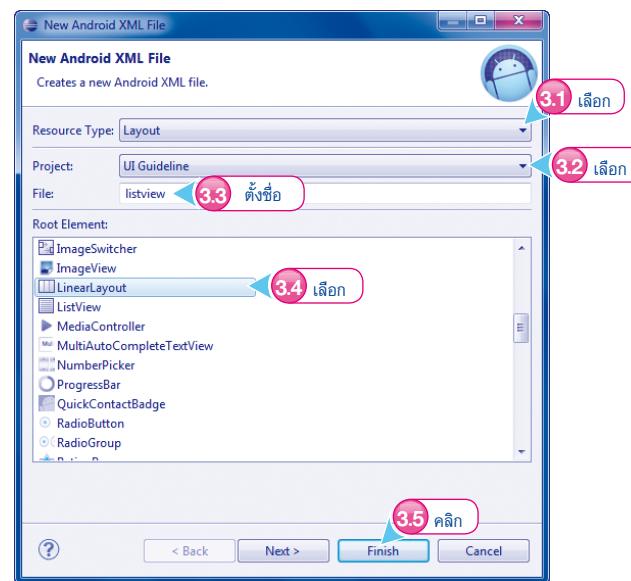
ListView เป็นการแสดงผลรายการ อิลิเมนต์ที่อยู่ภายใต้แบบรายการตามแนวตั้ง โดยมีขั้นตอนการสร้างดังต่อไปนี้

### 1. สร้างไฟล์ XML ขึ้นมาใหม่ โดยคลิก

เมนู File > New > Android XML File



2. จะปรากฏ dialogue ตัวเลือกเมื่อกด New Android XML File ขึ้นมา
3. ตั้งชื่อไฟล์เป็น listview และกำหนดคุณสมบัติดังนี้



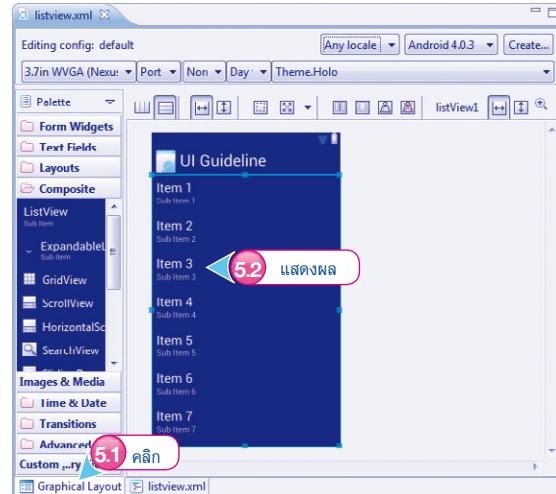
4. แก้ไขโค้ด XML ในไฟล์ listview.xml โดยเพิ่มอิเลเมนต์ ListView และกำหนด id เป็น listView1 ดังรูป

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical" 
6     >
7     <ListView
8         android:id="@+id/listView1"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content" 
11    >
12 </LinearLayout>
13

```

5. คลิกแท็บ Graphical Layout เพื่อดูหน้าจอแอพพลิเคชัน



## 6. เขียนโค้ด Java กำหนดการทำงานของแอพพลิเคชันในไฟล์ UIGuidelineActivity.java ดังนี้

```
package com.ui.guideline;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

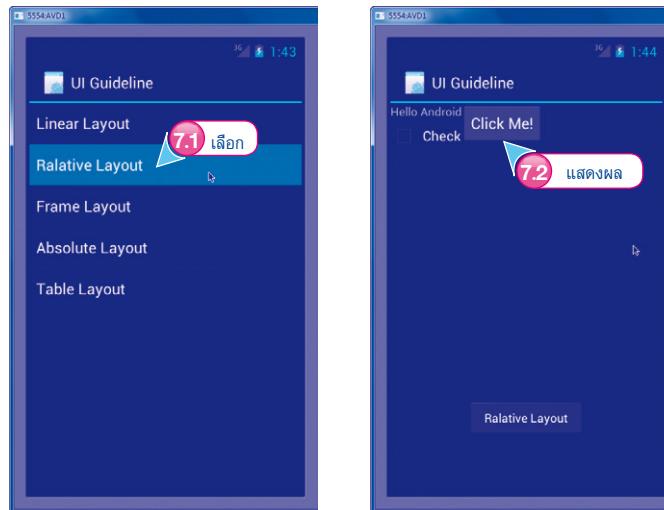
public class UIGuidelineActivity extends Activity {
    /** Called when the activity is first created. */
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.listview);
        ListView list_view1;
        final String page_name[] = { "Linear Layout", "Ralative Layout",
            "Frame Layout", "Absolute Layout", "Table Layout" };
        list_view1 = (ListView) findViewById(R.id.listView1);
        ArrayAdapter<String> array_adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, page_name);
        list_view1.setAdapter(array_adapter);
        list_view1.setOnItemClickListener(new OnItemClickListener() {

            public void onItemClick(AdapterView<?> arg0, View arg1, int num,
                long arg3) {
                Toast.makeText(getApplicationContext(), page_name[num],
                    Toast.LENGTH_LONG).show();
                switch (num + 1) {
                    case 1:
                        setContentView(R.layout.linear);
                        break;
                    case 2:
                        setContentView(R.layout.relative);
                        break;
                    case 3:
                        setContentView(R.layout.frame);
                        break;
                    case 4:
                        setContentView(R.layout.absolute);
                        break;
                    case 5:
                        setContentView(R.layout.table);
                        break;
                }
            }
        });
    }
}
```

กำหนดรูปแบบการแสดง Layout

จากโค้ดจะเป็นการแสดงรายการ Layout ชนิดต่างๆ ตามที่ผู้อ่านได้เคยเรียนรู้ในบทที่ผ่านมา และเมื่อคลิกที่ Layout ชนิดใด โปรแกรมก็จะเปลี่ยนหน้าจอเป็น Layout ชนิดนั้นๆ

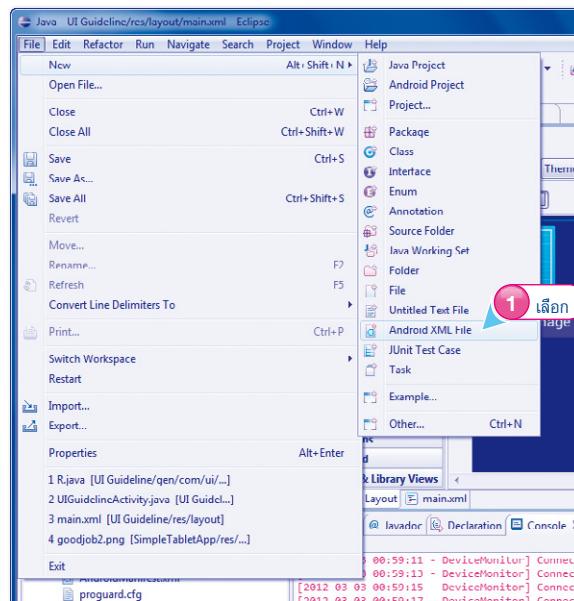
## 7. เมื่อรันแอพพลิเคชันจะได้ผลลัพธ์ดังนี้



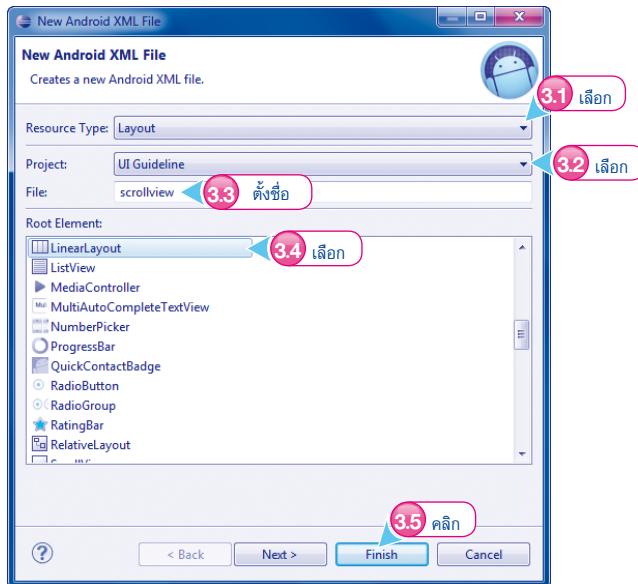
## ScrollView

ScrollView เป็นการกำหนดคุณสมบัติของหน้าจอ เพื่อให้สามารถเลื่อนดูได้ทุกส่วน โดยมีขั้นตอน การสร้างดังต่อไปนี้

### 1. สร้างไฟล์ XML ขึ้นมาใหม่ โดยคลิกเมนู File > New > Android XML File



- จะปรากฏ dialogue ออกบอช New Android XML File ขึ้นมา
- ตั้งชื่อเป็น scrollview และกำหนดคุณสมบัติดังนี้



#### 4. แก้ไขโค้ด XML ในไฟล์ scrollview.xml ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <TableLayout <span style="color:blue">←</span>
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:stretchColumns="0">
        <TableRow>
            <View
                android:layout_height="80dp"
                android:background="#000000"/>
                <TextView android:text="#000000"
                    android:paddingLeft="4dp"
                    android:layout_gravity="center_vertical"/>
            </TableRow>
            <TableRow>
                <View
                    android:layout_height="80dp"
                    android:background="#440000"/>
                    <TextView android:text="#440000"
                        android:paddingLeft="4dp"
                        android:layout_gravity="center_vertical"/>
                </TableRow>
                <TableRow>
                    <View
                        android:layout_height="80dp"
                        android:background="#884400"/>
                        <TextView android:text="#884400"
```

● กำหนดการแสดงเป็นแบบ TableLayout โดยมีทั้งหมด 7 แถว

● กำหนดคุณสมบัติแถวที่ 1

● กำหนดคุณสมบัติแถวที่ 2



```

        android:paddingLeft="4dp"
        android:layout_gravity="center_vertical"/>
    
```

</TableRow>

<TableRow>

<View>

android:layout\_height="80dp"

android:background="#aa8844"/>

<TextView android:text="#aa8844"
 android:paddingLeft="4dp"
 android:layout\_gravity="center\_vertical"/>

</TableRow>

<TableRow>

<View>

android:layout\_height="80dp"

android:background="#ffaa88"/>

<TextView android:text="#ffaa88"
 android:paddingLeft="4dp"
 android:layout\_gravity="center\_vertical"/>

</TableRow>

<TableRow>

<View>

android:layout\_height="80dp"

android:background="#ffffaa"/>

<TextView android:text="#ffffaa"
 android:paddingLeft="4dp"
 android:layout\_gravity="center\_vertical"/>

</TableRow>

<TableRow>

<View>

android:layout\_height="80dp"

android:background="#ffffff"/>

<TextView android:text="#ffffff"
 android:layout\_gravity="center\_vertical"
 android:paddingLeft="4dp"
 android:text="#ffffff" />

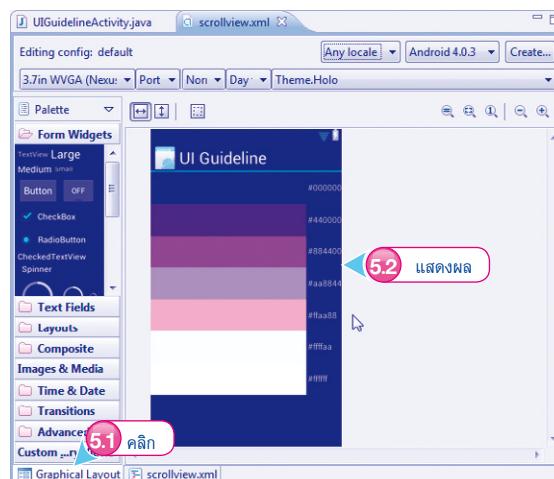
</TableRow>

</TableLayout>

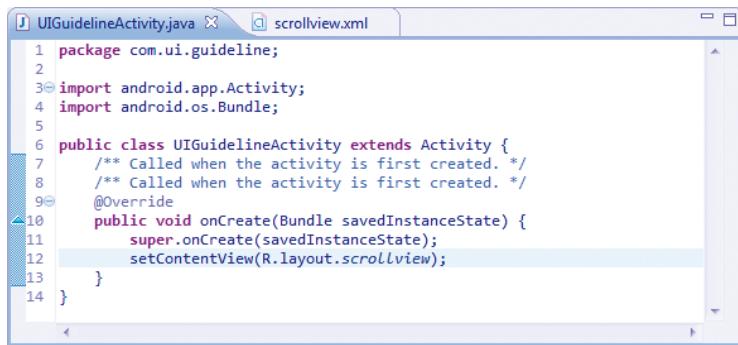
</ScrollView>

กำหนดคุณสมบัติได้ที่ 7

## 5. คลิกแท็บ Graphical Layout เพื่อดูหน้าจอแอพพลิเคชัน

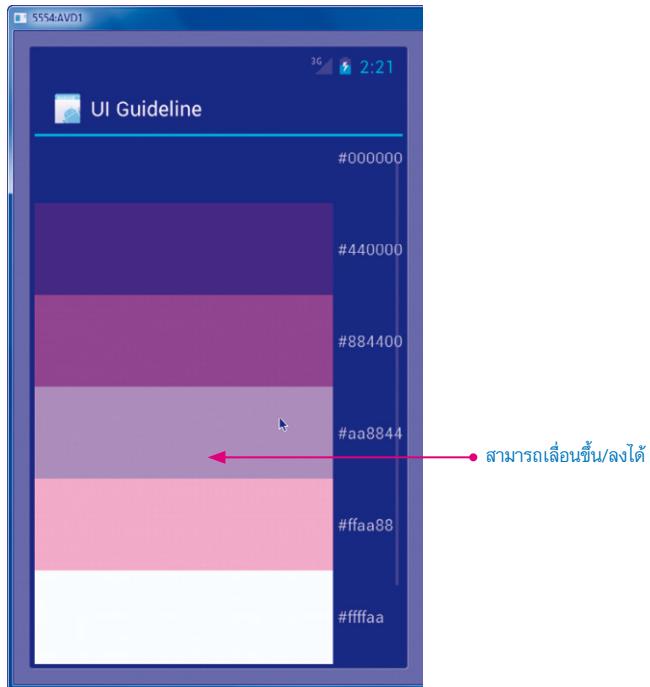


6. แก้ไขโค้ด Java ในไฟล์ `UIGuidelineActivity.java` โดยใช้ไฟล์ `scrollview.xml` มาแสดงเป็นหน้าจอ



```
1 package com.ui.guideline;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5
6 public class UIGuidelineActivity extends Activity {
7     /** Called when the activity is first created. */
8     /** Called when the activity is first created. */
9     @Override
10    public void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.scrollview);
13    }
14 }
```

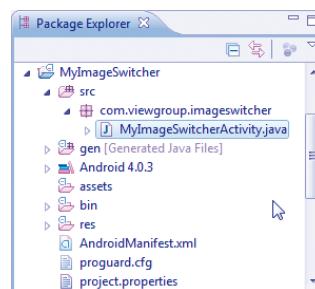
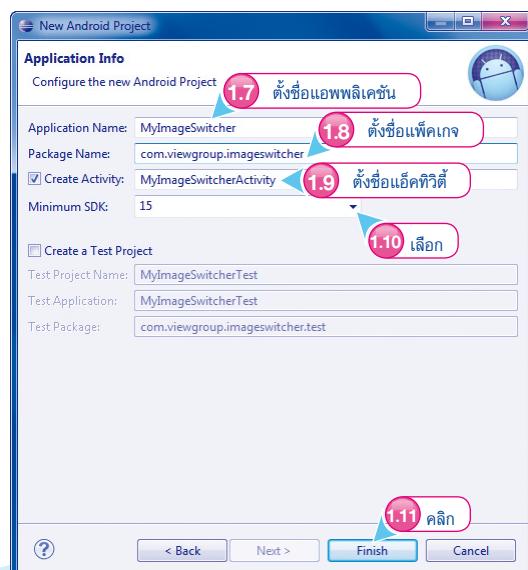
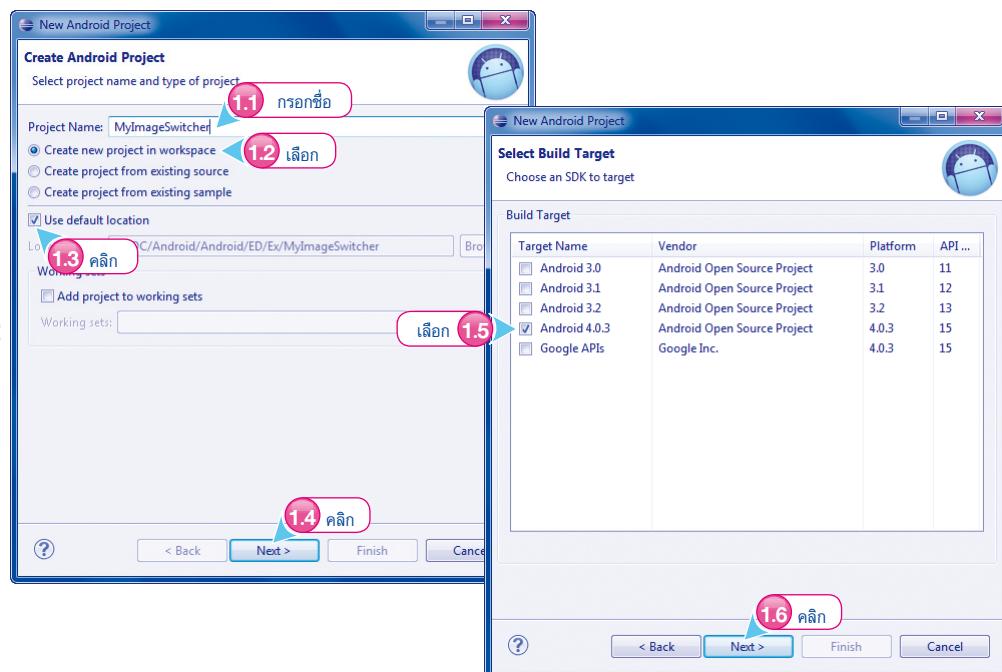
7. เมื่อรันแอพพลิเคชันจะได้ผลลัพธ์ดังนี้



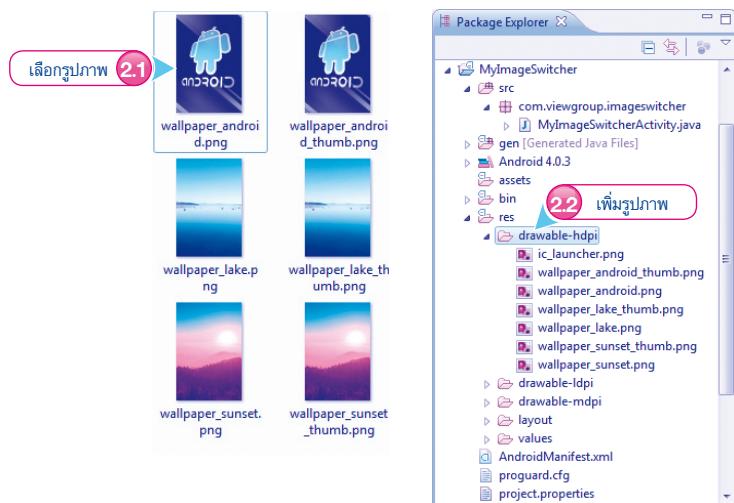
# Image Switcher View

Image Switcher View เป็นการกำหนดคุณสมบัติในการแสดงรูปภาพหลายรูป โดยสามารถเลือกแสดงรูปภาพที่แตกต่างกันสลับไปมาได้ โดยมีขั้นตอนการสร้างดังต่อไปนี้

- สร้างโปรเจกต์ใหม่ ในที่นี้ตั้งชื่อว่า MyImageSwitcher และกำหนดคุณสมบัติดังนี้



## 2. นำไฟล์รูปภาพที่ต้องการมาใส่ในโฟลเดอร์ res/drawable-hdpi ดังรูป



## 3. แก้ไขโค้ด XML ในไฟล์ main.xml ดังนี้



## 4. แก้ไขโค้ด Java ในไฟล์ MyImageSwitcherActivity.java ดังนี้

```
package com.viewgroup.imageswitcher;

import android.app.Activity;
import android.content.Context;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.AnimationUtils;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.LayoutParams;
import android.widget.ImageSwitcher;
import android.widget.ImageView;
```



```
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ViewSwitcher.ViewFactory;

public class MyImageSwitcherActivity extends Activity implements
    OnItemSelectedListener, ViewFactory {
    private Integer[] mThumbIds = { R.drawable.wallpaper_android_thumb,
        R.drawable.wallpaper_lake_thumb, R.drawable.wallpaper_sunset_thumb };

    private Integer[] mImageIds = { R.drawable.wallpaper_android,
        R.drawable.wallpaper_lake, R.drawable.wallpaper_sunset };
    private Context mContext;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mSwitcher = (ImageSwitcher) findViewById(R.id.imgswitcher);
        mSwitcher.setFactory(this);
        mSwitcher.setInAnimation(AnimationUtils.loadAnimation(this,
            android.R.anim.fade_in));
        mSwitcher.setOutAnimation(AnimationUtils.loadAnimation(this,
            android.R.anim.fade_out));

        Gallery g = (Gallery) findViewById(R.id.gallery);
        g.setAdapter(new ImageAdapter(this));
        g.setOnItemSelectedListener(this);
    }

    public void onItemSelected(AdapterView<?> parent, View v, int position,
        long id) {
        mSwitcher.setImageResource(mImageIds[position]);
    }

    public void onNothingSelected(AdapterView<?> parent) {
    }

    public View makeView() {
        ImageView i = new ImageView(this);
        i.setBackgroundColor(0xFF000000);
        i.setScaleType(ImageView.ScaleType.FIT_CENTER);
        i.setLayoutParams(new ImageSwitcher.LayoutParams(
            LayoutParams.FILL_PARENT, LayoutParams.FILL_PARENT));
        return i;
    }

    private ImageSwitcher mSwitcher;

    public class ImageAdapter extends BaseAdapter {
        public ImageAdapter(Context c) {
            mContext = c;
        }

        public int getCount() {
            return mThumbIds.length;
        }
    }
}
```

```

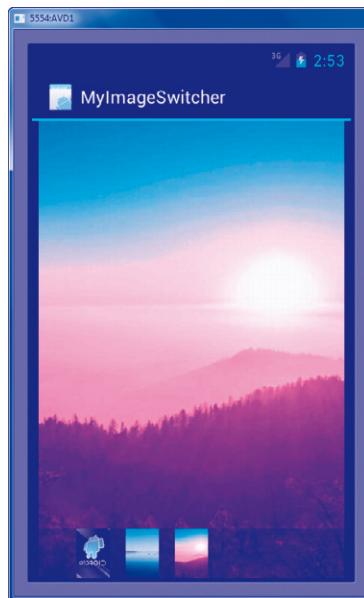
public Object getItem(int position) {
    return position;
}

public long getItemId(int position) {
    return position;
}

public View getView(int position, View convertView, ViewGroup parent) {
    ImageView i = new ImageView(mContext);
    i.setImageResource(mThumbIds[position]);
    i.setAdjustViewBounds(true);
    i.setLayoutParams(new Gallery.LayoutParams(
            LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT));
    i.setBackgroundColor(Color.WHITE);
    return i;
}
}
}

```

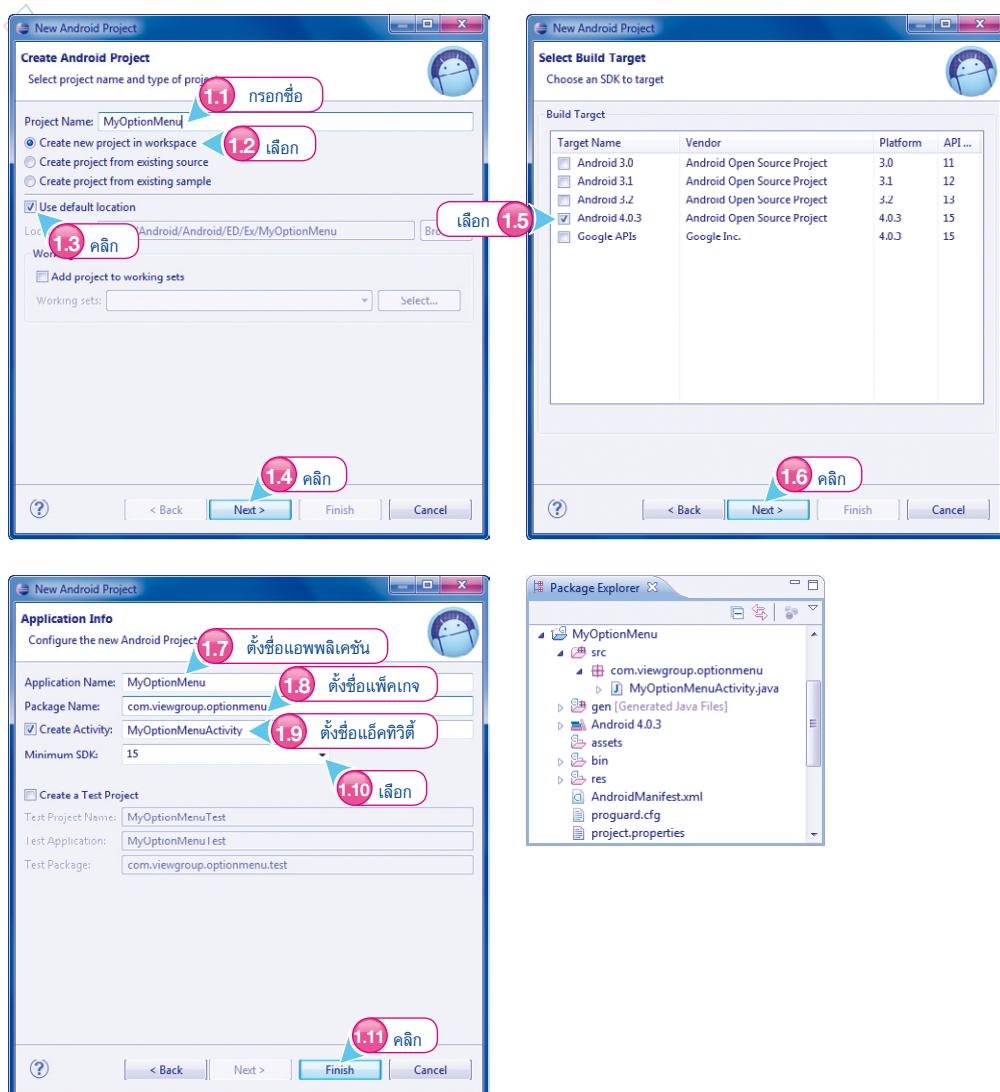
**5. เมื่อรันแอพพลิเคชันจะได้ผลลัพธ์ดังนี้**



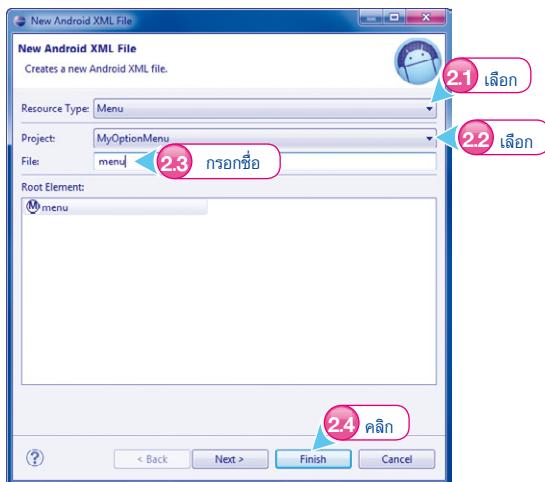
# Options Menu

Options Menu เป็นเมนูที่แสดงขึ้นมาเมื่อกดปุ่มเมนูบนโทรศัพท์ โดยมีขั้นตอนการสร้างดังต่อไปนี้

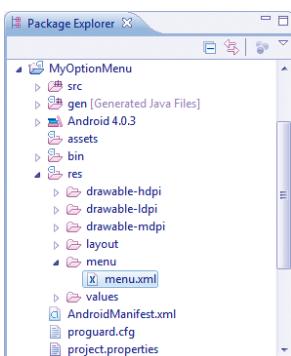
- สร้างโปรเจกต์ใหม่ ในที่นี้ตั้งชื่อว่า MyOptionMenu และกำหนดคุณสมบัติดังนี้



## 2. สร้างไฟล์ XML ใหม่สำหรับใช้เป็น Menu และกำหนดคุณสมบัติดังนี้



## 3. จะปรากฏไฟล์ menu.xml ในโฟลเดอร์ res/menu ที่หน้าต่าง Package Explorer



## 4. แก้ไขโค้ด XML ในไฟล์ menu.xml ดังนี้

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android" >
3
4     <item
5         android:id="@+id/help"
6         android:icon="@android:drawable/ic_menu_help"
7         android:title="Help">
8     </item>
9     <item
10        android:id="@+id/setting"
11        android:icon="@android:drawable/ic_menu_preferences"
12        android:title="Setting">
13     </item>
14
15 </menu>
```

โดยที่ @android:drawable/<name> เป็นการใช้รูปไอคอนมาตรฐานของ Android ซึ่งไม่ต้องเพิ่มไฟล์รูปเข้าไปเอง

## 5. แก้ไขโค้ด Java ในไฟล์ MyOptionMenuActivity.java เพื่อนำไฟล์ menu.xml มาใช้งานดังนี้

```
package com.viewgroup.optionmenu;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

public class MyOptionMenuActivity extends Activity {
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.help:
                Toast.makeText(getApplicationContext(), "Go to Help.", Toast.LENGTH_LONG)
                    .show();
                return true;
            case R.id.setting:
                Toast.makeText(getApplicationContext(), "Go to Setting.",
                    Toast.LENGTH_LONG).show();
                break;
        }
        return false;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

กำหนดการ  
แสดงผล

## 6. รันแอพพลิเคชัน แล้วคลิกปุ่มเมนูจะปรากฏ Menu ดังรูป

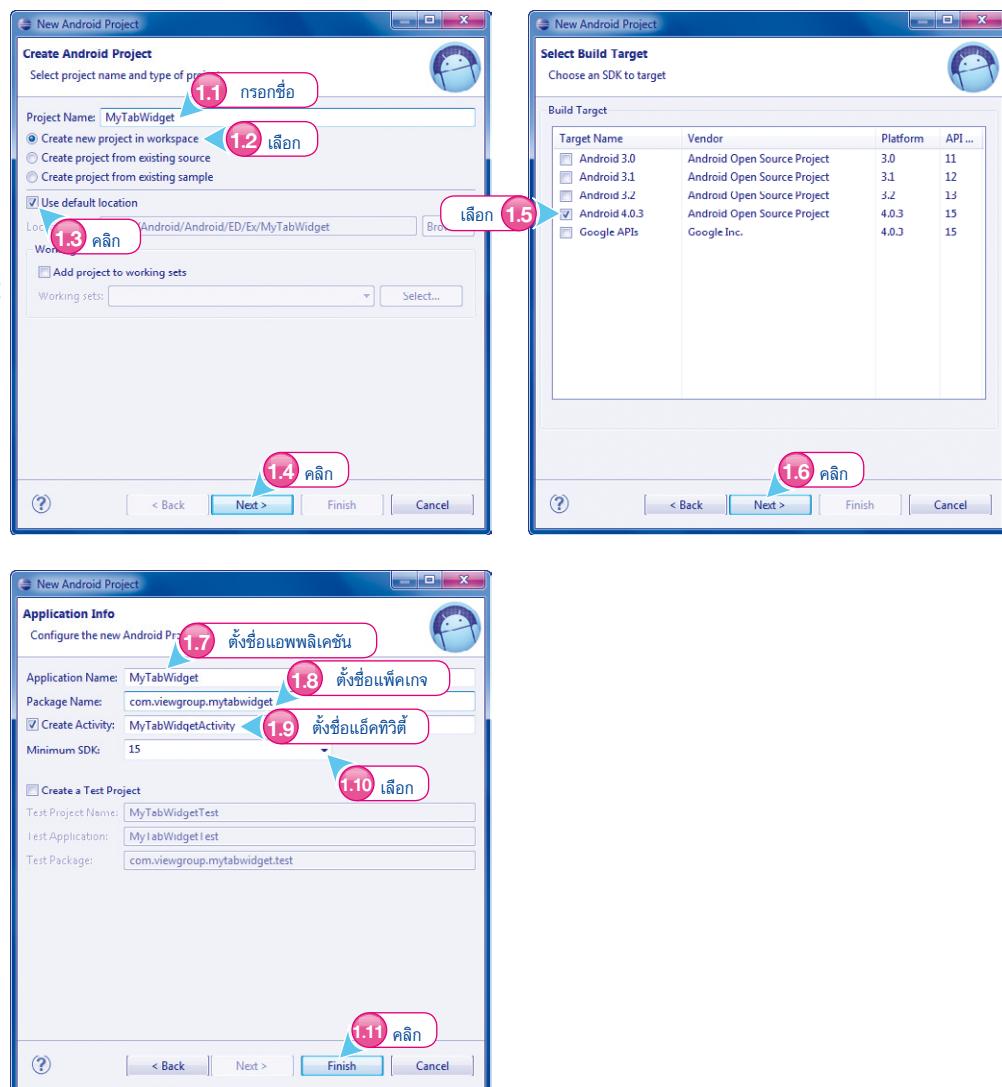


เมื่อคลิกปุ่ม Help จะแสดงข้อความ “Go to Help” และเมื่อคลิกปุ่ม Setting จะแสดงข้อความ “Go to Setting” ซึ่งผู้อ่านสามารถแก้ไขได้โปรแกรมในส่วนของเมธอด onOptionsItemSelected() เพื่อให้แอพพลิเคชันทำงานตามที่ต้องการได้

# TabWidget

TabWidget เป็นการแบ่งหน้าจอออกเป็นหลายๆ หน้า มีลักษณะเหมือนช้อนทับกันไว้ ซึ่งจะเป็นประโยชน์ในการสร้างแอปพลิเคชันที่มีเนื้อหามากๆ หรือในกรณีที่ต้องการแบ่งเนื้อหาออกเป็นหมวดหมู่ ทั้งนี้มีขั้นตอนการสร้างดังต่อไปนี้

- สร้างโปรเจกต์ใหม่ ในที่นี้ตั้งชื่อว่า MyTabWidget และกำหนดคุณสมบัติดังนี้



## 2. แก้ไขโค้ด XML ในไฟล์ main.xml ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TabHost
        android:id="@+id/tabhost"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <LinearLayout
            android:id="@+id/linearLayout1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical" >

            <TabWidget
                android:id="@+android:id/tabs"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" >
                </TabWidget>

                <FrameLayout
                    android:id="@+id/tabcontent"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" >
                    <!-- Tab 1 Content -->
                    <!-- Tab 2 Content -->
                    <!-- Tab 3 Content -->
                </FrameLayout>
            </LinearLayout>
        </TabHost>
    </LinearLayout>
```

กำหนดคุณสมบัติของ Tab ที่ 1

```
<LinearLayout
    android:id="@+id/tab1"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello Android."
        android:textAppearance="?android:attr/textAppearanceMedium" >
    </TextView>
</LinearLayout>
```

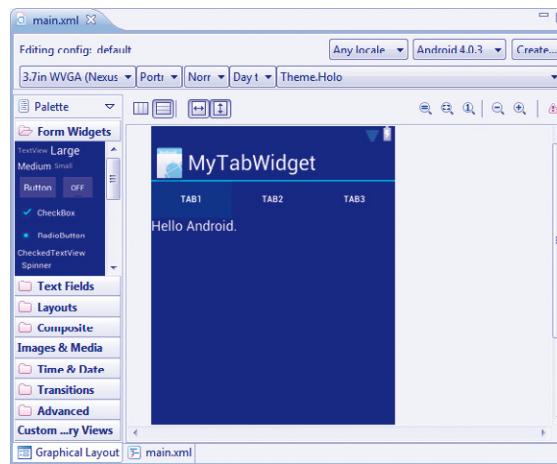
กำหนดคุณสมบัติของ Tab ที่ 2

```
<LinearLayout
    android:id="@+id/tab2"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
</LinearLayout>
```

กำหนดคุณสมบัติของ Tab ที่ 3

```
<LinearLayout
    android:id="@+id/tab3"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
</LinearLayout>
```

### 3. ได้หน้าตาของโปรแกรมดังนี้



### 4. แก้ไขโค้ด Java ในไฟล์ MyTabWidgetActivity.java ดังนี้

```
package com.viewgroup.mytabwidget;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TabHost;

public class MyTabWidgetActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        TabHost tabHost=(TabHost)findViewById(R.id.tabhost);
        tabHost.setup();
        TabSpec tabSpec=tabHost.newTabSpec("tab1");
        tabSpec.setContent(R.id.tab1);
        tabSpec.setIndicator("Home");
        tabHost.addTab(tabSpec);

        tabSpec=tabHost.newTabSpec("tab2");
        tabSpec.setContent(R.id.tab2);
        tabSpec.setIndicator("Setting");
        tabHost.addTab(tabSpec);

        tabSpec=tabHost.newTabSpec("tab3");
        tabSpec.setContent(new Intent(this,MyActivity.class));
        tabSpec.setIndicator("About",getResources().getDrawable(android.R.drawable.ic_dialog_info));
        tabHost.addTab(tabSpec);
    }
}
```

กำหนดคุณสมบัติ Tab ที่ 1

กำหนดคุณสมบัติ Tab ที่ 2

กำหนดคุณสมบัติ Tab ที่ 3

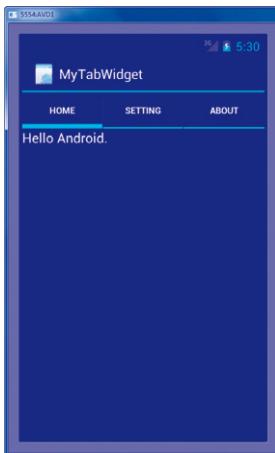
โดยที่

- **setContent()** เป็นเมธอดสำหรับกำหนดเนื้อหาภายในแท็บนั้นๆ ซึ่งผู้อ่านสามารถสร้างแอ็คทิวิตี้เตรียมไว้ก่อนนำมาแสดงในภายหลังได้ โดยกำหนดพารามิเตอร์ดังตัวอย่างต่อไปนี้

```
tabSpec.setContent(new Intent(this,<Activity Name>.class));
```

- **setIndicator()** เป็นเมธอดสำหรับกำหนดชื่อของแท็บ รวมไปถึงกำหนดรูปไอคอนของแท็บด้วย

## 5. เมื่อรันแอพพลิเคชันจะได้ผลลัพธ์ดังนี้



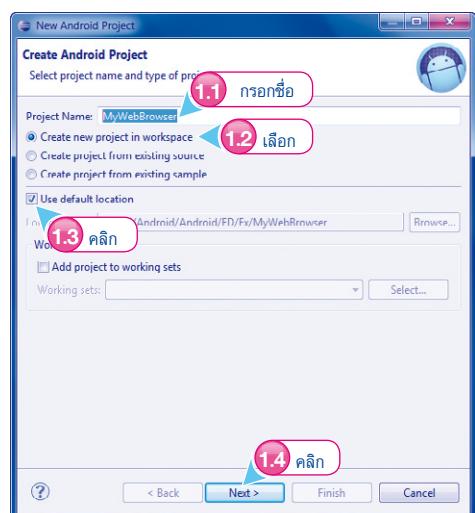
## WebView

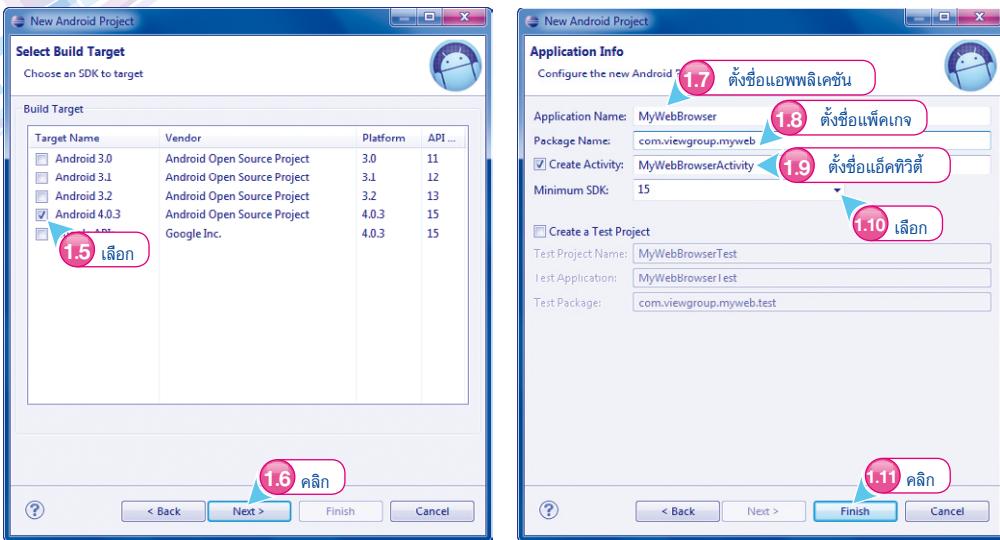
WebView เป็นตัวช่วยในการแสดงเว็บเพจ (Web Page) ในกรณีที่ผู้อ่านต้องการการแสดงผลเว็บเพจ (Web Page) ภายในแอพพลิเคชัน ที่สามารถทำได้โดยใช้ WebView

ในหัวข้อนี้ผู้เขียนจะอธิบายขั้นตอนการสร้างเว็บบราวเซอร์อย่างง่ายโดยใช้ WebView เพื่อเป็นตัวอย่างในการใช้งาน WebView ซึ่งมีขั้นตอนดังต่อไปนี้

### 1. สร้างโปรเจกต์ใหม่ ในที่นี้ชื่อว่า

MyWebBrowser และกำหนดคุณสมบัติ ดังนี้





## 2. แก้ไขโค้ด XML ในไฟล์ main.xml เพื่อจัดวางหน้าจอสำหรับแอพพลิเคชันดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <EditText
            android:id="@+id/editText1"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="0.3" >

            <requestFocus>
            </requestFocus>
        </EditText>

        <Button
            android:id="@+id/button1"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Go" >
    </Button>
</LinearLayout>
```

กำหนดคุณสมบัติของ LinearLayout

กำหนดคุณสมบัติของ EditText

กำหนดคุณสมบัติของ Button

```

<LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <WebView
        android:id="@+id/webView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
    </WebView>
</LinearLayout>

```

กำหนดคุณสมบัติของ LinearLayout

กำหนดคุณสมบัติของ WebView

3. แก้ไขโค้ด XML ในไฟล์ AndroidManifest.xml โดยเพิ่มแท็ก <uses-permission android:name="android.permission.INTERNET" /> เพื่อให้แอพพลิเคชันสามารถใช้งานอินเทอร์เน็ตได้ดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.viewgroup.myweb"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="12" />

    <uses-permission android:name="android.permission.INTERNET" /> ← เพิ่มแท็ก

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".MyWebBrowserActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

4. แก้ไขโค้ด Java ในไฟล์ MyWebBrowserActivity.java เพื่อให้แอพพลิเคชันสามารถทำงานได้ดังนี้

```

package com.viewgroup.myweb;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;

```



```

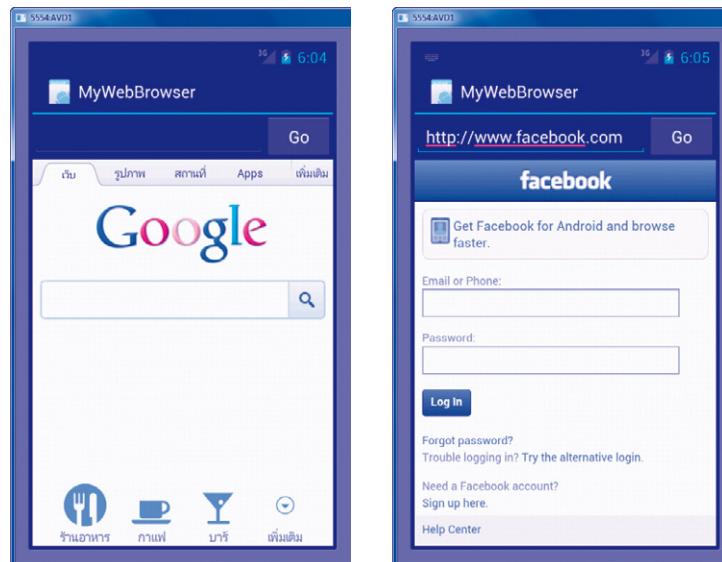
import android.widget.TextView;

public class MyWebBrowserActivity extends Activity {
    WebView webview;
    Button bt_go;
    TextView txt_address;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        webview = (WebView) findViewById(R.id.webView1);
        webview.setWebViewClient(new WebViewClient());
        webview.getSettings().setJavaScriptEnabled(true);
        txt_address = (TextView) findViewById(R.id.editText1);
        webview.loadUrl("http://www.google.com"); ← • กำหนด URL
        bt_go = (Button) findViewById(R.id.button1);
        bt_go.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                String url = txt_address.getText().toString().trim();
                if (!url.equals("")) { ← • ตรวจสอบการกรอก URL
                    webview.loadUrl(url);
                }
            }
        });
    }
}

```

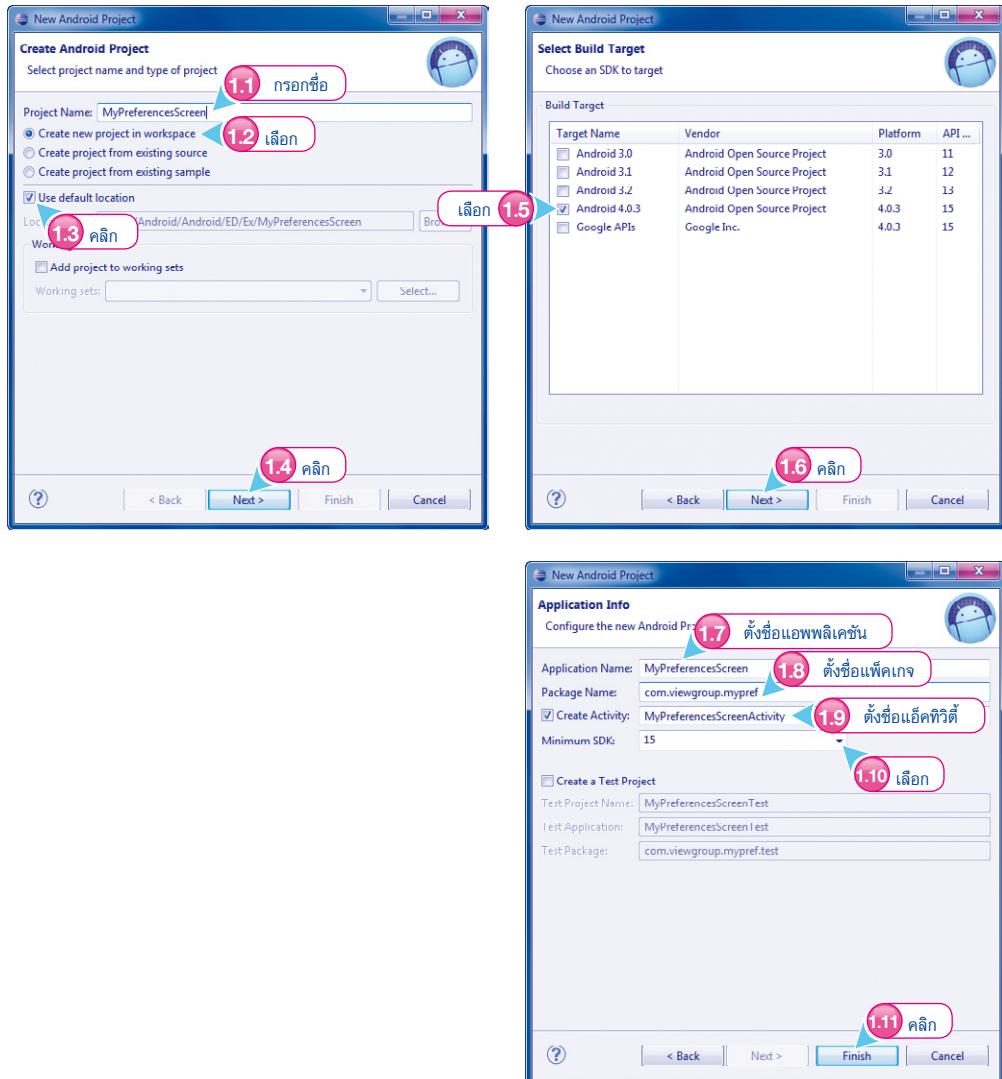
### 5. รันแอพพลิเคชันเพื่อทดสอบการทำงาน จะได้ผลลัพธ์ดังรูป



# Preferences Screen

Preferences Screen เป็นรูปแบบหน้าจอสำหรับตั้งค่าต่างๆ ของแอพพลิเคชัน ยกตัวอย่างเช่น การตั้งค่า Wireless & network ในแอนดรอยด์ ซึ่ง Preferences Screen ก็จะประกอบไปด้วยอิลิเมนต์ภายใน เช่น CheckBoxPreference, ListPreference และ EditTextPreference เป็นต้น สำหรับการสร้าง Preferences Screen ขึ้นมาใช้งานนั้น สามารถทำได้ตามขั้นตอนดังต่อไปนี้

- สร้างโปรเจกต์ใหม่ ในที่นี้ตั้งชื่อว่า MyPreferencesScreen และกำหนดคุณสมบัติดังนี้



2. สร้างไฟล์ arrays.xml ไว้ในลั่นของ res/values เพื่อเป็นข้อมูลใน ListPreferences ดังนี้



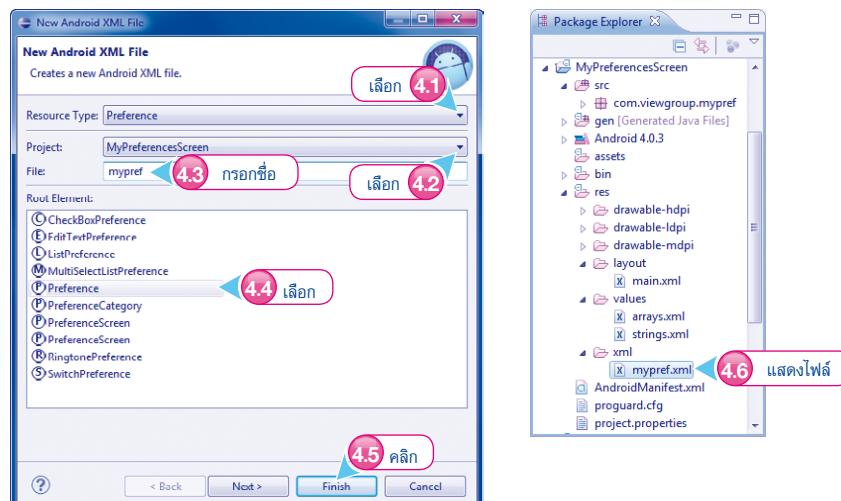
3. เพิ่มโค๊ด XML ในไฟล์ arrays.xml ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string-array name="listName">
        <item>Option 1</item>
        <item>Option 2</item>
        <item>Option 3</item>
    </string-array>
    <string-array name="listValue">
        <item>1</item>
        <item>2</item>
        <item>3</item>
    </string-array>

</resources>
```

4. สร้างไฟล์ mypref.xml เพื่อสร้างหน้าจอ Preference ดังนี้



## 5. เพิ่มโค้ด XML ในไฟล์ mypref.xml ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >

    <PreferenceCategory android:title="Preference Category" >

        <CheckBoxPreference
            android:defaultValue="false"
            android:key="checkboxPref"
            android:summary="true or false"
            android:title="Checkbox Preference" />

        <ListPreference
            android:defaultValue="0"
            android:entries="@array/listName"
            android:entryValues="@array/listValue"
            android:key="listPref"
            android:summary="Select value from array"
            android:title="List Preference" />

        <EditTextPreference
            android:name="EditText Preference"
            android:defaultValue="Nothing"
            android:key="editTextPref"
            android:summary="Please enter you string"
            android:title="Edit This Text" />

        <RingtonePreference
            android:name="Ringtone Preference"
            android:key="ringtonePref"
            android:summary="Select a ringtone"
            android:title="Ringtones" />

        <Preference
            android:key="customPref"
            android:summary="Please touch here to start MyActivity."
            android:title="Custom Preference" />
    </PreferenceCategory>

</PreferenceScreen>
```

จากโค้ด XML เป็นการสร้าง CheckBoxPreference, ListPreference, EditTextPreference, RingtonePreference และ CustomPreference

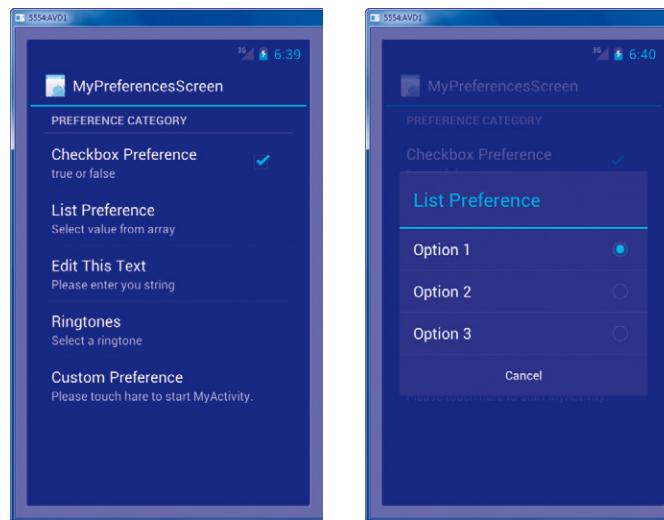
## 6. แก้ไขโค้ด Java ในไฟล์ MyPreferencesScreenActivity.java ดังนี้

```
package com.viewgroup.mypref;

import android.os.Bundle;
import android.preference.PreferenceActivity;

public class MyPreferencesScreenActivity extends PreferenceActivity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.mypref); ← ● กำหนดหน้าจอ
    }
}
```

7. เมื่อรันแอพพลิเคชันจะได้ผลลัพธ์ดังนี้



8. แก้ไขโค้ด Java ในไฟล์ MyPreferencesScreenActivity.java เพื่อนำค่าจาก Preference มาใช้งาน และเพิ่มเหตุการณ์เมื่อคลิกปุ่ม Custom Preference ดังนี้

```
package com.viewgroup.mypref;

import android.app.Activity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.Preference;
import android.preference.Preference.OnPreferenceClickListener;
import android.preference.PreferenceActivity;
import android.preference.PreferenceManager;
import android.widget.Toast;

public class MyPreferencesScreenActivity extends PreferenceActivity {
    boolean CheckboxPreference;
    String ListPreference;
    String editTextPreference;
    String ringtonePreference;
    String customPref;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.mypref);

        Preference customPref = (Preference) findPreference("customPref");
        customPref
            .setOnPreferenceClickListener(new OnPreferenceClickListener() {

                public boolean onPreferenceClick(Preference preference) {
                    getPrefValue();
                    Toast.makeText(

```

```

        getBaseContext(),
        "Custom preference has been clicked"
        + "\nCheck : " + CheckboxPreference
        + "\nList : " + ListPreference
        + "\nEditText : " + editTextPreference
        + "\nRingtone : " + ringtonePreference,
        Toast.LENGTH_LONG).show();

        return true;
    }

});

}

private void getPrefValue() {

    SharedPreferences prefs = PreferenceManager
        .getDefaultSharedPreferences(getApplicationContext());
    CheckboxPreference = prefs.getBoolean("checkboxPref", true);
    ListPreference = prefs.getString("listPref", "0");
    editTextPreference = prefs.getString("editTextPref",
        "Nothing has been entered");
    ringtonePreference = prefs.getString("ringtonePref",
        "DEFAULT_RINGTONE_URI");
    SharedPreferences mySharedPreferences = getSharedPreferences(
        "myCustomSharedPrefs", Activity.MODE_PRIVATE);
    customPref = mySharedPreferences.getString("cusomPref", "");
}
}

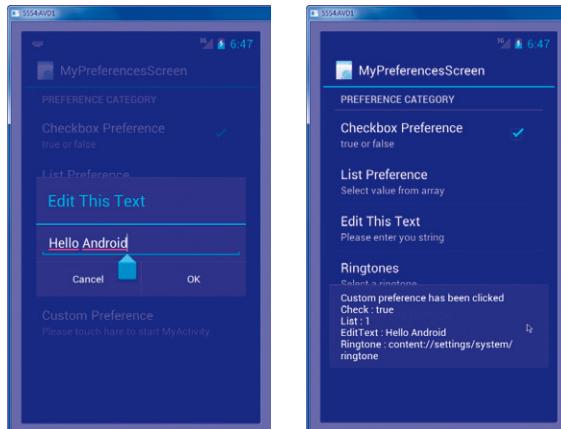
```

โดยที่

- **setOnPreferenceClickListener()** เป็นเมธอดสำหรับเพิ่มเหตุการณ์ เมื่อมีการคลิกปุ่ม Custom Reference ที่สร้างขึ้น
- **getPrefValue()** เป็นเมธอดที่สร้างขึ้นเพื่อดึงค่าปัจจุบันของแต่ละ Preference มาเก็บไว้ในตัวแปร

ในตัวอย่างนี้ผู้เขียนจะให้แอพพลิเคชันแสดงค่าปัจจุบันของแต่ละ Preference เมื่อคลิกปุ่ม Custom Reference

### 9. เมื่อรันแอพพลิเคชันจะได้ผลลัพธ์ดังนี้

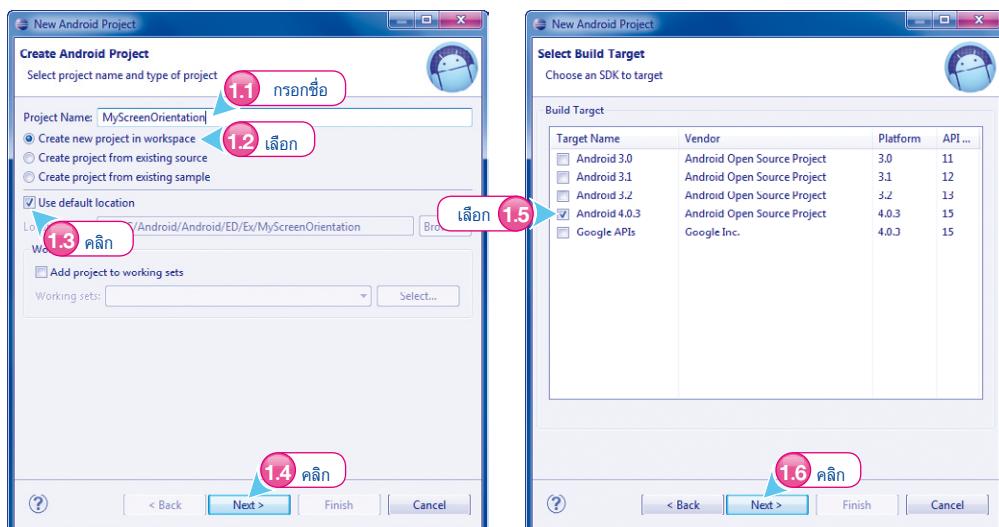


# การจัดวางหน้าจอ (Screen Orientation)

โดยทั่วไปแล้วแอพพลิเคชันจะสนับสนุนการทำงานหน้าจอแนวอน (Landscape) และแนวตั้ง (Portrait) ยกตัวอย่างเช่น เกม Angry Birds ที่จัดวางหน้าจอในแนวอนเพื่อให้เหมาะสมกับการเล่น

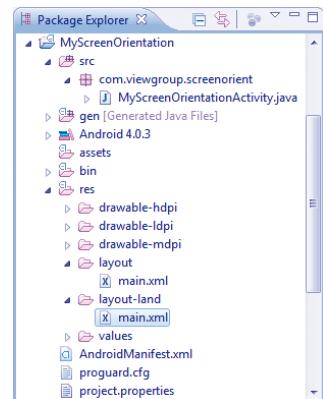
ในหัวข้อนี้ ผู้อ่านจะได้ศึกษาการพัฒนาแอพพลิเคชันให้สนับสนุนการจัดวางหน้าจอแนวอนและแนวตั้ง ซึ่งมีขั้นตอนดังนี้

- สร้างโปรเจกต์ใหม่ ในที่นี้ตั้งชื่อว่า MyScreenOrientation และกำหนดคุณสมบัติดังนี้



โดยปกติแล้วหลังจากที่สร้างโปรเจกต์ขึ้นมาใหม่ ไฟล์ main.xml ในส่วนของ res/layout จะเป็นไฟล์สำหรับลรังหน้าจอที่วางในแนวตั้ง

2. สร้างไฟล์เดอร์ layout-land ในล่างของ res (โดยคลิกขวาที่ res) และสร้างไฟล์ main.xml ไว้ภายในดังรูป

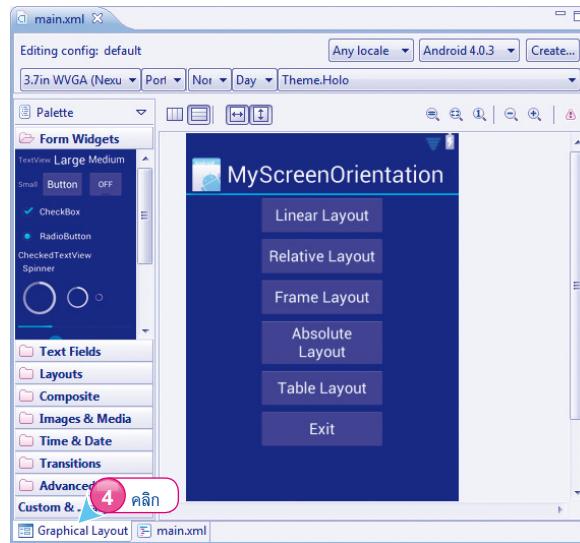


3. แก้ไขโค้ด XML ในไฟล์ main.xml เพื่อใช้สำหรับการจัดวางหน้าจอแนวตั้งดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_horizontal">

    <Button android:text="Linear Layout"
        android:id="@+id/button1"
        android:layout_width="120dp"
        android:layout_height="wrap_content">
    </Button>
    <Button android:text="Relative Layout"
        android:id="@+id/button2"
        android:layout_width="120dp"
        android:layout_height="wrap_content">
    </Button>
    <Button android:text="Frame Layout"
        android:id="@+id/button3"
        android:layout_width="120dp"
        android:layout_height="wrap_content">
    </Button>
    <Button android:text="Absolute Layout"
        android:id="@+id/button4"
        android:layout_width="120dp"
        android:layout_height="wrap_content">
    </Button>
    <Button android:text="Table Layout"
        android:id="@+id/button5"
        android:layout_width="120dp"
        android:layout_height="wrap_content">
    </Button>
    <Button android:text="Exit"
        android:id="@+id/button6"
        android:layout_width="120dp"
        android:layout_height="wrap_content">
    </Button>
</LinearLayout>
```

#### 4. คลิกแท็บ Graphical Layout จะปรากฏหน้าจอแนวตั้งดังรูป



#### 5. แก้ไขโค้ด XML ในไฟล์ main.xml ในส่วนของ res/layout-land เพื่อใช้สำหรับการจัดวางหน้าจอแนวอนดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <LinearLayout
        android:id="@+id/linearLayout3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical" > ←• กำหนดรูปแบบ Layout

        <LinearLayout
            android:id="@+id/linearLayout1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal" > ←• กำหนดรูปแบบ Layout

            <Button
                android:id="@+id/button1"
                android:layout_width="160dp"
                android:layout_height="wrap_content"
                android:text="Linear Layout" >
            </Button>

            <Button
                android:id="@+id/button2"
                android:layout_width="160dp"
```

```

        android:layout_height="wrap_content"
        android:text="Relative Layout" >
    </Button>

    <Button
        android:id="@+id/button3"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:text="Frame Layout" >
    </Button>
</LinearLayout>

<LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" > ←● ก้าวนดูรูปแบบ Layout

    <Button
        android:id="@+id/button4"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:text="Absolute Layout" >
    </Button>

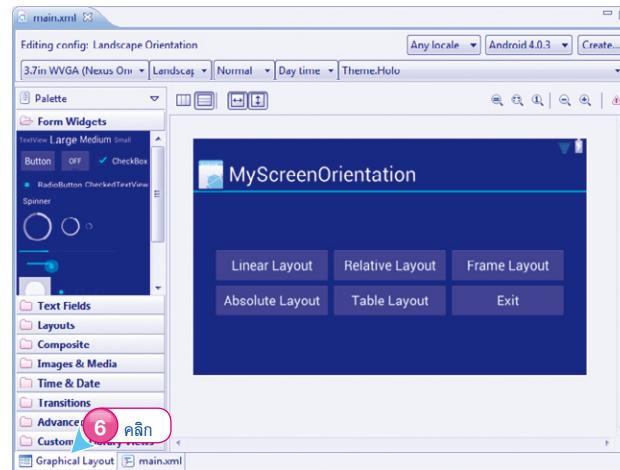
    <Button
        android:id="@+id/button5"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:text="Table Layout" >
    </Button>

    <Button
        android:id="@+id/button6"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:text="Exit" >
    </Button>
</LinearLayout>
</LinearLayout>

</LinearLayout>

```

**6. คลิกแท็บ Graphical Layout จะปรากฏหน้าจอแนวโน้มดังรูป**



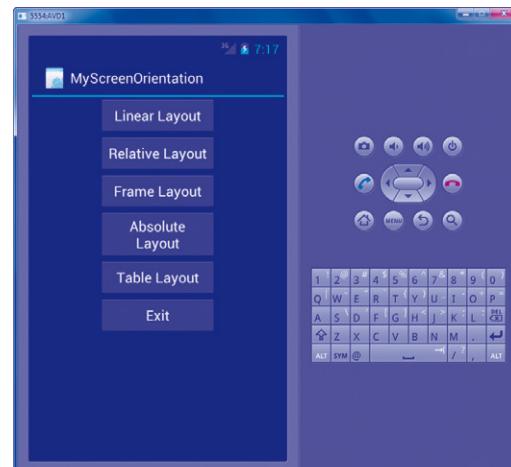
**7. แก้ไขโค้ด Java ในไฟล์ MyScreenOrientationActivity.java ดังนี้**

```
package com.viewgroup.screenorient;

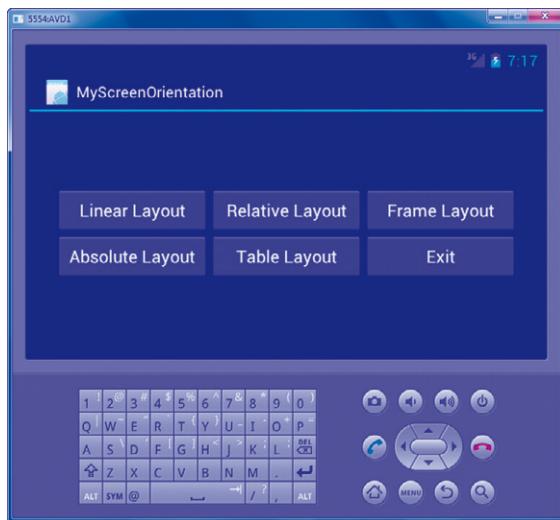
import android.app.Activity;
import android.os.Bundle;

public class MyScreenOrientationActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

**8. เมื่อรันแอพพลิเคชันจะแสดงหน้าจอในแนวตั้งดังรูป**



9. กดปุ่ม **Ctrl + F11** เพื่อเปลี่ยนหน้าจอเป็นแนวอน จะได้ผลลัพธ์ดังรูป



10. ในกรณีที่ผู้อ่านต้องการจัดวางหน้าจอในแนวอน หรือแนวตั้งเพียงอย่างเดียว สามารถทำได้โดยเพิ่มพร็อพเพอร์ตี้ `android:screenOrientation="landscape"` ของแท็ก `Activity` ในไฟล์ `AndroidManifest.xml` ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.viewgroup.screenorient"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="12" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".MyScreenOrientationActivity"
            android:screenOrientation="landscape" ←● เพิ่มโคด
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```



# สรุปหัวข้อ

ในบทนี้เป็นการอธิบายถึงการใช้งานวิวกรุ๊ปสำหรับพัฒนาแอพพลิเคชัน เพื่อออกแบบล่าวนติดต่อกับผู้ใช้งานได้อย่างเหมาะสม ทั้ง ListView, ScrollView, Image Switcher View, Options Menu และ TabWidget

## แบบฝึกหัดหัวข้อ

จงสร้างแอพพลิเคชันสำหรับแสดงภาพในรูปแบบแค็ตตาล็อก (Catalog) หรือ Photo Gallery อย่างง่าย โดยแบ่งเป็นหมวดหมู่ เช่น ภาพวิว ภาพถ่าย ภาพที่อยู่อาศัย และภาพการ์ตูน จากนั้นเมื่อผู้อ่านคลิกเลือก ในแต่ละหมวด แอพพลิเคชันก็จะเปิดคลังภาพในแต่ละหมวดขึ้นมาแสดง

## การพัฒนาแอพพลิเคชันรับส่งข้อความ (SMS)

เนื้อหาในบทที่ 8 จะเป็นล่วงของการฝึกพัฒนาแอพพลิเคชันรับส่งข้อความลับ (SMS) ซึ่งมีใช้งานอยู่ทั่วไปบนเครือข่ายของโทรศัพท์เคลื่อนที่

การรับส่งข้อความลับ หรือที่เรียกว่า SMS (Short Message Service) เป็นบริการทางการลือสารประเพณีหนึ่ง ซึ่งอนุญาตให้ผู้ใช้งานสามารถส่งข้อความลับๆ ไปยังอุปกรณ์สื่อสารอีกเครื่องหนึ่งได้ โดยทั่วไปแล้วจะเป็นบริการที่ใช้ในโทรศัพท์เคลื่อนที่หรือคอมพิวเตอร์ทั่วไป

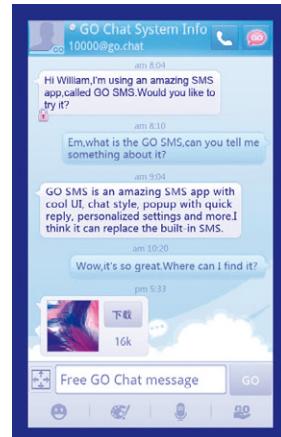
ทั้งนี้บริการ SMS นั้นได้ถูกพัฒนามาจากการส่งเพจ (Page) โดยใช้โปรโตคอล (Protocol) ตามมาตรฐานของโทรศัพท์เคลื่อนที่ และภายหลังได้ถูกพัฒนาขึ้นบนเทคโนโลยีโทรศัพท์เคลื่อนที่แบบ GSM (Global System for Mobile Communications) ซึ่งสามารถส่งข้อความได้มากที่สุดเป็นจำนวน 160 ตัวอักษร ซึ่งปัจจุบันการส่ง SMS นั้นสามารถส่งได้จากเครื่องหนึ่งไปยังเครื่องหนึ่ง (Unicast) หรือมากกว่า 1 เครื่อง (Broadcast) ได้

โดยเนื้อหาในบทนี้ จะแบ่งออกเป็นขั้นตอนให้ศึกษาดังนี้

1. หลักการทำงานของแอพพลิเคชันรับส่งข้อความ
2. เริ่มต้นสร้างโปรเจกต์
3. ออกแบบหน้าจอแอพพลิเคชัน
4. เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน
5. ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator)

# หลักการทำงานของแอพพลิเคชันรับส่งข้อความ

สำหรับการพัฒนาแอพพลิเคชันใดๆ ก็ตาม ผู้พัฒนาแอพพลิเคชัน จะต้องเข้าใจหลักการทำงานของแอพพลิเคชันในเบื้องต้นก่อน ซึ่งในบทนี้ ผู้อ่านจะได้ฝึกพัฒนาแอพพลิเคชันรับส่งข้อความ ซึ่งมีรายละเอียดของ แอพพลิเคชันดังต่อไปนี้



ตัวอย่างแอพพลิเคชันรับส่ง SMS ▶

## NOTE



สำหรับการพัฒนาแอพพลิเคชันบน Android แอพพลิเคชันนั้นๆ จะต้องมีการตั้งค่าการอนุญาต (Permission) ซึ่งก็คือ การกำหนดให้แอพพลิเคชันสามารถทำอะไรได้บ้าง เช่น ค่า INTERNET ที่ต้องการ อนุญาตให้แอพพลิเคชันนั้นๆ สามารถเชื่อมต่อ กับอินเทอร์เน็ตได้

สำหรับในกรณีของแอพพลิเคชันรับส่ง SMS นั้น ค่า SEND\_SMS จะเป็นการอนุญาตให้แอพพลิเคชัน รับส่ง SMS ได้ โดยสามารถส่งข้อความระหว่างกันได้ันั่นเอง ซึ่งรายละเอียดการกำหนดค่า Permission จะอธิบายในขั้นตอนการพัฒนาแอพพลิเคชันรับส่ง SMS ต่อไป

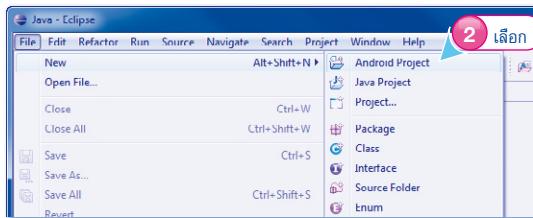
- หน้าจอโทรศัพท์จำลอง หรือ Emulator จะมีส่วนประกอบต่างๆ ดังนี้
  1. ช่องกรอกล่องข้อความสำหรับใส่เลขหมายโทรศัพท์ (Enter the phone number of recipient)
  2. ช่องกรอกล่องข้อความที่ใช้สำหรับพิมพ์ข้อความ (Message)
  3. ปุ่มกด (Button) ที่ใช้สำหรับส่งข้อความ (Send SMS)
- การทำงานของแอพพลิเคชันรับส่ง SMS ที่จะพัฒนา มีขั้นตอนการทำงานดังนี้
  1. เริ่มต้นจากผู้ใช้แอพพลิเคชันพิมพ์หมายเลขโทรศัพท์
  2. พิมพ์ข้อความที่ต้องการส่งไปยังหมายเลขปลายทาง
  3. ผู้ใช้คลิกปุ่ม Send SMS
  4. ข้อความนั้นจะถูกกล่าวไปยังหมายเลขปลายทาง

ในส่วนของการทดสอบแอพพลิเคชันรับส่ง SMS นั้น ให้จำลอง Emulator ขึ้นมาเป็นจำนวน 2 เครื่อง เพื่อใช้ทดสอบการรับส่งข้อความระหว่างกัน (Receive/Send SMS)

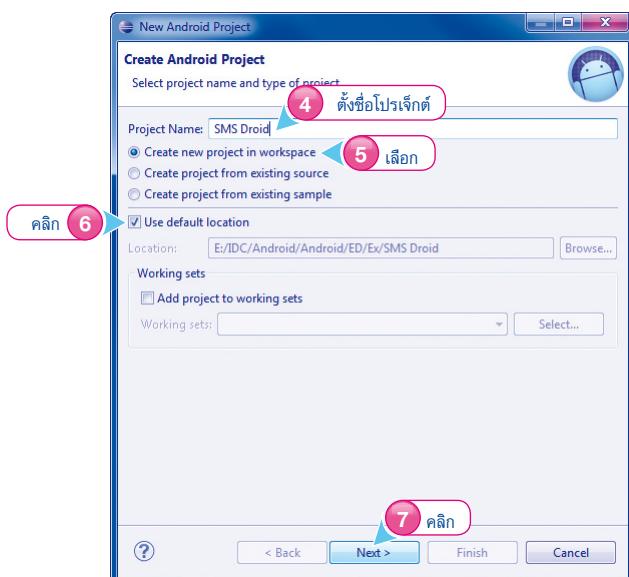
# เริ่มต้นสร้างโปรเจกต์

เริ่มต้นพัฒนาโปรแกรมโดยสร้างโปรเจกต์สำหรับเขียนโปรแกรม ซึ่งผู้อ่านสามารถทำได้ตามขั้นตอนดังต่อไปนี้

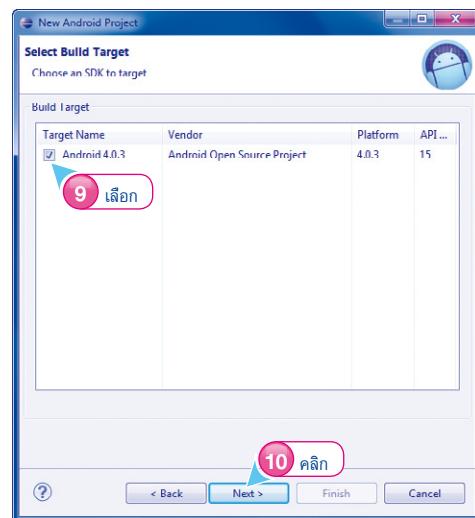
1. เปิดโปรแกรม Eclipse
2. คลิกเมนู File > New > Android Project



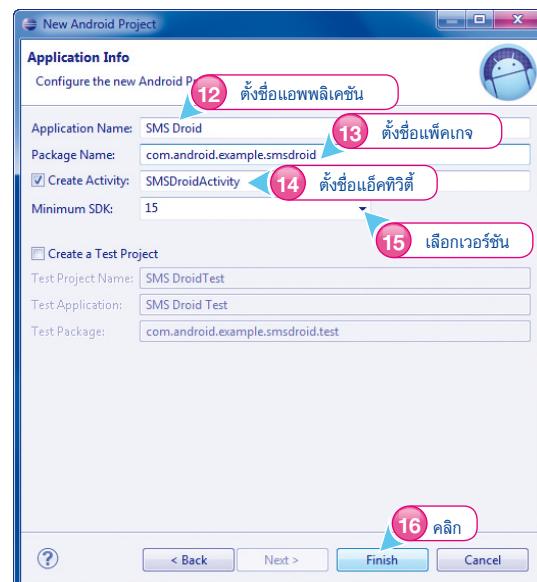
3. จะปรากฏ dialogue ให้ออกบ็อกซ์ New Android Project
4. ในส่วนแรกสำหรับกำหนดชื่อและรูปแบบของโปรเจกต์ ในที่นี้ตั้งชื่อโปรเจกต์ว่า SMS Droid
5. ในส่วนของ Contents คลิกเลือก Create new project in workspace เพื่อสร้างโปรเจกต์ใหม่
6. คลิกเครื่องหมายถูกหน้า Use default location เพื่อพัฒนาแอพพลิเคชันในตำแหน่งที่อยู่เดียวกัน กับโปรเจกต์ Eclipse
7. คลิกปุ่ม **Next >**



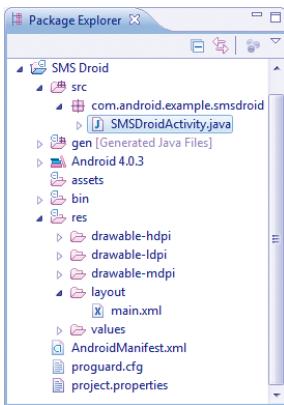
8. จะปรากฏ dialogue ตัวเลือก New Android Project ในส่วนที่ส่อง สำหรับเลือกเวอร์ชันของ Android
9. คลิกเลือกเวอร์ชันของ Android ที่จะพัฒนา ในที่นี่เลือก Android 4.0.3
10. คลิกปุ่ม **Next >**



11. จะปรากฏ dialogue ตัวเลือก New Android Project ในส่วนที่สาม สำหรับกำหนดรายละเอียดของแอพพลิเคชัน
12. ตั้งชื่อของแอพพลิเคชันคือ SMS Droid
13. ตั้งชื่อแพ็คเกจเป็น com.android.example.smsdroid
14. ตั้งชื่อแอ็คทิวิตี้ ในที่นี่ตั้งชื่อเป็น SMSDroidActivity
15. ในส่วนของ Minimum SDK: คลิกเลือกเวอร์ชัน 15 ซึ่งเป็นเวอร์ชันเดียวกันกับ API Level ที่เลือกในข้อ 9
16. คลิกปุ่ม **Finish** ซึ่งถือเป็นอันเสร็จลั่นการสร้างโปรเจกต์



## 17. จะปรากฏโปรเจกต์ที่ Package Explorer ซึ่งแสดงถึงโครงสร้างของโปรเจกต์ที่ได้สร้างขึ้นมา



หลังจากที่สร้างโปรเจกต์เสร็จลืนแล้ว ขั้นตอนต่อไปจะเป็นการแก้ไขไฟล์หลักๆ ซึ่งในบทนี้จะมี 4 ไฟล์ดังนี้

- ไฟล์ AndroidManifest.xml ซึ่งถูกใช้เพื่อกำหนดลิธีให้โปรแกรมสามารถรับและส่ง SMS ได้
- ไฟล์ main.xml ซึ่งจะใช้เป็นไฟล์สำหรับรับหน้าจอที่อยู่บนอุปกรณ์ โดยสามารถกำหนดให้มีรูปแบบตามความต้องการได้
- ไฟล์ SMSDroidActivity.java ซึ่งเป็นไฟล์หลักที่ใช้สำหรับกำหนดการทำงานของแอพพลิเคชันในส่วนของการส่งข้อความ
- ไฟล์ SMSDroidReceiver.java ซึ่งเป็นไฟล์หลักที่ใช้สำหรับกำหนดการทำงานของแอพพลิเคชันในส่วนของการรับข้อความ

## กำหนดสิทธิ์การทำงานให้แอพพลิเคชัน

การกำหนดลิธีการทำงานให้แอพพลิเคชันเป็นการแก้ไขโค้ด XML ในไฟล์ AndroidManifest.xml โดยจะแบ่งออกเป็น 2 ส่วนคือ

- ส่วนที่ 1** กำหนดให้แอพพลิเคชันมีขีดความสามารถต่างๆ เช่น ในกรณีของโปรแกรมรับส่ง SMS นั้นจะกำหนดลิธีให้แอพพลิเคชันสามารถรับส่ง SMS ได้จริงระหว่างแอพพลิเคชันสองเครื่อง ซึ่งจะมีการเพิ่มแทร็คการอนุญาตการให้ลิธี (Permission) ในไฟล์ AndroidManifest.xml ดังนี้

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```



- ส่วนที่ 2 จะเป็นการเพิ่มความสามารถให้ไฟล์ SMSDroidReceiver.java ให้สามารถตักจับ SMS ที่ถูกส่งเข้ามายังเครื่องได้ (ส่วนนี้จะเพิ่มในไฟล์ SMSDroidReceiver.java)

The screenshot shows the AndroidManifest.xml file in the Android Studio manifest editor. A red dashed box highlights the three permission tags: <uses-permission android:name="android.permission.READ\_SMS" />, <uses-permission android:name="android.permission.SEND\_SMS" />, and <uses-permission android:name="android.permission.RECEIVE\_SMS" />. A red arrow points from the text "เพิ่มโค้ด" (Add code) to this highlighted area.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.example.smsdroid"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="15" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <receiver android:name=".SmsDroidReceiver" >
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
                <action>
                </action>
            </intent-filter>
        </receiver>
        <activity
            android:name=".SMSDroidActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Manifest Application Permissions Instrumentation AndroidManifest.xml

## ອອກແບບหน້າຈອງໄວພວລີເຄເບັນ

การອອກແນບหน້າຈອງເອພພລິເຄຂັນເປັນການແກ້ໄຂໄຟລໍ XML ສໍາຫັບເອພພລິເຄຂັນຮັບສ່ງ SMS (Short Message Service) ນັ້ນສາມາດໃຫ້ໄດ້ໂດຍແກ້ໄຂໄຟລໍ main.xml ຊຶ່ງເປັນໄຟລໍທັກສໍາຫັບການອອກແນບໜ້າຈອງຂອງເອພພລິເຄຂັນ ໂດຍຜ້ອງໆສາມາດແກ້ໄຂໂຄດ XML ໃນໄຟລໍ main.xml ໄດ້ດັ່ງນີ້

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:padding="5dp" >

    <TextView <span style="color: red;">12

```

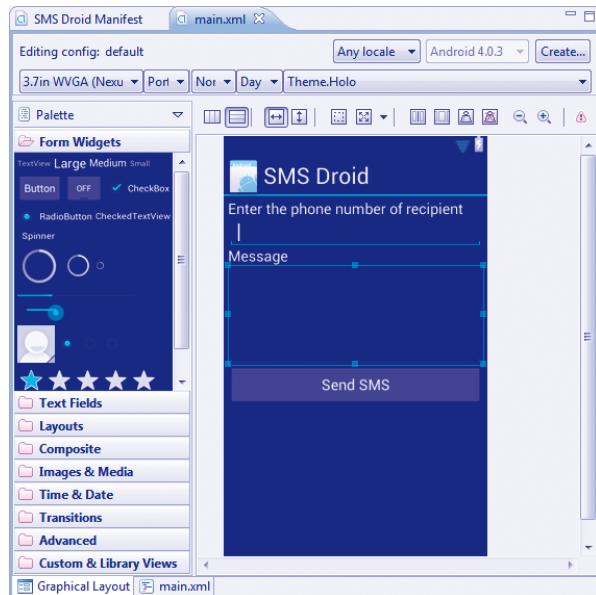
```

<requestFocus />
</EditText>

<TextView <span style="color: red;">123

```

ชี้ไปยังผลลัพธ์ดังนี้



▲ หน้าจอแอปพลิเคชันที่ต้องการพัฒนา

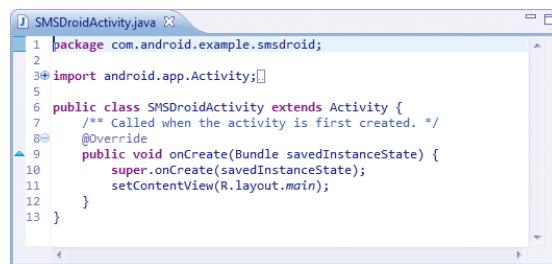
# เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน

ในการเขียนโค้ดกำหนดการทำงานของแอพพลิเคชันนี้ จะต้องเขียน 2 ส่วนคือ โค้ดในส่วนของการส่งข้อความและโค้ดในส่วนของการรับข้อความ ซึ่งมีรายละเอียดดังนี้

## การเขียนโค้ดส่วนของการส่งข้อความ

เมื่อผู้อ่านสร้างหน้าจอแอพพลิเคชันเรียบร้อยแล้ว ขั้นตอนต่อไปคือ การเขียนโค้ด Java ที่ไฟล์ SMSDroidActivity.java เพื่อใช้กำหนดการทำงานในการส่งข้อความระหว่าง Emulator ทั้ง 2 เครื่อง โดยมีขั้นตอนดังต่อไปนี้

- เริ่มต้นโดยการดับเบิลคลิกไฟล์ SMSDroidActivity.java ขึ้นมา ซึ่งจะมีโค้ดดังรูป



ทั้งนี้ setContentView(R.layout.main) จะเป็นส่วนของการนำไฟล์ main.xml มาด (Draw) เป็นหน้าจอ ในส่วนของ R.layout.main เป็นการอ้างอิงไปยังไฟล์ main.xml ซึ่งผู้อ่านสามารถสร้างไฟล์ main.xml ขึ้นมาใหม่ได้ทั้งหมด เพื่อออกแบบหน้าจอแอพพลิเคชันตามความต้องการได้เอง โดยไม่ต้องใช้ main.xml ที่ผู้เขียนสร้างไว้ก็ได้

- แก้ไขโค้ด Java ในไฟล์ SMSDroidActivity.java ดังนี้

```

package com.android.example.smsdroid;

import android.app.Activity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SMSDroidActivity extends Activity {

    Button btnSendSMS;
    EditText txtPhoneNo;
    EditText txtMessage;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
        txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
    }
}

```

```

txtMessage = (EditText) findViewById(R.id.txtMessage);
btnSendSMS.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        String phoneNo = txtPhoneNo.getText().toString();
        String message = txtMessage.getText().toString();
        if (phoneNo.length() > 0 && message.length() > 0) {
            SmsManager sms = SmsManager.getDefault();
            sms.sendTextMessage(phoneNo, null, message, null, null);
        } else {
            Toast.makeText(getApplicationContext(),
            "Please enter both phone number and message.",
            Toast.LENGTH_SHORT).show();
        }
    }
});
}
}

```

## อธิบายโค้ดการทำงานของแอพพลิเคชัน

- นำเข้าคลาส SmsManager ที่ใช้ในการรับส่ง SMS

```
import android.telephony.SmsManager;
```

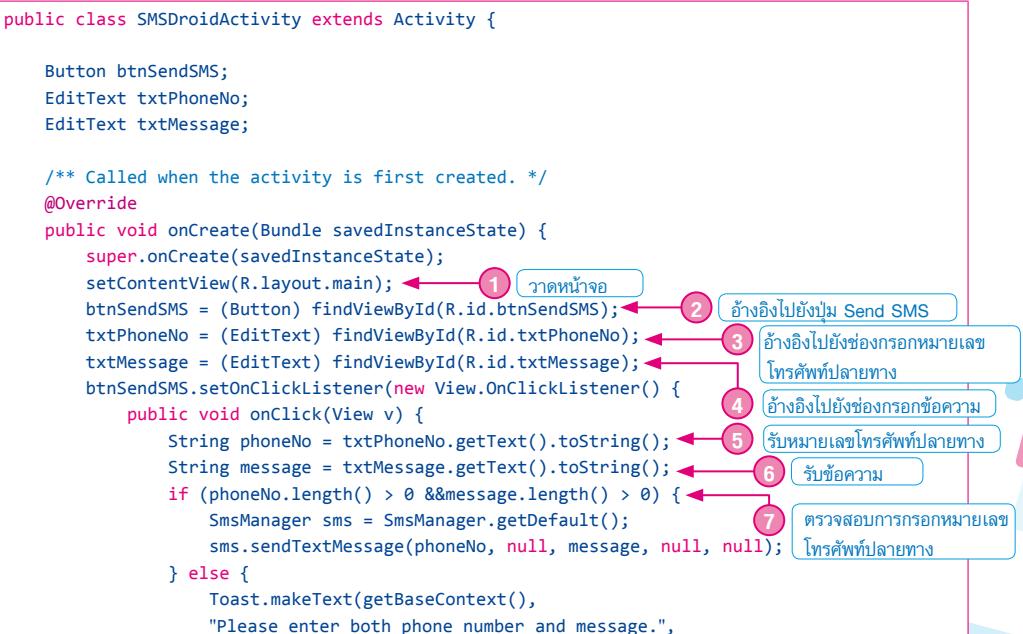
- เขียนโค้ดหลักเพื่อใช้อ้างอิงไปยัง SendSMS (Button), PhoneNo (EditText) และ Message (EditText) จากอิลิเมนต์ของไฟล์ main.xml ทั้งนี้ในการรับส่ง SMS นั้นผู้อ่านจะต้องใส่หมายเลขโทรศัพท์ และพิมพ์ข้อความที่จะส่งลงในกล่องข้อความ จากนั้นคลิกปุ่ม Send SMS ซึ่งการอ้างอิงนั้นก็เพื่อใช้จัดการกับค่าที่พิมพ์เข้ามาหนึ่ง โดยจะลังเกตได้ว่าจะเป็นการนำค่ามาใช้ในขั้นตอนต่อไป

```

public class SMSdroidActivity extends Activity {

    Button btnSendSMS;
    EditText txtPhoneNo;
    EditText txtMessage;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main); ← 1 วิธีหน้าจอ
        btnSendSMS = (Button) findViewById(R.id.btnSendSMS); ← 2 อ้างอิงไปยังปุ่ม Send SMS
        txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo); ← 3 อ้างอิงไปยังช่องกรอกหมายเลขโทรศัพท์ปลายทาง
        txtMessage = (EditText) findViewById(R.id.txtMessage); ← 4 อ้างอิงไปยังช่องกรอกข้อความ
        btnSendSMS.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                String phoneNo = txtPhoneNo.getText().toString(); ← 5 รับหมายเลขโทรศัพท์ปลายทาง
                String message = txtMessage.getText().toString(); ← 6 รับข้อความ
                if (phoneNo.length() > 0 && message.length() > 0) { ← 7 ตรวจสอบการกรอกหมายเลขโทรศัพท์ปลายทาง
                    SmsManager sms = SmsManager.getDefault();
                    sms.sendTextMessage(phoneNo, null, message, null, null);
                } else {
                    Toast.makeText(getApplicationContext(),
                    "Please enter both phone number and message.",
```



The diagram illustrates the flow of the application's logic:

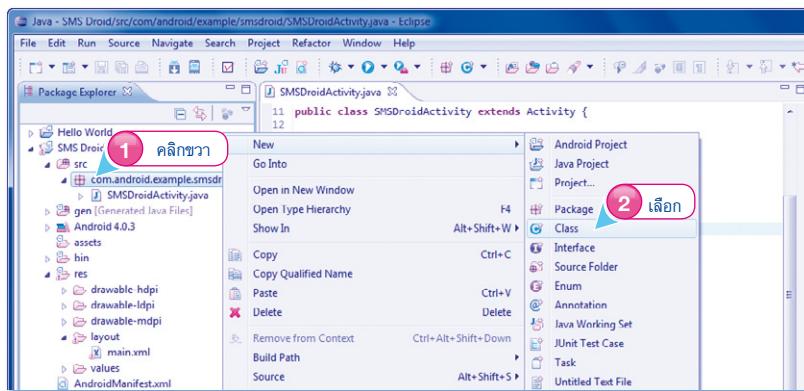
- Callout 1:** Points to the line `setContentView(R.layout.main);` with the label "วิธีหน้าจอ" (Way to screen).
- Callout 2:** Points to the line `btnSendSMS = (Button) findViewById(R.id.btnSendSMS);` with the label "อ้างอิงไปยังปุ่ม Send SMS" (Reference to the Send SMS button).
- Callout 3:** Points to the line `txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);` with the label "อ้างอิงไปยังช่องกรอกหมายเลขโทรศัพท์ปลายทาง" (Reference to the recipient phone number input field).
- Callout 4:** Points to the line `txtMessage = (EditText) findViewById(R.id.txtMessage);` with the label "อ้างอิงไปยังช่องกรอกข้อความ" (Reference to the message input field).
- Callout 5:** Points to the line `String phoneNo = txtPhoneNo.getText().toString();` with the label "รับหมายเลขโทรศัพท์ปลายทาง" (Receive phone number).
- Callout 6:** Points to the line `String message = txtMessage.getText().toString();` with the label "รับข้อความ" (Receive message).
- Callout 7:** Points to the condition `if (phoneNo.length() > 0 && message.length() > 0)` with the label "ตรวจสอบการกรอกหมายเลขโทรศัพท์ปลายทาง" (Check for phone number and message entry).

```
Toast.LENGTH_SHORT).show();
}
}
});
```

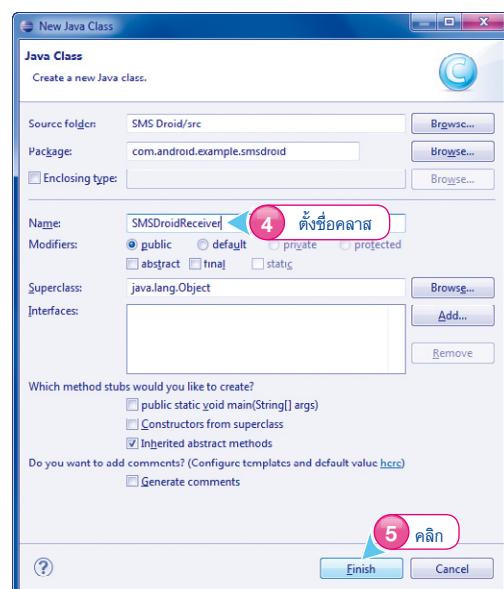
## การเขียนโค้ดส่วนของการรับข้อความ

หลังจากที่ผู้อ่านได้เขียนโปรแกรมสำหรับรับ SMS เรียบร้อยแล้ว ในส่วนนี้ผู้อ่านจะได้เขียนโปรแกรมสำหรับรับ SMS ที่ถูกส่งมาอย่างเครื่อง โดยมีขั้นตอนดังต่อไปนี้

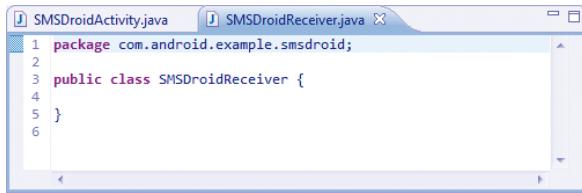
- สร้างไฟล์ SMSDroidReceiver.java โดยคลิกขวาที่ com.android.example.smsdroid
- เลือกคำสั่ง New > Class



- จะปรากฏ dialogue ให้ออกบันทึกชื่อ New Java Class
- ตั้งชื่อคลาสคือ SMSDroidReceiver
- คลิกปุ่ม **Finish**



## 6. จะปรากฏหน้าต่างสำหรับเขียนโค้ดลงในไฟล์ SMSDroidReceiver.java ดังรูป



## 7. แก้ไขโค้ด Java ในไฟล์ SMSDroidReceiver.java ดังนี้

```
package com.android.example.smsdroid;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.Toast;

public class SMSDroidReceiver extends BroadcastReceiver {
    private static final String SMS_RECEIVED = "android.provider.Telephony.SMS_RECEIVED";

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(SMS_RECEIVED)) {
            Bundle bundle = intent.getExtras();
            if (bundle != null) {
                Object[] pdus = (Object[]) bundle.get("pdus");
                final SmsMessage[] messages = new SmsMessage[pdus.length];
                for (int i = 0; i < pdus.length; i++) {
                    messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
                }
                if (messages.length > -1) {
                    Toast.makeText(context, "Message From :"
                            + messages[0].getOriginatingAddress()
                            + "\n" + messages[0].getMessageBody(),
                            Toast.LENGTH_LONG).show();
                }
            }
        }
    }
}
```

### อธิบายโค้ดการทำงานของแอพพลิเคชัน

- สร้างคลาส SMSDroidReceiver โดยขยายมาจากคลาส BroadcastReceiver เพื่อดักจับเหตุการณ์ ต่างๆ ของตัวเครื่องเมื่อมีการเปลี่ยนแปลง

```
public class SMSDroidReceiver extends BroadcastReceiver {
```

2. เขียนโค้ด Override เมธอด onReceive() เพื่อตรวจสอบว่าเครื่องได้รับ SMS และหากเครื่องได้รับ SMS เข้ามาใหม่ ก็จะแสดงข้อความที่ได้รับออกทางหน้าจอ

```

@Override
public void onReceive(Context context, Intent intent) {
    if (intent.getAction().equals(SMS_RECEIVED)) {
        Bundle bundle = intent.getExtras();
        if (bundle != null) {
            Object[] pdus = (Object[]) bundle.get("pdus");
            final SmsMessage[] messages = new SmsMessage[pdus.length];
            for (int i = 0; i < pdus.length; i++) {
                messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
            }
            if (messages.length > -1) {
                Toast.makeText(
                    context,
                    "Message From :"
                    + messages[0].getOriginatingAddress()
                    + "\n" + messages[0].getMessageBody(),
                    Toast.LENGTH_LONG).show();
            }
        }
    }
}

```

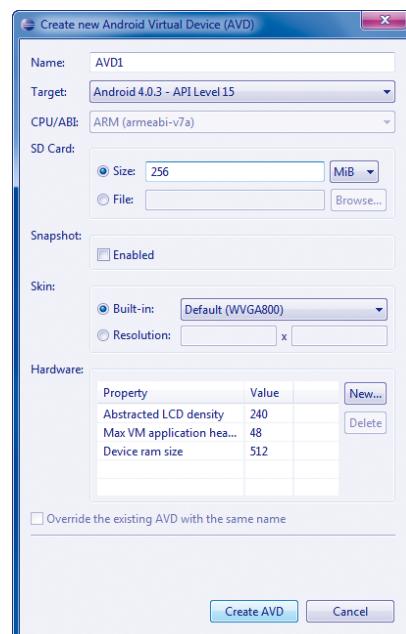
จับข้อความมาเก็บใน messages[i]

ส่วนแสดง Message

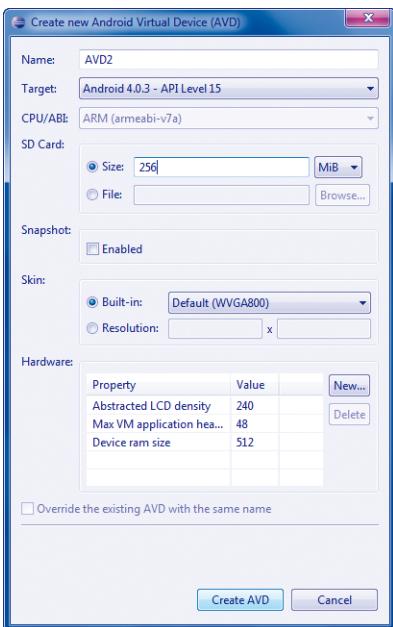
## ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator)

หลังจากการพัฒนาแอพพลิเคชันเสร็จลุ้นแล้ว ขั้นตอนต่อไปคือ ทดสอบการทำงานของแอพพลิเคชัน ซึ่งผู้เขียนจะไม่กล่าวซ้ำอีก โดยผู้อ่านสามารถอ่านบทหวานได้จากบทที่ผ่านมา อย่างไรก็ตามในหัวข้อนี้จะอธิบายถึงการรัน AVD จำนวน 2 เครื่องพร้อมๆ กัน เพื่อให้ล่งข้อความระหว่างกัน โดยมีขั้นตอนดังนี้ ดังต่อไปนี้

1. สร้าง AVD ใหม่ขึ้นมา ให้ตั้งชื่อว่า AVD1 โดยมีคุณสมบัติดังนี้



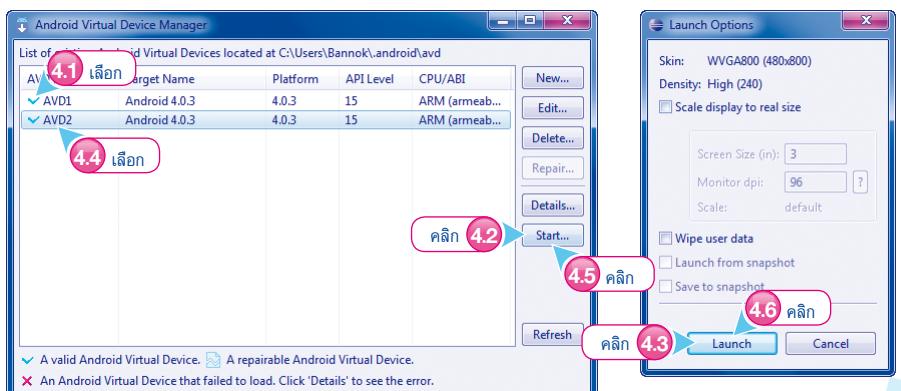
## 2. สร้าง AVD ใหม่ขึ้นมา ให้ตั้งชื่อว่า AVD2 โดยมีคุณสมบัติดังนี้



## 3. ผู้อ่านจะได้ AVD จำนวน 2 เครื่องคือ AVD1 และ AVD2



## 4. จากนั้นให้ผู้อ่านเปิดใช้เครื่อง AVD1 และ AVD2



## 5. จะประทับ Emulator 2 เครื่องคือ AVD1 และ AVD2 ดังรูป



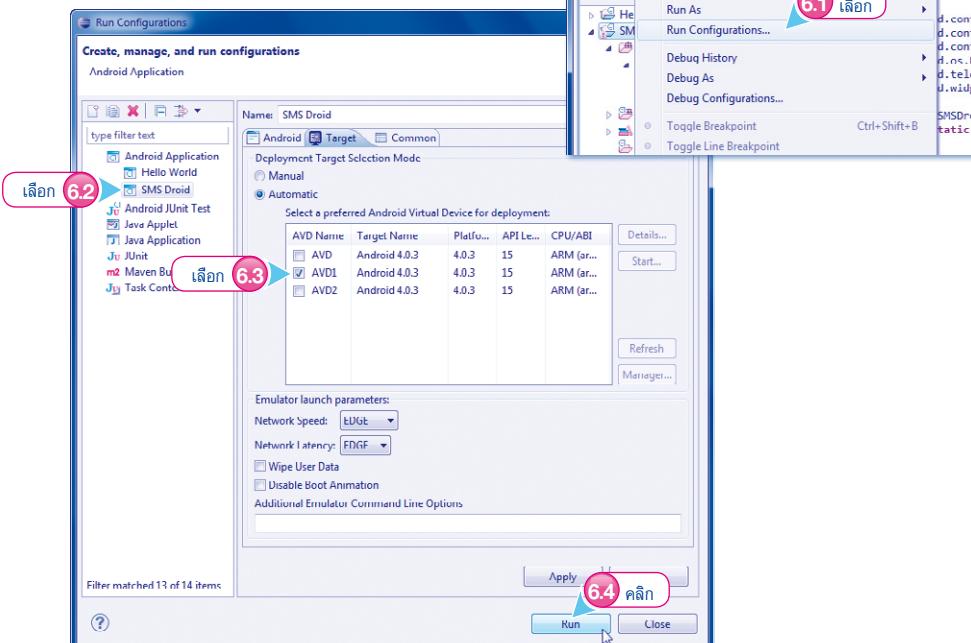
## 6. รันแอพพลิเคชั่นรับส่ง SMS จากໂປຣເຈັກຕໍ່າມຂັ້ນຕອນຕ່ອໄປນີ້

6.1 ຄລິກເມນູ Run > Run Configurations...

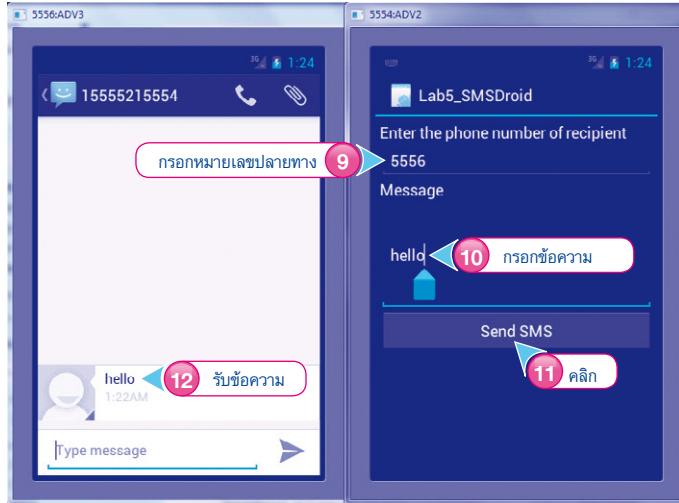
6.2 ຄລິກເລືອກ Android Application ເປັນ SMS Droid

6.3 ໃນແຕ່ບົນ Target ຄລິກເລືອກ AVD1

6.4 ຄລິກປຸ່ມ **Run**



7. รันแอพพลิเคชันรับส่ง SMS อีกครั้ง โดยเลือก Target เป็น AVD2 ตามขั้นตอนในข้อ 6
8. จะปรากฏเครื่อง Emulator ต้นทางและเครื่อง Emulator ปลายทางขึ้นมา
9. กรอกหมายเลขปลายทางในเครื่อง Emulator ต้นทาง
10. กรอกข้อความในเครื่อง Emulator ต้นทาง
11. คลิกปุ่ม Send SMS ในเครื่อง Emulator ต้นทางเพื่อส่งข้อความ
12. เครื่องปลายทางจะได้รับข้อความ และแสดงข้อความออกทางหน้าจอ



#### NOTE



สำหรับการทดสอบการทำงานของแอพพลิเคชันรับส่งข้อความ SMS นั้น ในช่องที่ใส่ตัวเลข 5556 จะหมายถึง หมายเลขโทรศัพท์ปลายทางที่จะส่งข้อความ SMS ไปถึง แต่ความเป็นจริงแล้วหมายเลข 5556 นี้จะเป็นหมายเลขโทรศัพท์เคลื่อนที่จริงปลายทาง แต่ในตัวอย่างนี้หมายเลข 5556 จะใช้เป็นการจำลองหมายเลขโทรศัพท์สำหรับการทดสอบบน Emulator เท่านั้น

หรืออีกนัยหนึ่งคือ ค่าพอร์ต (Port) ซึ่งเครื่องโทรศัพท์จำลองในแต่ละเครื่องก็จะมีหมายเลขประจำเครื่องด้วยเช่นกัน โดยจากตัวอย่างข้างต้นเป็นการรันแอพพลิเคชันที่เครื่องหมายเลข 5554 และจัดส่งข้อความไปยังเครื่องที่มีหมายเลข 5556 เป็นต้น

## สรุปภ้ายบท

ในบทนี้เป็นการอธิบายถึงตัวอย่างของการพัฒนาแอพพลิเคชัน ที่ใช้งานการรับส่งข้อความสั้น (SMS) ซึ่งมีการอนุญาตผ่านทางฟังก์ชัน SEND\_SMS ในไฟล์ AndroidManifest.xml

## แบบฝึกหัดก้ายบต

แบบฝึกหัดในบทนี้ให้ผู้อ่านพัฒนาแอพพลิเคชันเพิ่มเติม โดยเพิ่มการล่ง SMS ไปยังเครื่องหล่าย เครื่องพร้อมๆ กัน เช่น ข้อความจากเครื่อง 5554 จะถูกจัดส่งไปยังเครื่อง 5555 และ 5556 ด้วยข้อความเดียวกันและพร้อมๆ กัน

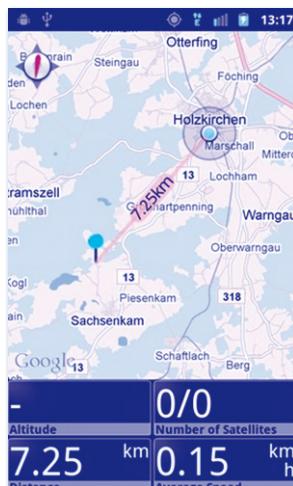
## การพัฒนาแอพพลิเคชันติดต่อกับ GPS และการอ้างอิงกับระบบแผนที่ (Map)

เนื้อหาในบทนี้ จะเป็นการฝึกพัฒนาแอพพลิเคชันติดต่อกับระบบ GPS (Global Positioning System) และการอ้างอิงกับระบบแผนที่ (Google Map) ซึ่งจุดประสงค์ของบทนี้จะเป็นการเรียกใช้งานคลาส LocationManager ที่ระบุถึงค่าพิกัดและนำค่า้นมาแสดงบนแผนที่

### ความรู้พื้นฐานเกี่ยวกับ GPS

ระบบกำหนดตำแหน่งบนโลก หรือ GPS (Global Positioning System) เป็นระบบที่ระบุตำแหน่งบนพื้นโลก โดยคำนวนจากค่าสัญญาณนาฬิกาที่ส่งมาจากดาวเทียมที่โคจรรอบโลก ทำให้ทราบตำแหน่งที่แน่นอน จึงสามารถระบุตำแหน่งต่างๆ ของอุปกรณ์รับสัญญาณต่างๆ ได้อย่างถูกต้อง

ดาวเทียม GPS เป็นดาวเทียมที่ใช้บนวงโคจรระดับกลาง หรือ MEO (Medium Earth Orbit) ซึ่งโดยทั่วไปนั้นจะมีวงโคจรที่ระดับความสูงประมาณ 20,000 กิโลเมตร



ซึ่งการยืนยันตำแหน่งต่างๆ จะอาศัยค่าพิกัดจากดาวเทียมอย่างน้อย 4 ดวง โดยปกติแล้วดาวเทียมแต่ละดวงจะโคจรรอบโลกเป็นเวลา 12 ชั่วโมงต่อหนึ่งวัน โดยมีความเร็วประมาณ 4 กิโลเมตร/วินาที (km/s) และการโคจรในแต่ละรอบนั้นจะแบ่งออกเป็น 6 ระยะ ซึ่งแต่ละระยะจะมีดาวเทียมจำนวน 4 ดวง โดยทำมุมที่ 55 องศา (Degree) ดังนั้น โดยรวมแล้วทั้งระบบจะต้องมีดาวเทียมอยู่ทั้งน้อยจำนวน 24 ดวง

โดยทั่วไปแล้วอุปกรณ์รับสัญญาณในปัจจุบัน จะพนักงานเข้ากับระบบแผนที่เพื่อใช้ระบุรายละเอียดของสถานที่ต่างๆ รวมไปถึงสามารถนำมาประยุกต์ใช้ในการนำทางได้อีกด้วย

◀ ตัวอย่างแอพพลิเคชันติดต่อกับ GPS และการอ้างอิงกับระบบแผนที่ (Map)

ในบทนี้ผู้เขียนจะแบ่งการศึกษาออกเป็น 2 ส่วนหลักๆ คือ

- การอ่านตำแหน่งพิกัดของระบบ GPS
- การแสดงผลค่าพิกัดนั้นๆ บน Google Map

## การอ่านตำแหน่งพิกัดของระบบ GPS

สำหรับการพัฒนาแอพพลิเคชันเพื่อใช้งานระบบ GPS บนระบบปฏิบัติการ Android นั้นมีขั้นตอนที่ไม่ซับซ้อนมากนัก ซึ่งสามารถทำได้โดยอาศัยคลาส LocationManager ซึ่งคลาஸดังกล่าวได้ผ่านอยู่ใน API (Application Programming Interface) เรียบร้อยแล้ว

ทั้งนี้การพัฒนาแอพพลิเคชันอ่านตำแหน่งพิกัดค่าละติจูด (Latitude) และค่าลองจิจูด (Longitude) ของระบบ GPS นั้น สามารถแบ่งออกได้เป็น 4 ขั้นตอนคือ

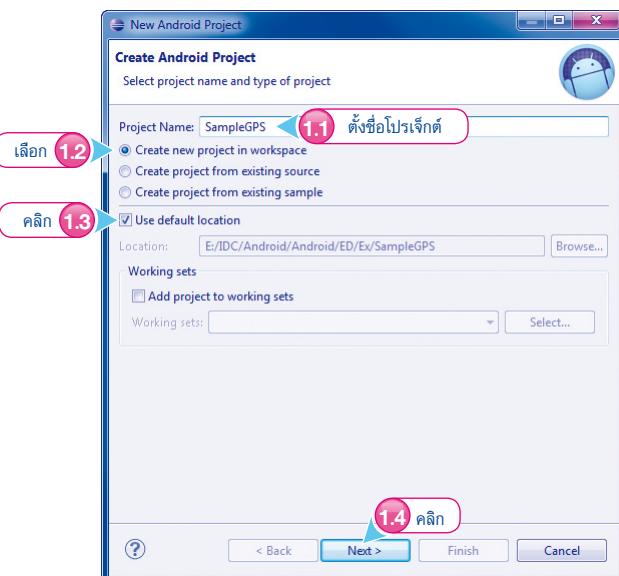
1. เริ่มต้นสร้างโปรเจกต์
2. ออกแบบหน้าจอแอพพลิเคชัน
3. เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน
4. ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator)

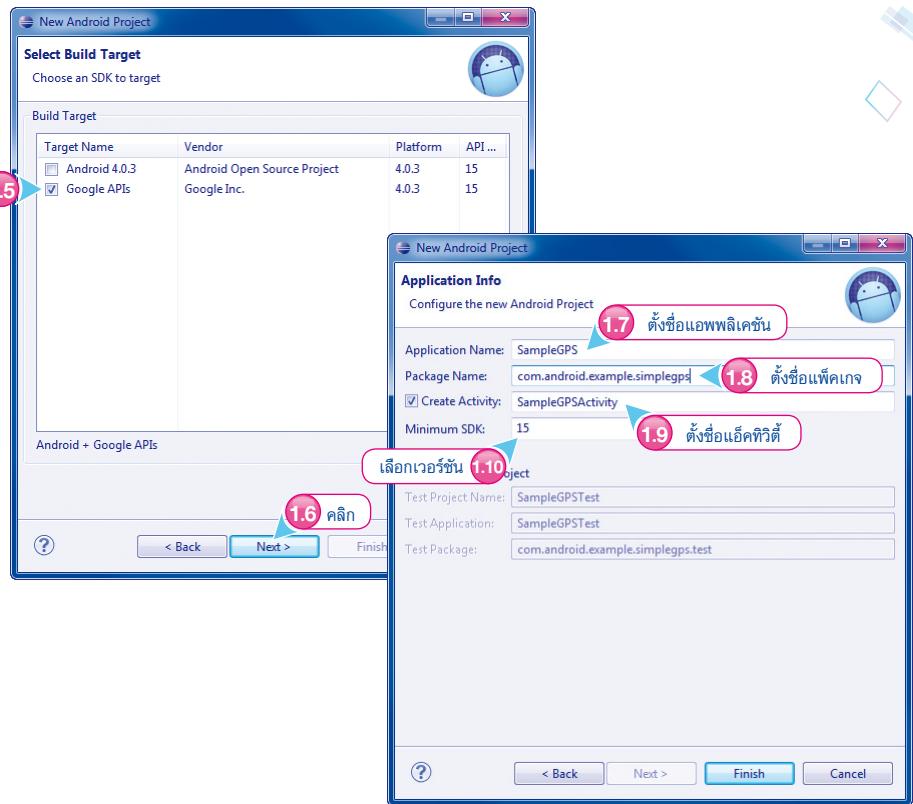
อย่างไรก็ตามขั้นตอนในบางขั้นตอน ผู้เขียนได้กล่าวไว้แล้วในบทที่ผ่านมา ดังนั้น ในบทนี้ผู้เขียนจะไม่กล่าวซ้ำในบางขั้นตอน ซึ่งผู้อ่านสามารถศึกษาบทท่านได้ในบทที่ 3 ถึงบทที่ 8

### เริ่มต้นสร้างโปรเจกต์

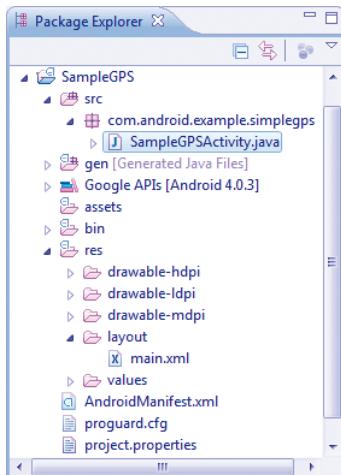
เริ่มต้นพัฒนาแอพพลิเคชันโดยสร้างโปรเจกต์สำหรับเขียนโปรแกรม ซึ่งผู้อ่านสามารถทำได้ตามขั้นตอนดังต่อไปนี้

1. เปิดโปรแกรม Eclipse สร้างโปรเจกต์ที่มีคุณสมบัติดังนี้





2. จะปรากฏโปรเจกต์ที่ Package Explorer ชี้ไปแสดงถึงโครงสร้างของโปรเจกต์ที่ได้สร้างขึ้นมา



## กำหนดสิทธิ์การทำงานให้แอพพลิเคชัน

กำหนดสิทธิ์การทำงานให้แอพพลิเคชัน เป็นการแก้ไขโค้ด XML ในไฟล์ AndroidManifest.xml ซึ่งเป็นการอนุญาตให้แอพพลิเคชันมีสิทธิ์ความสามารถต่างๆ เพิ่มขึ้น เช่น ในการถือของแอพพลิเคชันอ่านค่าพิกัดนั้น จะเป็นการกำหนดสิทธิ์ให้แอพพลิเคชันสามารถอ่านค่าพิกัด (ค่าละติจูดและค่าลองจิจูด) ได้จริง โดยเพิ่มโค้ดการอนุญาตการให้สิทธิ์ (Permission) ในไฟล์ AndroidManifest.xml ดังนี้

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

แสดงได้ดังรูป

The screenshot shows the AndroidManifest.xml file in the Eclipse IDE. A pink arrow points to the line of code: <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>. Below the code editor, there are tabs for Manifest, Application, Permissions, Instrumentation, and AndroidManifest.xml. The Permissions tab is highlighted.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.example.simplegps"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="15" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/> ← เพิ่มโค้ด
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".SampleGPSActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

## เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน

เมื่อผู้อ่านแก้ไขไฟล์ AndroidManifest.xml แล้ว ขั้นตอนต่อไปคือ การเขียนโค้ด Java ในไฟล์ SampleGPSActivity.java เพื่อใช้ในการอ่านค่าพิกัด โดยเรียกใช้งานคลาส LocationManager เพื่อขอเข้าใช้งาน การอ่านพิกัดของระบบ GPS ซึ่งมีโค้ดดังนี้

```

package com.android.example.simplegps;

import android.app.Activity;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager; ← เรียกใช้งานคลาส LocationListener และคลาส LocationManager
import android.os.Bundle;
import android.widget.Toast;

public class SampleGPSActivity extends Activity {

    private LocationManager lm;
    private LocationListener locationListener;

```

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    lm = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    locationListener = new MyLocationListener();
    lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 10000, 1,
        locationListener);
}

private class MyLocationListener implements LocationListener {
    public void onLocationChanged(Location loc) {
        if (loc != null) {
            Toast.makeText(
                getBaseContext(),
                "Location changed \nLat: " + loc.getLatitude()
                + "\nLng: " + loc.getLongitude(),
                Toast.LENGTH_SHORT).show();
        }
    }

    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub
    }

    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }
}
}

```

## อธิบายโค้ดการทำงานของแอพพลิเคชัน

- การสร้างออบเจกต์ของคลาส LocationManager นั้น สามารถเรียกใช้งานได้โดยตรงดังนี้

```
getSystemService(Context.LOCATION_SERVICE);
```

- การลงทะเบียนให้แอพพลิเคชัน สามารถรับการแจ้งเตือนบริการประเภทต่างๆ จากระบบได้นั้น จะมีรูปแบบค้าลั่งดังนี้

```
requestLocationUpdates (String provider, long minTime, float minDistance, LocationListener listener)
```

โดยจะต้องระบุพารามิเตอร์ต่างๆ ดังต่อไปนี้

- provider** คือ ชื่อสิ่งที่ต้องการขอรับบริการ ซึ่งในที่นี้ให้กำหนดเป็นค่า LocationManager.GPS\_PROVIDER เพราะแอพพลิเคชันนี้มีการเรียกใช้งาน GPS
- minTime** คือ ระยะเวลาที่ห่างกันในการแจ้งเตือนแต่ละวงรอบ ซึ่งมีหน่วยเป็นมิลลิวินาที (millisecond)
- minDistance** คือ การแจ้งเตือนเมื่อมีการเปลี่ยนตำแหน่งตามระยะห่างที่กำหนด ซึ่งมีหน่วย เป็นเมตร (meter)



- **listener** คือ อะบเจกต์ของคลาส LocationListener ที่ใช้สำหรับตักจับเมื่อมีการแจ้งเตือน  
มายังแอพพลิเคชัน
3. สร้างคลาส MyLocationListener ซึ่งพัฒนามาจากคลาส LocationListener ซึ่งเป็น Inner Class  
ให้สำหรับการตรวจจับในระบบ GPS โดยเฉพาะ ในกรณีที่มีการเปลี่ยนแปลงค่าตำแหน่ง ซึ่งมี  
จำนวน 4 เมธอด (Method) ดังต่อไปนี้
- onLocationChanged(Location loc) จะถูกเรียกใช้งานเมื่อมีการเปลี่ยนแปลงตำแหน่ง
  - onProviderDisabled(String provider) จะถูกเรียกใช้งานเมื่อปิดการใช้งาน
  - onProviderEnabled(String provider) จะถูกเรียกใช้งานเมื่อเปิดการใช้งาน
  - onStatusChanged(String provider, int status, Bundle extras) จะถูกเรียกใช้งานเมื่อมีการ  
เปลี่ยนแปลงสถานะ

```
private class MyLocationListener implements LocationListener {
    public void onLocationChanged(Location loc) {

    }

    public void onProviderDisabled(String provider) {

    }

    public void onProviderEnabled(String provider) {

    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
    }
}
```

4. ในส่วนของเมธอด onLocationChanged ของคลาส MyLocationListener มีหน้าที่แสดงตำแหน่ง  
พิกัดปัจจุบันออกมาทางหน้าจอ โดยผู้อ่านสามารถกำหนดค่าอื่นๆ ได้เพิ่มเติม เช่น
- getAccuracy() ซึ่งใช้ในกรณีที่มีค่าความคลาดเคลื่อนในหน่วยเมตร
  - getAltitude() ซึ่งใช้ในกรณีที่มีค่าระดับเหนือน้ำทะเล มีหน่วยเป็นฟุต
  - getSpeed() ซึ่งใช้ในกรณีที่มีค่าความเร็วในหน่วยเมตรต่อวินาที

```
public void onLocationChanged(Location loc) {
    if (loc != null) {
        Toast.makeText(
            getBaseContext(),
            "Location changed \nLat: " + loc.getLatitude()
            + "\nLng: " + loc.getLongitude(),
            Toast.LENGTH_SHORT).show();
    }
}
```

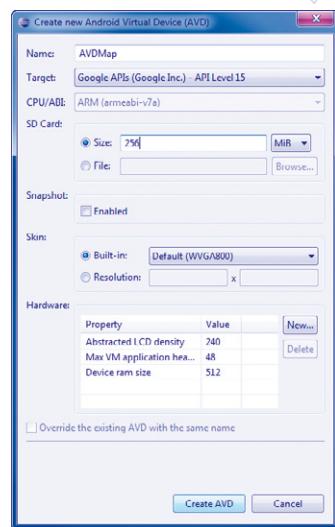
# ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator)

หลังจากการเขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน เลร์นจีนแล้ว ขั้นตอนต่อไปคือ ทดสอบการทำงานของแอพพลิเคชัน ทั้งนี้ผู้อ่านสามารถอ่านบทหวานได้จากบทที่ 2 โดยสร้าง Emulator ที่มีคุณสมบัติดังรูป

จากนั้นให้รันแอพพลิเคชันบน Emulator ที่สร้างขึ้นมา เมื่อรันแอพพลิเคชันขึ้นมาแล้ว ผู้อ่านจะต้องล็อกหน้าจอ ให้แอพพลิเคชัน เพื่อตรวจสอบว่าแอพพลิเคชันสามารถแสดงค่าพิกัดได้ถูกต้องหรือไม่

ในการนี้จะใช้งาน Emulator Control เป็นเครื่องมือในการส่งค่าพิกัด ซึ่งมีขั้นตอนการทดสอบดังต่อไปนี้ (ผู้อ่านสามารถศึกษาส่วนของ Emulator Control ได้ในบทที่ 14 เทคนิคในการแก้ไขปัญหาสำหรับการพัฒนาแอพพลิเคชัน)

## 1. คลิกเมนู Window > Open Perspective > DDMS

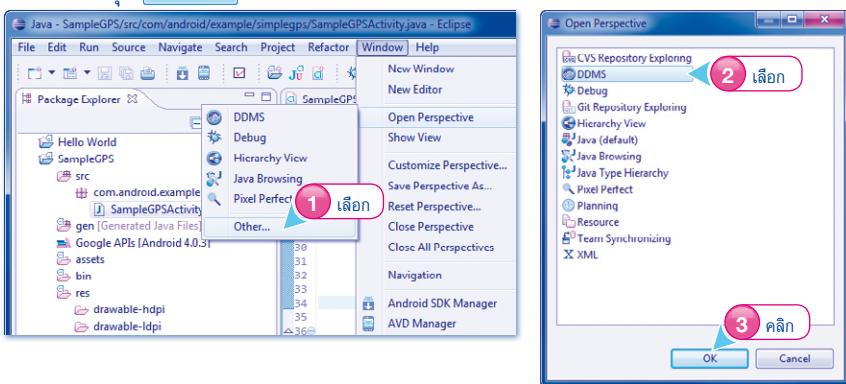


### NOTE

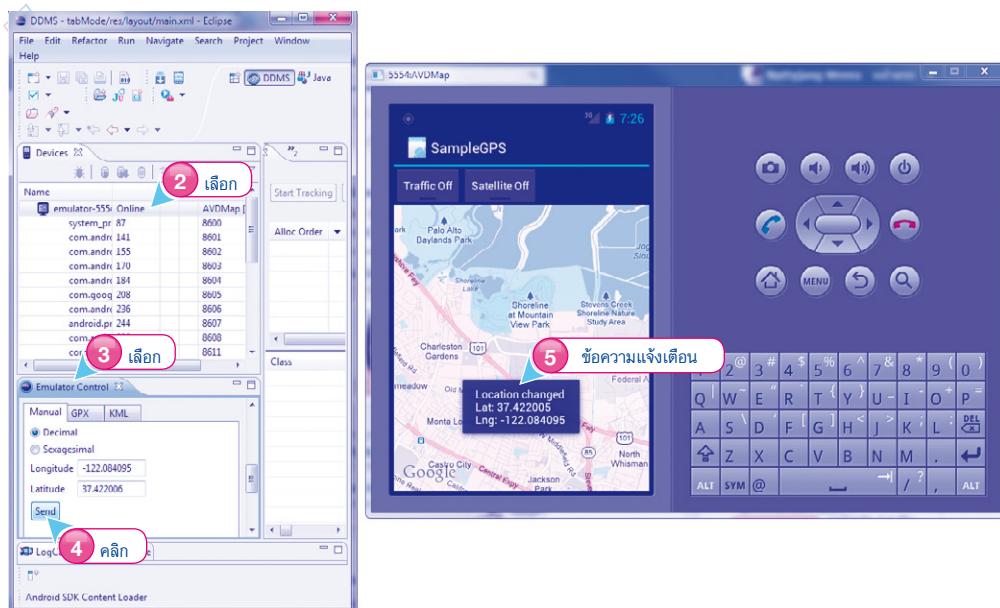


ในกรณีที่ผู้อ่านไม่พบเมนูดังข้อ 1 ให้ทำการขั้นตอนดังนี้

1. คลิกเมนู Window > Open Perspective > Other...
2. จะปรากฏ dialogue Open Perspective และคลิกเลือก DDMS
3. คลิกปุ่ม OK



2. คลิกเลือก Device ที่ต้องการทดสอบ
3. ให้ผู้อ่านมาที่แท็บ Emulator Control
4. คลิกปุ่ม **Send**
5. ลังเกตหน้าจอ Emulator จะปรากฏข้อความแจ้งเตือนสำหรับการเปลี่ยนค่าตำแหน่งดังรูป



### TIPS

สำหรับการใช้งานเครื่องมือ DDMS (Dalvik Debug Monitor Server) นั้นอาจจะไม่สะดวกในการสังเกตผลลัพธ์ของการทำงานเมื่อมีการพัฒนาแอพพลิเคชันบน Eclipse ดังนั้น ผู้อ่านสามารถนำเข้าลงช่องวิธีการส่งค่าพิเศษได้โดยวิธีหนึ่งคือ ให้เปิด Command Prompt จากนั้นพิมพ์คำสั่ง "C:\telnet localhost 5554"

สังเกตผลลัพธ์การทำงานบน Emulator ซึ่งก็จะได้ผลลัพธ์เช่นเดียวกันกับการทำงานบน Eclipse ทั้งนี้ ค่าพอร์ตที่ 5554 จะเข้าอยู่กับพอร์ตที่ Emulator เลือกใช้ โดยผู้อ่านลังเกตตัวเลขนี้ได้จาก Emulator ที่เปิดขึ้นใช้งาน และแสดงได้ดังตัวอย่างด้านในนี้

```
Administrator: Telnet localhost
Android Console: type 'help' for a list of commands
OK
geo fix 82.411629 28.054553
KO
```

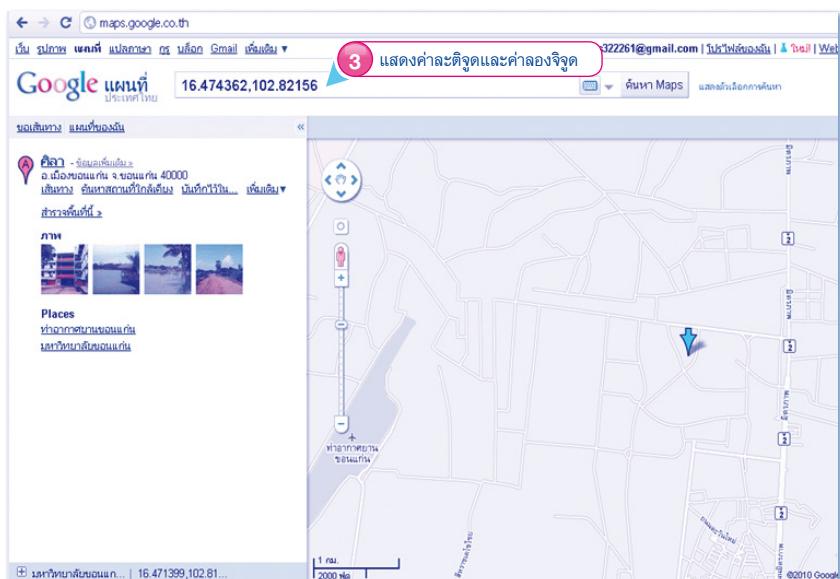
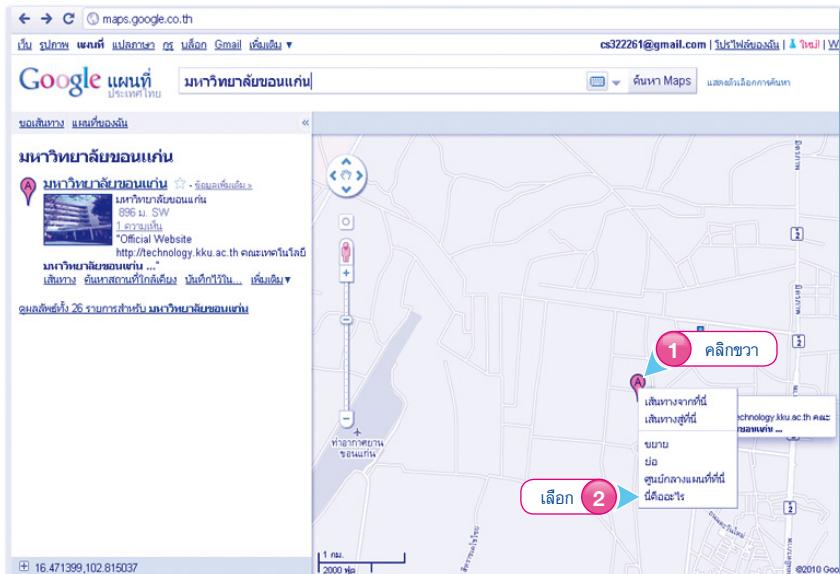
## TIPS



วิธีการหาค่าพิกัด (ค่าละติจูดและค่าลองจิจูด) สำหรับสถานที่ต่างๆ นั้นสามารถทำได้ผ่านเว็บไซต์ <http://maps.google.co.th> ด้วยวิธี เช่น แสดงการค้นหาที่อยู่ของมหาวิทยาลัยขอนแก่น ซึ่งมีขั้นตอนดังนี้

1. คลิกขวาที่หมุด จะเห็นกล่องข้อความปรากฏขึ้นมา
2. เลือกคำสั่ง "นี่คืออะไร"
3. จะปรากฏหน้าจอที่แสดงค่าละติจูดและค่าลองจิจูดบนกล่องข้อความค้นหาด้านบน ซึ่งในกรณีนี้ค่าที่ได้คือ 16.474362, 102.82156

ซึ่งสามารถนำค่าทั้งสองไปใส่ในช่อง Latitude และ Longitude ของเครื่องมือ Emulator Control ได้



# การแสดงผลค่าพิกัดบน Google Map

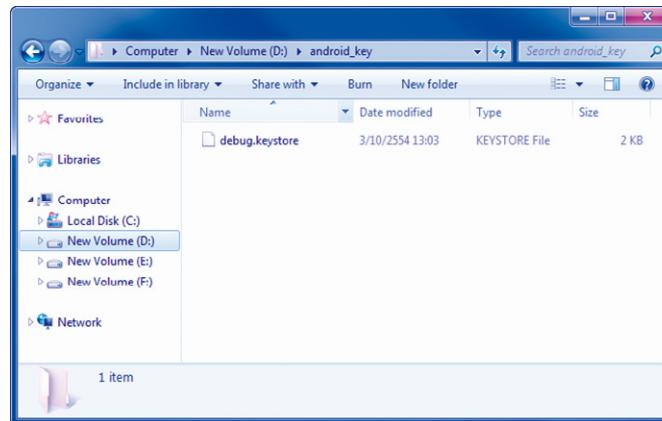
หลังจากที่เขียนโค้ดเพื่อกำหนดค่าพิกัดผ่านระบบ GPS ได้แล้ว ขั้นตอนต่อไปคือ การนำค่าพิกัดที่ได้ไปแสดงบนแผนที่ Google Map เพื่อแสดงตำแหน่งปัจจุบันหรือค่าพิกัดนั้นบนแผนที่ ซึ่งสามารถแบ่งออกเป็น 5 ขั้นตอนดังนี้

1. สร้าง Google API Key
2. ออกแบบหน้าจอแอพพลิเคชัน
3. เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน
4. ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator)
5. ปรับแต่งแอพพลิเคชันเพื่อแสดงแผนที่ในรูปแบบต่างๆ และเพิ่ม Marker

## สร้าง Google API Key

การสร้าง Google API Key เป็นขั้นตอนโดยรวมในการขออนุญาตเข้าใช้แผนที่ของ Google โดยมีขั้นตอนดังๆ ดังต่อไปนี้

1. ให้ผู้อ่านตัดลอกไฟล์ debug.keystore เก็บไว้ ชื่งโดยปกติแล้วจะอยู่ที่ไดเร็คทอรี C:\username\<username>.android แต่ถ้าไม่สามารถใช้ฟังก์ชันการค้นหาข้อมูลของ Windows ค้นหาไฟล์นี้ได้เช่นกัน ในที่นี่เก็บไฟล์ debug.keystore ไว้ที่ไดร์ D:\android\_key



2. เปิด Command Prompt และเข้าไปยังไดเร็คทอรีของ JDK ในที่นี่อยู่ที่ไดเร็คทอรี C:\Program Files\Java\jdk1.6.0\_24\bin
3. พิมพ์คำสั่งด้านล่างแล้วกดปุ่ม **Enter**

```
keytool.exe -list -alias androiddebugkey -keystore "D:\android_key\debug.keystore" -storepass android -keypass android -v
```

จากนั้นจะปรากฏ Certificate สำหรับใช้งาน Google API ขึ้นมาดังรูป

#### 4. คัดลอก Certificate Fingerprint (MD5) เก็บไว้

```

Administrator: C:\Windows\system32\cmd.exe
C:\Program Files\Java>cd \
C:\>cd "Program Files\Java\jdk1.7.0_01\bin"
C:\Program Files\Java\jdk1.7.0_01\bin>keytool -list -alias androiddebugkey -keystore "D:\android_key\debug.keystore" -storepass android -keypass android -v
Alias name: androiddebugkey
Creation date: Oct 3, 2011
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 4e89503b
Valid from: Mon Oct 03 13:03:39 ICT 2011 until: Tue Oct 02 13:03:39 ICT 2012
Certificate fingerprints:
        MD5: CE:C9:E3:4F:C1:B1:05:C8:75:71:33:F4:9F:A4:A9:2E
        SHA1: B1:80:FD:40:30:6F:4C:97:BC:D4:D1:32:2FA3:98:9E:5A:9C:43:33
        SHA256: 22:33:22:D0:60:59:5A:EC:9F:0E:56:57:B4:92:DB:69:61:00:A1:60:DB:39:11:46:D4:68:D6:A0:3B:5A:FF:19
        Signature algorithm name: SHA1withRSA
        Version: 3
C:\Program Files\Java\jdk1.7.0_01\bin>

```

5. ให้ผู้อ่านเข้าเว็บไซต์ <http://code.google.com/intl/th-TH/android/maps-api-signup.html> เพื่อสร้าง Google API Key
6. คลิกเครื่องหมายถูกหน้า I have read and agree with the terms and conditions ซึ่งจะเป็นการยอมรับข้อกำหนดต่างๆ ในการขอเข้าใช้งาน Google Map
7. กรอก Certificate Fingerprint ในช่อง My certificate's MD5 fingerprint:
8. คลิกปุ่ม Generate API Key



9. จะปรากฏ API Key และให้ผู้อ่านคัดลอก API Key เก็บไว้เพื่อใช้สำหรับการออกแบบหน้าจอแอพพลิเคชันในล่วงต่อไป



## กำหนดสิทธิ์การทำงานให้แอพพลิเคชัน

กำหนดสิทธิ์การทำงานให้แอพพลิเคชัน เป็นการแก้ไขโค้ด XML ในไฟล์ AndroidManifest.xml ซึ่งเป็นการอนุญาตให้แอพพลิเคชันสามารถใช้งานอินเทอร์เน็ต ได้โดยเพิ่ม permission สำหรับใช้งานอินเทอร์เน็ต ดังนี้

```
<uses-permission android:name="android.permission.INTERNET" />
```

และเพิ่ม permission สำหรับใช้งาน library maps ดังนี้

```
<uses-library android:name="com.google.android.maps" />
```

ซึ่งได้โค้ดที่แก้ไขแล้วดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.example.simplegps"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="15" />

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" /> ← เพิ่มโค้ด

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <uses-library android:name="com.google.android.maps" /> ← เพิ่มโค้ด

        <activity
            android:name=".SampleGPSActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

## ออกแบบหน้าจอแอพพลิเคชัน

การออกแบบหน้าจอแอพพลิเคชัน เป็นการแก้ไขโค้ด XML ในไฟล์ main.xml ในที่นี่คือ ออกแบบสำหรับแอพพลิเคชันเพื่อใช้แสดงค่าพิกัดบน Google Map ซึ่งจะไม่มีการวาดหน้าจอแอพพลิเคชัน เพียงแต่เป็นการใส่ค่าคุณสมบัติของค่า API Key ที่ได้รับจากเว็บไซต์ของ Google โดยแก้ไขโค้ด XML ในไฟล์ main.xml ดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <com.google.android.maps.MapView
        android:id="@+id/mapview1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey="<google_api_key>">
    />
</LinearLayout>

```

1 กำหนดคุณสมบัติของ Layout  
กรอก Key ที่ได้จากหัวข้อ สร้าง Google API Key

## เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน

เมื่อผู้อ่านออกแบบหน้าจอแอพพลิเคชันเรียบร้อยแล้ว ต่อไปจะเป็นการเขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน โดยแก้ไขโค้ด Java ในไฟล์ SampleGPSActivity.java เพื่อนำค่าพิกัด GPS มาแสดงบน Google Map ในแอพพลิเคชัน

โดยให้ผู้อ่านแก้ไขโค้ด Java จากไฟล์เดิมดังนี้

- แก้ไขจาก SampleGPSActivity extends Activity เป็น SampleGPSActivity extends MapActivity
- ให้ Override เมธอด isRouteDisplayed()  
ซึ่งมีโค้ดดังนี้

```

package com.android.example.simplegps;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;

import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.Toast;

public class SampleGPSActivity extends MapActivity {
    private LocationManager lm;
    private LocationListener locationListener;

    private MapView mapView;
    private MapController mc;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}

```

1 แก้ไขโค้ด



```
mapView = (MapView) findViewById(R.id.mapview1);
mapView.setBuiltInZoomControls(true);

mc = mapView.getController();
mc.setZoom(14);

lm = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
locationListener = new MyLocationListener();
lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 10000, 1,
    locationListener);
}

private class MyLocationListener implements LocationListener {
    public void onLocationChanged(Location loc) {
        if (loc != null) {
            Toast.makeText(getApplicationContext(), "Location changed\nLat: "
                + loc.getLatitude()
                + "\nLng: " + loc.getLongitude(),
                Toast.LENGTH_SHORT).show();

            int lat = (int) (loc.getLatitude() * 1E6);
            int lng = (int) (loc.getLongitude() * 1E6);
            GeoPoint point = new GeoPoint(lat, lng);
            mc.animateTo(point); // mapController.setCenter(point);

        }
    }

    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub
    }

    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }
}

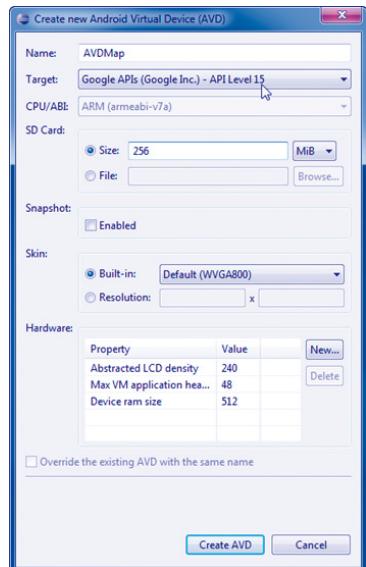
@Override ← 2 Override เมธอด
protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}
}
```

## ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator)

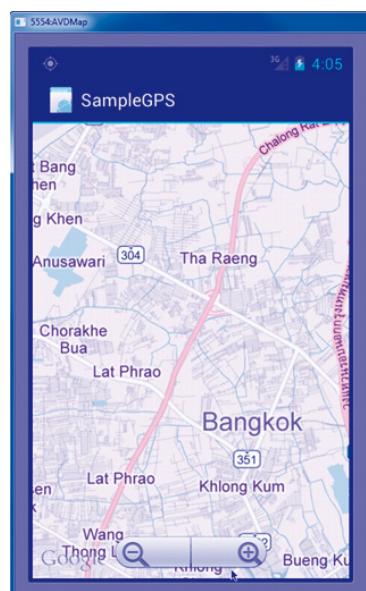
ในหัวข้อนี้จะเป็นการทดสอบแอพพลิเคชันที่ได้พัฒนาในบทนี้ ซึ่งผู้เขียนจะไม่ขออธิบายขั้นตอนการทดสอบซ้ำอีก ทั้งนี้ผู้อ่านสามารถอ่านการวันแอพพลิเคชันได้จากหัวข้อ ทดสอบการทำงานของแอพพลิเคชันผ่านโทรศัพท์จำลอง (Emulator) ในบทที่ 2 และ 3

อย่างไรก็ตามในหัวข้อนี้จะเป็นการรัน Emulator ขึ้นมา เพื่อนำค่าพิกัดไปแสดงบนแผนที่ของ Google ซึ่งมีขั้นตอนคร่าวๆ ดังนี้

1. ลรรจง AVD โดยกำหนด Target เป็น Google APIs (Google Inc.) – API Level 15 ทั้งนี้ค่า Target จะไม่ใช่ค่า Android 4.0.3 ดังรูป



2. รันแอพพลิเคชัน ซึ่งผู้อ่านสามารถจำลองการเปลี่ยนตำแหน่งค่าพิกัด GPS โดยใช้เครื่องมือ Emulator Control เช่น ใส่ค่าละติจูดและค่าลองจิจูดที่แตกต่างกัน โดยตัวอย่างของผลลัพธ์จะแสดงได้ดังรูป

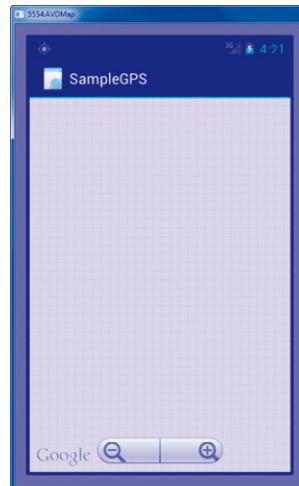


## NOTE



หากผู้อ่านรันแอพพลิเคชันแล้วไม่มีข้อผิดพลาดใดๆ แต่แผนที่ไม่แสดงออกมานะ โดยจะลังเกตเห็นเป็นลេនក្រុត (Grid) ดังรูป

สามารถแก้ไขได้โดยให้ผู้อ่านนำไฟล์ debug.keystore ไปสร้าง Google API Key ใหม่ และแก้ไขโค้ด XML ในไฟล์ main.xml จากนั้นรันแอพพลิเคชันใหม่อีกครั้ง



## ปรับแต่งแอพพลิเคชันเพื่อแสดงแผนที่ในรูปแบบต่างๆ

หลังจากที่ผู้อ่านสามารถพัฒนาโปรแกรมให้แสดงแผนที่อย่างง่ายได้เรียบร้อยแล้ว ในหัวข้อนี้จะอธิบายการปรับแต่งแอพพลิเคชันให้มีความสมบูรณ์ขึ้น โดยการปรับแต่งให้แสดงแผนที่ในรูปแบบต่างๆ (ในที่นี้ปรับแต่งจากแอพพลิเคชันเดิม) ซึ่งมีขั้นตอนดังนี้

1. แก้ไขโค้ด XML ในไฟล์ main.xml เพื่อเพิ่มปุ่มสำหรับแสดงแผนที่ในแต่ละแบบดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <ToggleButton
            android:id="@+id/bt_traffic"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textOff="Traffic Off"
            android:textOn="Traffic On" />

        <ToggleButton
            android:id="@+id/bt_satellite"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textOff="Satellite Off"
            android:textOn="Satellite On" />

    </LinearLayout>
```

← เพิ่มปุ่ม 2 ปุ่ม

```

<com.google.android.maps.MapView
    android:id="@+id/mapview1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey "<google_api_key>"
    android:clickable="true"
    android:enabled="true" />

</LinearLayout>

```

## 2. แก้ไขโค้ด Java ในไฟล์ SampleGPSActivity.java โดยแก้ไขเมธอด onCreate() ดังนี้

```

package com.android.example.simplegps;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;

import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;

import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.Toast;
import android.widget.ToggleButton;

public class SampleGPSActivity extends MapActivity {

    private LocationManager lm;
    private LocationListener locationListener;

    private MapView mapView;
    private MapController mc;

    public void onCreate(Bundle savedInstanceState) { ← 2 แก้ไขโค้ด
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mapView = (MapView) findViewById(R.id.mapview1);
        mapView.setBuiltInZoomControls(true);

        mc = mapView.getController();
        mc.setZoom(14);

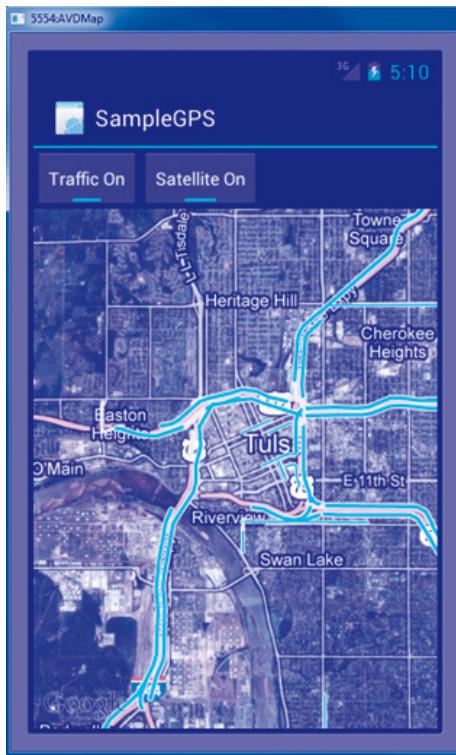
        lm = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        locationListener = new MyLocationListener();
        lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 10000, 1,
            locationListener);

        ToggleButton traffic = (ToggleButton) findViewById(R.id.bt_traffic);
        ToggleButton satellite = (ToggleButton) findViewById(R.id.bt_satellite);
    }
}

```

```
traffic.setOnCheckedChangeListener(new OnCheckedChangeListener() {  
  
    public void onCheckedChanged(CompoundButton buttonView,  
                                boolean isChecked) {  
        mapView.setTraffic(isChecked);  
    }  
});  
satellite.setOnCheckedChangeListener(new OnCheckedChangeListener() {  
  
    public void onCheckedChanged(CompoundButton buttonView,  
                                boolean isChecked) {  
        mapView.setSatellite(isChecked);  
    }  
});  
}  
  
private class MyLocationListener implements LocationListener {  
    public void onLocationChanged(Location loc) {  
        if (loc != null) {  
            Toast.makeText(  
                getBaseContext(),  
                "Location changed\nLat: " + loc.getLatitude()  
                + "\nLng: " + loc.getLongitude(),  
                Toast.LENGTH_SHORT).show();  
  
            int lat = (int) (loc.getLatitude() * 1E6);  
            int lng = (int) (loc.getLongitude() * 1E6);  
            GeoPoint point = new GeoPoint(lat, lng);  
            mc.animateTo(point); // mapController.setCenter(point);  
        }  
    }  
  
    public void onProviderDisabled(String provider) {  
        // TODO Auto-generated method stub  
    }  
  
    public void onProviderEnabled(String provider) {  
        // TODO Auto-generated method stub  
    }  
  
    public void onStatusChanged(String provider, int status, Bundle extras) {  
        // TODO Auto-generated method stub  
    }  
}  
  
@Override  
protected boolean isRouteDisplayed() {  
    // TODO Auto-generated method stub  
    return false;  
}  
}
```

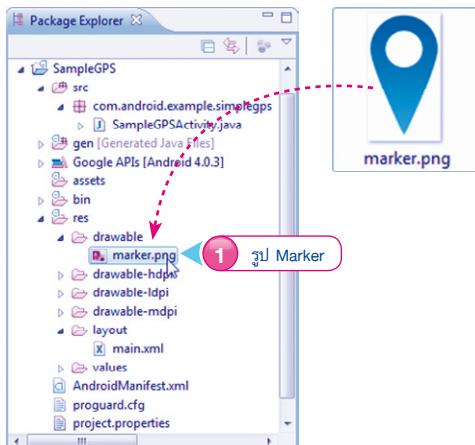
3. รันแอพพลิเคชันแล้วคลิกปุ่มเพื่อทดสอบ จะได้ผลลัพธ์ดังรูป



## การเพิ่ม Marker ลงบนแพทที่

ในหัวข้อนี้จะเป็นการเพิ่มความสมบูรณ์ให้แอพพลิเคชันมากขึ้นไปอีก เพื่อให้สามารถ Marker ในพิกัดที่ต้องการลงบนแพทที่ได้ (ในที่นี้ปรับแต่งจากแอพพลิเคชันเดิม) ซึ่งมีขั้นตอนดังนี้

1. คัดลอกไฟล์รูปภาพสำหรับทำตัว Marker มาไว้ที่ res/drawable





2. แก้ไขโค้ด Java ในไฟล์ SampleGPSActivity.java โดยเพิ่มคลาส MapOverlay ซึ่งขยายมาจากคลาส ItemizedOverlay เพื่อใช้สำหรับวัด Marker ดังนี้

```

package com.android.example.simplegps;

import java.util.ArrayList;
import java.util.List;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.ItemizedOverlay;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.OverlayItem;

import android.content.Context;
import android.graphics.drawable.Drawable;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;

import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.Toast;
import android.widget.ToggleButton;

public class SampleGPSActivity extends MapActivity {

    private LocationManager lm;
    private LocationListener locationListener;

    private MapView mapView;
    private MapController mc;
    GeoPoint p;
    List<Overlay> mapOverlays; ← เพิ่มโค้ด
    MapOverlay marker; ← เพิ่มโค้ด

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mapView = (MapView) findViewById(R.id.mapview1);
        mapView.setBuiltInZoomControls(true);

        mc = mapView.getController();
        mc.setZoom(14); ← เพิ่มโค้ด

        mapOverlays = mapView.getOverlays();
        Drawable drawable = this.getResources().getDrawable(R.drawable.marker);
        marker = new MapOverlay(drawable); ← เพิ่มโค้ด

        lm = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        locationListener = new MyLocationListener();
        lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 10000, 1,
            locationListener);
    }
}

```

```

ToggleButton traffic = (ToggleButton) findViewById(R.id.bt_traffic);
ToggleButton satellite = (ToggleButton) findViewById(R.id.bt_satellite);
traffic.setOnCheckedChangeListener(new OnCheckedChangeListener() {

    public void onCheckedChanged(CompoundButton buttonView,
                                boolean isChecked) {
        // TODO Auto-generated method stub
        mapView.setTraffic(isChecked);
    }
});
satellite.setOnCheckedChangeListener(new OnCheckedChangeListener() {

    public void onCheckedChanged(CompoundButton buttonView,
                                boolean isChecked) {
        // TODO Auto-generated method stub
        mapView.setSatellite(isChecked);
    }
});
}

private class MyLocationListener implements LocationListener {
    public void onLocationChanged(Location loc) {
        if (loc != null) {
            Toast.makeText(
                getBaseContext(),
                "Location changed\nLat: " + loc.getLatitude()
                + "\nLng: " + loc.getLongitude(),
                Toast.LENGTH_SHORT).show();

            int lat = (int) (loc.getLatitude() * 1E6);
            int lng = (int) (loc.getLongitude() * 1E6);
            p = new GeoPoint(lat, lng);

            OverlayItem overlayitem = new OverlayItem(p, "", "");
            marker.addOverlay(overlayitem);
            mapOverlays.add(marker);
            mc.animateTo(p);
        }
    }

    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub
    }

    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }
}

@Override
protected boolean isRouteDisplayed() {
    // TODO Auto-generated method stub
    return false;
}

```



```
class MapOverlay extends ItemizedOverlay { ← 2 เพิ่มโค้ด
    private ArrayList<OverlayItem> mOverlays = new ArrayList<OverlayItem>();
    Context mContext;

    public MapOverlay(Drawable src) {
        super(boundCenterBottom(src));
    }

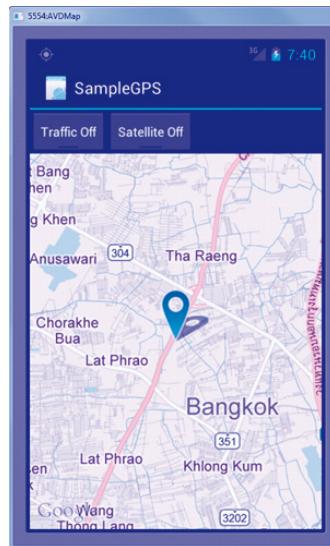
    @Override
    protected OverlayItem createItem(int i) {
        // TODO Auto-generated method stub
        return mOverlays.get(i);
    }

    public void addOverlay(OverlayItem overlay) {
        mOverlays.add(overlay);
        populate();
    }

    @Override
    public int size() {
        return mOverlays.size();
    }

}
```

3. รันแอพพลิเคชันแล้วคลิกปุ่มเพื่อทดสอบ จะได้ผลลัพธ์ดังรูป



## อธิบายโค้ดการทำงานของแอพพลิเคชัน

- เพิ่มคลาส MapOverlay ซึ่งขยายมาจากคลาส ItemizedOverlay เพื่อใช้สำหรับวัด Marker ดังนี้

```
mapOverlays = mapView.getOverlays();
Drawable drawable = this.getResources().getDrawable(R.drawable.marker);
marker = new MapOverlay(drawable);
```

- Marker ในตำแหน่งที่ต้องการ ซึ่งมีโค้ดดังนี้

```
OverlayItem overlayitem = new OverlayItem(p, "", "");
marker.addOverlay(overlayitem);
mapOverlays.add(marker);
mc.animateTo(p);
```

- ปรับแก้คลาส MapOverlay ดังนี้

```
class MapOverlay extends ItemizedOverlay {
    private ArrayList<OverlayItem> mOverlays = new ArrayList<OverlayItem>();
    Context mContext;

    public MapOverlay(Drawable src) {
        super(boundCenterBottom(src));
    }

    @Override
    protected OverlayItem createItem(int i) {
        // TODO Auto-generated method stub
        return mOverlays.get(i);
    }

    public void addOverlay(OverlayItem overlay) {
        mOverlays.add(overlay);
        populate();
    }

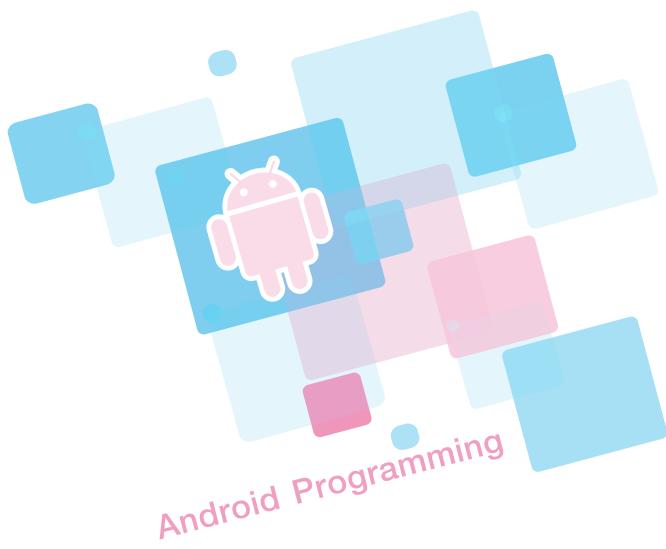
    @Override
    public int size() {
        return mOverlays.size();
    }
}
```

## สรุปก้ายบ

ในบทนี้เป็นการอธิบายการพัฒนาแอพพลิเคชันเพื่อใช้งาน GPS และเชื่อมโยงกับระบบแผนที่ (Map) โดยใช้ Google Map รวมไปถึงการสร้าง Google API Key เพื่อขออนุญาตในการเข้ามายัง Google ด้วย

## แบบฝึกหัดก้ายบ

หลังจากผู้อ่านทดลองพัฒนาโปรแกรมการใช้งานระบบกำหนดตำแหน่งบนโลก หรือระบบ GPS เรียบร้อยแล้ว แบบฝึกหัดในส่วนนี้คือ ให้ผู้อ่านเพิ่มความสามารถ Zoom In และ Zoom Out ในส่วนของแผนที่



## การพัฒนาแอพพลิเคชันติดต่อบนเครือข่าย อย่างง่าย

เนื้อหาในบทนี้ จะเป็นการฝึกพัฒนาแอพพลิเคชันติดต่อลือสารบันเครือข่ายคอมพิวเตอร์ เพื่อให้ผู้อ่าน มีความเข้าใจในการส่งข้อมูลผ่านเครือข่ายคอมพิวเตอร์ หรือเครือข่ายอินเทอร์เน็ตในเบื้องต้น ทั้งในส่วนของ คลื่อนที่และส่วนของเซิร์ฟเวอร์

การพัฒนาแอพพลิเคชันเชื่อมต่อบนเครือข่าย (Network Programming) เสมือนกับการพัฒนาโปรแกรม คลื่อนที่/เซิร์ฟเวอร์ (Client/Server) โดยทั่วไปแล้วหลักการทำงานของแอพพลิเคชันจะเริ่มจากการร้องขอการ เชื่อมต่อจากฝั่งคลื่อนที่ไปยังเซิร์ฟเวอร์ โดยที่โปรแกรมฝั่งเซิร์ฟเวอร์จะเฝ้ารอการร้องขอการเชื่อมต่อจากฝั่ง คลื่อนที่ตลอดเวลา

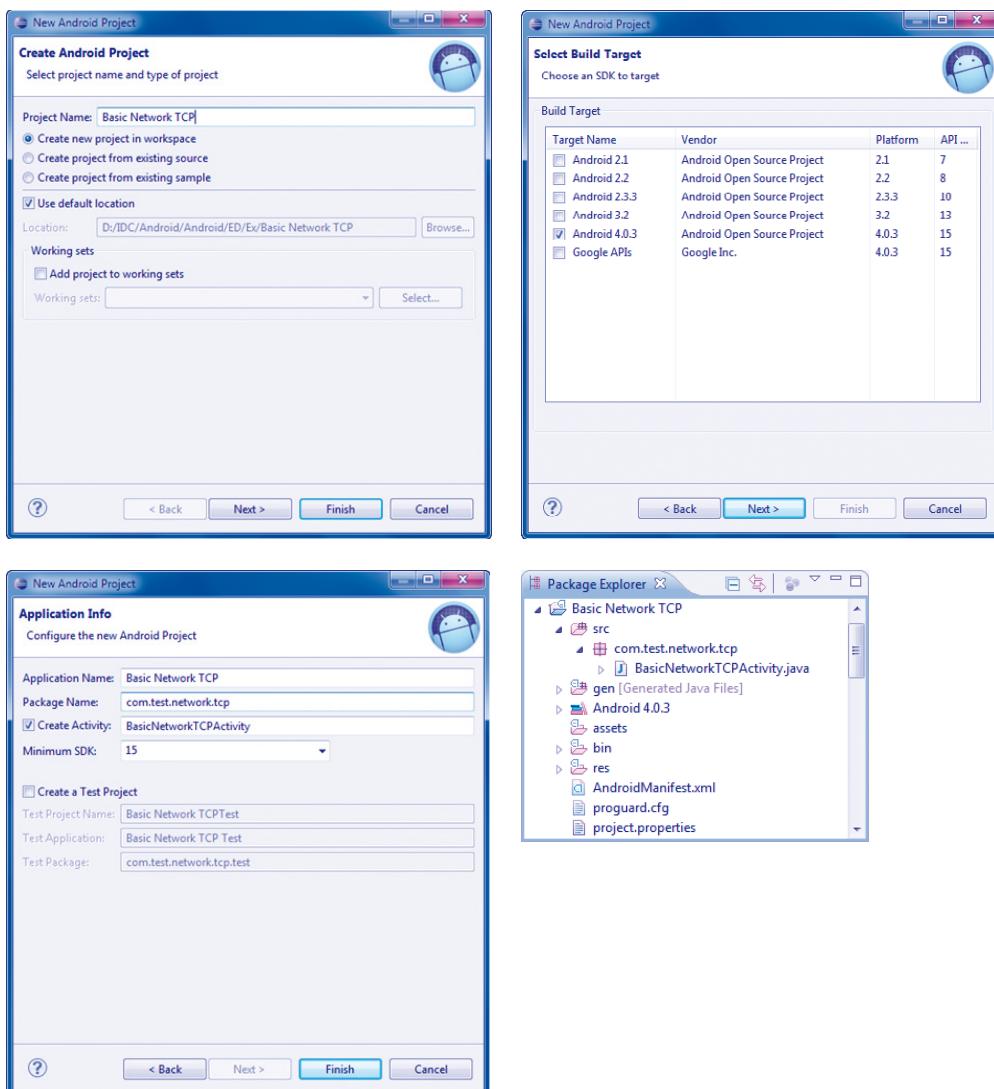


อย่างไรก็ตามเนื้อหาในบทนี้จะเน้นไปทางคลื่อนที่ เพื่อให้ผู้อ่านสามารถพัฒนาแอพพลิเคชันเชื่อมต่อ และส่งข้อความไปยังเซิร์ฟเวอร์ได้ จากนั้นเครื่องเซิร์ฟเวอร์จะลั่งข้อมูลกลับมายังเครื่องคลื่อนที่ โดยการพัฒนา โปรแกรมจะแสดงเป็นตัวอย่างของแอพพลิเคชันลงบนทนาย่างง่ายๆ โดยเนื้อหาแบ่งออกเป็น 2 ส่วนคือ

1. การสร้างแอพพลิเคชันส่วนเซิร์ฟเวอร์
2. การสร้างแอพพลิเคชันส่วนคลื่อนที่

# การสร้างแอพพลิเคชันส่วนเซิร์ฟเวอร์

ในส่วนของเซิร์ฟเวอร์นั้นจะทำหน้าที่ให้บริการแก่เครื่องเคลื่อนที่ที่เชื่อมต่อกัน ทั้งนี้การแลกเปลี่ยนหรือส่งข้อมูลระหว่างกันนั้นจะพัฒนาแอพพลิเคชันโดยใช้เทคนิคช่องเก็ต (Socket) โดยใช้บริการโปรโตคอล TCP (Transmission Control Protocol) ซึ่งขั้นตอนการสร้างแอพพลิเคชันส่วนเซิร์ฟเวอร์สามารถทำได้ตามขั้นตอนต่อไปนี้



1. สร้างโปรเจกต์ Android ใหม่ โดยกำหนดคุณสมบัติดังนี้
2. แก้ไขโคด XML ในไฟล์ main.xml ในส่วนของ res/layout เพื่อออกแบบหน้าจอดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <LinearLayout ← 1 กำหนดคุณสมบัติของ Layout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="10dip"
        >

        <Button ← 2 กำหนดคุณสมบัติของ Button
            android:id="@+id/run_as_server"
            android:layout_width="0dip"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Run as Server" />
        <Button
            android:id="@+id/run_as_client"
            android:layout_width="0dip"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Run as Client" />

    </LinearLayout>

    <TextView ← 3 กำหนดคุณสมบัติของ TextView
        android:id="@+id/textMode"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:layout_gravity="center_horizontal"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <LinearLayout ← 4 กำหนดคุณสมบัติของ Layout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="10dip"
        >

        <EditText ← 5 กำหนดคุณสมบัติ EditText
            android:id="@+id/editTextMessage"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" >

            <requestFocus />
        </EditText>
    </LinearLayout>

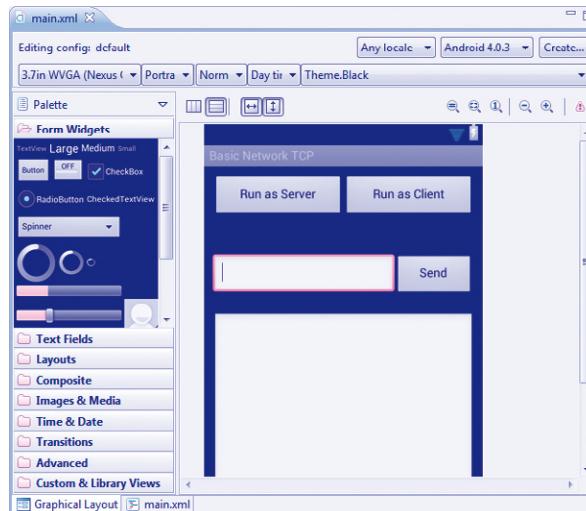
```

```

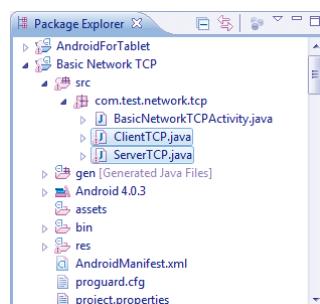
<Button <span style="color: red;">67

```

จะได้หน้าจอแอพพลิเคชันดังนี้



### 3. สร้างคลาส ServerTCP และ ClientTCP ไว้ภายในแพ็คเกจ com.test.network.tcp ดังนี้



#### 4. แก้ไขโค้ด Java ในไฟล์ BasicNetworkTCPActivity.java ดังนี้

```
package com.test.network.tcp;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class BasicNetworkTCPActivity extends Activity {

    private ServerTCP tcpServer; ← 1
    private ClientTCP tcpClient;

    final static String server_ip = "10.0.2.2";
    final static int connect_port = 7000;
    final static int timeout = 30000; // 30 sec
    final static int run_server_port = 7000;

    private int mode = 0;
    public static int SERVER_MODE = 1;
    public static int CLIENT_MODE = 2;

    EditText textSend;
    EditText textConver;
    TextView textMode;
    private Handler handler;

    public static String username = "TestClient";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        StrictMode.setThreadPolicy(new
        StrictMode.ThreadPolicy.Builder()
        .detectDiskReads().detectDiskWrites().detectNetwork()
        .penaltyLog().build());
        setContentView(R.layout.main);
        showGetNameDialog(); //-- get username
        handler = new Handler();
        textSend = (EditText) findViewById(R.id.editTextMessage);
        textConver = (EditText) findViewById(R.id.editTextConver);
        textMode = (TextView) findViewById(R.id.textMode);
        // --start server
        Button run_server = (Button) findViewById(R.id.run_as_server);
        Button run_client = (Button) findViewById(R.id.run_as_client);
        Button send = (Button) findViewById(R.id.buttonSend);
        run_server.setOnClickListener(new OnClickListener() {
```

```
public void onClick(View arg0) {
    startServer();

}
});  
run_client.setOnClickListener(new OnClickListener() {

    public void onClick(View arg0) {
        startClient();

    }
});  
send.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        sendMessage(username + " :"
                + textSend.getText().toString() + "\n");
    }
});  
}  
  
public void showGetNameDialog() {
    AlertDialog.Builder alert = new AlertDialog.Builder(this);

    alert.setTitle("Your Name");
    alert.setCancelable(false);
    // Set an EditText view to get user input
    final EditText input = new EditText(this);
    alert.setView(input);

    alert.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int whichButton) {
            username = input.getText().toString();

        }
    });
    alert.show();
}  
  
public void startServer() { 2
    //start server code
}  
  
public void startClient() { 3
    //start client code
}  
  
public void sendMessage(String text) {
    //send message to server
}  
  
//--update status
public void updateStatus(String text) { 4
```

```

        Toast.makeText(this, text, Toast.LENGTH_LONG).show();
    }
    //---update conversation ← 5
    public void updateConversation(final String text) {
        // use handler to update UI from thread;
        handler.post(new Runnable() {
            public void run() {
                textConver.append(text);
            }
        });
    }
}

```

### 5. แก้ไขโค้ดในไฟล์ ServerTCP.java เพื่อกำหนดการทำงานในส่วนของเซิร์ฟเวอร์ดังนี้

```

package com.test.network.tcp;

import java.io.BufferedReader;
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

import android.util.Log;

// -- TCP Server call when run as Server
public class ServerTCP implements Runnable {
    public static final String TAG = "ServerTCP Log";
    private ServerThread clients;
    private ServerSocket server = null;
    private Thread thread = null;
    private BasicNetworkTCPActivity _uclient;

    public ServerTCP(BasicNetworkTCPActivity uclient,int port) {
        _uclient = uclient;
        try {

            Log.d(TAG, "Binding to port " + port + ", please wait ...");
            server = new ServerSocket(port);
            Log.d(TAG, "Server started: " + server);
            start();
            _uclient.updateStatus("Server started: " + server);
        } catch (IOException ioe) {
            _uclient.updateStatus("Can not bind to port " + port + ":" + ioe.getMessage());
        }
    }

    public void run() {
        while (thread != null) {
            try {
                Log.d(TAG, "Waiting for a client ...");
                // add new client
                clients = new ServerThread(this,server.accept());
            }
        }
    }
}

```

```
        try {
            clients.open();
            clients.start();

        } catch (IOException ioe) {
            Log.d(TAG,"Error opening thread: " + ioe);
        }
    } catch (IOException ioe) {
        Log.d(TAG, "Server accept error: " + ioe);
        stop();
    }
}

public void start() {
    if (thread == null) {
        thread = new Thread(this);
        thread.start();
    }
}

public void stop() {
    if (thread != null) {
        thread.stop();
        thread = null;
    }
}
public void sendToClient(String text){
    try {
        clients.send(text);

    } catch (Exception e) {
        // TODO: handle exception
    }
}
//--handle message from client
public synchronized void handle(int ID, String input) {
    Log.d(TAG, "Receive from ID:" + ID + " msg:" + input);
    try {
        _uclient.updateConversation(input);
    } catch (Exception e) {

    }
}

public synchronized void remove(int ID) {
    try {
        clients.close();
    } catch (IOException ioe) {
        Log.d(TAG, "Error closing thread: " + ioe);
    }
    clients.stop();
}
```

```

// -- ServerThread create when client connect.
public class ServerThread extends Thread {
    private ServerTCP server = null;
    private Socket socket = null;
    private int ID = -1;

    private DataInputStream streamIn = null;
    private DataOutputStream streamOut = null;
    private boolean fstop = false;

    public ServerThread(ServerTCP _server, Socket _socket) {
        super();
        if (_server == null) {
            Log.e(TAG, "_server = null");
        }
        server = _server;
        socket = _socket;
        ID = socket.getPort();
    }

    public void tstop() {
        fstop = true;
    }

    public void send(String msg) { ←● 送ข้อความไปยัง Client
        try {
            streamOut.writeUTF(msg);
            streamOut.flush();
        } catch (IOException ioe) {
            Log.e(TAG, ID + " ERROR sending: " + ioe.getMessage());
            server.remove(ID);
            stop();
        }
    }

    public int getID() {
        return ID;
    }

    public void run() {
        Log.d(TAG, "Server Thread " + ID + " running.");
        while (!fstop) {
            try {
                server.handle(ID, streamIn.readUTF());
            } catch (IOException ioe) {
                Log.e(TAG, ID + " ERROR reading: " + ioe.getMessage());
                server.remove(ID);
                tstop();
            }
        }
    }

    public void open() throws IOException {
        streamIn = new DataInputStream(new BufferedInputStream(
            socket.getInputStream()));
        streamOut = new DataOutputStream(new BufferedOutputStream(
            socket.getOutputStream()));
    }
}

```

```

    public void close() throws IOException {
        if (socket != null)
            socket.close();
        if (streamIn != null)
            streamIn.close();
        if (streamOut != null)
            streamOut.close();
    }
}
}

```

## อธิบายโค้ดการทำงานของแอพพลิเคชัน

### 1. ประกาศตัวแปรหลักสำหรับการทำงานของแอพพลิเคชัน

```

private ServerTCP tcpServer;
private ClientTCP tcpClient;

final static String server_ip = "10.0.2.2"; ← กำหนด IP
final static int connect_port = 7000; ← กำหนด Port
final static int timeout = 30000; // 30 sec
final static int run_server_port = 7000;

private int mode = 0;
public static int SERVER_MODE = 1;
public static int CLIENT_MODE = 2;

```

2. เมธอด startServer() เป็นเมธอดสำหรับเริ่มต้นการทำงานเป็นเซิร์ฟเวอร์
3. เมธอด startClient() เป็นเมธอดสำหรับการทำงานเป็นไคลเอนท์
4. เมธอด updateStatus() เป็นเมธอดสำหรับแสดงสถานะการซ้อมต่อ

```

public void updateStatus(String text) {
    Toast.makeText(this, text, Toast.LENGTH_LONG).show();
}

```

5. เมธอด updateConversation() เป็นเมธอดสำหรับอัพเดตข้อความที่สุนทรากันระหว่างเซิร์ฟเวอร์ และไคลเอนท์

```

public void updateConversation(final String text) {
    // use handler to update UI from thread;
    handler.post(new Runnable() {
        public void run() {
            textConver.append(text);
        }
    });
}

```

เมื่อมีการสร้างออบเจกต์ของคลาส SeverTCP จะมีการแยกการทำงานออกเป็น-thread (Thread) สำหรับรับการเชื่อมต่อจากไคลเอนท์ และเมื่อได้รับการเชื่อมต่อจากไคลเอนท์จะมีการแยก-thread ของคลาส SeverThread อีกหนึ่ง-thread เพื่อรับข้อมูลที่ส่งมาจากไคลเอนท์ โดยในเบื้องต้นนี้จะสามารถเชื่อมต่อได้เครื่องเดียวเท่านั้น

**NOTE**

พอร์ต (Port) ใช้หลักการของ Port of Entry หรือช่องทางในการเข้ามาติดต่อบนเครื่องคอมพิวเตอร์นั้นๆ ซึ่งในกรณีที่มีการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ทั้ง 2 เครื่องนั้น จะสามารถติดต่อกันได้ผ่านช่องทางนี้เท่านั้น โดยปกติแล้วการบริการต่างๆ (Service) จะมีความลับพันธ์กับค่าพอร์ตนี้ เช่น

- การให้บริการเว็บโดยใช้โปรโตคอล HTTP (Hypertext Transfer Protocol) จะให้บริการที่พอร์ต 80
- การให้บริการล่งข้อมูลแบบ FTP (File Transfer Protocol) จะใช้พอร์ตที่ 20 และ 21
- การให้บริการ DNS (Domain Name System) จะใช้พอร์ตที่ 53 เป็นต้น

## การสร้างแอพพลิเคชันส่วน个体ลเอนท์

ในหัวข้อนี้จะเป็นการสร้างส่วนของคลาสส์ที่เข้าไปเชื่อมต่อกับเครื่องเซิร์ฟเวอร์ ซึ่งมีขั้นตอนการสร้างคลาสส์ดังนี้

### 1. แก้ไขโค้ด Java ในไฟล์ ClientTCP.java ดังนี้

```
package com.test.network.tcp;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.SocketAddress;
import java.net.SocketTimeoutException;
import java.net.UnknownHostException;

import android.util.Log;

public class ClientTCP {
    public static final String TAG = "ClientTCP";
    BasicNetworkTCPActivity _client;
    private Socket socket = null;
    private DataOutputStream streamOut = null;
    private ClientThread _tclient = null;

    public ClientTCP(BasicNetworkTCPActivity client, String serverName,
                    int serverPort, int timeout) {
        _client = client;
        _client.updateStatus("Establishing connection. Please wait ...");
        try {
            SocketAddress sockaddr = new InetSocketAddress(serverName,
                serverPort);
            socket = new Socket();
            socket.connect(sockaddr, timeout);
            Log.d(TAG, "Connected : " + socket);
            streamOut = new DataOutputStream(socket.getOutputStream());
            _tclient = new ClientThread(this, socket);
        }
        _client.updateStatus("Connected");
    }
}
```



```
        } catch (UnknownHostException e) {
            _client.updateStatus("Unknown host " + serverName);
            Log.e(TAG, "Unknown host " + serverName);
            e.printStackTrace();
        } catch (SocketTimeoutException e) {
            _client.updateStatus("Connect timeout");
            Log.e(TAG, "Connect timeout");
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void sendMessage(String text) {
        if (text == null)
            return;
        if (text != "") {
            try {
                if (streamOut != null) {
                    streamOut.writeUTF(text);
                    streamOut.flush();
                }
            } catch (IOException ioe) {
                Log.e(TAG, "Sending error: " + ioe.getMessage());
                close();
            }
        }
    }

    public void close() {
        try {
            if (streamOut != null)
                streamOut.close();
            if (socket != null)
                socket.close();
        } catch (IOException ioe) {
            Log.e(TAG, "Error closing ...");
        }
    }

    public void handle(String msg) {
        Log.i(TAG, "Revieve from server : " + msg);
        _client.updateConversation(msg);
    }

    public class ClientThread extends Thread {
        public static final String TAG = "Client Thread";
        private Socket socket = null;
        private ClientTCP client = null;
        private DataInputStream streamIn = null;
```

```

public boolean sflag = false;

public ClientThread(ClientTCP _client, Socket _socket) {
    client = _client;
    socket = _socket;
    open();
    start();
}

public void open() {
    try {
        streamIn = new DataInputStream(socket.getInputStream());
    } catch (IOException ioe) {
        Log.i(TAG, "Error getting input stream: " + ioe);
        client.close();
    }
}

public void close() {
    try {
        if (streamIn != null)
            streamIn.close();
    } catch (IOException ioe) {
        Log.d(TAG, "Error closing input stream: " + ioe);
    }
}

public void run() {
    while (!sflag) {
        try {
            client.handle(streamIn.readUTF());
        } catch (IOException ioe) {
            Log.d(TAG, "Listening error: " + ioe.getMessage());
            client.close();
            sflag = true;
        }
    }
}
}

```

การทำงานของโคลเอนท์จะมีลักษณะคล้ายกับเซิร์ฟเวอร์ โดยโคลเอนท์จะเชื่อมต่อไปยังเซิร์ฟเวอร์ ด้วยการกำหนดไอพี (IP) และพอร์ต (Port) ที่ต้องการเชื่อมต่อ เมื่อเชื่อมต่อสำเร็จจะมีการแยกเทรดออกจากเพื่อรับข้อมูลที่ส่งมาจากเซิร์ฟเวอร์ และแสดงข้อมูลที่ได้รับบนหน้าจอ

## 2. เขียนโค้ดเพิ่มในเมธอด startServer() ของคลาส BasicNetworkTCPActivity ดังนี้

```

public void startServer() {
    //start server code
    tcpServer = new ServerTCP(this, run_server_port);
    mode = SERVER_MODE;
    textMode.setText("Server Mode");
}

```

**3. เขียนโค้ดเพิ่มในเมธอด startClient() ของคลาส BasicNetworkTCPActivity ดังนี้**

```
public void startClient() {
    //start client code
    tcpClient = new ClientTCP(this, server_ip, connect_port, timeout);
    mode = CLIENT_MODE;
    textMode.setText("Client Mode");
}
```

**4. เขียนโค้ดเพิ่มในเมธอด sendMessage() ของคลาส BasicNetworkTCPActivity ดังนี้**

```
public void sendMessage(String text) {
    //send message to server

    if (mode == SERVER_MODE) {
        if (tcpServer != null) {
            updateConversation(text);
            tcpServer.sendToClient(text); // send message to client
        }
    } else if (mode == CLIENT_MODE) {

        if (tcpClient != null) {
            updateConversation(text);
            tcpClient.sendMessage(text);
        }
    }
}
```

**5. แก้ไขโค้ด XML ในไฟล์ AndroidManifest.xml โดยการเพิ่มสิทธิ์ให้โปรแกรมสามารถใช้งานผ่านเครือข่ายอินเทอร์เน็ต (Internet Permission) ได้โดยการเพิ่มโค้ดดังนี้**

```
<uses-permission android:name="android.permission.INTERNET"/>
```

ซึ่งจะได้โค้ดในไฟล์ AndroidManifest.xml ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.test.network.tcp"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="15" />
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".BasicNetworkTCPActivity"
            android:label="@string/app_name"
            android:theme="@android:style/Theme.Black"
            >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

```

```

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>

```

## ทดลองการทำงานของแอพพลิเคชัน

ในการทดสอบการทำงานของแอพพลิเคชันนั้นสามารถทดสอบได้ทั้งบน Emulator และเครื่องโทรศัพท์เคลื่อนที่จริง โดยการทดสอบบนเครื่องจริงนั้น จะต้องกำหนดไอพีเป็นไอพีของเครื่องที่รันเป็นเซิร์ฟเวอร์ในคลาส BasicNetworkTCPActivity ดังนี้

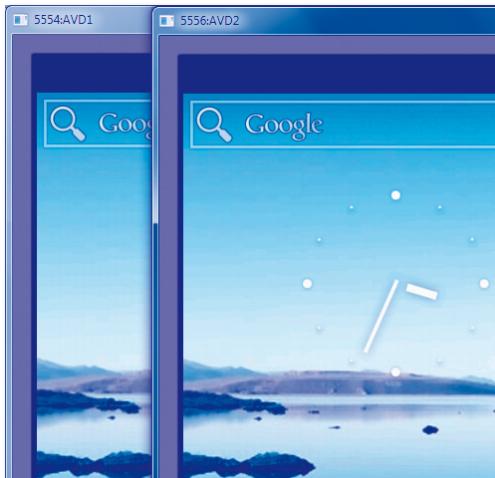
```

final static String server_ip = "10.0.2.2";
final static int connect_port = 7000;
final static int timeout = 30000; // 30 sec
final static int run_server_port = 7000;

```

ในที่นี่ค่าต่างๆ เป็นค่าที่กำหนดขึ้นเป็นตัวอย่างให้ผู้อ่านมองเห็นรูปแบบการกำหนดค่าเท่านั้น  
ในกรณีที่ผู้อ่านไม่สะดวกที่จะทดสอบบนเครื่องโทรศัพท์เคลื่อนที่จริง ในตัวอย่างนี้ผู้เขียนจะแสดงขั้นตอนการทดสอบแอพพลิเคชันบน Emulator ทั้ง 2 เครื่อง ซึ่งมีขั้นตอนดังนี้

1. สร้าง Emulator จำนวน 2 เครื่อง สำหรับใช้รันเป็นเซิร์ฟเวอร์และไคลเอนท์
2. รัน Emulator ทั้งสอง แล้วรอจนกว่าจะพร้อมใช้งาน ในที่นี่ให้ผู้อ่านรัน Emulator หมายเลขพอร์ต 5554 และ 5556 โดยจะใช้เครื่อง 5554 เป็นเซิร์ฟเวอร์



3. เปิดโปรแกรม Command Prompt และพิมพ์คำสั่งดังนี้

```
telnet 127.0.0.1 5554 ← หมายเลขอุปกรณ์
```

```
redir add tcp:7000:7000
redir list tcp
```

#### 4. จะได้ผลลัพธ์ดังนี้

```
Telnet 127.0.0.1
Android Console: type 'help' for a list of commands
OK
redir add tcp:7000:7000
OK
redir list tcp
tcp:7000 => 7000
OK
```

สำหรับคำสั่ง `redir add tcp:<localport>:<emulatorport>` จะเป็นการกำหนดการทำงานของคอมพิวเตอร์ คือ เมื่อมีการเชื่อมต่อไปยังพอร์ต 7000 ของเครื่องคอมพิวเตอร์นั้นๆ ก็จะถูกส่งต่อไปยังพอร์ต 7000 ของ Emulator โดยผู้อ่านจะต้องกำหนด IP เป็น “10.0.2.2” เพื่อเชื่อมต่อจาก Emulator ไปยังเครื่องคอมพิวเตอร์ เช่น `redir add tcp:5000:6000`

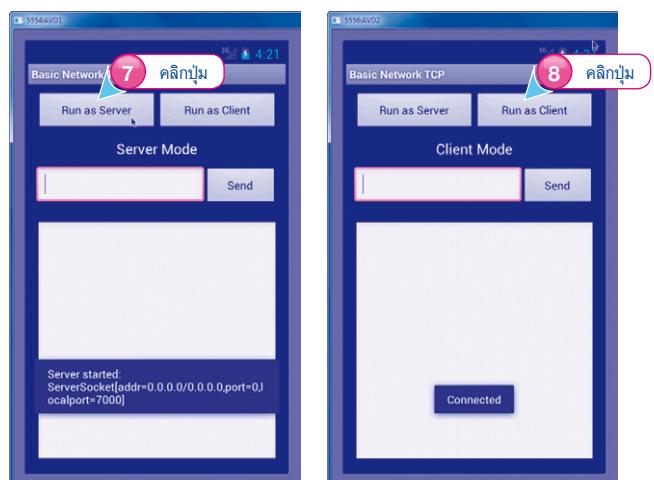
#### 5. รันแอพพลิเคชันบน Emulator ทั้ง 2 เครื่อง

#### 6. กรอกชื่อผู้ใช้งานดังนี้

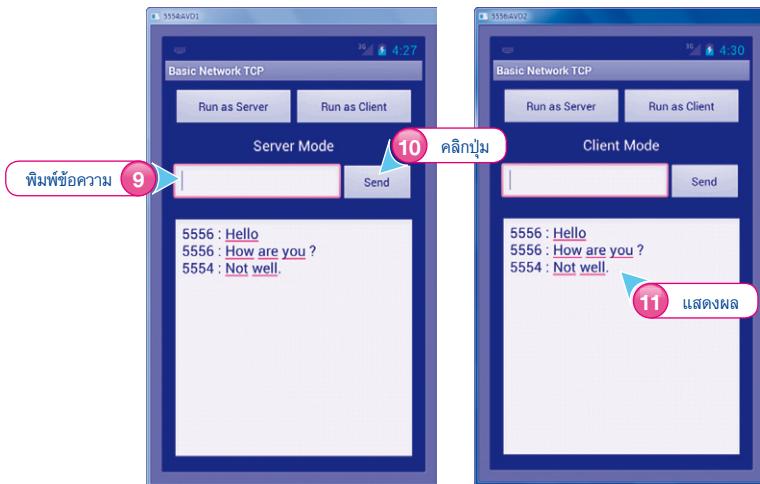


#### 7. ที่เครื่อง 5554 คลิกปุ่ม Run as Server

#### 8. ที่เครื่อง 5556 คลิกปุ่ม Run as Client เพื่อเชื่อมต่อไปยังเซิร์ฟเวอร์



9. พิมพ์ข้อความที่ต้องการ
10. คลิกปุ่ม Send
11. ข้อความจะถูกส่งไปยังเครื่องปลายทาง ซึ่งได้ผลลัพธ์ดังนี้

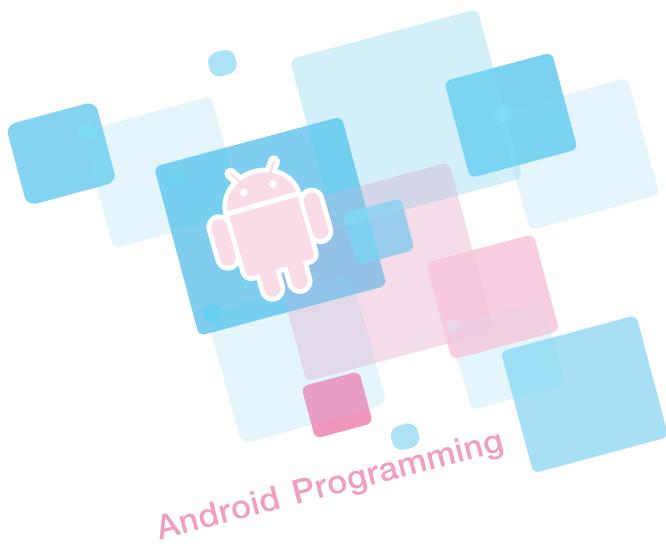


## สรุปท้ายบท

ในบทนี้เป็นการอธิบายการพัฒนาแอพพลิเคชัน เพื่อทำงานเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์/โทรศัพท์เคลื่อนที่ Android ในรูปแบบของโคลเอนท์/เซิร์ฟเวอร์ โดยใช้งาน Socket Programming

## แบบฝึกหัดท้ายบท

จงพัฒนาแอพพลิเคชันเพื่อแลกเปลี่ยนข้อความระหว่างกันทั้ง 2 ฝ่าย ในส่วนของผู้ใช้โคลเอนท์และเซิร์ฟเวอร์ (ทั้งสองฝ่ายสามารถส่งข้อความหากันได้) โดยที่เซิร์ฟเวอร์รองรับการเชื่อมต่อของโคลเอนท์ได้มากกว่า 1 โคลเอนท์ และเมื่อโคลเอนท์เครื่องใดส่งข้อมูล เซิร์ฟเวอร์จะกระจาย (Broadcast) ข้อมูลไปยังเครื่องโคลเอนท์ทุกเครื่องยกเว้นเครื่องที่ส่งข้อมูลได้



## การพัฒนาแอพพลิเคชันทำงานกับฐานข้อมูล

เนื้อหาในบทนี้จะเป็นการฝึกพัฒนาแอพพลิเคชันทำงานกับฐานข้อมูล (Database) ที่ติดตั้งบนระบบปฏิบัติการ Android เพื่อเป็นพื้นฐานให้ผู้อ่านสามารถนำไปประยุกต์ใช้งานในการพัฒนาแอพพลิเคชันขั้นสูงได้อย่างมีประสิทธิภาพ

### ความรู้เบื้องต้นเกี่ยวกับการทำงานกับฐานข้อมูล

ถึงแม้ว่าในปัจจุบันฐานข้อมูลที่มีใช้อยู่นั้นจะมีหลากหลายตัวอย่าง เช่น MySQL, SQLite และ Microsoft Access แต่ในบทเรียนนี้ผู้เรียนจะใช้อ้างอิงกับฐานข้อมูลของ SQLite

ฐานข้อมูลของ SQLite เป็นระบบฐานข้อมูลที่ได้รับความนิยมเป็นอย่างมากใน Smart Phone และระบบปฏิบัติการ Android เนื่องจาก SQLite มีโครงสร้างไม่ซับซ้อน รวมไปถึงคุณลักษณะของ SQLite ที่มีการทำงานแบบเครื่องเดียว (Standalone) ซึ่งจะเหมาะสมสำหรับฐานข้อมูลที่มีขนาดไม่ใหญ่มากนัก เนื่องจากโทรศัพท์เคลื่อนที่นั้นมีข้อจำกัดในหลายๆ ด้าน เช่น หน่วยความจำและความสามารถของการประมวลผล เป็นต้น

ระบบจัดการฐานข้อมูล หรือ DBMS (Database Management System) สามารถแบ่งลักษณะการทำงานออกได้เป็นหลายๆ ลักษณะ เช่น แอพพลิเคชันที่ใช้จัดการฐานข้อมูล อุปกรณ์หน่วยความจำ (Device Memory) และการรักษาความมั่นคงปลอดภัย (Security) เป็นต้น

ในปัจจุบันมี DBMS ที่นิยมใช้งานมากมาย เช่น MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, Sybase และ dBASE เป็นต้น ซึ่งปกติแล้ว DBMS แต่ละชนิดจะมีลักษณะเชื่อมต่อให้ฐานข้อมูลสามารถทำงานร่วมกับฐานข้อมูลประเภทอื่นๆ ได้ ซึ่งเรียกว่า **ODBC (Open Database Connectivity)**

งานอย่างหนึ่งที่ถือได้ว่า มีความสำคัญกับระบบจัดการฐานข้อมูลเป็นอย่างยิ่งก็คือ การออกแบบฐานข้อมูลโดยเฉพาะในด้านประสาทวิภาค การทำงานเมื่อคลังข้อมูลมีขนาดใหญ่ ซึ่งสามารถแบ่งแนวคิดการออกแบบฐานข้อมูลได้เป็นหลายประเภท เช่น

- รูปแบบข้อมูลแบบโครงสร้างแบบลำดับชั้น (Hierarchical data model) – One to Many
- รูปแบบข้อมูลแบบเครือข่าย (Network data model) – Many to Many
- รูปแบบความล้มพันธ์ข้อมูล (Relation data model) – Row and Column

ทั้งนี้ผู้อ่านสามารถศึกษาเพิ่มเติมเกี่ยวกับทฤษฎีต่างๆ ที่เกี่ยวข้องกับฐานข้อมูลได้จากเว็บไซต์ หรือหนังสือต่างๆ เช่น Database Systems : The Complete Book โดย Hector Garcia-Molina, Jeffrey D. Ullman และ Jennifer Widom เป็นต้น

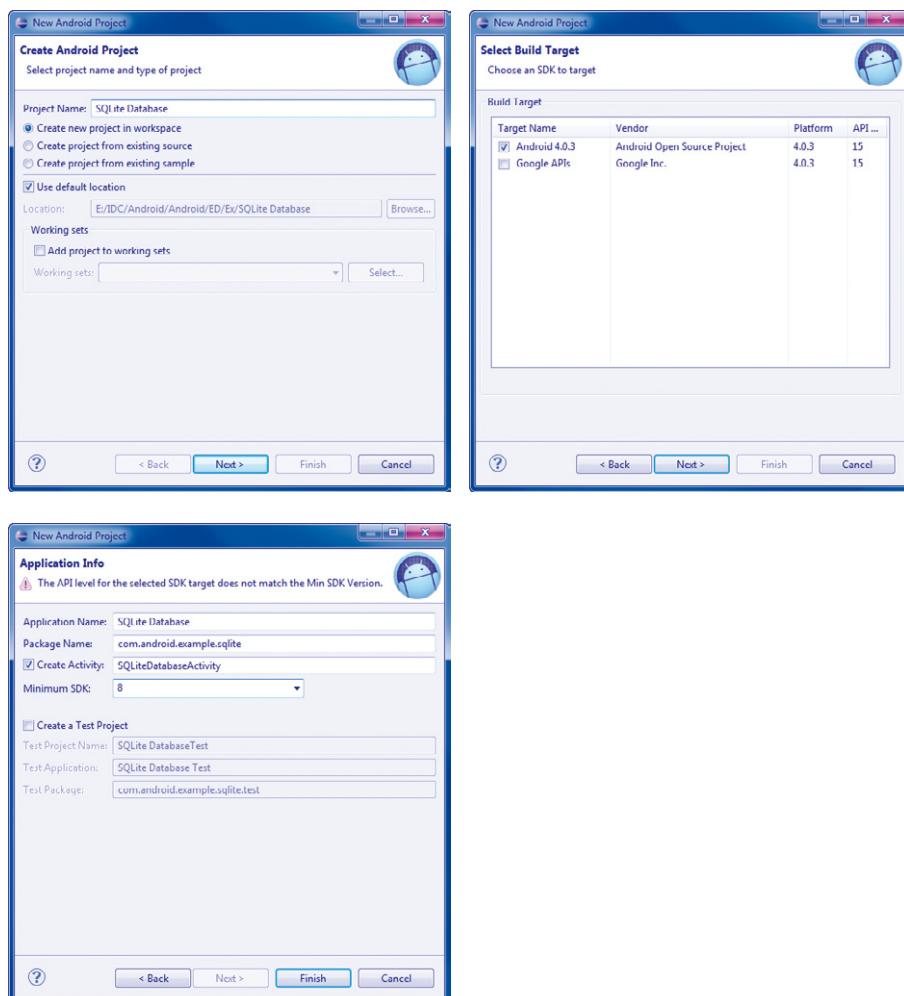
ทั้งนี้เนื้อหาในบทนี้จะแบ่งออกเป็น 2 ส่วนคือ

1. การสร้างล้วนสำหรับเชื่อมต่อและใช้งานฐานข้อมูล
2. การใช้งานฐานข้อมูลผ่านยูสเซอร์รีเฟิร์ส (User Interface)

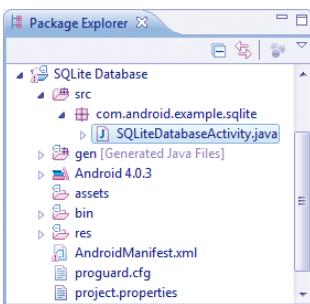
## การสร้างส่วนเชื่อมต่อและใช้งานฐานข้อมูล

ในหัวข้อนี้จะเป็นการสร้างล้วนเชื่อมต่อกับฐานข้อมูลที่เรียกว่า **Database Adapter** ซึ่งสามารถสร้างได้ตามขั้นตอนดังนี้

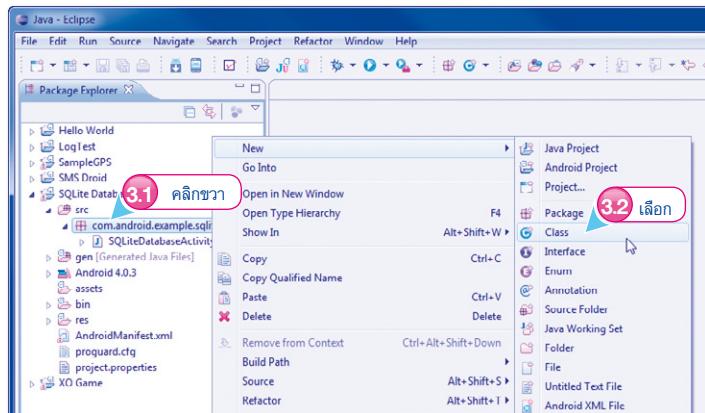
1. สร้างโปรเจกต์ใหม่ชื่อว่า SQLite Database ซึ่งมีคุณสมบัติดังนี้



2. จะได้ไปรูปแบบดังนี้



3. สร้างคลาส (Class) SimpleSQLiteHelper โดยคลิกขวาที่ไฟล์ com.android.example.sqlite และเลือกคำสั่ง New > Class



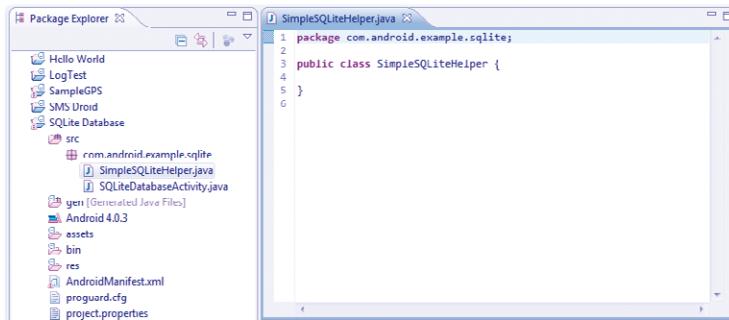
4. จะปรากฏ dialogue ให้ออกบัน្តอกซ์ New Java Class ขึ้นมา

5. ตั้งชื่อคลาสเป็น SimpleSQLiteHelper

6. คลิกปุ่ม **Finish**



## 7. จะประยุกต์ไฟล์ SimpleSQLiteHelper.java ขึ้นมาดังรูป



## 8. แก้ไขโค้ด Java ในไฟล์ SimpleSQLiteHelper.java โดยเพิ่มเติมเดิมดังนี้

```

package com.android.example.sqlite;

import android.content.Context; ← 2
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class SimpleSQLiteHelper extends SQLiteOpenHelper { ← 1

    public static final String TABLE_BOOKS = "book"; ← กำหนดชื่อตาราง
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_ISBN = "isbn";
    public static final String COLUMN_TITLE = "title";
    public static final String COLUMN_PRICE = "price";
    public static final String COLUMN_PUBLISHER = "publisher";
    private static final String DATABASE_NAME = "books.db"; ← กำหนดชื่อรากฐานข้อมูล
    private static final int DATABASE_VERSION = 1; ← 3

    // Database creation sql statement
    private static final String DATABASE_CREATE = "create table "
        + TABLE_BOOKS + "("
        + COLUMN_ID + " integer primary key autoincrement, "
        + COLUMN_ISBN + " text,"
        + COLUMN_TITLE + " text,"
        + COLUMN_PRICE + " numeric,"
        + COLUMN_PUBLISHER + " text" + ");";

    public SimpleSQLiteHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override ← 4
    public void onCreate(SQLiteDatabase database) {
        database.execSQL(DATABASE_CREATE);
    }

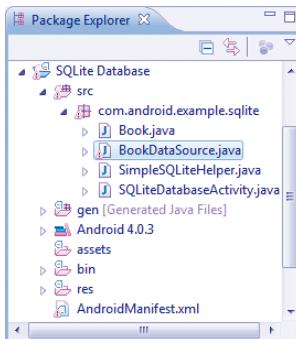
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.w(SimpleSQLiteHelper.class.getName(),
    
```

```

        "Upgrading database from version " + oldVersion + " to "
        + newVersion + ", which will destroy all old data");
    db.execSQL("DROP TABLE IF EXISTS" + TABLE_BOOKS);
    onCreate(db);
}
}

```

### 9. สร้างคลาส Book และคลาส BookDataSource ดังนี้



### 10. เขียนโค้ดกำหนดการทำงานของคลาส Book ดังนี้

```

package com.android.example.sqlite;

public class Book {
    private long id;
    private String isbn;
    private String title;
    private double price;
    private String publisher;

    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public String getIsbn() {
        return isbn;
    }
    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public double getPrice() {
        return price;
    }
}

```

```

public void setPrice(double price) {
    this.price = price;
}
public String getPublisher() {
    return publisher;
}
public void setPublisher(String publisher) {
    this.publisher = publisher;
}
// Will be used by the ArrayAdapter in the ListView
@Override
public String toString() {
    return title;
}
}

```

11. เขียนโค้ดกำหนดการทำงานของคลาส BookDataSource เพื่อใช้สำหรับเพิ่ม ลบ แก้ไข และ อ่านข้อมูลจากฐานข้อมูลดังนี้

```

package com.android.example.sqlite;

import java.util.ArrayList;
import java.util.List;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;

public class BookDataSource {
    // Database fields
    public static final String TAG = "SQLite Database";
    private SQLiteDatabase database;
    private SimpleSQLiteHelper dbHelper;
    private String[] allColumns = { SimpleSQLiteHelper.COLUMN_ID,
        SimpleSQLiteHelper.COLUMN_ISBN,
        SimpleSQLiteHelper.COLUMN_TITLE,
        SimpleSQLiteHelper.COLUMN_PRICE,
        SimpleSQLiteHelper.COLUMN_PUBLISHER };

    public BookDataSource(Context context) {
        dbHelper = new SimpleSQLiteHelper(context);
    }

    public void open() throws SQLException {
        database = dbHelper.getWritableDatabase();
    }

    public void close() {
        dbHelper.close();
    }
}

```

```

public Book insertBook(Book book) { ← 5 ส่วนเพิ่มรายการหนังสือ
    ContentValues values = new ContentValues();
    values.put(SimpleSQLiteHelper.COLUMN_ISBN, book.getIsbn());
    values.put(SimpleSQLiteHelper.COLUMN_TITLE, book.getTitle());
    values.put(SimpleSQLiteHelper.COLUMN_PRICE, book.getPrice());
    values.put(SimpleSQLiteHelper.COLUMN_PUBLISHER, book.getPublisher());
    long insertId = database.insert(SimpleSQLiteHelper.TABLE_BOOKS, null,
        values);

    Cursor cursor = database.query(SimpleSQLiteHelper.TABLE_BOOKS,
        allColumns, SimpleSQLiteHelper.COLUMN_ID + " = " + insertId,
        null, null, null);
    cursor.moveToFirst();
    return cursorToBook(cursor);
}

public void deleteBook(Book book) { ← 6 ส่วนลบรายการหนังสือ
    long id = book.getId();
    Log.d(TAG,"Book deleted with id: " + id);
    database.delete(SimpleSQLiteHelper.TABLE_BOOKS,
        SimpleSQLiteHelper.COLUMN_ID + " = " + id, null);
}

public void updateBook(Book book) { ← 7 ส่วนอัปเดตรายการหนังสือ
    ContentValues values = new ContentValues();
    values.put(SimpleSQLiteHelper.COLUMN_ISBN, book.getIsbn());
    values.put(SimpleSQLiteHelper.COLUMN_TITLE, book.getTitle());
    values.put(SimpleSQLiteHelper.COLUMN_PRICE, book.getPrice());
    values.put(SimpleSQLiteHelper.COLUMN_PUBLISHER, book.getPublisher());
    database.update(SimpleSQLiteHelper.TABLE_BOOKS, values,
        SimpleSQLiteHelper.COLUMN_ID + "=" + book.getId(), null);
}

public List<Book> getAllBook() { ← 8 ส่วนแสดงรายการหนังสือ
    List<Book> comments = new ArrayList<Book>();
    Cursor cursor = database.query(SimpleSQLiteHelper.TABLE_BOOKS,
        allColumns, null, null, null, null, null);
    cursor.moveToFirst();
    while (!cursor.isAfterLast()) {
        Book comment = cursorToBook(cursor);
        comments.add(comment);
        cursor.moveToNext();
    }
    cursor.close();
    return comments;
}

private Book cursorToBook(Cursor cursor) {
    Book book = new Book();
    book.setId(cursor.getLong(0));
    book.setIsbn(cursor.getString(1));
    book.setTitle(cursor.getString(2));
    book.setPrice(cursor.getDouble(3));
    book.setPublisher(cursor.getString(4));
    return book;
}
}

```

## อธิบายโค้ดการทำงานของแอพพลิเคชัน

### 1. สร้างคลาส SimpleSQLiteHelper โดยขยายมาจากคลาส SQLiteOpenHelper

```
public class SimpleSQLiteHelper extends SQLiteOpenHelper {

    public SimpleSQLiteHelper(Context context) {
        super(context, null, null, null);
    }

    @Override
    public void onCreate(SQLiteDatabase database) {
        // ...
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // ...
    }
}
```

### 2. เขียนโค้ดเพื่อนำเข้า Namespace ที่จะใช้งาน เช่น android.database.\*

```
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
```

### 3. ประกาศค่าคงที่ต่างๆ ที่จำเป็นในการทำงานของแอพพลิเคชัน ซึ่งผู้เขียนได้ประกาศค่าคงที่สำหรับ อ้างอิงกับฟิลด์ที่สำคัญๆ ในฐานข้อมูล เช่น ID, ISBN, TITLE, PRICE และ PUBLISHER เป็นต้น ทั้งนี้ผู้อ่านสามารถเพิ่มเติมหรือแก้ไขเองได้ตามความเหมาะสม

```
public static final String TABLE_BOOKS = "book";
public static final String COLUMN_ID = "_id";
public static final String COLUMN_ISBN = "isbn";
public static final String COLUMN_TITLE = "title";
public static final String COLUMN_PRICE = "price";
public static final String COLUMN_PUBLISHER = "publisher";
private static final String DATABASE_NAME = "mybook.db";
private static final int DATABASE_VERSION = 1;

// Database creation sql statement
private static final String DATABASE_CREATE = "create table "
    + TABLE_BOOKS + "("
    + COLUMN_ID + " integer primary key autoincrement, "
    + COLUMN_ISBN + " text,"
    + COLUMN_TITLE + " text,"
    + COLUMN_PRICE + " numeric,"
    + COLUMN_PUBLISHER + " text" + ");";
```

#### 4. Override เมธอด onCreate() และ onUpgrade เพื่อสร้างและปรับปรุงฐานข้อมูล

```
@Override  
public void onCreate(SQLiteDatabase database) {  
    database.execSQL(DATABASE_CREATE);  
}  
  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    Log.w(SimpleSQLiteHelper.class.getName(),  
        "Upgrading database from version " + oldVersion + " to "  
        + newVersion + ", which will destroy all old data");  
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_BOOKS);  
    onCreate(db);  
}
```

#### 5. ส่วนเพิ่มรายการหนังสือในฐานข้อมูล

```
public Book insertBook(Book book) {  
    ContentValues values = new ContentValues();  
    values.put(SimpleSQLiteHelper.COLUMN_ISBN, book.getIsbn());  
    values.put(SimpleSQLiteHelper.COLUMN_TITLE, book.getTitle());  
    values.put(SimpleSQLiteHelper.COLUMN_PRICE, book.getPrice());  
    values.put(SimpleSQLiteHelper.COLUMN_PUBLISHER, book.getPublisher());  
    long insertId = database.insert(SimpleSQLiteHelper.TABLE_BOOKS, null,  
        values);  
  
    Cursor cursor = database.query(SimpleSQLiteHelper.TABLE_BOOKS,  
        allColumns, SimpleSQLiteHelper.COLUMN_ID + " = " + insertId,  
        null, null, null, null);  
    cursor.moveToFirst();  
    return cursorToBook(cursor);  
}
```

#### 6. ส่วนลบรายการหนังสือในฐานข้อมูล

```
public void deleteBook(Book book) {  
    long id = book.getId();  
    Log.d(TAG, "Book deleted with id: " + id);  
    database.delete(SimpleSQLiteHelper.TABLE_BOOKS,  
        SimpleSQLiteHelper.COLUMN_ID + " = " + id, null);  
}
```

#### 7. ส่วนแก้ไขรายการหนังสือในฐานข้อมูล

```
public void updateBook(Book book) {  
    ContentValues values = new ContentValues();  
    values.put(SimpleSQLiteHelper.COLUMN_ISBN, book.getIsbn());  
    values.put(SimpleSQLiteHelper.COLUMN_TITLE, book.getTitle());  
    values.put(SimpleSQLiteHelper.COLUMN_PRICE, book.getPrice());  
    values.put(SimpleSQLiteHelper.COLUMN_PUBLISHER, book.getPublisher());  
    database.update(SimpleSQLiteHelper.TABLE_BOOKS, values,  
        SimpleSQLiteHelper.COLUMN_ID + " = " + book.getId(), null);  
}
```



# การใช้งานบล็อกผ่าน User Interface

เมื่อสร้างส่วนเชื่อมต่อกับฐานข้อมูลแล้ว ต่อไปจะเป็นการเข้าใช้งานฐานข้อมูลผ่าน User Interface ซึ่งประกอบด้วยการอ่านข้อมูล การเพิ่มข้อมูล การลบข้อมูล และการแก้ไขข้อมูล

## ออกแบบหน้าจอแอพพลิเคชัน

การออกแบบหน้าจอแอพพลิเคชัน เป็นการแก้ไขโค้ดในไฟล์ XML เพื่อแสดงรายการข้อมูลและการรับข้อมูลจากผู้ใช้งานดังนี้

1. แก้ไขโค้ด XML ในไฟล์ main.xml เพื่อออกแบบหน้าจอในส่วนแสดงข้อมูลดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <LinearLayout ← 1 กำหนดคุณสมบัติของ Layout
        android:id="@+id/group"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <Button ← 2 กำหนดคุณสมบัติของ Button
            android:id="@+id/buttonAdd"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Add New" />

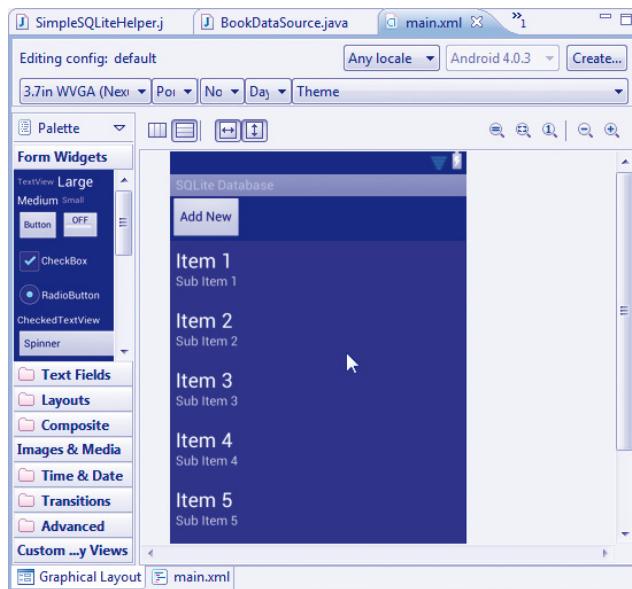
    </LinearLayout>

    <ListView ← 3 กำหนดคุณสมบัติของ ListView
        android:id="@+id/list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>

```

## จะได้ผลลัพธ์ดังนี้



## 2. สร้างไฟล์ add\_book.xml สำหรับใช้รับข้อมูลจากผู้ใช้งานดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="250dip"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:padding="10dip" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ISBN :" />

    <EditText <span style="color: red;">1
```

```

<EditText ← 2 กำหนดคุณสมบัติของ Title
    android:id="@+id/editTextTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Price :" />

<EditText ← 3 กำหนดคุณสมบัติของ Price
    android:id="@+id/editTextPrice"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Publisher :" />

<EditText ← 4 กำหนดคุณสมบัติของ Publisher
    android:id="@+id/editTextPublisher"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

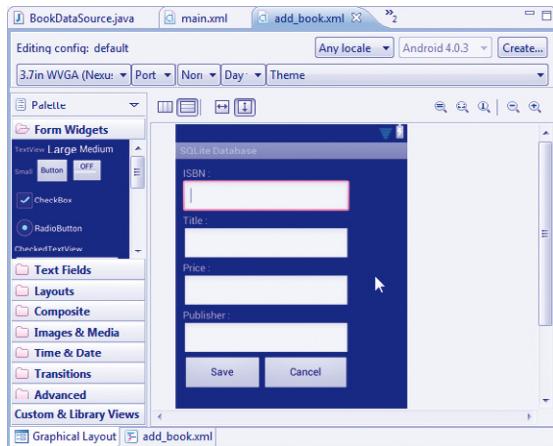
    <Button ← 5 กำหนดคุณสมบัติของปุ่ม Save
        android:id="@+id/buttonSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Save" />

    <Button ← 6 กำหนดคุณสมบัติของปุ่ม Cancel
        android:id="@+id/buttonCancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Cancel" />
</LinearLayout>

</LinearLayout>

```

## จะได้ผลลัพธ์ดังนี้



### 3. สร้างไฟล์ detail\_book.xml สำหรับแสดงรายละเอียดของข้อมูลดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp" >

    <TextView > 1 กำหนดคุณสมบัติของส่วนแสดง ISBN
        android:id="@+id/textBookISBN"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ISBN" />

    <TextView > 2 กำหนดคุณสมบัติของส่วนแสดง Title
        android:id="@+id/textBookTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Title" />

    <TextView > 3 กำหนดคุณสมบัติของส่วนแสดง Price
        android:id="@+id/textBookPrice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Price" />

    <TextView > 4 กำหนดคุณสมบัติของส่วนแสดง Publisher
        android:id="@+id/textBookPublisher"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Publisher" />

```

```

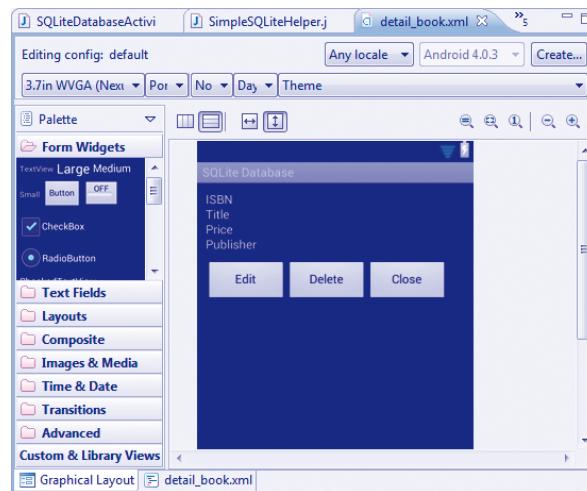
<LinearLayout ← 6 กำหนดคุณสมบัติของ Layout
    android:id="@+id/linearLayout1"
    android:layout_marginTop="10dip"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    >

    <Button ← 7 กำหนดคุณสมบัติของปุ่ม Edit
        android:id="@+id/buttonEdit"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1"
        android:text="Edit" />
    <Button ← 8 กำหนดคุณสมบัติของปุ่ม Delete
        android:id="@+id/buttonDelete"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1"
        android:text="Delete" />
    <Button ← 9 กำหนดคุณสมบัติของปุ่ม Close
        android:id="@+id/buttonClose"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1"
        android:text="Close" />
</LinearLayout>

</LinearLayout>

```

จะได้ผลลัพธ์ดังนี้



## ເປີຍໂຄດກໍາຫຼາດການກໍາທຳງານຂອງແອພພລິເຄຊັນ

ຫວ່າງໝັ້ນນີ້ເປັນການແກ້ໄຂໂຄດ Java ໃນໄຟລ් SQLiteDatabaseActivity.java ເພື່ອກໍາຫຼາດໃຫ້ແອພພລິເຄຊັນກໍາທຳງານກັບສ້າງຂໍ້ມູນລາຍມາທີ່ຕ້ອງການ ຂຶ້ນສິ້ນຄອນດังນີ້

1. ເພີ່ມໂຄດໃນໄຟລ් SQLiteDatabaseActivity.java ດັ່ງນີ້

```
package com.android.example.sqlite;

import java.util.List;

import android.app.Dialog;
import android.app.ListActivity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;

public class SQLiteDatabaseActivity extends ListActivity {
    final static String TAG = "SQLite";
    private BookDataSource datasource;
    List<Book> values;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button add_book = (Button) findViewById(R.id.buttonAdd);
        add_book.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                addNewBook();
            }
        });
        datasource = new BookDataSource(this);
        datasource.open();

        showAllBook(); // show all book.
    }

    public void showAllBook() {
        // show all book code
    }

    @Override
    protected void onListItemClick(ListView l, View v, int position, long id) {
        super.onListItemClick(l, v, position, id);
        showBookDetail(position); // View book by id
    }

    public void showBookDetail(final int id) {
        // show book detail code
    }
}
```

```

public void editBook(final Book book) {
    // edit book code
}

public void addNewBook() {
    // add new book code
}

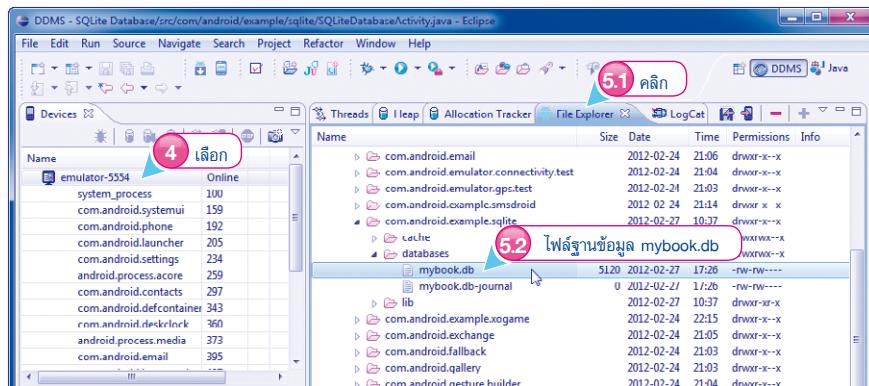
@Override
protected void onResume() {
    datasource.open();
    super.onResume();
}

@Override
protected void onPause() {
    datasource.close();
    super.onPause();
}
}

```

โค้ดในส่วนนี้เป็นการสร้างออบเจกต์ของคลาส BookDataSource เพื่อเข้าใช้งานในส่วนของการเชื่อมต่อกับฐานข้อมูลที่ได้สร้างไว้ในหัวข้อที่ผ่านมา

2. รันแอพพลิเคชันเพื่อตรวจสอบการสร้างฐานข้อมูลใหม่
3. เปิดใช้งาน DDMS โดยคลิกเมนู Window > Open Perspective > DDMS ที่โปรแกรม Eclipse เพื่อตรวจสอบว่าฐานข้อมูลได้ถูกสร้างไว้จริงหรือไม่
4. คลิกเลือก Emulator ที่ใช้ในการทดสอบแอพพลิเคชัน
5. จากนั้นให้ผู้อ่านคลิกแท็บ File Explorer และค้นหาแพ็คเกจที่ผู้อ่านได้สร้างขึ้น ซึ่งจะปรากฏเป็นฐานข้อมูลชื่อ mybook.db และเป็นการยืนยันว่าฐานข้อมูลนั้นได้ถูกสร้างขึ้นมาแล้ว



ปกติแล้วฐานข้อมูลจะถูกสร้างขึ้นที่ไดร์กทอรี /data/data/<package\_name>/databases folder แต่ในกรณีที่ผู้อ่านทำตามขั้นตอนที่กล่าวมาข้างต้น ฐานข้อมูลจะถูกสร้างที่ไดร์กทอรี /data/data/com.android.example.sqlite/databases ทั้งนี้ข้อลังก์เต็คือ ผู้อ่านจะต้องรันแอพพลิเคชันก่อนจึงจะเข้าถึงฐานข้อมูลได้

## TIPS



ในการทดสอบแอพพลิเคชัน ผู้อ่านสามารถลบไฟล์ฐานข้อมูลในอีเมลเตอร์ทึ้งได้ โดยพิมพ์คำสั่งที่ Command Prompt ดังนี้

```
adb shell  
cd /data/data/  
cd <your.application.java.package>  
cd databases  
rm <your db name>
```

ตัวอย่างเช่น

```
Administrator: C:\Windows\system32\cmd.exe - adb shell  
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation.  
C:\Users\Bannok>adb shell  
# cd /data/data/  
# cd com.android.example.sqlite  
# cd com.android.example.sqlite  
# cd databases  
# rm mybook.db  
rm mybook.db  
#
```

แสดงผลการลบฐานข้อมูล ซึ่งล้างเกตได้ว่าจะไม่มีฐานข้อมูล mybook.db ดังนี้

Name	Size	Date	Time	Permissions
com.android.email		2012-02-24	21:06	drwxr-x--x
com.android.emulator.connecti		2012-02-24	21:04	drwxr-x--x
com.android.emulator.gps.test		2012-02-24	21:03	drwxr-x--x
com.android.example.smsdroid		2012-02-24	21:14	drwxr-x--x
com.android.example.sqlite		2012-02-27	10:37	drwxr-x--x
cache		2012-02-27	10:37	drwxrwx--x
databases	0	2012-02-27	18:03	drwxrwx--x
mybook.db-journal	0	2012-02-27	17:26	-rw-rw----
lib		2012-02-27	10:37	drwxr-xr-x
com.android.example.xogame		2012-02-24	22:15	drwxr-x--x
com.android.exchange		2012-02-24	21:05	drwxr-x--x
com.android.fallback		2012-02-24	21:03	drwxr-x--x

6. เขียนโค้ดเพิ่มเติม เพื่อใช้สำหรับการดึงข้อมูลจากฐานข้อมูล เพิ่มข้อมูลลงในฐานข้อมูล ดูรายละเอียดข้อมูล แก้ไขข้อมูล และลบข้อมูล ให้ผู้อ่านแก้ไขโค้ดเพิ่มเติมดังนี้

```
package com.android.example.sqlite;  
  
import java.util.List;  
  
import android.app.Dialog;  
import android.app.ListActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;
```

```
public class SQLiteDatabaseActivity extends ListActivity {
    final static String TAG = "SQLite";
    private BookDataSource datasource;
    List<Book> values;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button add_book = (Button) findViewById(R.id.buttonAdd);
        add_book.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                addNewBook();
            }
        });
        datasource = new BookDataSource(this);
        datasource.open();
        showAllBook(); // show all book.
    }

    public void showAllBook() {
        values = datasource.getAllBook();
        ArrayAdapter<Book> adapter = new ArrayAdapter<Book>(this,
            android.R.layout.simple_list_item_1, values);
        setListAdapter(adapter);
    }

    @Override ← 2
    protected void onListItemClick(ListView l, View v, int position, long id) {
        // TODO Auto-generated method stub
        super.onListItemClick(l, v, position, id);
        showBookDetail(position); // View book by id
    }

    public void showBookDetail(final int id) {
        @SuppressWarnings("unchecked")
        final ArrayAdapter<Book> adapter = (ArrayAdapter<Book>) getListAdapter();
        final Book book = (Book) getListAdapter().getItem(id);

        final Dialog dialog = new Dialog(SQLiteDatabaseActivity.this);
        dialog.setContentView(R.layout.detail_book);
        dialog.setTitle("Book Detail");
        dialog.setCancelable(true);

        TextView txt_isbn = (TextView) dialog.findViewById(R.id.textBookISBN);
        TextView txt_title = (TextView) dialog.findViewById(R.id.textBookTitle);
        TextView txt_price = (TextView) dialog.findViewById(R.id.textBookPrice);
        TextView txt_publisher = (TextView) dialog
            .findViewById(R.id.textBookPublisher);

        txt_isbn.setText("ISBN :\t" + book.getIsbn());
        txt_title.setText("Title :\t" + book.getTitle());
        txt_price.setText("Price :\t" + book.getPrice() + "");
        txt_publisher.setText("Publisher :\t" + book.getPublisher());
    }
}
```

```

//edit book
Button button_edit = (Button) dialog.findViewById(R.id.buttonEdit);
button_edit.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        if (getListAdapter().getCount() > 0) {
            editBook(book);
            dialog.dismiss();
        }
    }
});
//delete book
Button button_delete = (Button) dialog.findViewById(R.id.buttonDelete);
button_delete.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        if (getListAdapter().getCount() > 0) {
            datasource.deleteBook(book); // delete book
            adapter.remove(book);
            dialog.dismiss();
            Toast.makeText(SQLiteDatabaseActivity.this, "Delete data succeed.", Toast.LENGTH_LONG).show();
        }
    }
});
// close dialog
Button button_cancel = (Button) dialog.findViewById(R.id.buttonClose);
button_cancel.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        dialog.dismiss();
    }
});

dialog.show();
}

public void editBook(final Book book) { ←—————③
    final Dialog dialog = new Dialog(SQLiteDatabaseActivity.this);
    dialog.setContentView(R.layout.add_book);
    dialog.setTitle("Edit Book");
    dialog.setCancelable(true);

    final EditText isbn = (EditText) dialog.findViewById(R.id.editTextISBN);
    final EditText title = (EditText) dialog
        .findViewById(R.id.editTextTitle);
    final EditText price = (EditText) dialog
        .findViewById(R.id.editTextPrice);
    final EditText publisher = (EditText) dialog
        .findViewById(R.id.editTextPublisher);

    isbn.setText(book.getIsbn());
    title.setText(book.getTitle());
    price.setText(String.valueOf(book.getPrice()));
    publisher.setText(book.getPublisher());
    // setup button
    Button button_save = (Button) dialog.findViewById(R.id.buttonSave);
    button_save.setText("Update");
    button_save.setOnClickListener(new OnClickListener() {

```



```
public void onClick(View v) {
    // get string
    String value_isbn = isbn.getText().toString();
    String value_title = title.getText().toString();
    double value_price = Double.parseDouble(price.getText()
        .toString());
    String value_publisher = publisher.getText().toString();
    book.setIsbn(value_isbn);
    book.setTitle(value_title);
    book.setPrice(value_price);
    book.setPublisher(value_publisher);
    datasource.updateBook(book);
    showAllBook();
    dialog.cancel();
}
});
Button button_cancel = (Button) dialog.findViewById(R.id.buttonCancel);
button_cancel.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        dialog.cancel();
    }
});
dialog.show();
}

public void addNewBook() { ← 1
    final Dialog dialog = new Dialog(SQLiteDatabaseActivity.this);
    dialog.setContentView(R.layout.add_book);
    dialog.setTitle("Add New Book");
    dialog.setCancelable(true);
    // setup button
    Button button_save = (Button) dialog.findViewById(R.id.buttonSave);
    button_save.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            // get string
            EditText isbn = (EditText) dialog
                .findViewById(R.id.editTextISBN);
            EditText title = (EditText) dialog
                .findViewById(R.id.editTextTitle);
            EditText price = (EditText) dialog
                .findViewById(R.id.editTextPrice);
            EditText publisher = (EditText) dialog
                .findViewById(R.id.editTextPublisher);

            String value_isbn = isbn.getText().toString();
            String value_title = title.getText().toString();
            double value_price = Double.parseDouble(price.getText()
                .toString());
            String value_publisher = publisher.getText().toString();
            @SuppressWarnings("unchecked")
            ArrayAdapter<Book> adapter = (ArrayAdapter<Book>) getListAdapter();
            Book book = new Book();
            book.setIsbn(value_isbn);
            book.setTitle(value_title);
            book.setPrice(value_price);
            book.setPublisher(value_publisher);
```

```

        book = datasource.insertBook(book);
        adapter.add(book);
        dialog.cancel();
    }
});
Button button_cancel = (Button) dialog.findViewById(R.id.buttonCancel);
button_cancel.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        dialog.cancel();
    }
});
dialog.show();
}

@Override
protected void onResume() {
    datasource.open();
    super.onResume();
}

@Override
protected void onPause() {
    datasource.close();
    super.onPause();
}
}
}

```

### ອອីបាយໂគ្គការកំណត់នូវការងាររបស់ខ្លួន

1. គឺជាគេតាដែលត្រូវបានបង្កើតឡើងដើម្បីអាចបញ្ជីតាមលក្ខណៈដែលបានបង្កើតឡើង។ Add New នៃព័ត៌មាននេះត្រូវបានបង្កើតឡើងដោយប្រើប្រាស់ ISBN, Title, Price និង Publisher

```

public void addNewBook() {
    final Dialog dialog = new Dialog(SQLiteDatabaseActivity.this);
    dialog setContentView(R.layout.add_book);
    dialog.setTitle("Add New Book");
    dialog.setCancelable(true);
    // setup button
    Button button_save = (Button) dialog.findViewById(R.id.buttonSave);
    button_save.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            // get string
            EditText isbn = (EditText) dialog
                .findViewById(R.id.editTextISBN);
            EditText title = (EditText) dialog
                .findViewById(R.id.editTextTitle);
            EditText price = (EditText) dialog
                .findViewById(R.id.editTextPrice);
            EditText publisher = (EditText) dialog
                .findViewById(R.id.editTextPublisher);

            String value_isbn = isbn.getText().toString();
            String value_title = title.getText().toString();
            double value_price = Double.parseDouble(price.getText()
                .toString());
        }
    });
}
}

```

```

        String value_publisher = publisher.getText().toString();
        @SuppressWarnings("unchecked")
        ArrayAdapter<Book> adapter = (ArrayAdapter<Book>) getListAdapter();
        Book book = new Book();
        book.setIsbn(value_isbn);
        book.setTitle(value_title);
        book.setPrice(value_price);
        book.setPublisher(value_publisher);
        book = datasource.insertBook(book);
        adapter.add(book);
        dialog.cancel();
    }
});
Button button_cancel = (Button) dialog.findViewById(R.id.buttonCancel);
button_cancel.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        dialog.cancel();
    }
});
dialog.show();
}

```

2. โค้ดสำหรับแสดงรายละเอียดของข้อมูลในแต่ละรายการเมื่อคลิกที่ไอเท็ม (Item) นั้นๆ และภายในจะประกอบไปด้วยปุ่ม 3 ปุ่มคือ Edit, Delete และ Cancel ซึ่งเมื่อคลิกปุ่ม Edit ก็จะแสดง dialogue ล็อกบีกซึ่งสำหรับแก้ไขข้อมูล และเมื่อคลิกปุ่ม Delete ก็จะลบข้อมูลรายการนั้น

```

@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
    // TODO Auto-generated method stub
    super.onListItemClick(l, v, position, id);
    showBookDetail(position); // View book by id
}

public void showBookDetail(final int id) {
    @SuppressWarnings("unchecked")
    final ArrayAdapter<Book> adapter = (ArrayAdapter<Book>) getListAdapter();
    final Book book = (Book) getListAdapter().getItem(id);

    final Dialog dialog = new Dialog(SQLiteDatabaseActivity.this);
    dialog.setContentView(R.layout.detail_book);
    dialog.setTitle("Book Detail");
    dialog.setCancelable(true);

    TextView txt_isbn = (TextView) dialog.findViewById(R.id.textBookISBN);
    TextView txt_title = (TextView) dialog.findViewById(R.id.textBookTitle);
    TextView txt_price = (TextView) dialog.findViewById(R.id.textBookPrice);
    TextView txt_publisher = (TextView) dialog
        .findViewById(R.id.textBookPublisher);

    txt_isbn.setText("ISBN :\t" + book.getIsbn());
    txt_title.setText("Title :\t" + book.getTitle());
    txt_price.setText("Price :\t" + book.getPrice() + "");
    txt_publisher.setText("Publisher :\t" + book.getPublisher());
}

```

```

//edit book
Button button_edit = (Button) dialog.findViewById(R.id.buttonEdit);
button_edit.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        if (getListAdapter().getCount() > 0) {
            editBook(book);
            dialog.dismiss();
        }
    }
});
//delete book
Button button_delete = (Button) dialog.findViewById(R.id.buttonDelete);
button_delete.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        if (getListAdapter().getCount() > 0) {
            datasource.deleteBook(book); // delete book
            adapter.remove(book);
            dialog.dismiss();
            Toast.makeText(SQLiteDatabaseActivity.this, "Delete data succeed.", Toast.LENGTH_LONG).show();
        }
    }
});
// close dialog
Button button_cancel = (Button) dialog.findViewById(R.id.buttonClose);
button_cancel.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        dialog.dismiss();
    }
});

dialog.show();
}

```

3. โค้ดสำหรับปรับปรุงข้อมูลเมื่อคลิกปุ่ม Update ในกรณีที่ต้องการแก้ไขข้อมูล

```

public void editBook(final Book book) {
    final Dialog dialog = new Dialog(SQLiteDatabaseActivity.this);
    dialog.setContentView(R.layout.add_book);
    dialog.setTitle("Edit Book");
    dialog.setCancelable(true);

    final EditText isbn = (EditText) dialog.findViewById(R.id.editTextISBN);
    final EditText title = (EditText) dialog
        .findViewById(R.id.editTextTitle);
    final EditText price = (EditText) dialog
        .findViewById(R.id.editTextPrice);
    final EditText publisher = (EditText) dialog
        .findViewById(R.id.editTextPublisher);

    isbn.setText(book.getIsbn());
    title.setText(book.getTitle());
    price.setText(String.valueOf(book.getPrice()));
    publisher.setText(book.getPublisher());
    // setup button
    Button button_save = (Button) dialog.findViewById(R.id.buttonSave);

```

```

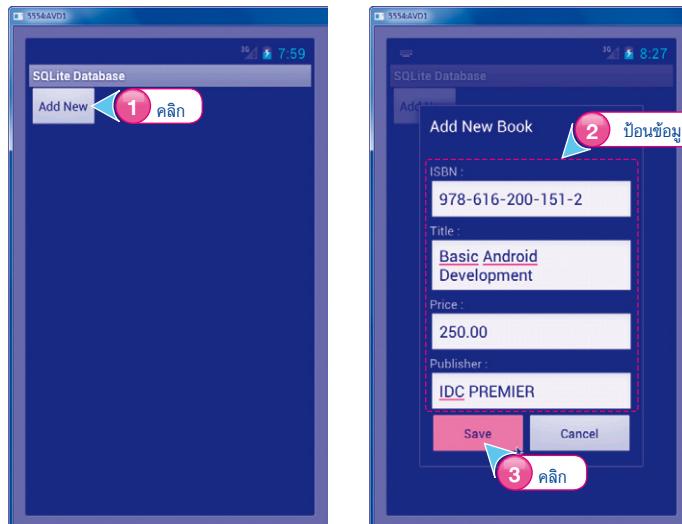
button_save.setText("Update");
button_save.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // get string
        String value_isbn = isbn.getText().toString();
        String value_title = title.getText().toString();
        double value_price = Double.parseDouble(price.getText()
            .toString());
        String value_publisher = publisher.getText().toString();
        book.setIsbn(value_isbn);
        book.setTitle(value_title);
        book.setPrice(value_price);
        book.setPublisher(value_publisher);
        datasource.updateBook(book);
        showAllBook();
        dialog.cancel();
    }
});
Button button_cancel = (Button) dialog.findViewById(R.id.buttonCancel);
button_cancel.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        dialog.cancel();
    }
});
dialog.show();
}
}

```

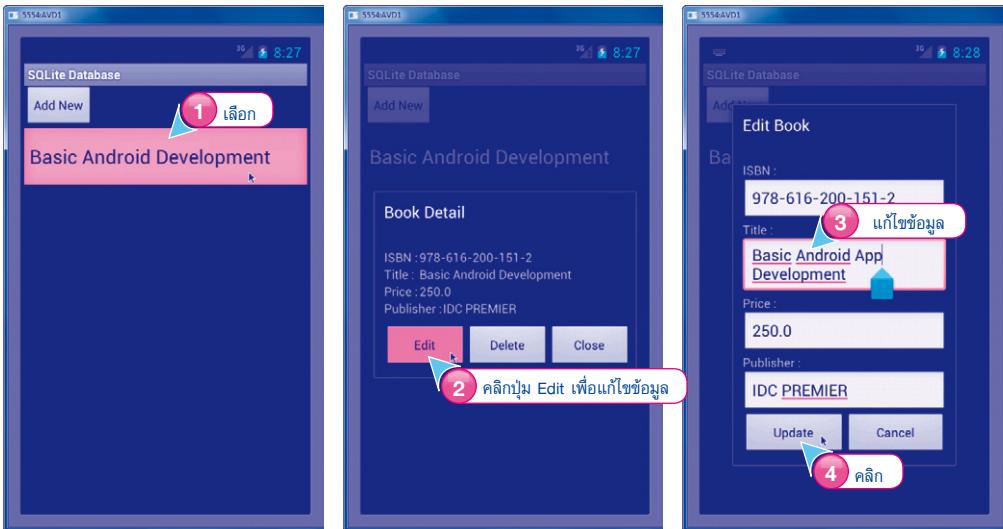
## ทดลองการทำงานของแอพพลิเคชัน

ทุกครั้งเมื่อพัฒนาแอพพลิเคชันเสร็จสมบูรณ์แล้ว ผู้อ่านควรทดสอบการทำงานของแอพพลิเคชันทุกครั้ง ซึ่งตัวอย่างนี้มีรายละเอียดดังนี้

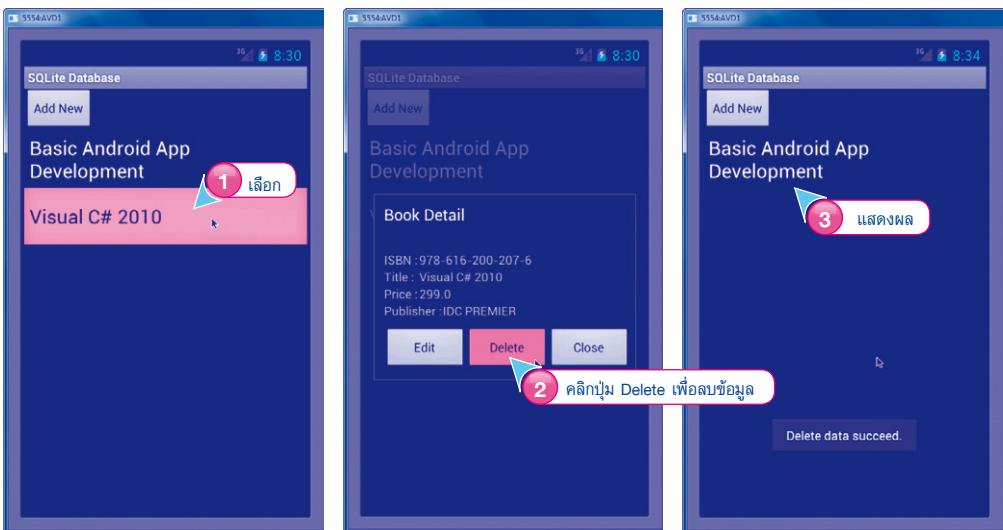
- ทดลองการเพิ่มข้อมูล



● ทดสอบการแก้ไขข้อมูล



● ทดสอบการลบข้อมูล



## สรุปภ้ายบท

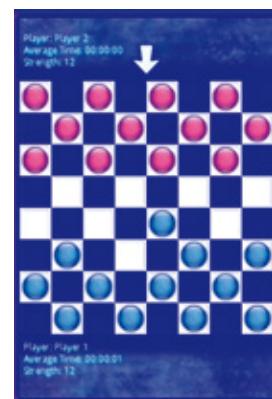
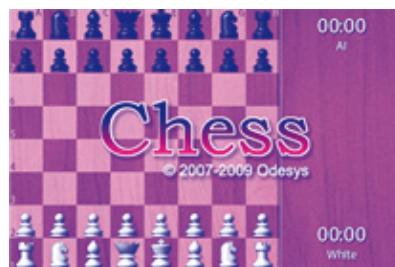
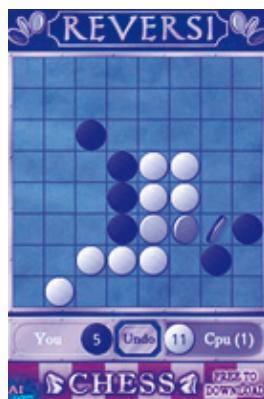
ในบทนี้เป็นการอธิบายถึงการพัฒนาแอพพลิเคชัน เพื่อประยุกต์ใช้งานฐานข้อมูล SQLite อย่างง่ายๆ โดยใช้ตัวอย่างของการเก็บข้อมูลรายชื่อหนังสือลงในฐานข้อมูล และสามารถค้นหาหนังสือ เพิ่ม แก้ไข หรือลบรายการหนังสือนั้นๆ ได้

## แบบฝึกหัดกা�ヤบต

หลังจากผู้อ่านได้พัฒนาแอพพลิเคชันติดต่อ กับฐานข้อมูลเบื้องต้นแล้ว แบบฝึกหัดในส่วนนี้ให้ผู้อ่านสร้างฐานข้อมูลและพัฒนาแอพพลิเคชันให้สามารถเพิ่มข้อมูล ลบข้อมูล แก้ไขข้อมูล และเรียงลำดับข้อมูลได้ จากนั้นให้ผู้อ่านพัฒนาแอพพลิเคชันเรียกใช้งานผ่านเครือข่ายอินเทอร์เน็ต หรือเครือข่ายโทรศัพท์เคลื่อนที่เพื่อติดต่อ กับฐานข้อมูลนั้นๆ ทั้งแบบโคลอ่อนท์/เซิร์ฟเวอร์ (Client/Server) และแบบ Web Service Programming บนระบบปฏิบัติการ Android

## การพัฒนาเกมแบบ 2 มิติ

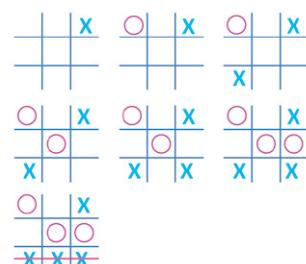
เนื้อหาในบทนี้จะเป็นส่วนของการฝึกพัฒนาแอพพลิเคชันที่เกี่ยวข้องกับการใช้งานกราฟิก (Graphic) แบบ 2 มิติ โดยผู้เขียนจะใช้การพัฒนาเกมอย่างง่ายมาเป็นตัวอย่างในการอธิบาย ซึ่งถือได้ว่าเกมเป็นแอพพลิเคชันที่ได้รับความนิยมสูงสุดในประเภทของแอพพลิเคชันทั้งหมด



▲ ตัวอย่างเกม 2 มิติ

### วิธีการเล่นเกม

ตัวอย่างที่นำมาใช้อธิบายในบทนี้ จะเป็นการพัฒนาเกม Tic-Tac-Toe หรือเกม OX หรือบางท่านอาจจะเรียกว่า เกม XO ซึ่งเกม OX นั้นเป็นเกมกระดานชนิดหนึ่ง ซึ่งจะเล่นอยู่บนตารางขนาด  $3 \times 3$  โดยที่ผู้เล่นถูกแบ่งออกเป็น 2 ฝ่าย แต่ละฝ่ายจะผลัดกันเขียนรูปวงกลม หรือเครื่องหมาย O และกาหนาท หรือเครื่องหมาย X ในตาราง ซึ่งมีรูปแบบการเล่นดังรูป



สำหรับการออกแบบตารางนั้นมีขนาดกว้าง 3 ช่อง ยาว 3 ช่อง และจะไม่มีการกำหนดขนาดของช่องในแต่ละช่อง อย่างไรก็ตามทุกช่องจะต้องมีขนาดเท่ากันและมีขนาดเพียงพอสำหรับการวางเครื่องหมาย O และเครื่องหมาย X ลงไปได้พอดี

เพื่อให้ผู้อ่านทุกท่านเข้าใจการทำงานของเกม ผู้เขียนจะขออธิบายวิธีการเล่นเพื่อให้เข้าใจตรงกันดังนี้

1. ผู้เล่นทั้ง 2 ฝ่ายกำหนดเครื่องหมายที่ใช้ในการเล่นที่แตกต่างกัน
2. กำหนดดาวฝ่ายใดจะเป็นฝ่ายเริ่มเล่นก่อน
3. ฝ่ายที่เริ่มเล่นก่อนจะเขียนเครื่องหมาย O (หรือ X ก็ได้ตามที่กำหนดไว้)
4. อีกฝ่ายหนึ่งจะเขียนเครื่องหมายลงในช่องที่เหลือ ทั้งนี้ผู้เล่นไม่สามารถเขียนเครื่องหมายของตัวเองซ้ำซึ่งกันไม่ได้
5. ทั้ง 2 ฝ่ายก็จะผลัดกันเขียนเครื่องหมายของตัวเองจนเต็มกระดาน

สำหรับการตัดสินผู้ชนะคือ ฝ่ายหนึ่งฝ่ายใดจะชนะเกมได้ ก็ต่อเมื่อผู้เล่นฝ่ายหนึ่งสามารถเขียนเครื่องหมายของตัวเองเรียงเป็นแนวตั้ง แนวนอน หรือแนวทแยงติดต่อกันเป็นจำนวน 3 เครื่องหมายได้สำเร็จก่อน ในกรณีที่มีเครื่องหมายเต็มตารางแล้ว แต่ไม่มีฝ่ายใดสามารถที่จะวางเครื่องหมายให้เรียงไปตามแนวตั้ง แนวนอน หรือแนวทแยงได้ ก็จะถือได้ว่าทั้ง 2 ฝ่ายเสมอ กัน และเริ่มเกมใหม่

นอกจากเกม XO แล้วยังมีการพัฒนาเกมรูปแบบต่างๆ ซึ่งเป็นส่วนขยายจากเทคนิคที่ใช้ในการเล่นเกมนี้ เช่น เกม Link 5, 3-dimensional Tic-Tac-Toe และ Tic-Tac-Chess เป็นต้น ทั้งนี้ผู้อ่านที่สนใจสามารถศึกษาเพิ่มเติมໄ่จากเว็บไซต์ แต่เนื้อหาในบทนี้จะมุ่งเน้นไปยังการพัฒนาเกม XO เมื่อต้นเท่านั้น โดยเนื้อหาในบทนี้จะถูกแบ่งออกเป็น 5 ขั้นตอนคือ

1. การสร้างคลาสหลักในการพัฒนาเกม XO
2. สร้างหน้าจอเมนูของเกม
3. การสร้างกราฟิก (Graphic) เมื่อต้น
4. การจัดการหน้าจอสัมผัส
5. การปรับแต่งการแสดงผลเมื่อจบเกม

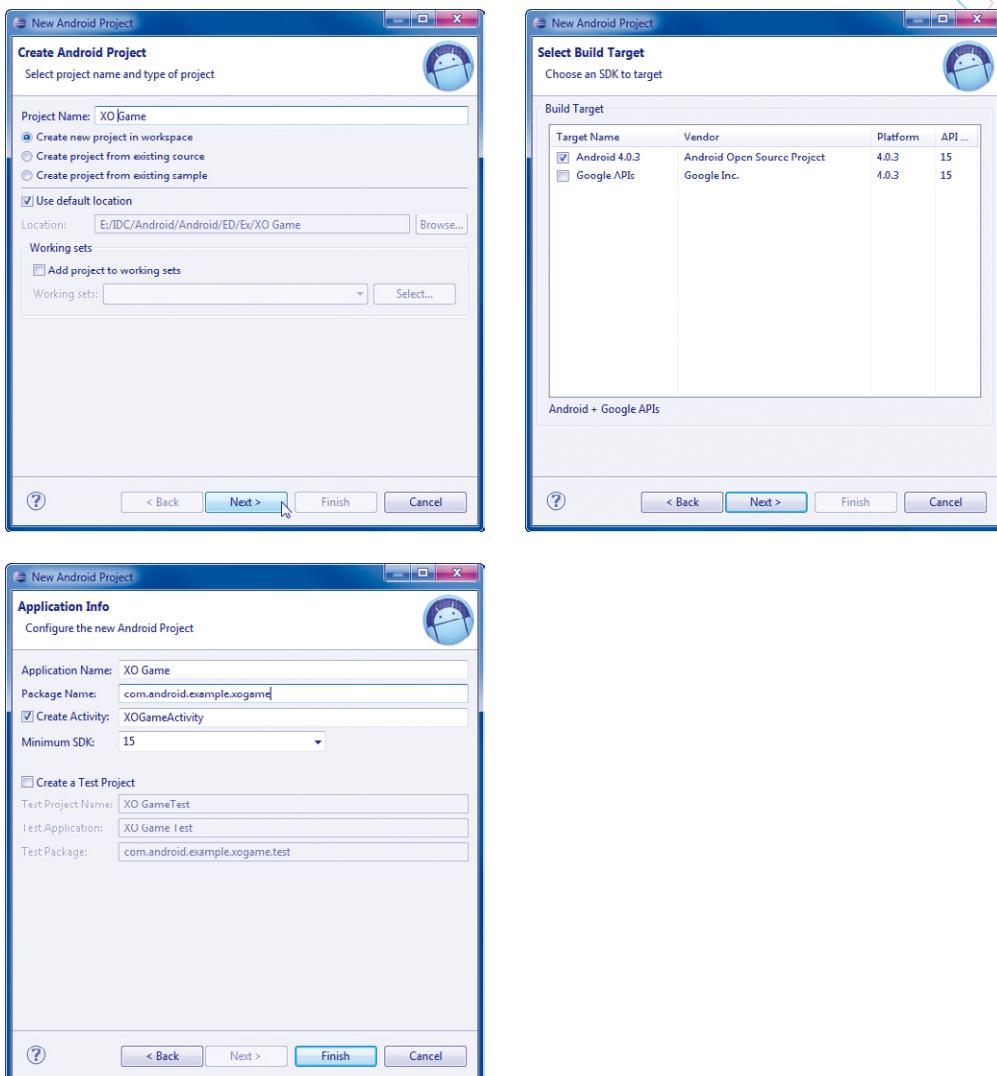
## การสร้างคลาสหลักในการพัฒนาเกม XO

ในส่วนนี้จะเป็นการสร้างคลาสหลักเพื่อใช้ในการพัฒนาเกม สำหรับใช้ควบคุมการทำเครื่องหมายของผู้เล่น และคำนวณการทำเครื่องหมายในกรณีที่เล่นคนเดียว ซึ่งมีขั้นตอนดังนี้

1. เขียนโค้ดกำหนดการวาดเครื่องหมายในไฟล์ SimpleXO.java
2. เขียนโค้ดคำนวณการทำเครื่องหมายในไฟล์ SimpleXOAI.java

ชี้งสามารถอธิบายรายละเอียดเป็นขั้นตอนอย่างๆ ได้ดังนี้

1. สร้างโปรเจ็คโดยกำหนดคุณสมบัติต่างๆ ดังนี้



2. สร้างไฟล์ SimpleXO.java และเขียนโค้ดกำหนดการวาดเครื่องหมาย O หรือเครื่องหมาย X ดังนี้

```
package com.android.example.xogame;

public class SimpleXO {
    private int[][] board;
    // get value for draw line when finish game
    private int win_row = -1;
    private int win_col = -1;
    private int win_dia = -1;
```



```
public SimpleXO(int size) {
    board = new int[size][size];
    clearBoard();
}

public SimpleXO(SimpleXO copy) {
    int row, column;
    int size = copy.getSize();

    board = new int[size][size];
    for (row = 0; row < size; row++)
        for (column = 0; column < size; column++)
            board[row][column] = copy.board[row][column];
}

public void clearBoard() { ← เมธอดตรวจสอบช่องว่างในตาราง
    int row, column;
    for (row = 0; row < getSize(); row++)
        for (column = 0; column < getSize(); column++) {
            board[row][column] = 0;
        }
}

public int getSize() { ← เมธอดอ่านค่าขนาดตาราง
    return board.length;
}

public int getLocation(int row, int column) { ← เมธอดอ่านค่า O หรือ X
    return board[row][column];
}

public boolean isSpaceLeft() {
    int row, column;
    for (row = 0; row < getSize(); row++)
        for (column = 0; column < getSize(); column++) {
            if (board[row][column] == 0)
                return true;
        }
    return false;
}

public boolean makeMove(int row, int column, int player) {
    if ((player < 1) || (player > 2))
        return false;
    if ((row < 0) || (row >= getSize()))
        return false;
    if ((column < 0) || (column >= getSize()))
        return false;
    if (board[row][column] != 0)
        return false;
    board[row][column] = player;
    return true;
}
```

```

// Return 0 == none, 1 == X, 2 == O and set number of row,column
public int getWinner() { ←———— เมื่อต้องการสอบถามผู้ชนะ

    int i, row;
    int winner;
    for (i = 0; i < getSize(); i++) {
        winner = getWinnerRow(i);
        if (winner != 0) {
            win_row = i;
            return winner;
        }
        winner = getWinnerColumn(i);
        if (winner != 0) {
            win_col = i;
            return winner;
        }
    }

    // diagonals top-left to right-bottom
    winner = board[0][0];
    for (i = 1; i < getSize(); i++) {
        if (board[i][i] != winner) {
            winner = 0;
            break;
        }
    }
    if (winner != 0) {
        win_dia = 1;
        return winner;
    }

    row = getSize() - 1;
    winner = board[row][0];
    for (i = 1; i < getSize(); i++) {
        if (board[row - i][i] != winner) {
            winner = 0;
            break;
        }
    }
    if (winner != 0) {
        win_dia = 2;
        return winner;
    }

    return 0;
}

public int[][] getBoard() {
    return board;
}

public void setBoard(int[][] board) {
    this.board = board;
}

public int getWin_row() {
    return win_row;
}

```

```
public void setWin_row(int win_row) {
    this.win_row = win_row;
}

public int getWin_col() {
    return win_col;
}

public void setWin_col(int win_col) {
    this.win_col = win_col;
}

public int getWin_dia() {
    return win_dia;
}

public void setWin_dia(int win_dia) {
    this.win_dia = win_dia;
}

private int getWinnerRow(int row) {
    int column;
    for (column = 1; column < getSize(); column++) {
        if (board[row][column] != board[row][0]) {
            return 0;
        }
    }
    return board[row][0];
}

private int getWinnerColumn(int column) {
    int row;
    for (row = 1; row < getSize(); row++) {
        if (board[row][column] != board[0][column]) {
            return 0;
        }
    }
    return board[0][column];
}

public static void getComputerMove(SimpleXO game, int player) {
    SimpleXOAI.calculateMove(game, player);
}
}
```

ในการสร้างคลาส SimpleXO.java จะเกิดข้อผิดพลาดในบรรทัด SimpleXOAI.calculateMove(game, player) เนื่องจากผู้อ่านยังไม่ได้สร้างคลาส SimpleXOAI โดยคลาส SimpleXO จะถูกสร้างขึ้นมาเพื่อควบคุมการทำเครื่องหมายของผู้เล่น รวมไปถึงการตรวจสอบผู้ชนะ ซึ่งจะถูกเรียกใช้ในการแสดงผล

### 3. สร้างไฟล์ SimpleXOAI.java และเขียนโค้ดคำนวณการทำเครื่องหมายดังนี้

```
package com.android.example.xogame;

public class SimpleXOAI {
    public static int calculateMove(SimpleXO game, int player) {
        if (game.isSpaceLeft() == false)
            return 0;

        int row, column, score, winner;
        int bestRow, bestColumn, bestScore, bestWinner;
        int otherPlayer = (player == 1) ? 2 : 1;
        bestRow = -2;
        bestColumn = -2;
        bestScore = -2;
        bestWinner = -2;

        for (row = 0; row < game.getSize(); row++) {
            for (column = 0; column < game.getSize(); column++) {
                if (game.getLocation(row, column) == 0) {

                    SimpleXO copy = new SimpleXO(game);
                    copy.makeMove(row, column, player);
                    if (copy.getWinner() == player) {

                        score = 1;
                        winner = player;
                        game.makeMove(row, column, player);
                        return winner;
                    }

                    winner = calculateMove(copy, otherPlayer);
                    if (winner == player) {
                        score = 1;
                    } else if (winner == 0) {
                        score = 0;
                    } else {
                        score = -1;
                    }

                    if (score > bestScore) {
                        bestRow = row;
                        bestColumn = column;
                        bestWinner = winner;
                        bestScore = score;
                    }
                }
            }
        }

        game.makeMove(bestRow, bestColumn, player);
        return bestWinner;
    }
}
```

คลาส SimpleXOAI ถูกสร้างเพื่อคำนวณการทำเครื่องหมายของผู้เล่นในกรณีที่ผู้เล่นเลือกเล่นแบบ คนเดียว

# การสร้างหน้าจอส่วนต่างๆ ของเกม

ในส่วนนี้จะเป็นการสร้างหน้าจอส่วนต่างๆ ของเกม ซึ่งแบ่งออกเป็น 5 ส่วนดังนี้

1. การสร้างหน้าจอมenuสำหรับเลือกรูปแบบการเล่น
2. การสร้างหน้าจอลำารับป้อมชื่อผู้เล่น
3. การสร้างหน้าจอลำารับเล่นเกม
4. การสร้างหน้าจอมีอุปกรณ์
5. การแก้ไขไฟล์ AndroidManifest.xml

## การสร้างหน้าจอมenuสำหรับเลือกรูปแบบการเล่น

ในส่วนนี้จะเป็นการสร้างหน้าจอลำารับเลือกรูปแบบการเล่น โดยจะแบ่งออกเป็น 2 แบบคือ

- แบบเล่นคนเดียว
- แบบเล่น 2 คน

ซึ่งมีขั้นตอนดังต่อไปนี้

1. สร้างไฟล์ home.xml ในส่วนของ res/layout และแก้ไขหน้าจอดังโค้ด XML ดังต่อไปนี้

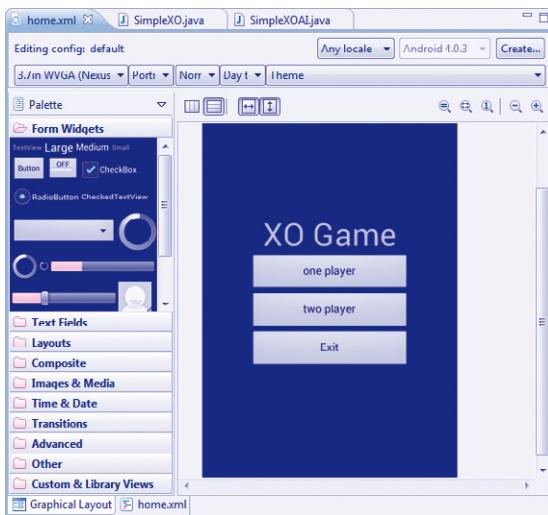
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="XO Game"
        android:textSize="40sp" />

    <Button <span style="color: red;">123

```

จะได้หน้าจอแอพพลิเคชันดังนี้



## 2. สร้างไฟล์ GameMenuActivity.java และเขียนโค้ด Java ดังนี้

```
package com.android.example.xogame;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class GameMenuActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.home);

        Button one_player = (Button) findViewById(R.id.button_one_player);
        one_player.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),
                    GamePlayerActivity.class);
                intent.putExtra("PLAYER", 1);
                startActivity(intent);
            }
        });
        Button two_player = (Button) findViewById(R.id.button_two_player);
        two_player.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),
                    GamePlayerActivity.class);
            }
        });
    }
}
```

```
        intent.putExtra("PLAYER", 2);
        startActivity(intent);
    }
});

Button bt_exit = (Button) findViewById(R.id.button_exit);
bt_exit.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        final AlertDialog alertDialog = new AlertDialog.Builder(
            GameMenuActivity.this).create();
        alertDialog.setMessage("Are you sure you want to exit?");
        alertDialog.setButton("Yes",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                    int which) {
                    finish();
                }
            });
        alertDialog.setButton2("Cancel",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                    int which) {
                    alertDialog.cancel();
                }
            });
        alertDialog.setIcon(R.drawable.ic_launcher);
        alertDialog.show();
    }
});
}
```

หลังจากที่ผู้อ่านเขียนโน้ตเดลเรียบร้อย โน้ตในบรรทัดที่มีการใช้งานคลาส GamePlayerActivity จะยังมี Error ออย เนื่องจากผู้อ่านยังไม่ได้สร้างคลาสดังกล่าว ซึ่งผู้เขียนจะกล่าวถึงในหัวข้อดีไป

## อริบายโค้ดการทำงานของเกม

จากโค้ดด้านบนมีรายละเอียดที่น่าสนใจดังนี้

- การส่งข้อมูลระหว่างแอ็คทิวิตี้ ผู้อ่านจะลังเลก่อนว่ามีการเขียนโค้ดด้านบนเพื่อให้เปิดหน้าจอของคลาส GamePlayerActivity ในรูป one\_player และ two\_player และมีการส่งข้อมูลชื่อ (name) เป็น “PLAYER” 1 และ 2 ซึ่งจะใช้ค่านี้ในการระบุว่าผู้เล่นเลือกเล่นแบบคนเดียว หรือ 2 คนนั่นเอง

```
Intent intent = new Intent(getApplicationContext(),GamePlayerActivity.class);
        intent.putExtra("PLAYER", 1);
        startActivity(intent);
```

2. การสร้าง Dialog ยืนยันการออกจากเกม เมื่อคลิกปุ่ม Exit จะมีการแสดงคำว่า “Are you sure you want to exit?” หากคลิกปุ่ม Yes ก็จะเรียกใช้เมธอด finish() เพื่อปิดเกม

```

final AlertDialog alertDialog = new AlertDialog.Builder(
        GameMenuActivity.this).create();
alertDialog.setMessage("Are you sure you want to exit?");
alertDialog.setButton("Yes",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
                    int which) {
                finish();
            }
        });
alertDialog.setButton2("Cancel",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
                    int which) {
                alertDialog.cancel();
            }
        });
alertDialog.setIcon(R.drawable.ic_launcher);
alertDialog.show();

```

## การสร้างหน้าจอสำหรับป้อนชื่อผู้เล่น

ในส่วนนี้จะเป็นการออกแบบหน้าจอสำหรับให้ผู้เล่นป้อนชื่อผู้เล่นคนแรกและคนที่สองซึ่งมีรายละเอียดดังนี้

### 1. สร้างไฟล์ player.xml ในส่วนของ res/layout และเขียนโค้ด XML ดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="10dip" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="player 1"
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <EditText
            android:id="@+id/editText_player1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <requestFocus />
        </EditText>
    
```

กำหนดคุณสมบัติของ Player 1

```

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Player 2"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
    android:id="@+id/editText_player2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

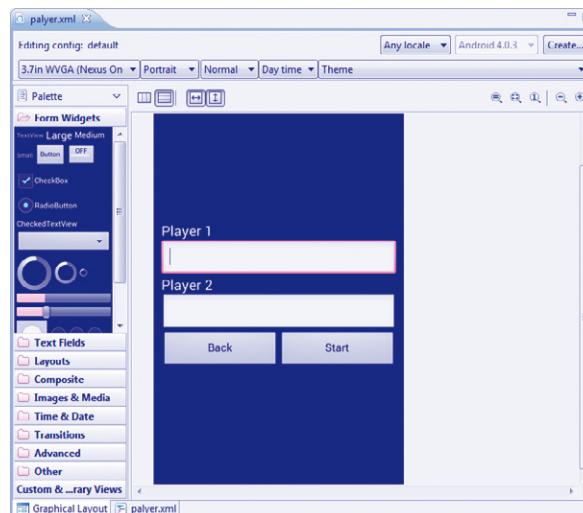
<LinearLayout ← 1 กำหนดคุณสมบัติของ Layout
    android:id="@+id/linearLayout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <Button ← 2 กำหนดคุณสมบัติของปุ่ม Back
        android:id="@+id/button_back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Back" />

    <Button ← 3 กำหนดคุณสมบัติของปุ่ม Start
        android:id="@+id/button_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Start" />
</LinearLayout>
</LinearLayout>

```

จะได้หน้าจอแอพพลิเคชันดังนี้



## 2. สร้างไฟล์ GamePlayerActivity.java และเขียนโค้ด Java ดังนี้

```
package com.android.example.xogame;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class GamePlayerActivity extends Activity {
    int player = 0;
    EditText player1;
    EditText player2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);

        setContentView(R.layout.palyer);

        Bundle extras = getIntent().getExtras();
        if(extras !=null) {
            player = extras.getInt("PLAYER");
        }

        player1 = (EditText) findViewById(R.id.editText_player1);
        player2 = (EditText) findViewById(R.id.editText_player2);

        if(player==1){
            player2.setText("android");
        }

        Button bt_back = (Button) findViewById(R.id.button_back);
        Button bt_start = (Button) findViewById(R.id.button_start);
        bt_back.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                finish();
            }
        });
        bt_start.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),XOGameActivity.class);
                intent.putExtra("PLAYER", player);
                intent.putExtra("PLAYER_ONE_NAME", player1.getText().toString());
                intent.putExtra("PLAYER_TWO_NAME", player2.getText().toString());
                startActivity(intent);
            }
        });
    }
}
```

ผู้อ่านจะลังเลตัวโค้ดบรรทัด XOGameActivity ซึ่งเป็นหน้าจอหลักสำหรับเล่นเกมจะยังมี Error อีก  
โดยผู้เขียนจะกล่าวถึงขั้นตอนการสร้างในหัวข้อดังไป

### อธิบายโค้ดการทำงานของเกม

ผู้อ่านจะลังเลตัวโค้ดเห็นว่ามีการเขียนโปรแกรมลักษณะคล้ายกับคลาส GameMenuActivity ซึ่งสร้าง  
ขึ้นในหัวข้อก่อนหน้านี้ ซึ่งส่วนที่แตกต่างกันคือ มีการสร้างการเชื่อมโยงไปยังคลาส XOGameActivity เมื่อ  
คลิกปุ่ม bt\_start โดยจะมีการส่งข้อมูล 3 ค่า ได้แก่

- "PLAYER" เก็บรูปแบบการเล่น
- "PLAYER\_ONE\_NAME" เก็บชื่อผู้เล่นคนแรก
- "PLAYER\_TWO\_NAME" เก็บชื่อผู้เล่นคนที่สอง

```
Intent intent = new Intent(getApplicationContext(), XOGameActivity.class)
intent.putExtra("PLAYER", player);
intent.putExtra("PLAYER_ONE_NAME", player1.getText().toString());
intent.putExtra("PLAYER_TWO_NAME", player2.getText().toString());
startActivity(intent);
```

### การสร้างหน้าจอสำหรับเล่นเกม

ในหัวข้อนี้ผู้เขียนจะอธิบายการสร้างหน้าจอสำหรับเล่นเกม โดยสร้างเป็นหน้าจอเบล่าๆ ไว้ก่อน  
ซึ่งรายละเอียดจะเขียนโค้ดเพิ่มเติมในภายหลัง

ในการสร้างหน้าจอสำหรับเล่นเกมนี้ จะไม่ใช้งานไฟล์ XML ในการออกแบบหน้าจอ แต่จะสร้างด้วย  
โค้ด Java แทน Graphic โดยแก้ไขโค้ดในไฟล์ XOGameActivity.java ดังนี้

```
package com.android.example.xogame;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Paint.Cap;
import android.graphics.Point;
import android.os.Bundle;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;

public class XOGameActivity extends Activity {
    String TAG = "OX Game";
    public Paint paint;
    public static float block_width = 0;
    public static int current_player = 1;
    public static boolean one_player = false;
    public static int count = 0;
    public static int winner = -1;
    public boolean pause_flag = false;
```

← เพิ่มคลาส

```

public static String[] player_name = { "Player 1", "Player 2" };
public static int[] player_score = { 0, 0 };

public static int line_padding = 20;

SimpleXO game = new SimpleXO(3);

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // get data from GamePlayerActivity
    Bundle extras = getIntent().getExtras();
    if (extras != null) {
        int porson = extras.getInt("PLAYER");
        if(porson==1){
            one_player = true;
        }else{
            one_player = false;
        }
        player_name[0] = extras.getString("PLAYER_ONE_NAME");
        player_name[1] = extras.getString("PLAYER_TWO_NAME");
    }
    setContentView(new Panel(this));
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    // new game
}

// start GameOverActivity when game finish.
public void gameOver(int win) {

}

class Panel extends View implements OnTouchListener {
    public Panel(Context context) {
        super(context);
        setFocusable(true);
        setFocusableInTouchMode(true);
        this.setOnTouchListener(this);
    }

    public void drawBoard(Canvas canvas) {
        // draw line when game finish
    }

    public void onDraw(Canvas canvas) {

    }

    public void drawAllElement(Canvas canvas) {

```

```

    }

    public void drawElementX(Canvas canvas, int i, int j) {
    }

    public void drawElementO(Canvas canvas, int i, int j) {
    }

    public void switchPlayer() {
        if (current_player == 1)
            current_player = 2;
        else
            current_player = 1;
    }

    public boolean onTouch(View view, MotionEvent event) {
        return true;
    }
}
}

```

จากโค้ดเป็นการสร้างตัวแปรที่ใช้สำหรับหน้าจอเล่นเกม โดยมีการสร้างคลาส Panel เพื่อใช้สร้างหน้าจอเองโดยไม่ต้องเรียกใช้งานไฟล์ XML

### อธิบายโค้ดการทำงานของเกม

ในเมธอด onCreate() จะมีการเขียนโค้ดเพื่อรับข้อมูลที่ส่งมาจากคลาส GamePlayerActivity โดยมีลักษณะตรงกันข้ามกับการส่งข้อมูล ในที่นี้จะส่งข้อมูลมาในแบบตัวเลขและข้อความ ซึ่งจะใช้เมธอด getInt() และ getString() สำหรับรับข้อมูลตามลำดับ

```

// get data from GamePlayerActivity
Bundle extras = getIntent().getExtras();
if (extras != null) {
    int person = extras.getInt("PLAYER");
    if(person==1){
        one_player = true;
    }else{
        one_player = false;
    }
    player_name[0] = extras.getString("PLAYER_ONE_NAME");
    player_name[1] = extras.getString("PLAYER_TWO_NAME");
}

```

### การสร้างหน้าจอเมื่อจบเกม

ในส่วนนี้เป็นการสร้างหน้าจอสำหรับแสดงผลเมื่อผู้เล่นเล่นจนจบเกม โดยแสดงข้อความว่าผู้เล่นคนใดเป็นผู้ชนะ และแสดงปุ่มสำหรับเริ่มเล่นเกมใหม ซึ่งมีรายละเอียดดังนี้

## 1. สร้างไฟล์ popup.xml และแก้ไขโค้ด XML ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:background="#aa000000"
    android:orientation="vertical" >

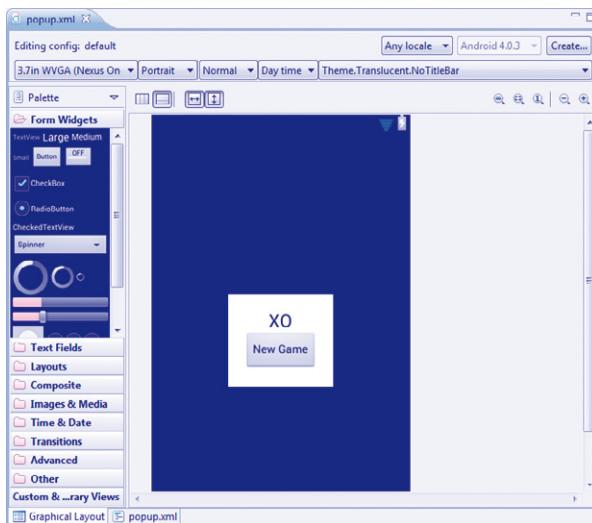
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#ffffffff" ← กำหนดสีดำ
        android:gravity="center"
        android:padding="20dip"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/text_winner"
            android:textColor="#000000" ← กำหนดสีขาว
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="XO"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <Button ← 1 กำหนดคุณสมบัติของปุ่ม New Game
            android:id="@+id/bt_newgame"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="New Game" />
    </LinearLayout>

</LinearLayout>
```

จะได้หน้าจอแอพพลิเคชันดังนี้



## 2. สร้างไฟล์ GameOverActivity.java และแก้ไขโค้ด Java ดังนี้

```
package com.android.example.xogame;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class GameOverActivity extends Activity {
    String msg ;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.popup);

        Bundle extras = getIntent().getExtras();
        if (extras != null) {
            msg = extras.getString("GAME_MSG");
        }
        TextView text_msg = (TextView) findViewById(R.id.text_winner);
        text_msg.setText(msg);
        Button bt_newgame = (Button) findViewById(R.id.bt_newgame);
        bt_newgame.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                setResult(RESULT_OK);
                finish();
            }
        });
    }
}
```

จากโค้ดเมื่อคลิกปุ่ม bt\_newgame จะมีการเรียกใช้งานเมธอด setResult() และกำหนดผลลัพธ์เป็น RESULT\_OK เพื่อลบส่วนของเกมที่ผ่านมา ซึ่งจะใช้ในการเริ่มเกมใหม่ หลังจากนั้นก็ปิดแอ็คทิวิตี้โดยเรียกใช้งานเมธอด finish()

### กำหนดรายละเอียดให้แอ็คทิวิตี้ต่างๆ

ในส่วนนี้จะเป็นการแก้ไขไฟล์ AndroidManifest.xml เพื่อลงทะเบียนแอ็คทิวิตี้ต่างๆ ที่สร้างขึ้นมาทั้งหมด ซึ่งได้แก่ คลาส GameMenuActivity, GameOverActivity, GamePlayerActivity และ XOGameActivity โดยแก้ไขโค้ด XML ในไฟล์ AndroidManifest.xml ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.example.xogame"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8" />
```

```

<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity android:name=".GameMenuActivity" >
        <intent-filter >
            <action android:name="android.intent.action.MAIN" />

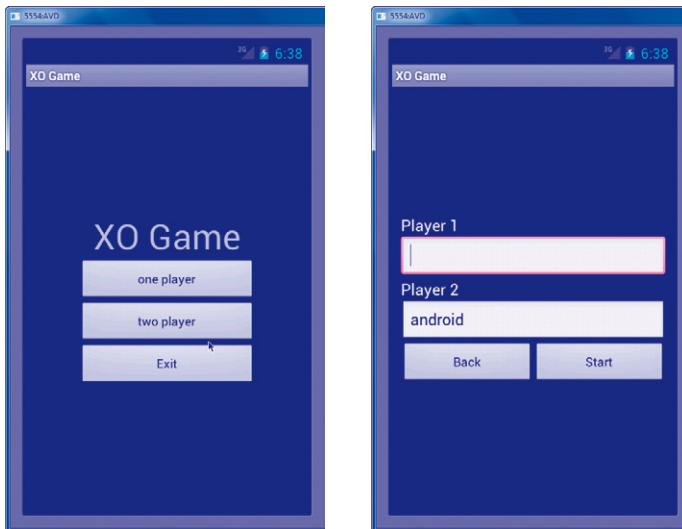
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".GamePlayerActivity" >
    </activity>
    <activity android:name=".XOGameActivity"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name=".GameOverActivity"
        android:theme="@android:style/Theme.Translucent.NoTitleBar" >
    </activity>
</application>

</manifest>

```

จากโค้ดจะมีการเพิ่มแอ็ตทริบิวต์ android:screenOrientation="portrait" เข้าไปเพื่อกำหนดหน้าจอของคลาส XOGameActivity ให้วางหน้าจอในแนวตั้งเพียงอย่างเดียวเท่านั้น

เมื่อผู้อ่านลองรันโปรแกรมเพื่อทดสอบการทำงานของเกม จะปรากฏหน้าจอของเกมดังรูป





# การสร้างกราฟิก (Graphic) เบื้องต้น

ในส่วนนี้จะเป็นการสร้างกราฟิกเบื้องต้น ซึ่งมีขั้นตอนโดยสรุปสำหรับการสร้างกราฟิกเพื่อร้องรับการใช้งานของเกม XO ดังนี้

ให้ผู้อ่านเขียนโค้ด Java เพิ่มเติมในส่วนของคลาส Panel ที่อยู่ในไฟล์ XOGameActivity.java ดังนี้

```
class Panel extends View implements OnTouchListener {  
    public Panel(Context context) {  
        super(context);  
        setFocusable(true);  
        setFocusableInTouchMode(true);  
        this.setOnTouchListener(this);  
    }  
  
    public void drawBoard(Canvas canvas) { ←———— วาดตาราง  
        int line_size = 5;  
        float w = this.getWidth();  
        float h = this.getHeight();  
        float s = w;  
        if (w > h)  
            s = h;  
  
        block_width = s;  
  
        int bw = (int) (s / 3);  
        int bh = (int) (s / 3);  
        paint = new Paint();  
        paint.setColor(Color.BLACK);  
        canvas.drawRect(0, 0, w, h, paint);  
        paint.setColor(Color.WHITE);  
        paint.setAntiAlias(true);  
        paint.setStrokeWidth(10);  
        paint.setStrokeCap(Cap.ROUND);  
        // draw board  
        canvas.drawLine(bw, line_size, bw, s, paint);  
        canvas.drawLine(2 * bw, line_size, 2 * bw, s, paint);  
        canvas.drawLine(line_size, bh, w - line_size, bh, paint);  
        canvas.drawLine(line_size, 2 * bh, w - line_size, 2 * bh, paint);  
  
        paint.setTextSize(40);  
        canvas.drawText(player_name[0] + " : " + player_score[0], 10,  
                        s + 60, paint);  
        canvas.drawText(player_name[1] + " : " + player_score[1], 10,  
                        s + 110, paint);  
    }  
  
    public void onDraw(Canvas canvas) {  
        drawBoard(canvas);  
        drawAllElement(canvas);  
        // draw line when game finish  
        if (game.getWinner() != 0) {  
    }
```

```

        int block = (int) (block_width / 3);
        int winrow = game.getWin_row();
        int wincol = game.getWin_col();
        int windia = game.getWin_dia();

        int x1 = 0, x2 = 0, y1 = 0, y2 = 0;

        if (winrow != -1) {
            x1 = block * winrow + (block / 2);
            x2 = block * winrow + (block / 2);
            y1 = line_padding;
            y2 = (int) block_width - line_padding * 2;
        } else if (wincol != -1) {
            y1 = block * wincol + (block / 2);
            y2 = block * wincol + (block / 2);
            x1 = line_padding;
            x2 = (int) block_width - line_padding * 2;
        } else if (windia != -1) {
            Log.i(TAG, "windia = " + windia);
            if (windia == 1) {
                x1 = line_padding;
                y1 = line_padding;
                x2 = (int) block_width - line_padding * 2;
                y2 = (int) block_width - line_padding * 2;
            } else if (windia == 2) {
                x1 = line_padding;
                y1 = (int) block_width - line_padding * 2;
                x2 = (int) block_width - line_padding * 2;
                y2 = line_padding;
            }
        }
        paint.setColor(Color.argb(200, 244, 250, 80));
        paint.setStrokeWidth(40);
        canvas.drawLine(x1, y1, x2, y2, paint);
    }
}

public void drawAllElement(Canvas canvas) {
    int value[][] = game.getBoard();
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            switch (value[i][j]) {
                case 1: // player 1
                    drawElementX(canvas, i, j);
                    break;
                case 2: // player 2
                    drawElementO(canvas, i, j);
                    break;
            }
        }
    }
}

public void drawElementX(Canvas canvas, int i, int j) {
    int bw = (int) (block_width / 3);
    int bh = (int) (block_width / 3);
}

```

```

        paint.setColor(Color.rgb(246, 107, 22));
        canvas.drawLine(i * bw + bw / 2 + bh / 2 - 20, j * bh + bh / 2 + bh
            / 2 - 20, i * bw + bw / 2 - bh / 2 + 20, j * bh + bh / 2
            - bh / 2 + 20, paint);
        canvas.drawLine(i * bw + bw / 2 + bh / 2 - 20, j * bh + bh / 2 - bh
            / 2 + 20, i * bw + bw / 2 - bh / 2 + 20, j * bh + bh / 2
            + bh / 2 - 20, paint);
    }

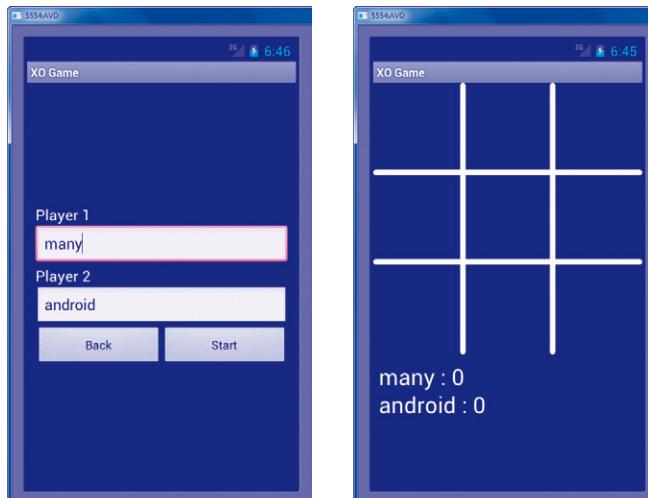
    public void drawElement0(Canvas canvas, int i, int j) {
        int bw = (int) (block_width / 3);
        int bh = (int) (block_width / 3);
        paint.setColor(Color.rgb(54, 130, 242));
        canvas.drawCircle(i * bw + bw / 2, j * bh + bh / 2, (bh - 20) / 2,
            paint);
        paint.setColor(Color.BLACK);
        canvas.drawCircle(i * bw + bw / 2, j * bh + bh / 2, (bh - 40) / 2,
            paint);
    }

    public void switchPlayer() {
        if (current_player == 1)
            current_player = 2;
        else
            current_player = 1;
    }

    public boolean onTouch(View view, MotionEvent event) {
        return true;
    }
}

```

เมื่อผู้อ่านลองรันเกมและเลือกป้อนชื่อผู้เล่น เมื่อคลิกปุ่ม Start จะได้ผลลัพธ์เป็นเล้นตารางที่ยังไม่สามารถคลิกเพื่อทำเครื่องหมายได้ดังรูป

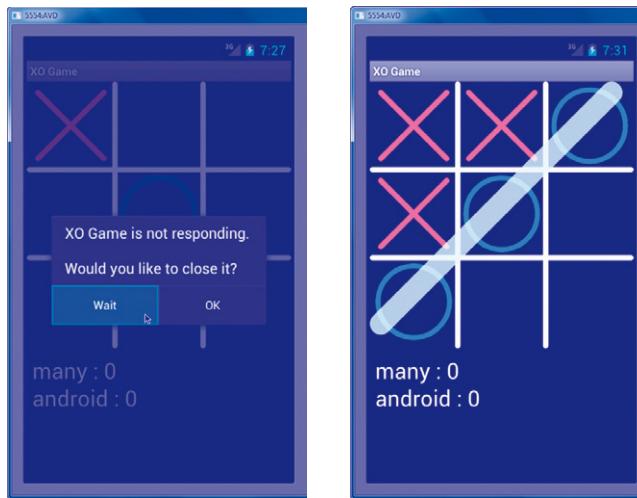


# การจัดการเกี่ยวกับหน้าจอสัมผัส (TouchScreen)

จากหัวข้อที่ผ่านมา การพัฒนาเกมจะใช้คลาส View มาสร้างเป็นหน้าจอเกม ซึ่งการดักจับเหตุการณ์ เมื่อผู้เล่นล้มปลั๊กที่หน้าจอสามารถทำได้ด้วยการอิมเพลเม้นต์คลาส OnTouchListener โดยเขียนโค้ด Java เพิ่ม ในเมธอด onTouch() ของคลาส Panel ดังนี้

```
public boolean onTouch(View view, MotionEvent event) {  
    if (event.getAction() == MotionEvent.ACTION_DOWN) {  
        Point point = new Point();  
        point.x = (int) event.getX();  
        point.y = (int) event.getY();  
  
        int i = (int) (1.0f * point.x / (block_width / 3));  
        int j = (int) (1.0f * point.y / (block_width / 3));  
        Log.i("Point", "(" + i + " " + j + ")");  
        game.makeMove(i, j, current_player);  
  
        if (one_player) {  
            switchPlayer();  
            SimpleOAI.calculateMove(game, current_player);  
            switchPlayer();  
        } else {  
            switchPlayer();  
        }  
  
        invalidate(); // refresh screen;  
  
        if (game.getWinner() != 0 && pause_flag != true) {  
            pause_flag = true;  
            gameOver(game.getWinner());  
        } else if (!game.isSpaceLeft()) {  
            pause_flag = true;  
            gameOver(0);  
        }  
    }  
  
    return true;  
}
```

เมื่อผู้อ่านรันเกมเพื่อทดสอบ และคลิกที่ช่องเพื่อทำเครื่องหมายจะปรากฏเครื่องหมาย X และ O ในกรณีที่ผู้อ่านเลือกแบบเล่นคนเดียว หลังจากคลิกที่ช่องเพื่อทำเครื่องหมาย จะปรากฏหน้าต่าง “XO Game is not responding” เนื่องจากความล่าช้าในการประมวลผลของ Emulator ให้ผู้อ่านคลิกปุ่ม “Wait” และเมื่อเล่นจนจบเกมจะปรากฏเลนส์เหลืองวีดทับเครื่องหมายของฝ่ายผู้ชนะดังรูป



ในการคลิกที่หน้าจอแต่ละครั้ง ก็จะมีการคำนวนเพื่อเพิ่มการรัวด์ X หรือ O โดยในระหว่างเล่นนั้น จะใช้ตัวแปร game ของคลาส SimpleXO ในการตรวจสอบว่าผู้เล่นเล่นจนจบเกมแล้วหรือไม่ และในกรณีที่เล่น คนเดียวจะใช้งานคลาส SimpleXOAI ใช้ในการคำนวนการเดินของฝ่ายโปรแกรม

## การปรับแต่งการแสดงผลเมื่อจบเกม

ในหัวข้อก่อนหน้านี้เมื่อผู้เล่นเล่นจบเกมแล้ว เกมก็จะไม่สามารถดำเนินการได้ ต่อได้ ซึ่งในกรณีนี้ผู้อ่านจะต้องเขียนโปรแกรมเพื่อแสดงข้อความแจ้งเดือนว่า ฝ่ายใดเป็นผู้ชนะ และมีปุ่มสำหรับเริ่มเล่นเกมใหม่ ซึ่งผู้อ่านสามารถทำได้ตามขั้นตอนดังต่อไปนี้

1. เขียนโค้ด Java เพิ่มในเมธอด gameOver() ของคลาส XOGameActivity ดังนี้

```
// start GameOverActivity when game finish.
public void gameOver(int win) {

    if (win != 0) {
        player_score[win-1]++;
        Intent intent = new Intent(getApplicationContext(), GameOverActivity.class);
        intent.putExtra("GAME_MSG", player_name[win-1] + " Win!");
        startActivityForResult(intent, 10);
    } else {
        Intent intent = new Intent(getApplicationContext(), GameOverActivity.class);
        intent.putExtra("GAME_MSG", "XO");
        startActivityForResult(intent, 10);
    }
}
```

จากโค้ดจะเพิ่มคะแนนให้ฝ่ายที่ชนะ และเรียกใช้งานแอ็คทิวิตี้ GameOverActivity โดยมีรูปแบบ การแสดงผลดังนี้

- ในกรณีที่มีฝ่ายชนะก็จะส่งชื่อของฝ่ายที่ชนะ เพื่อแสดงผลออกทางหน้าจอ
  - ในกรณีที่เสมอ ก็จะส่งข้อความ “XO” แทน
- ซึ่งคลาส GameOverActivity จะเป็นผู้ทำหน้าที่นำข้อความไปแสดงผล

## 2. ให้ Override เมธอด onActivityResult() ในคลาส XOGameActivity ดังนี้

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    // new game  
    if (resultCode == RESULT_OK && requestCode == 10) {  
        game = new SimpleXO(3);  
        pause_flag = false;  
        setContentView(new Panel(this));  
    }  
}
```

จากโค้ดจะเป็นการเริ่มเกมใหม่เมื่อผู้เล่นคลิกปุ่ม New Game ของแอ็คทิวิตี้ GameOverActivity ซึ่งผู้อ่านจะเห็นว่า ในการใช้งานแอ็คทิวิตี้ GameOverActivity จะใช้งานเมธอด startActivityResult() นั้นหมายกึ่งการเริ่มแอ็คทิวิตี้ใหม่ และรอผลลัพธ์จากแอ็คทิวิตี้นั้นด้วย เมื่อได้ผลลัพธ์เป็น RESULT\_OK ก็จะเริ่มเกมใหม่ เมื่อผู้อ่านเขียนโค้ดในไฟล์ XOGameActivity เสร็จเรียบร้อยแล้ว จะได้โค้ดที่เสร็จสมบูรณ์แล้วดังนี้

```
package com.android.example.xogame;  
  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Paint.Cap;  
import android.graphics.Point;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.View.OnTouchListener;  
  
public class XOGameActivity extends Activity {  
    String TAG = "OX Game";  
    public Paint paint;  
    public static float block_width = 0;  
    public static int current_player = 1;  
    public static boolean one_player = false;  
    public static int count = 0;  
    public static int winner = -1;  
    public boolean pause_flag = false;  
  
    public static String[] player_name = { "Player 1", "Player 2" };  
    public static int[] player_score = { 0, 0 };  
  
    public static int line_padding = 20;  
    SimpleXO game = new SimpleXO(3);  
  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // get data from GamePlayerActivity  
        Bundle extras = getIntent().getExtras();  
    }
```



```
    if (extras != null) {
        int porson = extras.getInt("PLAYER");
        if (porson == 1) {
            one_player = true;
        } else {
            one_player = false;
        }
        player_name[0] = extras.getString("PLAYER_ONE_NAME");
        player_name[1] = extras.getString("PLAYER_TWO_NAME");
    }
    setContentView(new Panel(this));

}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    // new game
    if (resultCode == RESULT_OK && requestCode == 10) {
        game = new SimpleXO(3);
        pause_flag = false;
        setContentView(new Panel(this));
    }
}

// start GameOverActivity when game finish.
public void gameOver(int win) {

    if (win != 0) {
        player_score[win - 1]++;
        Intent intent = new Intent(getApplicationContext(), GameOverActivity.class);
        intent.putExtra("GAME_MSG", player_name[win - 1] + " Win!");
        startActivityForResult(intent, 10);
    } else {
        Intent intent = new Intent(getApplicationContext(), GameOverActivity.class);
        intent.putExtra("GAME_MSG", "XO");
        startActivityForResult(intent, 10);
    }
}

class Panel extends View implements OnTouchListener {
    public Panel(Context context) {
        super(context);
        setFocusable(true);
        setFocusableInTouchMode(true);
        this.setOnTouchListener(this);
    }

    public void drawBoard(Canvas canvas) {
        int line_size = 5;
        float w = this.getWidth();
        float h = this.getHeight();
        float s = w;
        if (w > h)
            s = h;

        block_width = s;

        int bw = (int) (s / 3);

        canvas.drawLine(0, 0, 0, s, line_size);
        canvas.drawLine(0, 0, w, 0, line_size);
        canvas.drawLine(w, 0, w, s, line_size);
        canvas.drawLine(0, s, w, s, line_size);
        canvas.drawLine(w / 3, 0, w / 3, s, line_size);
        canvas.drawLine((w * 2) / 3, 0, (w * 2) / 3, s, line_size);
        canvas.drawLine(0, (s * 2) / 3, w, (s * 2) / 3, line_size);
        canvas.drawLine(0, (s * 2) / 3, w / 3, (s * 2) / 3, line_size);
        canvas.drawLine(0, (s * 2) / 3, (w * 2) / 3, (s * 2) / 3, line_size);
        canvas.drawLine(w / 3, (s * 2) / 3, w / 3, (w * 2) / 3, line_size);
        canvas.drawLine((w * 2) / 3, (s * 2) / 3, (w * 2) / 3, (w * 2) / 3, line_size);
    }

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (game != null) {
            int x = event.getX();
            int y = event.getY();
            if (x < 0 || x > w || y < 0 || y > s)
                return true;
            if (x < w / 3) {
                if (y < s / 3)
                    game.setCell(0, 0, 'X');
                else if (y > (s * 2) / 3)
                    game.setCell(0, 2, 'X');
                else
                    game.setCell(0, 1, 'X');
            } else if (x > (w * 2) / 3) {
                if (y < s / 3)
                    game.setCell(2, 0, 'X');
                else if (y > (s * 2) / 3)
                    game.setCell(2, 2, 'X');
                else
                    game.setCell(2, 1, 'X');
            } else {
                if (y < s / 3)
                    game.setCell(1, 0, 'X');
                else if (y > (s * 2) / 3)
                    game.setCell(1, 2, 'X');
                else
                    game.setCell(1, 1, 'X');
            }
        }
        return true;
    }
}
```

```

        int bh = (int) (s / 3);
        paint = new Paint();
        paint.setColor(Color.BLACK);
        canvas.drawRect(0, 0, w, h, paint);
        paint.setColor(Color.WHITE);
        paint.setAntiAlias(true);
        paint.setStrokeWidth(10);
        paint.setStrokeCap(Cap.ROUND);
        // draw board
        canvas.drawLine(bw, line_size, bw, s, paint);
        canvas.drawLine(2 * bw, line_size, 2 * bw, s, paint);
        canvas.drawLine(line_size, bh, w - line_size, bh, paint);
        canvas.drawLine(line_size, 2 * bh, w - line_size, 2 * bh, paint);

        paint.setTextSize(40);
        canvas.drawText(player_name[0] + " : " + player_score[0], 10,
                        s + 60, paint);
        canvas.drawText(player_name[1] + " : " + player_score[1], 10,
                        s + 110, paint);

    }

    public void onDraw(Canvas canvas) {
        drawBoard(canvas);
        drawAllElement(canvas);
        // draw line when game finish
        if (game.getWinner() != 0) {

            int block = (int) (block_width / 3);
            int winrow = game.getWin_row();
            int wincol = game.getWin_col();
            int windia = game.getWin_dia();

            int x1 = 0, x2 = 0, y1 = 0, y2 = 0;

            if (winrow != -1) {
                x1 = block * winrow + (block / 2);
                x2 = block * winrow + (block / 2);
                y1 = line_padding;
                y2 = (int) block_width - line_padding * 2;
            } else if (wincol != -1) {
                y1 = block * wincol + (block / 2);
                y2 = block * wincol + (block / 2);
                x1 = line_padding;
                x2 = (int) block_width - line_padding * 2;
            } else if (windia != -1) {
                Log.i(TAG, "windia = " + windia);
                if (windia == 1) {
                    x1 = line_padding;
                    y1 = line_padding;
                    x2 = (int) block_width - line_padding * 2;
                    y2 = (int) block_width - line_padding * 2;
                } else if (windia == 2) {
                    x1 = line_padding;
                    y1 = (int) block_width - line_padding * 2;
                    x2 = (int) block_width - line_padding * 2;
                    y2 = line_padding;
                }
            }
        }
    }
}

```

```
        }
        paint.setColor(Color.argb(200, 244, 250, 80));
        paint.setStrokeWidth(40);
        canvas.drawLine(x1, y1, x2, y2, paint);
    }
}

public void drawAllElement(Canvas canvas) {
    int value[][] = game.getBoard();
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            switch (value[i][j]) {
                case 1: // player 1
                    drawElementX(canvas, i, j);
                    break;
                case 2: // player 2
                    drawElementO(canvas, i, j);
                    break;
            }
        }
    }
}

public void drawElementX(Canvas canvas, int i, int j) {
    int bw = (int) (block_width / 3);
    int bh = (int) (block_width / 3);
    paint.setColor(Color.rgb(246, 107, 22));
    canvas.drawLine(i * bw + bw / 2 + bh / 2 - 20, j * bh + bh / 2 + bh
        / 2 - 20, i * bw + bw / 2 - bh / 2 + 20, j * bh + bh / 2
        - bh / 2 + 20, paint);
    canvas.drawLine(i * bw + bw / 2 + bh / 2 - 20, j * bh + bh / 2 - bh
        / 2 + 20, i * bw + bw / 2 - bh / 2 + 20, j * bh + bh / 2
        + bh / 2 - 20, paint);

}

public void drawElementO(Canvas canvas, int i, int j) {
    int bw = (int) (block_width / 3);
    int bh = (int) (block_width / 3);
    paint.setColor(Color.rgb(54, 130, 242));
    canvas.drawCircle(i * bw + bw / 2, j * bh + bh / 2, (bh - 20) / 2,
        paint);
    paint.setColor(Color.BLACK);
    canvas.drawCircle(i * bw + bw / 2, j * bh + bh / 2, (bh - 40) / 2,
        paint);
}

public void switchPlayer() {
    if (current_player == 1)
        current_player = 2;
    else
        current_player = 1;
}

public boolean onTouch(View view, MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        Point point = new Point();
        point.x = (int) event.getX();
        point.y = (int) event.getY();
    }
}
```

```

        int i = (int) (1.0f * point.x / (block_width / 3));
        int j = (int) (1.0f * point.y / (block_width / 3));
        Log.i("Point", "(" + i + "" + j + ")");
        game.makeMove(i, j, current_player);

        if (one_player) {
            switchPlayer();
            SimpleXOAI.calculateMove(game, current_player);
            switchPlayer();
        } else {
            switchPlayer();
        }

        invalidate(); // refresh screen;

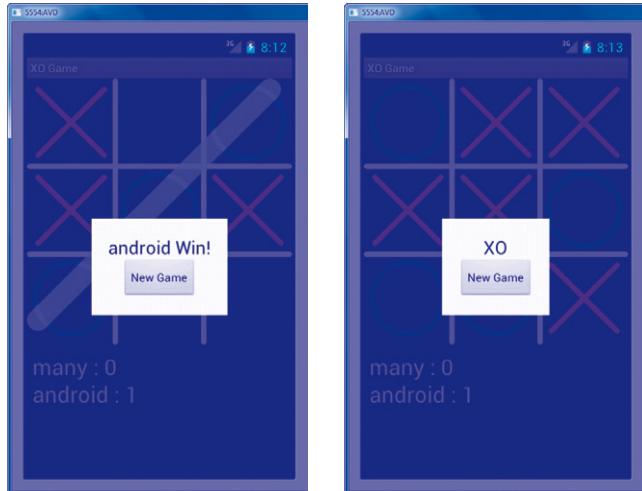
        if (game.getWinner() != 0 && pause_flag != true) {
            pause_flag = true;
            gameOver(game.getWinner());
        } else if (!game.isSpaceLeft()) {
            pause_flag = true;
            gameOver(0);
        }
    }

    return true;
}
}
}

```

## ทดลองการทำงานของเกม

ให้ผู้อ่านทดสอบเกมที่เขียนขึ้นทั้งหมด โดยการรันโปรแกรมจากนั้นให้กดลองเล่นเกม ซึ่งจะสังเกตได้ว่าหากฝ่ายใดฝ่ายหนึ่งสามารถคว่ำเครื่องหมายในแนวตั้ง แนวอน หรือแนวทแยงได้ก่อน ฝ่ายนั้นก็จะเป็นผู้ชนะดังรูป



จากรูปทางซ้ายมือ ฝ่าย O เป็นฝ่ายชนะ เนื่องจากวงเครื่องหมาย O ในแนวทแยงได้ก่อนนั่นเอง หรือจากรูปทางขวา มือ ในกรณีที่ไม่มีฝ่ายใดเป็นผู้ชนะ

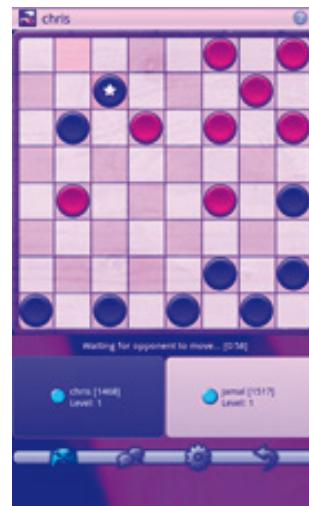
# สรุปก้ายบต

ในบทนี้เป็นการอธิบายการพัฒนาแอพพลิเคชัน โดยประยุกต์ใช้กราฟิกแบบ 2 มิติมาพัฒนาเกมในแบบ 2 มิติ ซึ่งใช้งานฟังก์ชันสำคัญๆ เช่น TouchScreen, Canvas และ Panel เป็นต้น

## แบบฝึกหัดก้ายบต

หลังจากผู้อ่านได้ทดลองพัฒนาเกม XO เรียนร้อยแล้ว แบบฝึกหัดในส่วนนี้ให้ผู้อ่านพัฒนาเกมมากอีก ที่มีหน้าจอขนาดของตาราง  $8 \times 8$  ทั้งนี้จะมีผู้เล่น 2 ฝ่าย แต่ละฝ่ายจะมีตัวหมากฝ่ายละ 8 ตัว โดยที่ตารางมีการทำเครื่องหมายช่องเว้นช่องสลับกัน ทั้งนี้หากอยู่มีเกติก้าและเทคนิคการเล่นดังนี้

- วิธีเล่นจะเริ่มจากแต่ละฝ่ายจะวางตัวหมากทั้ง 8 เป็น 2 แถว และจะเดินได้เฉพาะช่องที่ทำเครื่องหมายไว้เท่านั้น ข้อลังก์เกตคือ ตัวหมากธรรมดามีความสามารถเดินถอยหลังได้ อย่างไรก็ตาม เมื่อเดินไปถึงสุดตารางของฝ่ายตรงข้ามจะได้ลิทธิพิเศษเรียกว่า ยอด ซึ่งในกรณีนี้จะสามารถเดินหน้าหรือถอยหลังก็ได้
- ในการเล่นจะมีการหลอกล่อให้ฝ่ายตรงข้ามหลงกล หรือให้เข้ามายกนพรี หรือมีการหลอกล่อออก มาเพื่อให้สามารถนำหมากของอีกฝ่ายเข้ายกอลได้
- เกมนี้แต่ละฝ่ายจะต้องกินฝ่ายตรงข้ามให้หมดก่อน ผู้ที่ยังมีหมากเหลืออยู่บนตารางจะเป็นผู้ชนะ หรือกรณีที่เล่นฝ่ายตรงข้ามจนมุมและยอมแพ้ไปเอง

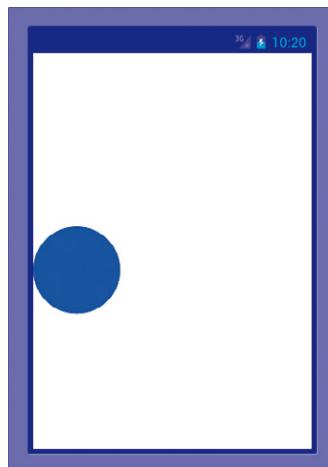


▲ ตัวอย่างแบบฝึกหัด

## การใช้งานเซนเซอร์ (Sensor) ใน Android

เนื้อหาในบทนี้ จะเป็นล้วนของการเรียกใช้งานฟังก์ชันสนับสนุนการทำงานของเซนเซอร์ (Sensor) หรือเครื่องรับรู้ของโทรศัพท์เคลื่อนที่ Android โดยตัวอย่างแอพพลิเคชันการใช้งานในบทนี้ จะเป็นแอพพลิเคชัน สะท้อนลูกบอลไปมา หรือ Bouncing Ball

**Bouncing Ball** เป็นแอพพลิเคชันที่ประกอบด้วยลูกบอลทรงกลมหนึ่งลูก ซึ่งจะกลิ้งหรือเคลื่อนที่ไปมา (ซ้าย ขวา บน ลง) บนหน้าจอของแอพพลิเคชัน โดยการเคลื่อนที่นั้นจะขึ้นอยู่กับลักษณะของการถือโทรศัพท์ เคลื่อนที่ เช่น เมื่อมีการเอียงโทรศัพท์เคลื่อนที่ไปทางซ้าย ลูกบอลก็จะกลิ้งไปทางด้านซ้าย หรือเมื่อลูบคลิ๊ก กับขอบโทรศัพท์เคลื่อนที่ ลูกบอลก็จะกระดอนหรือสะท้อนกลับไปในทิศทางตรงกันข้าม สำหรับทิศทางและความเร็วของการเคลื่อนที่นั้นจะถูกควบคุมหรือกำหนดไว้ด้วย Sensor



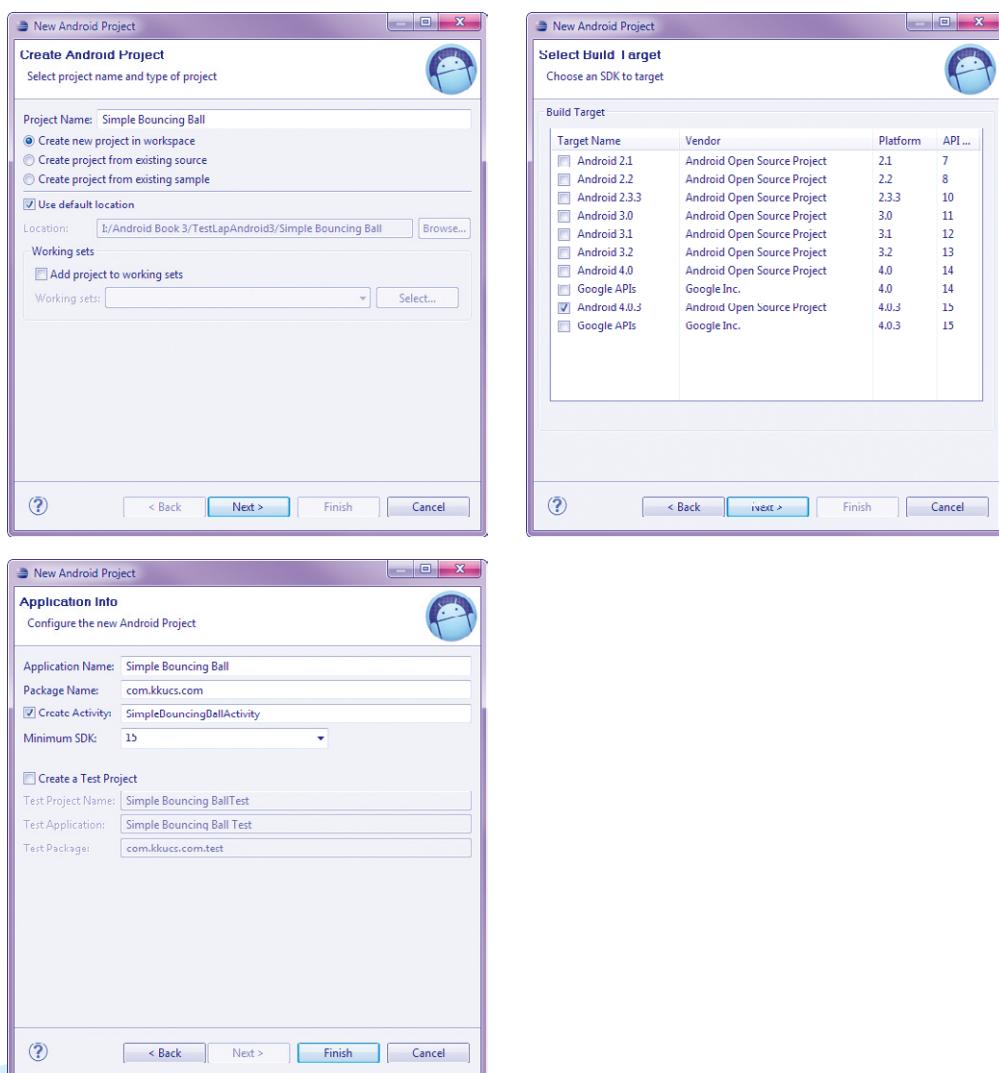
▲ ตัวอย่างแอพพลิเคชัน

โดยเนื้อหาในบทนี้จะแบ่งออกเป็น 5 ส่วนดังนี้

- การสร้างโปรเจกต์ (Project)
- การแก้ไขไฟล์ (File) เอ็กซ์เต้มแอล (XML—Extensible Markup Language)
- การออกแบบหน้าจอของแอพพลิเคชัน
- การแก้ไขแฟ้มหรือไฟล์หลัก หรือไฟล์ Java
- การรัน (Run) โปรแกรมบนโทรศัพท์จริง

## การสร้างโปรเจกต์ (Project)

ขั้นตอนแรกของการเริ่มพัฒนาแอพพลิเคชันทุกชนิดก็คือ การสร้างโปรเจกต์ (ผู้อ่านสามารถศึกษาขั้นตอนการสร้างโปรเจกต์ได้ในบทที่ 2) โดยตั้งชื่อโปรเจกต์นี้ว่า Simple Bouncing Ball



## กำหนดสิทธิ์การทำงานให้ออพพลิเคชัน

ในหัวข้อนี้จะเป็นการแก้ไขไฟล์ AndroidManifest.xml เพื่อเพิ่มการอนุญาตการใช้สิทธิ์ (Permission) สำหรับการเรียกใช้งาน Sensor เมื่อลูกบลอกกระแทกบนของโทรศัพท์เคลื่อนที่ ก็จะทำให้มีการลั่นของโทรศัพท์เคลื่อนที่ เมื่อเป็นการเตือนของการชนกันนั่นเอง

ซึ่งสามารถกำหนดสิทธิ์ได้โดยให้ผู้อ่านเพิ่มโค้ด XML ในไฟล์ AndroidManifest.xml ที่เรียกว่า **android.permission.VIBRATE** เข้าไปดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.kkucs.com"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".SimpleBouncingBallActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.VIBRATE"></uses-permission>
</manifest>
```

สำหรับการแก้ไขไฟล์ AndroidManifest.xml ผู้อ่านจะต้องตรวจสอบส่วนของการเชื่อมต่อกับไฟล์ Java โดยสามารถสังเกตได้จาก activity android:name=".SimpleBouncingBallActivity" ในกรณีนี้ SimpleBouncingBall จะเป็นชื่อของคลาส SimpleBouncingBallActivity.java

## ออกแบบหน้าจอแอพพลิเคชัน

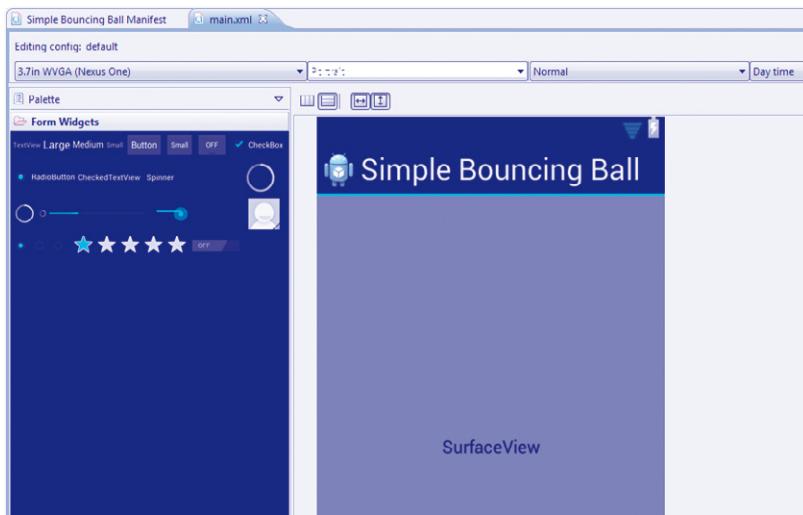
สำหรับการสร้างแอพพลิเคชันเพื่อเรียกใช้การทำงานของ Sensor นั้น ผู้เขียนจะออกแบบหน้าจอของแอพพลิเคชันให้มีรูปแบบที่ไม่ซับซ้อน เพื่อรับรับการแสดงผลสำหรับภาพเคลื่อนไหวเท่านั้น ดังนั้น การออกแบบหลักจะอยู่ในส่วนของไฟล์ Java ซึ่งโค้ด XML ในไฟล์ main.xml มีดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <SurfaceView
```

```

        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:id="@+id/bouncing_ball_surface"
    />
</LinearLayout>
```

ผู้อ่านจะลังเลติดใจว่าจะมีการเพิ่มอิลิเม้นต์ (Element) SurfaceView เข้าไปเพื่อใช้ในการสนับสนุนการแสดงผลของภาพเคลื่อนไหวนั้นเอง ซึ่งได้ผลลัพธ์ดังนี้



## เขียนโค้ดกำหนดการทำงานของแอพพลิเคชัน

ในหัวข้อนี้เป็นการแก้ไขไฟล์ Java ซึ่งเป็นการแก้ไขไฟล์หลัก เพื่อกำหนดการทำงานของแอพพลิเคชัน ในที่นี้จะแบ่งออกเป็น 2 ไฟล์คือ Ball.java และ SimpleBouncingBall.java

### สร้างคลาสควบคุมการเปลี่ยนแปลงตำแหน่งของลูกบอล

ลิงแรกที่ผู้อ่านต้องทำในการสร้างคลาสควบคุมการเปลี่ยนแปลงตำแหน่งของลูกบอลคือ ให้สร้างคลาสที่ชื่อว่า Ball ขึ้นมา ซึ่งคลาสนี้จะทำหน้าที่คำนวนและควบคุมการเปลี่ยนแปลงตำแหน่งของลูกบอล โดยใช้ค่าความเร็วที่ได้รับจากคลาสหลัก หรือคลาส SimpleBouncingBallActivity (ซึ่งจะกล่าวในหัวข้อถัดไป) เมื่อสร้างคลาส Ball เรียบร้อยแล้ว ให้ผู้อ่านเขียนโค้ดกำหนดการทำงานของคลาส Ball ดังนี้

```

package com.kkucs.com;

import java.util.concurrent.atomic.AtomicReference;
import android.os.Vibrator; ← เพิ่มคลาส

public class Ball {
    private final float pixelsPerMeter = 5;
    private final int ballRadius;
    public float ballPixelX, ballPixelY;
    private int pixelWidth, pixelHeight;
```

```

private float velocityX, velocityY;
private float accelX, accelY;
private static final float rebound = 0.8f;
private static final float STOP_BOUNCING_VELOCITY = 2f;
private volatile long lastTimeMs = -1;
public final Object LOCK = new Object();
private AtomicReference<Vibrator> vibratorRef = new AtomicReference<Vibrator>();

public Ball(int ballRadius) {
    this.ballRadius = ballRadius;
}

public void setAccel(float ax, float ay) {
    synchronized (LOCK) {
        this.accelX = ax;
        this.accelY = ay;
    }
}

public void setSize(int width, int height) {
    synchronized (LOCK) {
        this.pixelWidth = width;
        this.pixelHeight = height;
    }
}

public int getBallRadius() {
    return ballRadius;
}

public void moveBall(int ballX, int bally) {
    synchronized (LOCK) {
        this.ballPixelX = ballX;
        this.ballPixelY = bally;
        velocityX = 0;
        velocityY = 0;
    }
}

public void updatePhysics() {
    float lWidth, lHeight, lBallX, lBallY, lAx, lAy, lVx, lVy;
    synchronized (LOCK) {
        lWidth = pixelWidth;
        lHeight = pixelHeight;
        lBallX = ballPixelX;
        lBallY = ballPixelY;
        lVx = velocityX;
        lVy = velocityY;
        lAx = accelX;
        lAy = -accelY;
    }

    if (lWidth <= 0 || lHeight <= 0) {
        return;
    }
}

```

```
long curTime = System.currentTimeMillis();
if (lastTimeMs < 0) {
    lastTimeMs = curTime;
    return;
}

long elapsedMs = curTime - lastTimeMs;
lastTimeMs = curTime;

lVx += ((elapsedMs * lAx) / 1000) * pixelsPerMeter;
lVy += ((elapsedMs * lAy) / 1000) * pixelsPerMeter;

lBallX += ((lVx * elapsedMs) / 1000) * pixelsPerMeter;
lBallY += ((lVy * elapsedMs) / 1000) * pixelsPerMeter;

boolean bouncedX = false;
boolean bouncedY = false;

if (lBallY - ballRadius < 0) {
    lBallY = ballRadius;
    lVy = -lVy * rebound;
    bouncedY = true;
} else if (lBallY + ballRadius > lHeight) {
    lBallY = lHeight - ballRadius;
    lVy = -lVy * rebound;
    bouncedY = true;
}

if (bouncedY && Math.abs(lVy) < STOP_BOUNCING_VELOCITY) {
    lVy = 0;
    bouncedY = false;
}

if (lBallX - ballRadius < 0) {
    lBallX = ballRadius;
    lVx = -lVx * rebound;
    bouncedX = true;
} else if (lBallX + ballRadius > lWidth) {
    lBallX = lWidth - ballRadius;
    lVx = -lVx * rebound;
    bouncedX = true;
}

if (bouncedX && Math.abs(lVx) < STOP_BOUNCING_VELOCITY) {
    lVx = 0;
    bouncedX = false;
}

synchronized (LOCK) {
    ballPixelX = lBallX;
    ballPixelY = lBallY;

    velocityX = lVx;
    velocityY = lVy;
}
```

```

        if (bouncedX || bouncedY) {
            Vibrator v = vibratorRef.get();
            if (v != null) {
                v.vibrate(20L);
            }
        }
    }

    public void setVibrator(Vibrator v) {
        vibratorRef.set(v);
    }
}

```

## เขียนโค้ดกำหนดการทำงานในคลาสหลัก

เนื่องจากในบทนี้จะเป็นการใช้งาน Sensor ที่เกี่ยวข้องกับความเร่ง โดยมีการตรวจสอบค่าความเร่ง จากแกน x และแกน y ซึ่งมีการแปลงค่าเป็นความเร็ว จากนั้นนำค่าที่ได้ไปคำนวนหาค่าดำเนินการ อีกครั้งหนึ่ง และในการนี้ผู้เขียนเรียกใช้คลาส SensorManager เข้ามาช่วยในการจัดการกับ Sensor

- สำหรับการพัฒนาคลาส SensorListener นั้น ผู้เขียนจะ Override เมธอดต่างๆ ที่เกี่ยวข้องดังต่อไปนี้
- `onAccuracyChanged(int sensor, int accuracy)` เมธอดนี้จะถูกเรียกใช้งานเมื่อค่าความแม่นยำมีการเปลี่ยนแปลง
  - `onSensorChanged(int sensor, float[] values)` ในการนี้ค่า Sensor มีการเปลี่ยนแปลง เมธอดนี้ ก็จะถูกเรียกใช้งาน ในที่นี่เราจะใช้ Sensor ความเร่ง โดยกำหนดให้ `values` เป็นตัวแปรอาร์เรย์ (Array) เช่น `values[0]`, `values[1]` และ `values[2]` ที่ใช้สำหรับเก็บค่าความเร่งตามแกน x, y และ z ตามลำดับ

ดังนั้น ให้ผู้อ่านเขียนโค้ดกำหนดการทำงานของคลาส SimpleBouncingBallActivity ดังนี้

```

package com.kkucs.com;

import static android.hardware.SensorManager.DATA_X;
import static android.hardware.SensorManager.DATA_Y;
import static android.hardware.SensorManager.SENSOR_ACCELEROMETER;
import static android.hardware.SensorManager.SENSOR_DELAY_GAME;

import java.util.concurrent.TimeUnit;

import android.app.Activity;
import android.content.Context;
import android.content.pm.ActivityInfo;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.hardware.SensorListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.os.Vibrator;
import android.view.SurfaceHolder;
import android.view.SurfaceView;

```

← เพิ่มคลาส



```
import android.view.Window;
import android.view.WindowManager;
import android.view.SurfaceHolder.Callback;

public class SimpleBouncingBallActivity extends Activity implements Callback,
        SensorListener {
    private static final int BALL_RADIUS = 50;
    private SurfaceView surface;
    private SurfaceHolder holder;
    private final Ball ball = new Ball(BALL_RADIUS);
    private MyThread thread ;
    private Paint backgroundPaint;
    private Paint ballPaint;
    private SensorManager sensorMgr;
    private long lastSensorUpdate = -1;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        setContentView(R.layout.main);
        surface = (SurfaceView) findViewById(R.id.bouncing_ball_surface);
        holder = surface.getHolder();
        surface.getHolder().addCallback(this);

        backgroundPaint = new Paint();
        backgroundPaint.setColor(Color.WHITE);

        ballPaint = new Paint();
        ballPaint.setColor(Color.BLUE);
        ballPaint.setAntiAlias(true);
    }

    protected void onPause() {
        super.onPause();

        ball.setVibrator(null);
        sensorMgr.unregisterListener(this, SENSOR_ACCELEROMETER);
        sensorMgr = null;
        ball.setAccel(0, 0);
    }

    protected void onResume() {
        super.onResume();

        sensorMgr = (SensorManager) getSystemService(SENSOR_SERVICE);
        boolean accelSupported = sensorMgr.registerListener(this,
                SENSOR_ACCELEROMETER, SENSOR_DELAY_GAME);

        if (!accelSupported) {
            // on accelerometer on this device
            sensorMgr.unregisterListener(this, SENSOR_ACCELEROMETER);
        }
        Vibrator vibrator = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
        ball.setVibrator(vibrator);
    }
}
```

```

public void surfaceChanged(SurfaceHolder holder, int format, int width,
                           int height) {
    ball.setSize(width, height);
}

public void surfaceCreated(SurfaceHolder holder) {
    thread = new MyThread ();
    thread.start();
}

private void draw() {
    Canvas c = null;
    try {
        c = holder.lockCanvas();
        if (c != null) {
            doDraw(c);
        }
    } finally {
        if (c != null) {
            holder.unlockCanvasAndPost(c);
        }
    }
}

private void doDraw(Canvas c) {
    int width = c.getWidth();
    int height = c.getHeight();
    c.drawRect(0, 0, width, height, backgroundPaint);

    float ballX, ballY;
    synchronized (ball.LOCK) {
        ballX = ball.ballPixelX;
        ballY = ball.ballPixelY;

    }
    c.drawCircle(ballX, ballY, BALL_RADIUS, ballPaint);
}

public void surfaceDestroyed(SurfaceHolder holder) {
    try {
        ball.setSize(0, 0);
        thread .safeStop();
    } finally {
        thread = null;
    }
}

private class MyThread extends Thread {
    private volatile boolean running = true;
    public void run() {
        while (running) {
            try {
                TimeUnit.MILLISECONDS.sleep(5);
                draw();
                ball.updatePhysics(); // update ตำแหน่งของลูกบอล
            }
        }
    }
}

```

```

        } catch (InterruptedException ie) {
            running = false;
        }
    }

    public void safeStop() {
        running = false;
        interrupt();
    }
}

public void onAccuracyChanged(int sensor, int accuracy) {
}

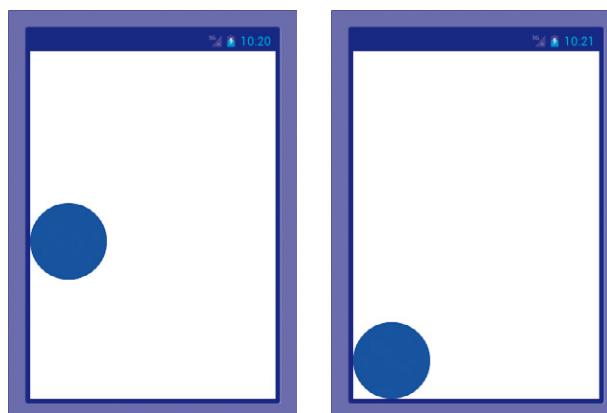
public void onSensorChanged(int sensor, float[] values) {
    if (sensor == SENSOR_ACCELEROMETER) {
        long curTime = System.currentTimeMillis();
        if (lastSensorUpdate == -1 || (curTime - lastSensorUpdate) > 50) {
            lastSensorUpdate = curTime;
            ball.setAccel(values[DATA_X], values[DATA_Y]);
        }
    }
}
}

```

## ทดลองการทำงานของแอพพลิเคชัน

เมื่อผู้อ่านพัฒนาแอพพลิเคชันเรียบร้อยแล้ว ต่อไปจะเป็นการทดสอบการทำงานของแอพพลิเคชัน ซึ่งผู้อ่านต้องติดตั้งแอพพลิเคชันบนโทรศัพท์เคลื่อนที่จริงเล่ายก่อน (ศึกษาได้จากบทที่ 16)

หลังจากการติดตั้งและเปิดใช้งานแอพพลิเคชัน ผู้อ่านจะสังเกตเห็นลูกบล็อกเคลื่อนที่อยู่บนหน้าจอ ดังรูป



นอกจากนี้เมื่อผู้อ่านเอียงเครื่องโทรศัพท์ ลูกบล็อกก็จะค่อยๆ เคลื่อนที่บนหน้าจอตามทิศทางที่ผู้อ่านเอียงเครื่องโทรศัพท์ และเมื่อลูกบล็อกกระแทกกับขอบของหน้าจอ ก็จะเกิดการสั่นของโทรศัพท์ และลูกบล็อกก็จะกระดอนกลับไปในทิศทางตรงกันข้าม

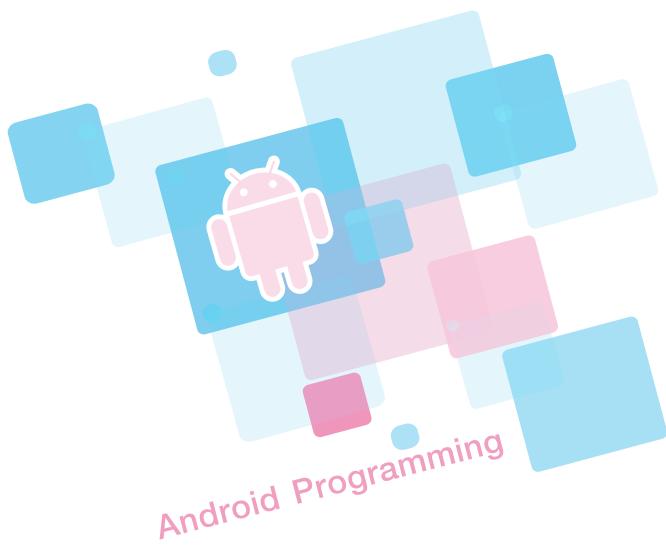
# สรุปท้ายบท

ในบทนี้เป็นการอธิบายถึงการพัฒนาแอพพลิเคชันโดยเรียกใช้งานฟังก์ชัน Sensor บนเครื่องโทรศัพท์ Android ซึ่งแอพพลิเคชันมีลักษณะเสมือนเกม 2 มิติที่มีลูกบอลกลิ้งไปมาบนหน้าจอ

## แบบฝึกหัดท้ายบท

แบบฝึกหัดในบทนี้ ให้ผู้อ่านปรับแต่งหน้าจอส่วนของลูกบอลดังนี้

- ในกรณีที่ลูกบอลมีการกระแทกกับขอบของโทรศัพท์ ให้ลูกบอลเปลี่ยนลีบเป็นอีกสีหนึ่งที่ไม่เหมือนกัน เช่น แดง เหลือง เขียว น้ำเงินตามลำดับ
- ให้ผู้อ่านสร้างลูกบอลจำนวน 2 ลูก โดยมีการเคลื่อนที่ที่เป็นอิสระต่อกัน (เช่น ความเร็วไม่เท่ากัน) และเมื่อชนกันให้ลูกบอลเกิดการสะท้อนไปในทิศทางที่ตรงกันข้าม

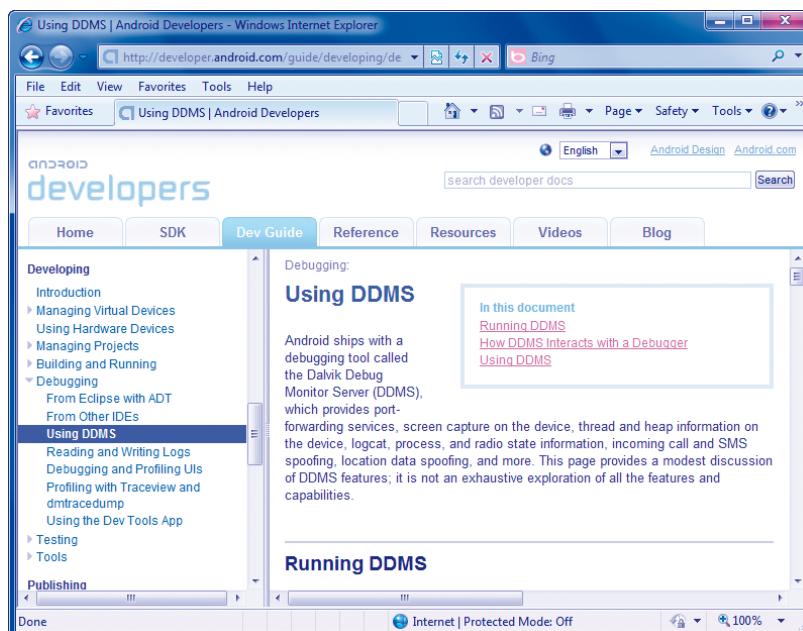


## เทคนิคในการแก้ไขปัญหาสำหรับการพัฒนาแอปพลิเคชัน

เนื้อหาในบทนี้ จะเป็นส่วนเพิ่มเติมเพื่อช่วยให้ผู้อ่านนำเทคนิคต่างๆ ไปประยุกต์ใช้ในการแก้ไขปัญหา หรือจุดบกพร่องที่อาจจะเกิดขึ้นได้ในระหว่างการพัฒนาแอปพลิเคชัน รวมไปถึงแนะนำเครื่องมือช่วยเหลือต่างๆ ด้วย

ในบทนี้จะแบ่งเนื้อหาออกเป็น 2 ส่วนหลักๆ คือ การใช้เครื่องมือ Emulator Control และการใช้งาน LogCat อย่างไรก็ตามผู้อ่านที่สนใจสามารถศึกษาเพิ่มเติมได้ที่เว็บไซต์ต่างๆ ดังนี้

- <http://developer.android.com/guide/developing/tools/ddms.html>



- <http://developer.android.com/guide/developing/tools/adb.html>

The screenshot shows a Windows Internet Explorer window displaying the "Android Debug Bridge | Android Developers" page. The URL in the address bar is <http://devcloper.android.com/guide/developing/tools/adb.html>. The page content is about the Android Debug Bridge (adb), which is a versatile command line tool for communicating with emulators or connected devices. It includes sections on ADB quickview, issuing ADB commands, querying for emulator/device instances, directing commands to specific instances, installing applications, forwarding ports, copying files, listing adb commands, issuing shell commands, and enabling local logging.

- <http://developer.android.com/guide/developing/tools/emulator.html>

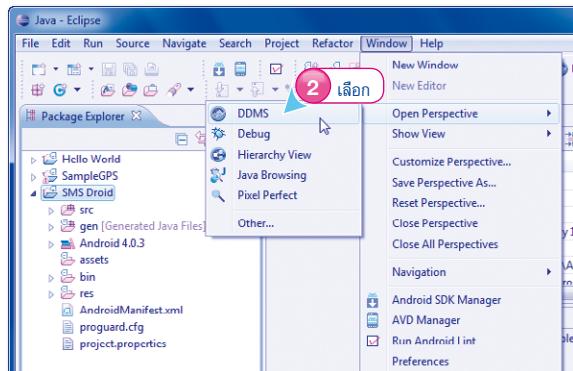
The screenshot shows a Windows Internet Explorer window displaying the "Android Emulator | Android Developers" page. The URL in the address bar is <http://devcloper.android.com/guide/developing/tools/cmulator.html>. The page content is about the Android Emulator, which is a software component of the Android SDK that runs on a host computer to simulate a mobile device. It includes sections on emulator startup options, keyboard mapping, and links to using the emulator and managing virtual devices.

# การใช้งานเครื่องมือ Dalvik Debug Monitor Server (DDMS)

ในบางกรณีจำเป็นต้องรับการสั่งและตรวจสอบการทำงานของแอปพลิเคชันบนระบบปฏิบัติการ Android ที่ติดตั้งบนเครื่องพกพา (Event) ต่างๆ เช่น การเปลี่ยนตำแหน่งค่าพิกัด GPS ดังที่ได้กล่าวไว้ในบทที่ 9 ซึ่งเครื่องมือ Dalvik Debug Monitor Server (DDMS) ถูกสร้างขึ้นมาเพื่อสนับสนุนการทำงานประมวลผลนี้

นอกจากนี้เครื่องมือนี้ยังสามารถจัดการ SMS และจัดการไฟล์ต่างๆ ของ Emulator ทั้งสองเครื่องได้อีกด้วย โดยมีขั้นตอนการเรียกใช้งานเครื่องมือ DDMS ดังนี้

1. เปิดเครื่อง Emulator ขึ้นมา หรือถ้าในกรณีที่ไม่ได้สร้างไว้ให้ผู้อ่านสร้าง AVD ขึ้นมาก่อน จากนั้น จึงรัน AVD
2. คลิกเมนู Window > Open Perspective > DDMS เพื่อเรียกใช้งาน DDMS (Dalvik Debug Monitor Server)

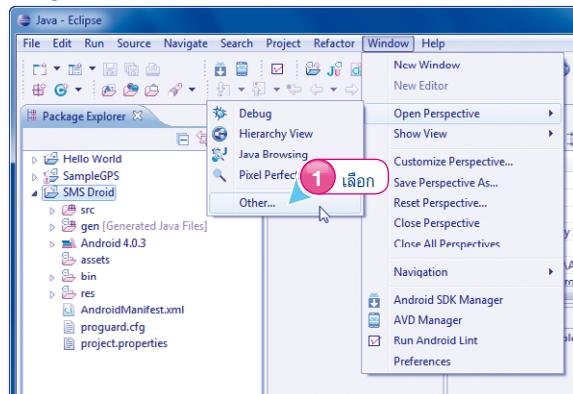


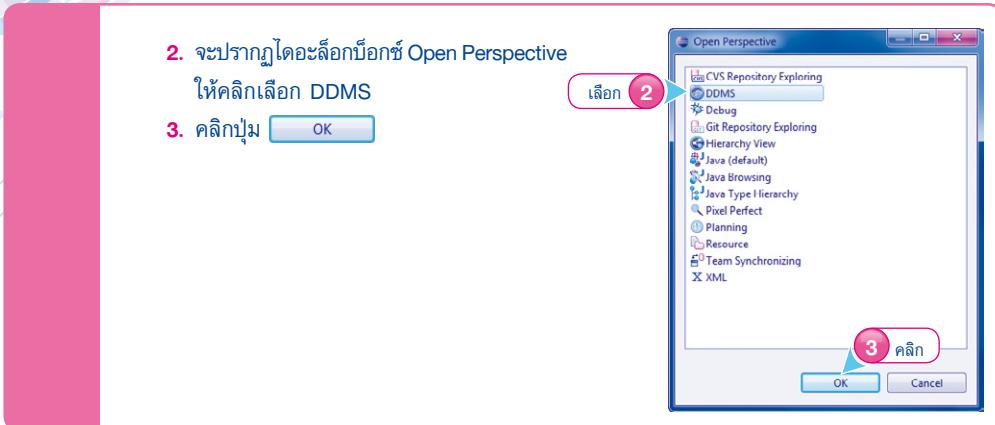
## NOTE



ในกรณีที่ผู้อ่านไม่พบเมนูดังข้อ 2 ให้ทำตามขั้นตอนดังนี้

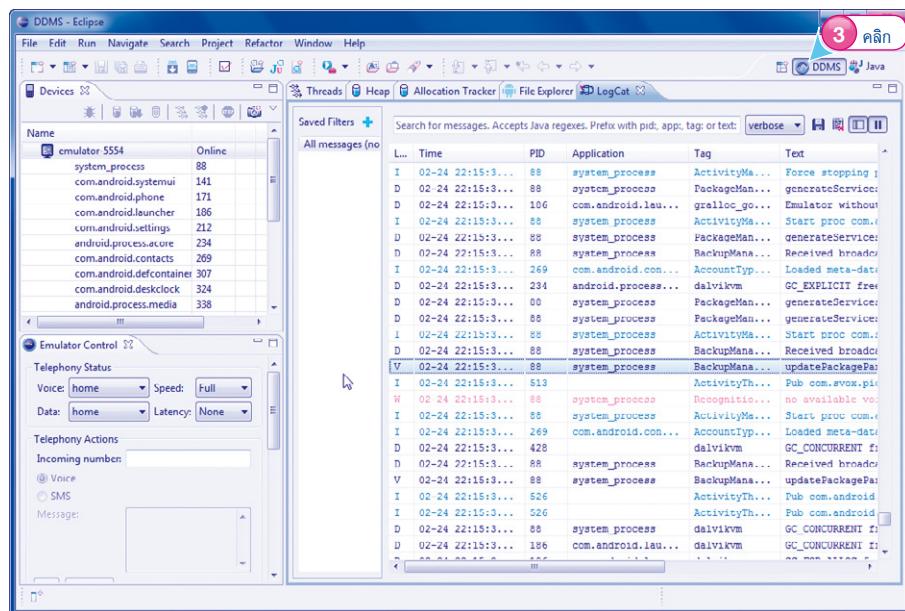
1. คลิกเมนู Window > Open Perspective > Other...





2. จะปรากฏ dialogue ออกมือชี้ Open Perspective  
ให้คลิกเลือก DDMS
3. คลิกปุ่ม **OK**
3. ผู้อ่านสามารถเรียกใช้งาน DDMS ได้จากการคลิกปุ่ม **DDMS** ที่มุมบนขวา มีอักษรของหน้าต่างโปรแกรมได้ เช่นเดียวกันดังรูป

#### 4. จะปรากฏหน้าต่าง DDMS



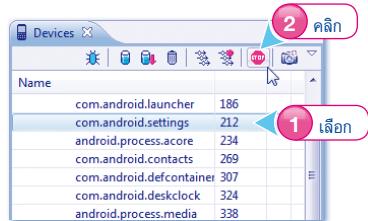
เมื่อผู้อ่านเปิด DDMS และ Emulator ให้ทำงานเรียบร้อยแล้ว เราสามารถเปลี่ยนการทำงานของ DDMS ออกเป็น 3 ล่วงคือ

- แท็บ (Tab) Devices ทางด้านซ้ายบน
- แท็บ Emulator Control ทางด้านซ้ายล่าง
- แท็บ LogCat ทางด้านขวา มือทั้งนี้ในล่วนของ LogCat จะอธิบายในหัวข้อถัดไป

## Devices

Devices เป็นเครื่องมือที่ช่วยแสดงรายละเอียดของอุปกรณ์ ซึ่งสามารถกำหนดลักษณะต่างๆ ให้กับ Process ที่ทำงานอยู่บน Emulator นั้นๆ ได้ เช่น

- การปิด Process ที่ทำงานอยู่ ซึ่งสามารถปิดได้โดยคลิกเลือก Process ที่ต้องการปิด จากนั้น คลิกปุ่ม ที่หน้าต่าง Devices
- การจับภาพหน้าจอจากเครื่องโทรศัพท์เคลื่อนที่จริง (เมื่อมีการเชื่อมต่อ) ทั้งนี้เครื่องมือนี้จะกล่าวถึง อีกครั้งในบทที่ 16

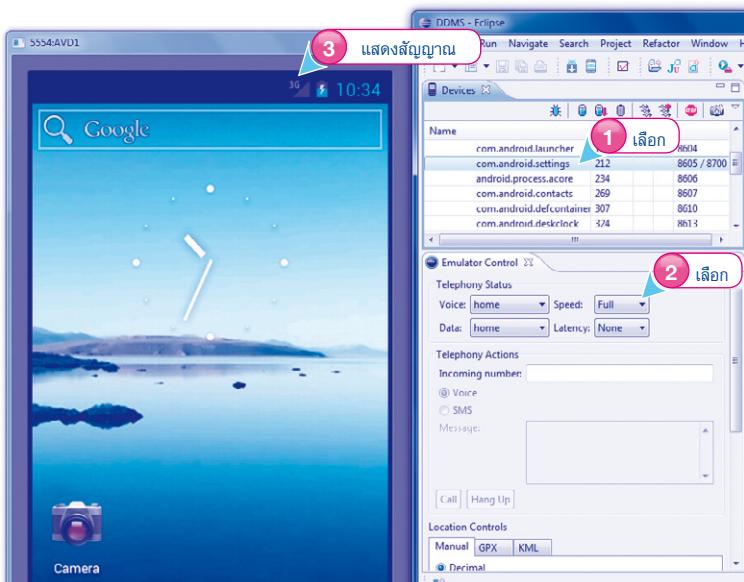


## Emulator Control

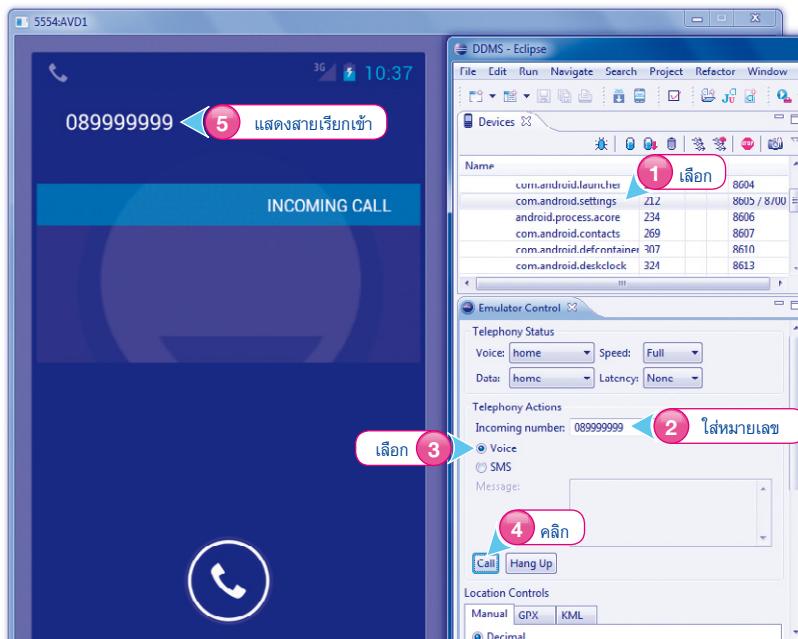
Emulator Control เป็นเครื่องมือเสริมที่ช่วยให้เครื่อง Emulator มีความสมบูรณ์เหมือนโทรศัพท์จริงมากยิ่งขึ้น เช่น สามารถจำลองเหตุการณ์ที่สายเรียกเข้าได้, จำลองเหตุการณ์รับข้อความ SMS ได้ เป็นต้น ในหัวข้อนี้จะกล่าวถึงเฉพาะส่วนที่มีการใช้งานบ่อยครั้งเท่านั้น โดยจะมีรายละเอียดของเครื่องมือหลักๆ ที่สำคัญดังนี้

- Telephony Status** เครื่องมือนี้ถูกใช้สำหรับควบคุมสถานะต่างๆ ของ Emulator เช่น
  - Voice: การปรับแต่งสถานะของลัญญาณโทรศัพท์
  - Speed: การปรับแต่งความเร็วของระบบเครือข่าย
  - Data: การปรับแต่งสภาวะการณ์เมื่อมีการส่งข้อมูลของเครื่องโทรศัพท์
  - Latency: การกำหนดรูปแบบของค่าหน่วงเวลา (Delay) ในการส่งข้อมูล

ทั้งนี้จะสังเกตได้ว่าเมื่อมีการปรับแต่งสถานะต่างๆ แล้ว ระบบจะมีการเปลี่ยนแปลงในส่วนของแถบแสดงสถานะ (Status bar) ที่อยู่บนโทรศัพท์จำลองอีกด้วย

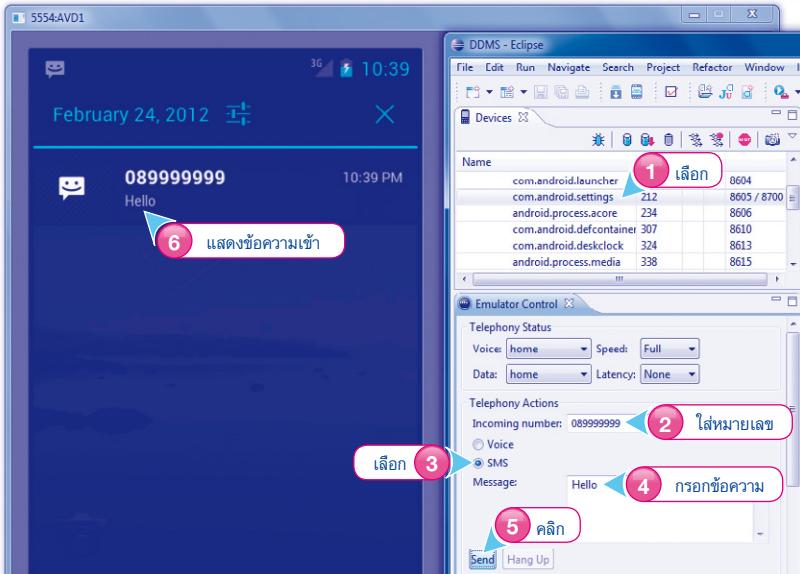


- **Telephony Actions** เครื่องมือนี้ถูกใช้สำหรับการจำลองเหตุการณ์เมื่อมีสายเรียกเข้า ซึ่งสามารถใช้งานได้ดังนี้
  - คลิกเลือก Emulator
  - ใส่หมายเลขโทรศัพท์ที่กล่องข้อความ Incoming number:
  - คลิกเลือก Voice
  - คลิกปุ่ม Call
  - Emulator แสดงสายเรียกเข้า



นอกจากนี้ผู้อ่านยังสามารถจำลองการส่งข้อความได้ เช่นเดียวกัน ซึ่งสามารถใช้งานได้ดังนี้

- คลิกเลือก Emulator
- ใส่หมายเลขโทรศัพท์ที่กล่องข้อความ Incoming number:
- คลิกเลือก SMS
- กรอกข้อความที่ต้องการส่ง
- คลิกปุ่ม Send
- Emulator แสดงข้อความเข้า



- **Location Controls** เครื่องมือนี้ถูกใช้สำหรับจำลองการเปลี่ยนตำแหน่งพิกัด GPS ซึ่งจะใช้จำลองสำหรับแอพพลิเคชันที่มีการใช้งานระบบ GPS เท่านั้น ซึ่งมีวิธีใช้งานดังนี้

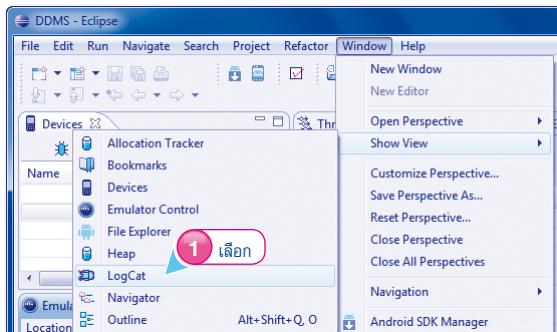
1. คลิกเลือก Decimal
2. ใส่ค่าพิกัดลองจิจูด (Longitude) และละติจูด (Latitude)
3. คลิกปุ่ม Send



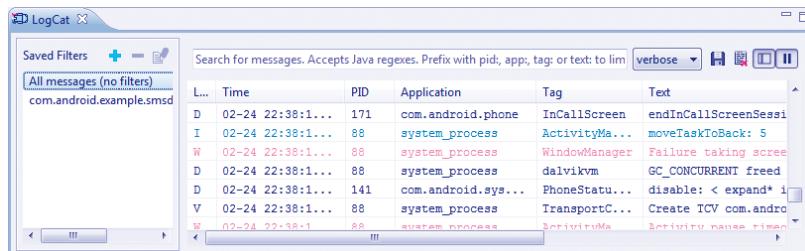
## การใช้งาน LogCat

สำหรับการพัฒนาแอพพลิเคชันนั้นมักอยู่ครั้งที่นักพัฒนาประสบกับปัญหา หรือข้อผิดพลาด (Error) ดังนั้น Android SDK ได้ผนวกเอาเครื่องมือสำหรับช่วยค้นหาข้อผิดพลาดที่เกิดขึ้นเข้ามาไว้ใน DDMS ด้วย ซึ่งเรียกเครื่องมือนี้ว่า **LogCat** (อ่านว่า “ล็อกแแคท”) โดยผู้อ่านสามารถเปิดใช้งาน LogCat ได้ดังนี้

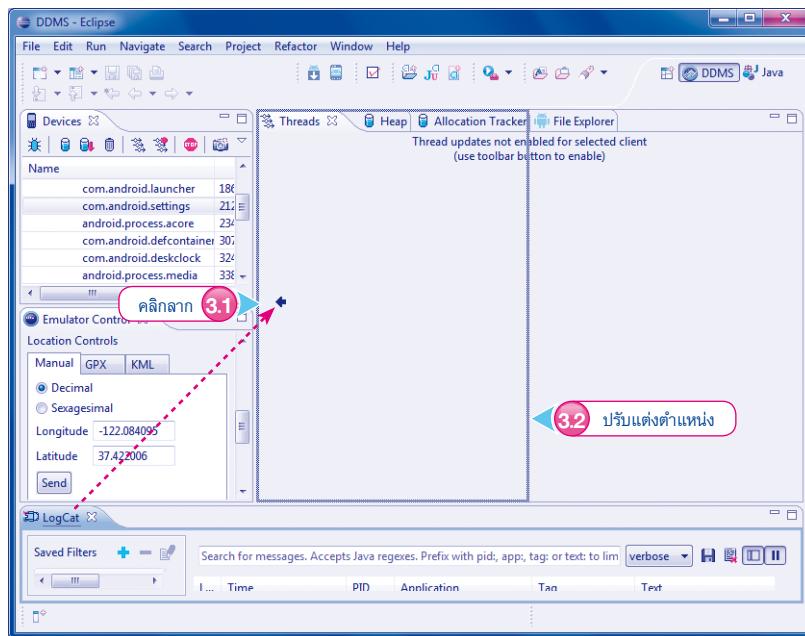
1. คลิกเมนู Window > Show View > LogCat



## 2. จะประภากหน้าต่าง LogCat ดังรูป



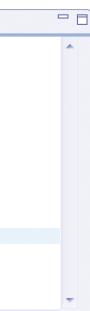
3. LogCat จะถูกเปิดขึ้นมาที่หน้าต่างด้านล่างของ DDMS ซึ่งผู้อ่านสามารถปรับแต่งตำแหน่งและขนาดของเครื่องมือได้ โดยคลิกลากและปล่อย (Drag and Drop) ที่หน้าต่างไปยังตำแหน่งที่ต้องการ



ทั้งนี้ LogCat นั้นสามารถสนับสนุนการตรวจสอบข้อมูลต่างๆ ได้โดยการเขียนล็อก (Log) และในส่วนนี้นักพัฒนาสามารถตรวจสอบการทำงานของแอพพลิเคชันได้โดยง่ายคือ การสร้างล็อกเพิ่มในโปรแกรม Java โดยใช้คลาส Log ซึ่งมีรูปแบบคำสั่งดังนี้

บันทึกของ Log	คำสั่ง	สีตัวอักษร
DEBUG	Log.d(<tag>,<message>);	น้ำเงิน
ERROR	Log.e(<tag>,<message>);	แดง
INFO	Log.i(<tag>,<message>);	เขียว
VERBOSE	Log.v(<tag>,<message>);	ดำ
WARN	Log.w(<tag>,<message>);	ล้วม

โดยมีตัวอย่างการเขียนล็อก เพื่อตรวจสอบการทำงานของแอพพลิเคชันในกรณีที่มีการวนลูปทำซ้ำดังนี้

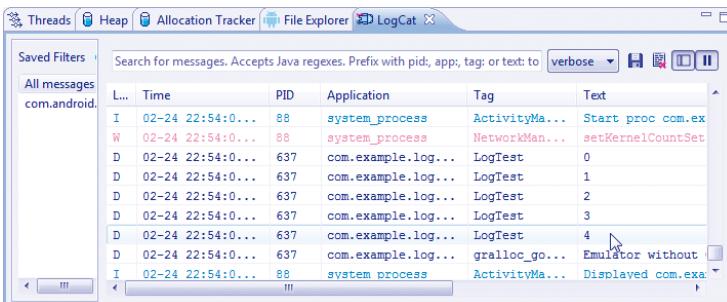


```

1 package com.example.logtest;
2
3 import android.app.Activity;
4
5 public class LogMainActivity extends Activity {
6
7     final static String TAG = "LogTest";
8
9     @Override
10    public void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.main);
13        for(int i=0;i<5;i++){
14            Log.d(TAG, String.valueOf(i));
15        }
16    }
17 }

```

จากนั้นเมื่อรันแอพพลิเคชัน ผู้อ่านสามารถเปลี่ยนหน้าต่างไปยังหน้าต่าง LogCat ซึ่งจะปรากฏหน้าจอแสดงค่า n ที่เพิ่มขึ้นในการวนซ้ำแต่ละรอบดังรูป

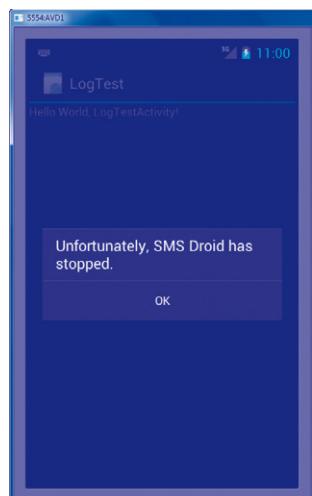


L...	Time	PID	Application	Tag	Text
I	02-24 22:54:0...	88	system_process	ActivityMa...	Start proc com.ex...
W	02-24 22:54:0...	88	system_process	NetworkMan...	setKernelCountSet
D	02-24 22:54:0...	637	com.example.log...	LogTest	0
D	02-24 22:54:0...	637	com.example.log...	LogTest	1
D	02-24 22:54:0...	637	com.example.log...	LogTest	2
D	02-24 22:54:0...	637	com.example.log...	LogTest	3
D	02-24 22:54:0...	637	com.example.log...	LogTest	4
D	02-24 22:54:0...	637	com.example.log...	gralloc_go...	Emulator without
I	02-24 22:54:0...	88	system process	ActivityMa...	Displayed com.ex...

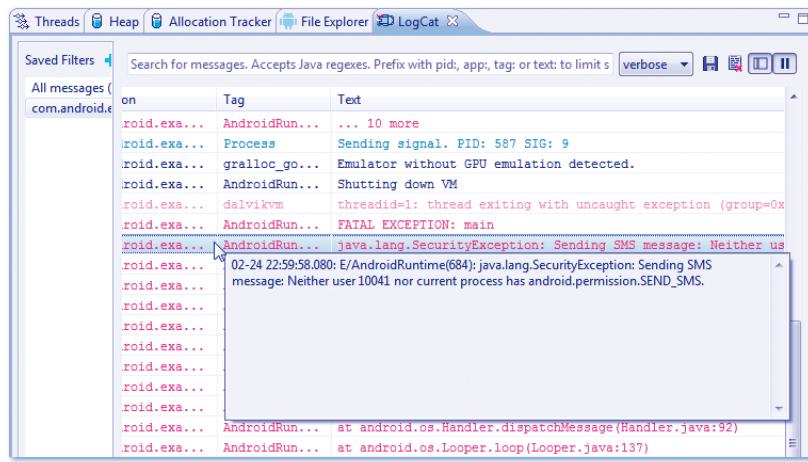
## การตรวจสอบข้อผิดพลาดของแอพพลิเคชัน โดยการอ่านค่า Log

หลังจากที่ผู้อ่านได้เรียนรู้เรื่องวิธีการสร้าง Log เรียบร้อยแล้ว ในหัวข้อนี้ผู้อ่านจะได้เรียนรู้ถึงวิธีการอ่านค่า Log สำหรับการแก้ไขข้อผิดพลาดของแอพพลิเคชันที่อาจจะเกิดขึ้นได้ ทั้งนี้ผู้เขียนจะแสดงถึงตัวอย่างที่มีข้อผิดพลาดในเบื้องต้นดังต่อไปนี้

**กรณีตัวอย่าง** กรณีที่นักพัฒนาไม่ได้เพิ่มคำสั่งการอนุญาต (Permission) หรือการทำหนัดลิทฟ์ให้กับแอพพลิเคชัน เช่น ตัวอย่างแอพพลิเคชันการส่ง SMS ที่ไม่ได้กำหนดลิทฟ์ “permission SEND\_SMS” ที่ไฟล์ AndroidManifest.xml เมื่อรันแอพพลิเคชันก็จะเกิดข้อผิดพลาดซึ่งแสดงได้ดังรูป



หลังจากนั้นผู้อ่านจะสามารถเข้าไปตรวจสอบความผิดพลาด ที่เกิดขึ้นได้จากกล่องเครื่องมือ LogCat โดยจะเห็นข้อผิดพลาด ซึ่งแสดงได้ดังรูป

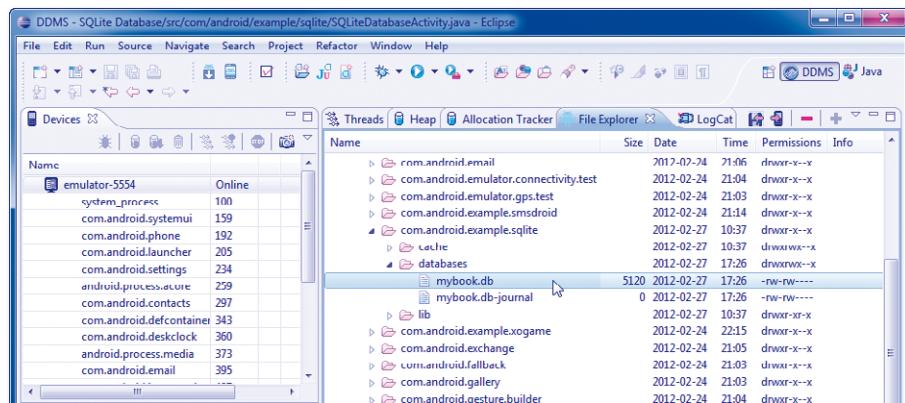


ในกรณีนี้ข้อผิดพลาดดังกล่าวแสดงให้เห็นว่า แอพพลิเคชันที่ได้สร้างขึ้นนั้นไม่ได้รับการให้สิทธิ์สำหรับการส่ง SMS

ทั้งนี้จำกตัวอย่างข้างต้นเป็นเพียงการตรวจสอบข้อผิดพลาดเบื้องต้นเท่านั้น อย่างไรก็ตามสำหรับการพัฒนาแอพพลิเคชันขั้นสูงนั้น การแก้ไขข้อผิดพลาดต่างๆ จะต้องใช้ประสบการณ์มากเลยทีเดียว

## ค้นหาตำแหน่งของไฟล์ด้วย File Explorer

นอกจากเครื่องมือ LogCat ที่ผูกเข้ากับ Android SDK แล้ว เครื่องมืออีกประเภทหนึ่งที่ใช้ในการค้นหา (Search) ตำแหน่งของไฟล์คือ หน้าต่าง File Explorer ซึ่งแสดงได้ดังรูป



ตัวอย่างนี้จะเป็นการค้นหาไฟล์ที่สร้างขึ้นจากตัวอย่างในบทที่ 11 ซึ่งคือไฟล์ mybook.db

## TIPS



ในบางครั้งเมื่อพัฒนาแอปพลิเคชันเสร็จลิ้นแล้ว อาจจะเกิดข้อผิดพลาด ซึ่งแสดงในรูปแบบต่างๆ หรือแม้แต่ไม่สามารถรัน Emulator ได้ ดังนั้น วิธีที่อาจจะช่วยแก้ปัญหาได้ก็คือ

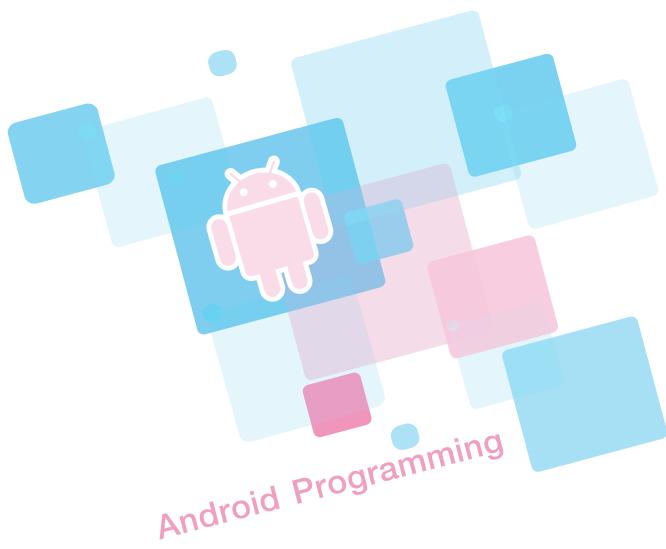
- คลิกเมนู Project และตามด้วย Clean และลอง Build โปรเจกต์อีกครั้งหนึ่ง
- หรือบางครั้งผู้อ่านสามารถปิด Emulator และเปิดขึ้นมาใหม่อีกครั้ง รวมไปถึงปิดโปรแกรม Eclipse และเปิดขึ้นมาใหม่อีกครั้งหนึ่งด้วยเช่นกัน
- หรือบางครั้งอาจจะต้องใช้ DDMS ในการ Reset เครื่องมือเอเดบี (adb)
- หรือบางครั้งต้องเข้าสู่ Command Prompt และพิมพ์ "adb kill-server"
- หรือบางครั้งต้องเข้าสู่ Command Prompt และพิมพ์ "adb uninstall com.kku.app" ทั้งนี้ com.kku.app จะเปลี่ยนไปตามโปรเจกต์ที่นักพัฒนาเปิดใช้งานอยู่

## สรุปภัยบ

ในบทนี้เป็นการอธิบายถึงเทคนิคการแก้ไขปัญหาสำหรับการพัฒนาแอปพลิเคชัน ด้วยเครื่องมือหลัก 3 ประเภทคือ Emulator Control (Telephony Actions และ Location Controls), LogCat และ File Explorer

## แบบฝึกหัดภัยบ

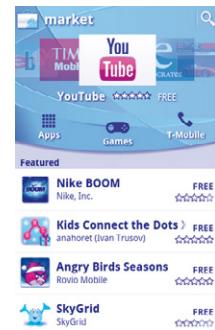
หลังจากผู้อ่านได้ศึกษาการใช้งานเครื่องมือที่ช่วยในการหาข้อผิดพลาดของแอปพลิเคชันแล้ว แบบฝึกหัดในบทนี้ให้ผู้อ่านจำลองข้อผิดพลาดของแอปพลิเคชัน เช่น ไม่ได้กำหนดลิขิท์ Permission INTERNET ให้กับแอปพลิเคชันการลงข้อมูลบนเครือข่าย (บทที่ 10) จากนั้นให้ผู้อ่านระบุบรรทัดที่แจ้งเตือนข้อผิดพลาด



## การอัพโหลดแอพพลิเคชันเข้า Android Market

Android Market เป็นแพลตฟอร์มร้านค้าแอพพลิเคชันออนไลน์ของระบบ Android โดยบริษัท Google มีลักษณะการทำงานคล้ายคลึงกับ AppStore ของบริษัท Apple หรือ AppWorld ของค่าย BlackBerry นั่นเอง

ในการเข้าไปดาวน์โหลดแอพพลิเคชันต่างๆ ใน Android Market เหล่านี้ได้นั้น จะต้องใช้โทรศัพท์เคลื่อนที่ระบบ Android เท่านั้น โดยเข้าถึง Android Market ผ่านแอพพลิเคชันบนเครื่องโทรศัพท์เคลื่อนที่หรือแท็บเล็ตที่ใช้ระบบปฏิบัติการ Android ที่ชื่อว่า Market



แอพพลิเคชันใน Android Market นั้นจะถูกแบ่งออกเป็น 2 ส่วนหลักๆ คือ

- แอพพลิเคชันทั่วไป (General Application)
- เกม (Game)

ซึ่งทั้ง 2 ส่วนนี้ยังถูกแบ่งเป็นลักษณะรูปแบบการดาวน์โหลดได้ 2 รูปแบบคือ ไม่มีค่าใช้จ่าย (Free) และแบบมีค่าใช้จ่าย

เนื้อหาในบทนี้ จะเป็นการอธิบายถึงขั้นตอนการนำแอพพลิเคชันที่ผู้อ่านพัฒนาขึ้นไปไว้ยัง Android Market ได้ โดยเนื้อหาในบทนี้ผู้เขียนจะแบ่งออกเป็น 3 ขั้นตอนคือ

- การจัดเตรียมไฟล์ต่างๆ สำหรับอัพโหลดแอพพลิเคชัน
- การอัพโหลดแอพพลิเคชันเข้า Android Market
- การอัพเดตเวอร์ชัน (Version) ของแอพพลิเคชัน

# การจัดเตรียมไฟล์ต่างๆ สำหรับอัพโหลดแอพพลิเคชัน

ในการอัพโหลดแอพพลิเคชันเข้าใน Android Market นั้นจะประกอบไปด้วยไฟล์ 2 ส่วนคือ

- ไฟล์ประเททรูภาพหรือไฟล์วิดีโอ
- ไฟล์ .apk สำหรับติดตั้งแอพพลิเคชัน

## การจัดเตรียมไฟล์รูปภาพและวิดีโอสำหรับอัพโหลดแอพพลิเคชัน

ให้ผู้อ่านจัดเตรียมไฟล์รูปภาพสำหรับอัพโหลดแอพพลิเคชัน ดังนี้

- ภาพตัวอย่างหน้าจอ โดยกำหนดความกว้างและความสูงเป็น 320x480, 480x800, 480x854, 128x720 หรือ 1280x800 และมีนามสกุลไฟล์เป็น PNG หรือ JPEG ขนาด 24 บิต ซึ่งผู้อ่านสามารถใช้เครื่องมือ Screen Capture ที่อยู่ใน DDMS ช่วย Capture รูปภาพได้
- ไอคอน (ICON) แอพพลิเคชันความละเอียดสูง โดยกำหนดความกว้างและความสูงเป็น 512x512 และมีนามสกุลไฟล์เป็น PNG หรือ JPEG ขนาด 32 บิตที่มีขนาดสูงสุด 1024 KB
- กราฟิกโปรโมทเป็นป้ายแบนเนอร์ (Banner) และบน Market App ความกว้าง 180xความสูง 120 และมีนามสกุลไฟล์เป็น PNG หรือ JPEG ขนาด 24 บิต
- กราฟิกแบนเนอร์แสดงที่เว็บไซต์ <http://market.android.com> โดยกำหนดความกว้างและความสูง เป็น 1024x500 และมีนามสกุลไฟล์เป็น PNG หรือ JPEG ขนาด 24 บิต
- วิดีโอโปรโมท ลิงค์วิดีโอที่อัพโหลดไว้ที่ YouTube

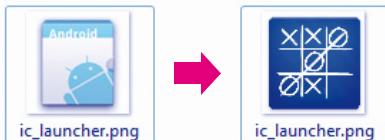


◀ ดาวน์โหลด  
Android Market

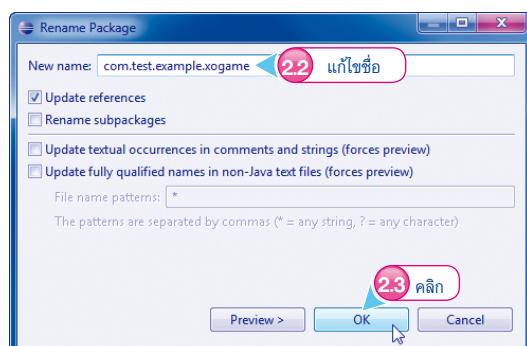
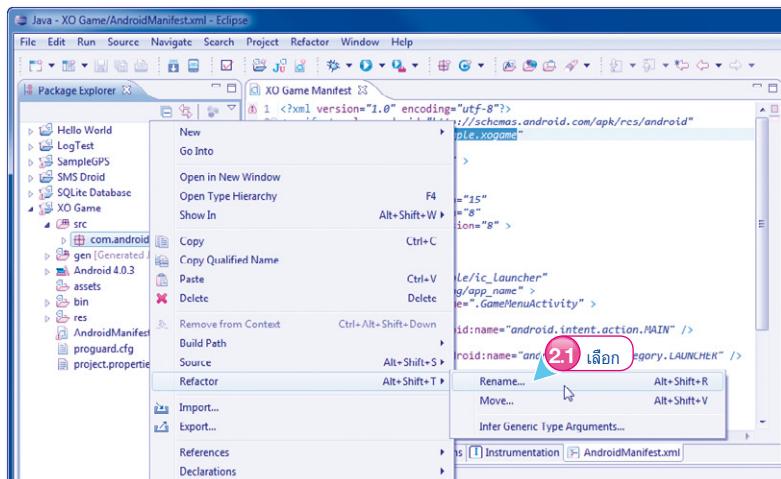
## การส่งออกไฟล์ .apk สำหรับอัพโหลดแอพพลิเคชัน

หลังจากที่ผู้อ่านได้จัดเตรียมไฟล์รูปภาพสำหรับอัพโหลดแอพพลิเคชันเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการส่งออกไฟล์ .apk ซึ่งมีขั้นตอนดังนี้ ดังนี้

- เปลี่ยนรูปไอคอน (Icon) ของแอพพลิเคชันให้เรียบร้อย ในที่นี่ผู้เขียนจะเปลี่ยนรูปไอคอนเกม XO ที่พัฒนาในบทที่ 12



- ตรวจสอบชื่อแพ็คเกจของโปรเจกต์ จะต้องไม่ชื่นตันด้วย com.android, com.google, android, com.example ในที่นี่ผู้เขียนจะเปลี่ยนชื่อแพ็คเกจจาก com.android.example.xogame เป็น com.test.example.xogame





3. ตรวจสอบการกำหนด `versionCode` และ `versionName` ในไฟล์ `AndroidManifest.xml` ทั้งนี้ `versionCode` จะเป็นตัวเลขระบุเวอร์ชันของโค้ด ซึ่งผู้อ่านจะต้องรับไปเรื่อยๆ เมื่อมีการอัปโหลดไฟล์ `.apk` เวอร์ชันใหม่ และในส่วนของ `versionName` นั้นจะเป็นตัวบอกเวอร์ชันสำหรับแสดงใน Android Market

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.test.example.xogame"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="8" >
    </uses-sdk>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity android:name=".GameMenuActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
        <activity android:name=".GamePlayerActivity" >
        </activity>
        <activity
            android:name=".XOGameActivity"
            android:screenOrientation="portrait" >
            <activity
                android:name="com.test.example.xogame.GameOverActivity"
                android:theme="@android:style/Theme.Translucent.NoTitleBar" >
            </activity>
        </application>
</manifest>

```

4. กำหนดค่า API Level เพื่อรับรอง Android เวอร์ชันที่สนับสนุน โดยแก้ `maxSdkVersion` เวอร์ชันสูงสุดที่สนับสนุน `minSdkVersion` เวอร์ชันต่ำสุด และ `targetSdkVersion` ในไฟล์ `AndroidManifest.xml` ดังนี้

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.test.example.xogame"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:maxSdkVersion="15"
        android:minSdkVersion="8"
        android:targetSdkVersion="8" >
    </uses-sdk>

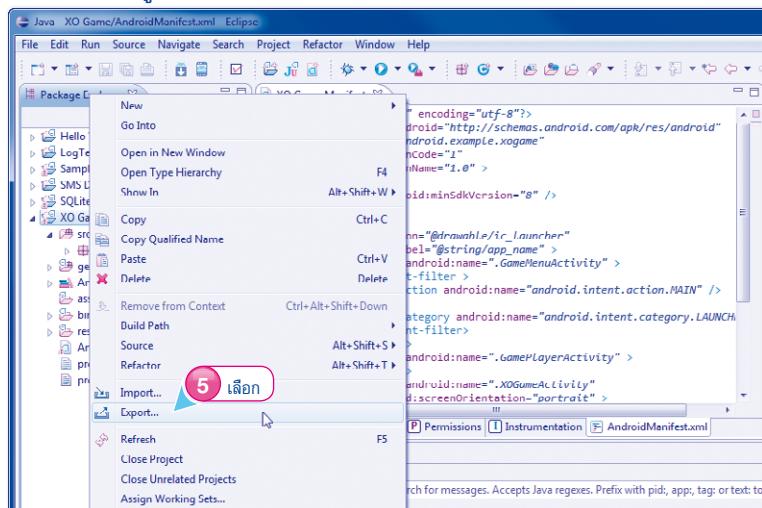
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity android:name=".GameMenuActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
        <activity android:name=".GamePlayerActivity" >
        </activity>
        <activity
            android:name=".XOGameActivity"
            android:screenOrientation="portrait" >
            <activity
                android:name="com.test.example.xogame.GameOverActivity"
                android:theme="@android:style/Theme.Translucent.NoTitleBar" >
            </activity>
        </application>
</manifest>

```

ผู้อ่านสามารถดู API Level ของ Android เวอร์ชันต่างๆ ได้จากตารางด้านล่างนี้

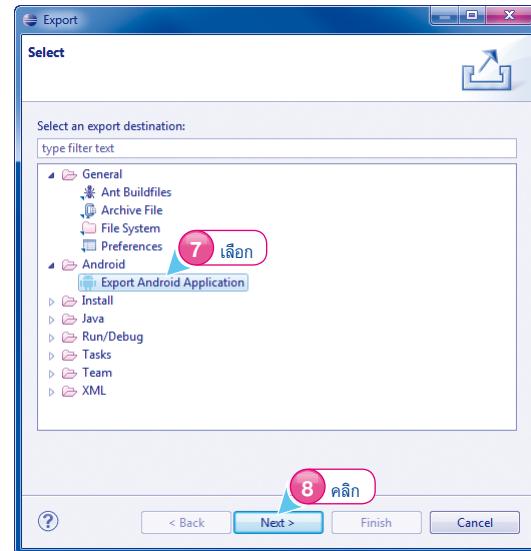
Platform Version	API Level	VERSION_CODE
Android 4.0.3	15	ICE_CREAM SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4	10	GINGERBREAD_MR1
Android 2.3.3		
Android 2.3.2	9	GINGERBREAD
Android 2.3.1		
Android 2.3		
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

5. คลิกขวาที่โปรเจกต์ แล้วเลือกคำสั่ง Export... ในตัวอย่างนี้จะเป็นการส่งออกของแอพพลิเคชันเกม XO ที่ได้ถูกพัฒนาขึ้นในบทที่ 12

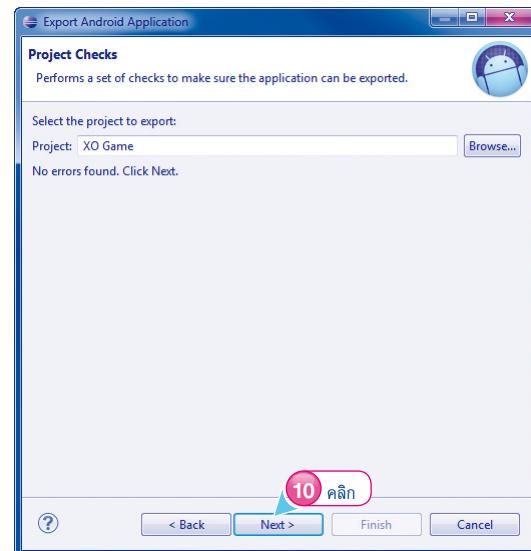




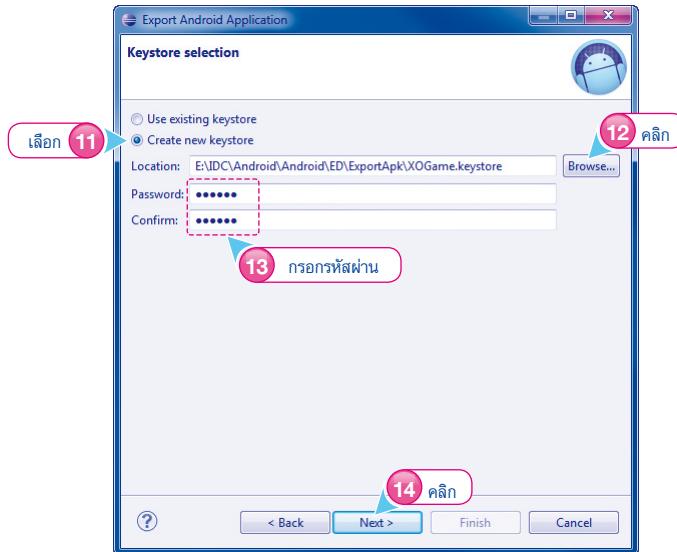
6. จะปรากฏได้จะเลือกนี่อ กซ์ Export ขึ้นมา
7. คลิกเลือก Export Android Application
8. คลิกปุ่ม Next >



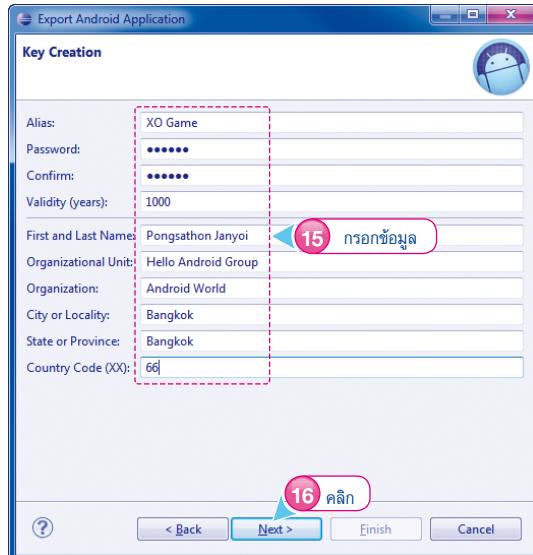
9. จะปรากฏได้จะเลือกนี่อ กซ์ Export Android Application ขึ้นมา
10. ตรวจสอบชื่อของโปรเจกต์ว่าถูกต้องหรือไม่ จากนั้นคลิกปุ่ม Next >



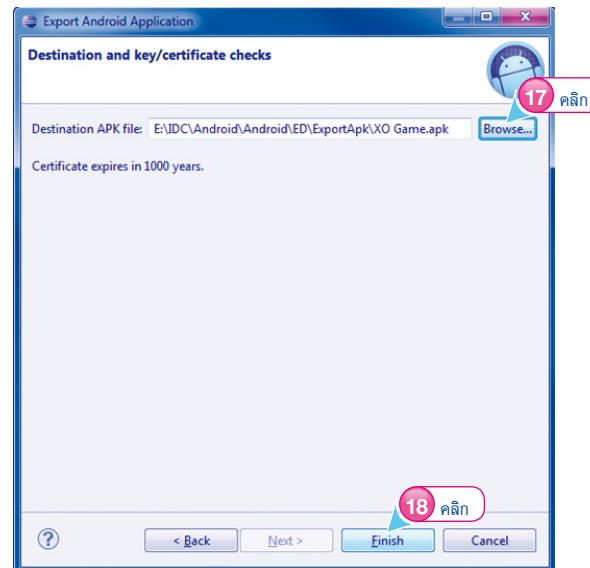
11. คลิกเลือก Create new keystore
12. คลิกปุ่ม **Browse...** เพื่อเลือกไดเร็คทอรีสำหรับจัดการเก็บไฟล์ Keystore
13. ในส่วนของ Password: ให้ใส่ Password ที่ต้องการและพิมพ์รหัสผ่านซ้ำในส่วนของ Confirm: โดยทั้งสองส่วนจะต้องตรงกัน
14. คลิกปุ่ม **Next >**



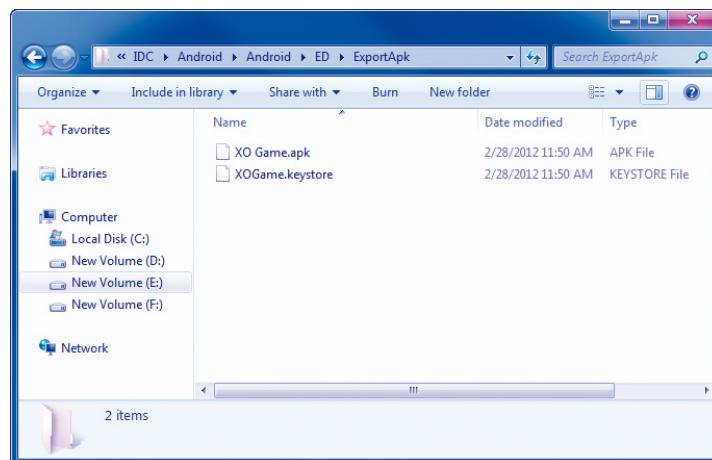
15. ให้ผู้อ่านใส่รายละเอียดต่างๆ ของผู้พัฒนาแอพพลิเคชัน ซึ่งจะถูกจัดเก็บอยู่ในไฟล์ Keystore ในที่นี้กรอกรายละเอียดต่างๆ ดังนี้
16. คลิกปุ่ม **Next >**



17. คลิกปุ่ม **Browse...** เพื่อเลือกไฟล์ .apk ที่จะถูกออก ในกรณีนี้คือ ไฟล์ XOGame.apk โดยจะถูกจัดเก็บอยู่ที่ไดเร็คทอรีของโปรเจกต์ ในที่นี่คือ E:\IDC\Android\Android\ED\ExportApk
18. คลิกปุ่ม **Finish**



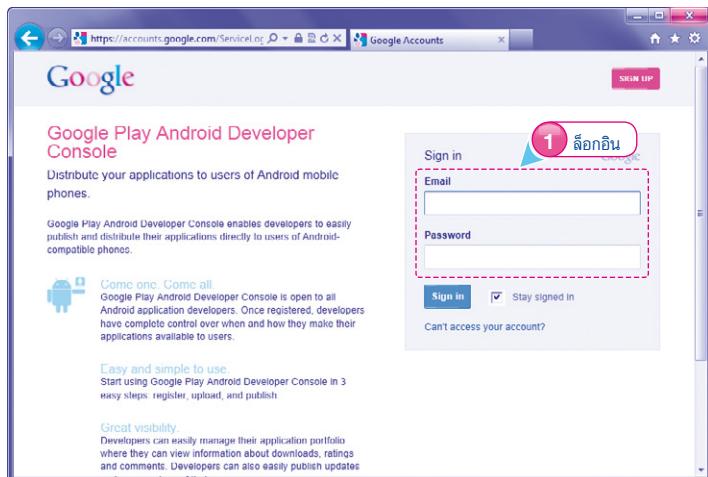
19. ในกรณีนี้การรับรองจะสามารถใช้ได้เป็นระยะเวลา 1,000 ปี
20. หลังจากที่ผู้อ่านได้สร้างไฟล์ Keystore และไฟล์ .apk เรียบร้อยแล้ว ผู้อ่านสามารถเข้าไปตรวจสอบในไดเร็คทอรีที่ระบุค่าไว้ในข้อ 12 โดยสังเกตได้ว่าจะปรากฏไฟล์ 2 ไฟล์คือไฟล์ XOGame.keystore และไฟล์ XO Game.apk ซึ่งผู้อ่านสามารถนำไฟล์ .apk นี้ไปอัพโหลดเข้า Android Market ได้



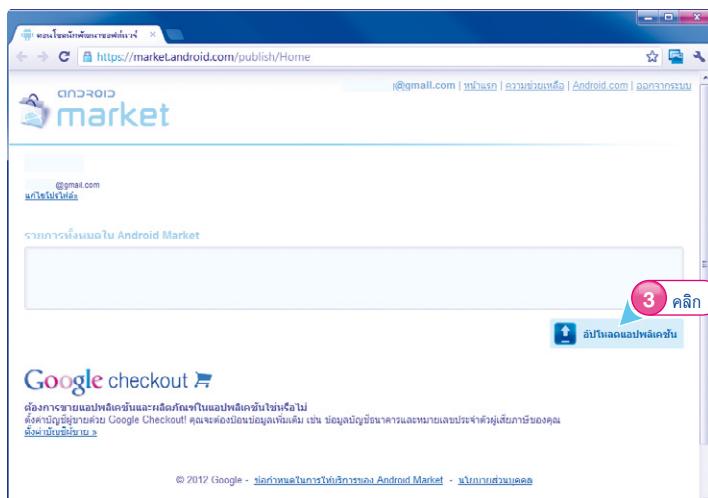
# การอัพโหลดแอพพลิเคชันเข้า Android Market

หลังจากที่ผู้อ่านเตรียมไฟล์รูปภาพต่างๆ และไฟล์ .apk เล็งจเรียบร้อยแล้ว ในส่วนนี้จะเป็นการนำไฟล์ที่เตรียมไว้ไปอัพโหลดเข้า Android Market ซึ่งผู้อ่านจะต้องลงทะเบียนเป็น Developer ก่อน โดยจะมีค่าใช้จ่าย \$25 สำหรับขั้นตอนการอัพโหลดแอพพลิเคชันเข้า Android Market มีดังนี้

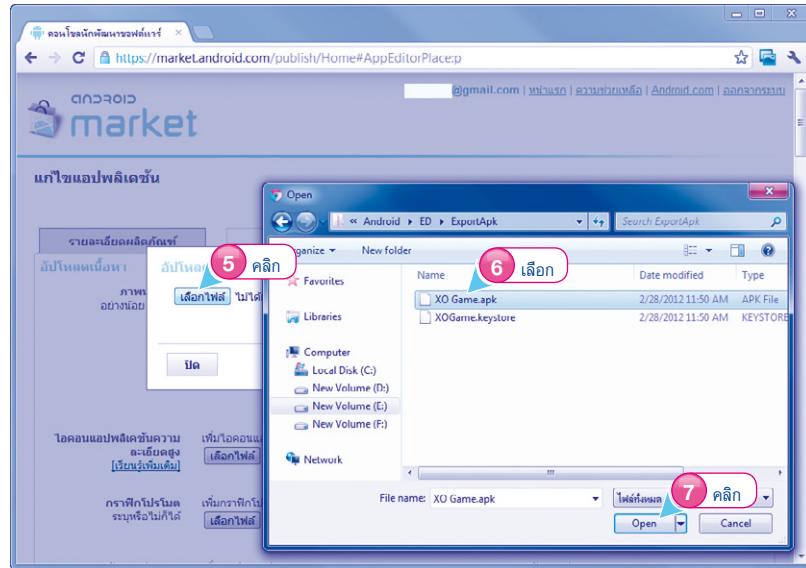
1. เปิดเบราว์เซอร์ <https://market.android.com/publish/> และล็อกอิน (Login) ในกรณีที่ไม่มีรหัสผ่านให้ลงทะเบียนก่อน



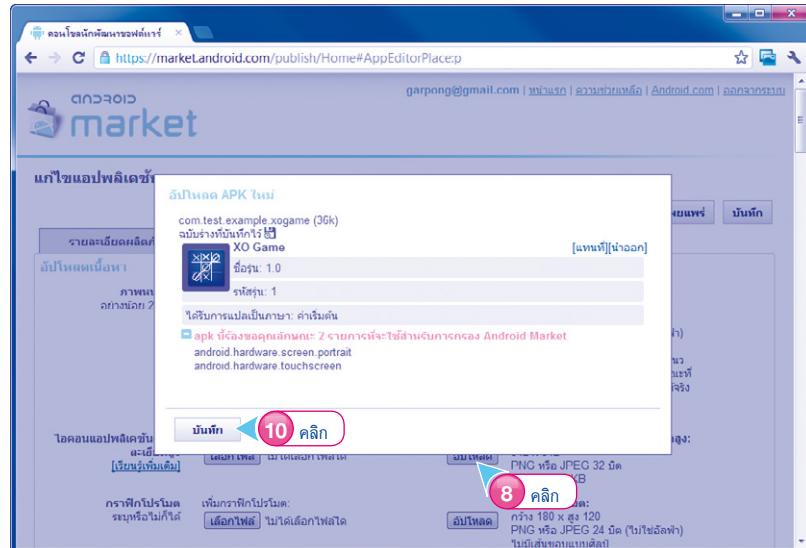
2. จะปรากฏหน้าหลัก (Home)
3. คลิกปุ่ม อัปโหลดแอพพลิเคชัน



4. จะปรากฏหน้าต่างสำหรับอัปโหลดไฟล์
5. คลิกปุ่ม **[เลือกไฟล์]**
6. คลิกเลือกไฟล์ XO Game.apk
7. คลิกปุ่ม **[Open ▾]**



8. คลิกปุ่ม **[อัปโหลด]**
9. จะปรากฏไดอะล็อกบ็อกซ์ อัปโหลด APK ใหม่
10. คลิกปุ่ม **[บันทึก]**



## 11. อัพโหลดรูปภาพที่ผู้อ่านเตรียมไว้ดังรูป



## 12. แก้ไขรายละเอียดต่างๆ ของแอปพลิเคชันดังรูป

The screenshot shows the 'market' application's edit page for the previously submitted item. The form includes the following fields:

- รายละเอียดรายการ**
  - ภาษา**: English (en)  
เพิ่มภาษา [เขียนลงในช่องเดียว]
  - Title (หัวข้อ)**: Tic Game
  - Description (อธิบาย)**: Tic Tac Toe for the Android platform lets you test your skills against the wits of the robot. Put three X-marks in a line horizontally, vertically, or diagonally before the computer can line up three O-marks and you win.  
220 วิรษะ (สูงสุด 4000)
  - Recent Changes (อัปเดต)**: bugs fixed  
220 วิรษะ (สูงสุด 4000)
  - Recent Changes (อัปเดต)**: bugs fixed  
10 วิรษะ (สูงสุด 500)
  - Promo Text (หัวข้อ)**: Tic Tac Toe for Android  
23 วิรษะ (สูงสุด 80)
  - ประเภทแอปพลิเคชัน**: เกม

### **13. กำหนดเงื่อนไขการเผยแพร่แอพพลิเคชันดังรูป**

#### 14. คลิกเพื่อยอมรับข้อตกลง

ชื่อบุคคลเด็ก

เงินเดือน

**คลิก 14**

อีเมล

 และเพิ่มเติมชื่อเป็นไปตามหลักเกณฑ์ที่ด้านหน้าของแบบร้อยด้วย

โทรศัพท์

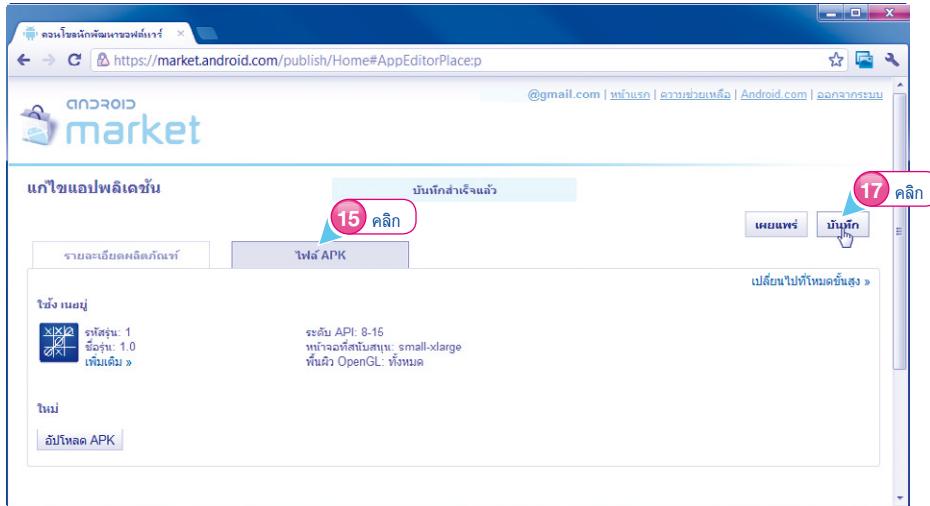
 ฉันทราบว่าและเพิ่มเติมชื่อที่ได้ระบุของฉันอาจถูกนำไปใช้ในการดำเนินการทางกฎหมาย

**บันทึก**

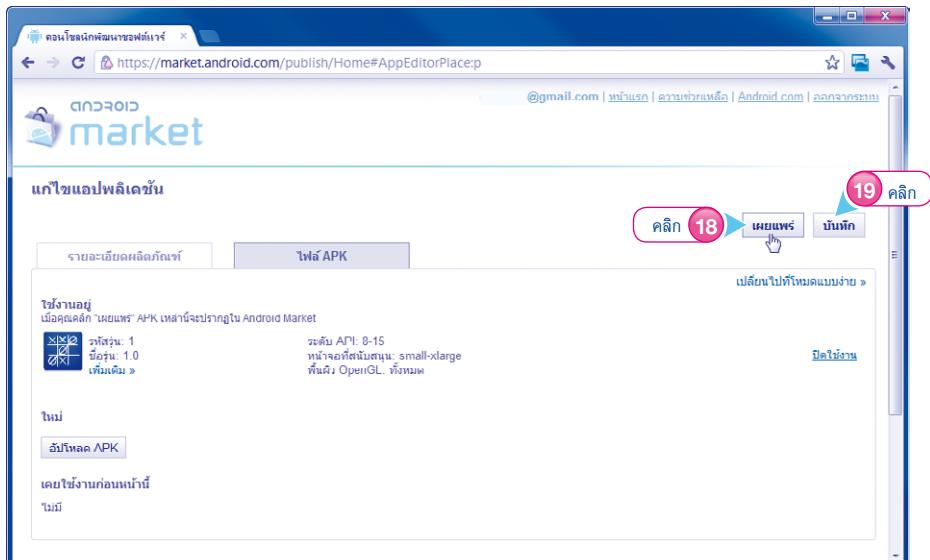
ฉันเห็นใจเด็กนี้เป็นไปตามหลักเกณฑ์ที่ด้านหน้าของแบบร้อยด้วย

ฉันทราบว่าและเพิ่มเติมชื่อที่ได้ระบุของฉันอาจถูกนำไปใช้ในการดำเนินการทางกฎหมาย

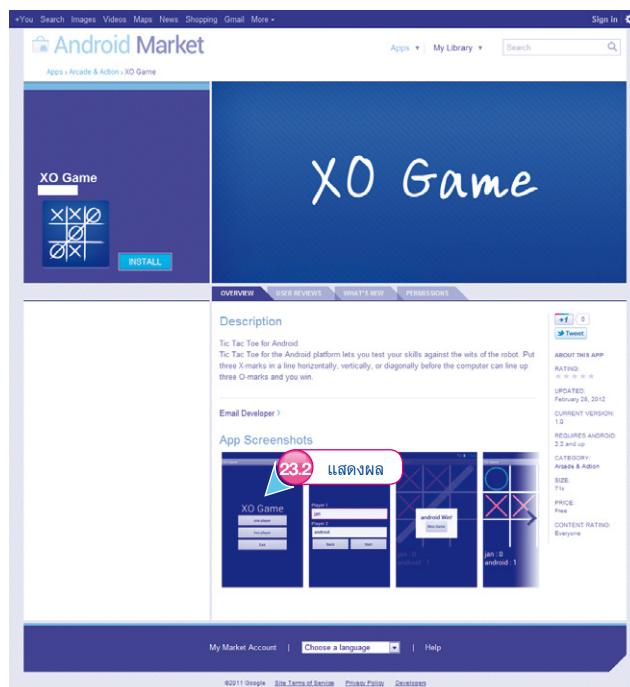
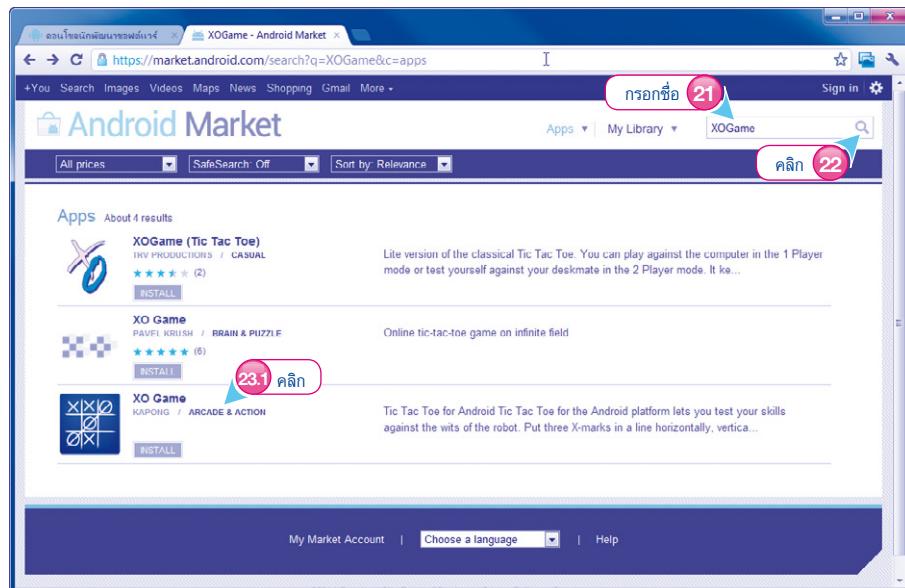
15. คลิกເທັນ ໄຟລ໌ APK
16. คลิกທີ່ລຶ່ງນີ້ ເປີດໃຊ້ງານ (ຈະປາກວູນໃນການອັພໂຫລດຮັງແຮກ)
17. ທັດການນັ້ນຄຸກປຸ່ມ ນັ້ນທິກ



18. คลิกປຸ່ມ ເໝຍພັກ ເພື່ອໃຫ້ເອພພລິເຄຊັນປາກວູນ Android Market
19. คลิกປຸ່ມ ນັ້ນທິກ ອີກຮັງເພື່ອບັນທຶກການເພີ່ມແພີ່ມ



20. รอประมาณ 10-30 นาที และเปิดเว็บไซต์ <http://market.android.com/> เพื่อตรวจสอบแอพพลิเคชันที่อัปโหลดเข้า Android Market
21. พิมพ์ XOGame ที่ช่องค้นหา
22. คลิกปุ่ม เพื่อค้นหาแอพพลิเคชัน
23. จะพบแอพพลิเคชันที่อัปโหลดเข้ามา ซึ่งก็คือ XO Game



เมื่อค้นหาใน Android Market Apps ในสมาร์ทโฟน ก็จะได้ผลลัพธ์เช่นเดียวกัน



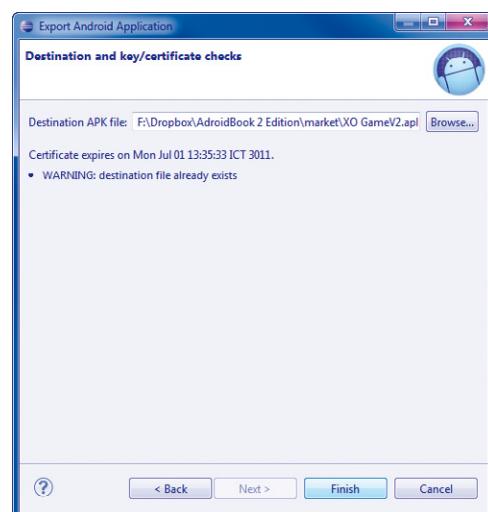
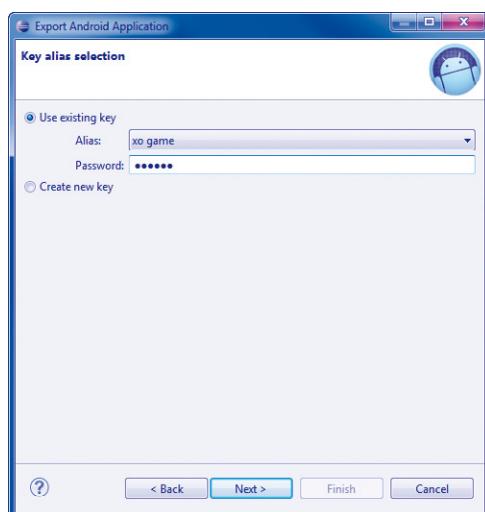
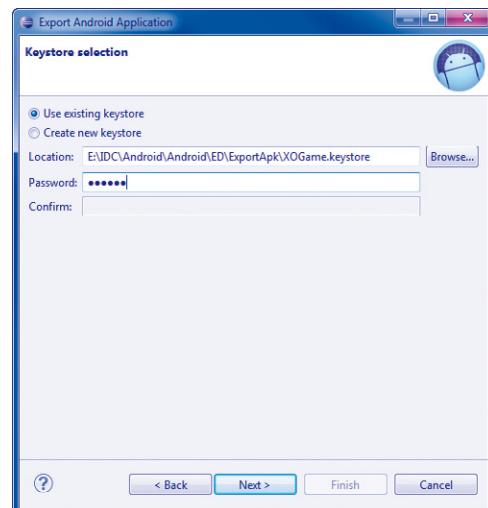
## การอัพเดตเวอร์ชันแอพพลิเคชัน

โดยปกติแล้วเมื่อมีการแก้ไขข้อผิดพลาด หรือมีการปรับปรุงแอพพลิเคชัน ผู้อ่านต้องมีการอัพเดตไฟล์ .apk ใหม่เข้าไปแทนที่เวอร์ชันเดิม ซึ่งมีขั้นตอนดังต่อไปนี้

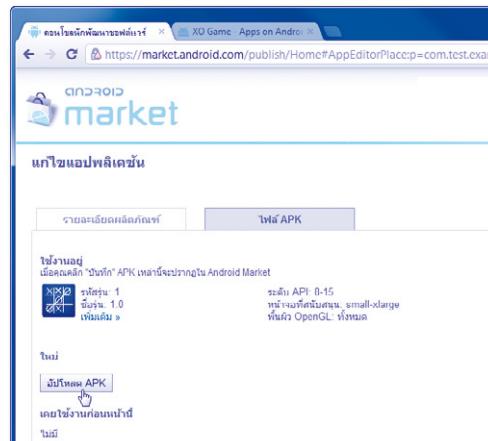
1. แก้ไข versionCode และ versionName ในไฟล์ AndroidManifest.xml ในที่นี่จะเปลี่ยน versionCode เป็น 2 และ versionName เป็น 1.1

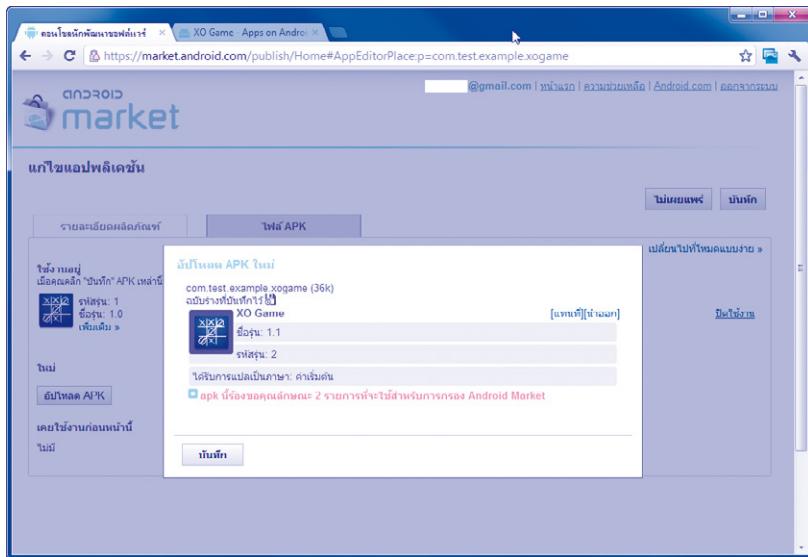
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
 package="com.test.example.xogame"
 android:versionCode="2"
 android:versionName="1.1" >

2. ให้ผู้อ่าน Export ไฟล์ .apk โดยเลือกไฟล์ Keystore เดิมที่เคยสร้างไว้ แล้ว ใส่รหัสผ่านเดิม (ตามขั้นตอนที่กล่าวไว้ ในหัวข้อการล่งออกไฟล์ .apk สำหรับ อัพโหลดแอพพลิเคชัน)

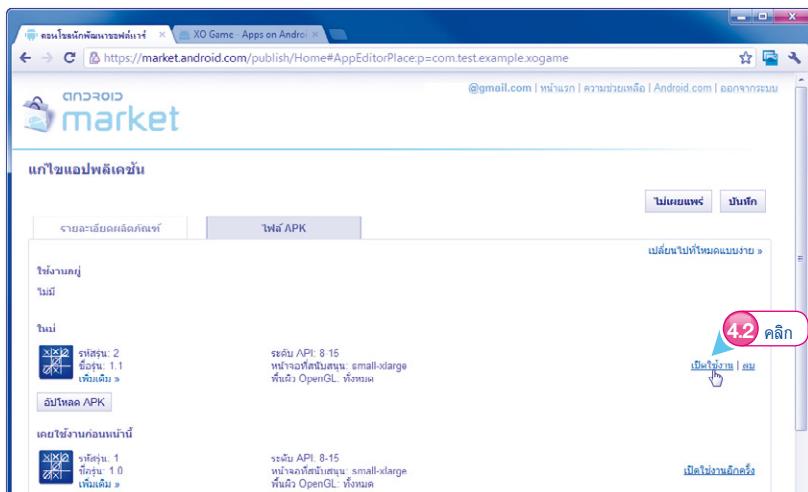
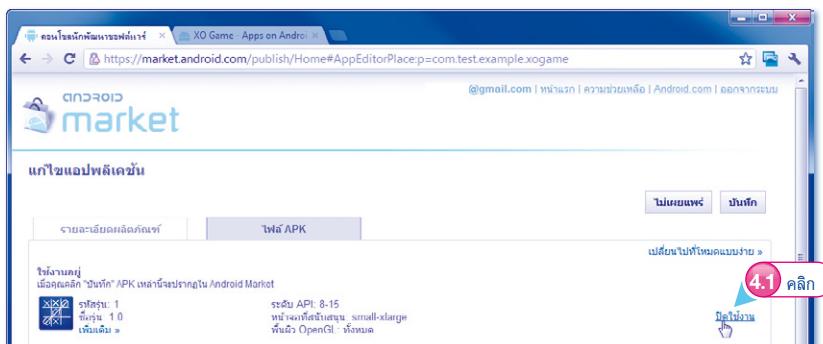


3. อัพโหลดไฟล์ .apk เวอร์ชันใหม่เพิ่มเข้าไป (ตามขั้นตอนที่กล่าวไว้ในหัวข้อ การอัพโหลดแอพพลิเคชันเข้า Android Market)

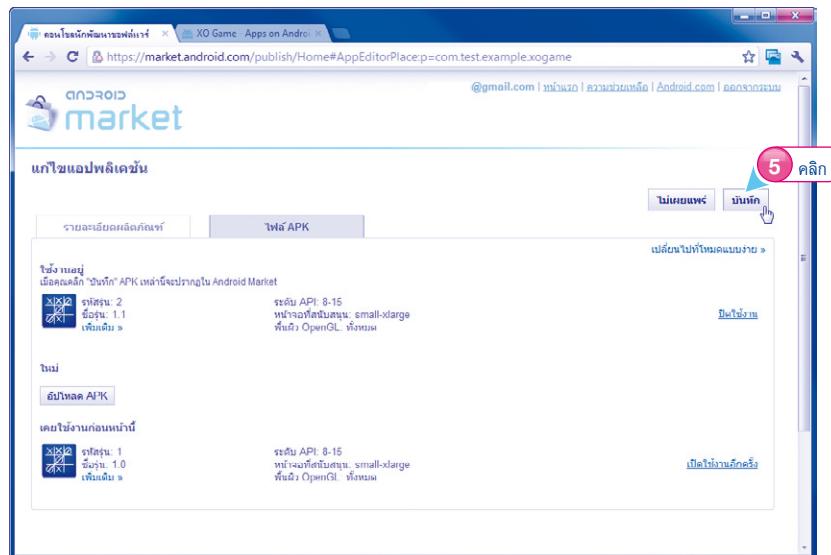




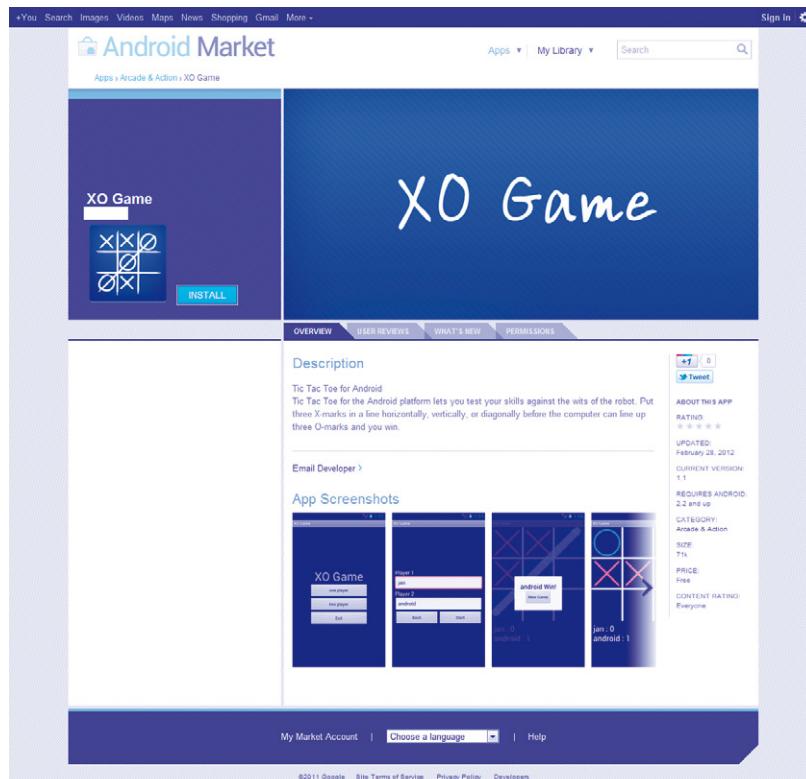
#### 4. ปิดการใช้งานไฟล์ APK เวอร์ชันเดิม และเปิดใช้งานไฟล์ APK เวอร์ชันใหม่



## 5. คลิกปุ่ม บันทึก เพื่อจบการอัพเดตเวอร์ชัน



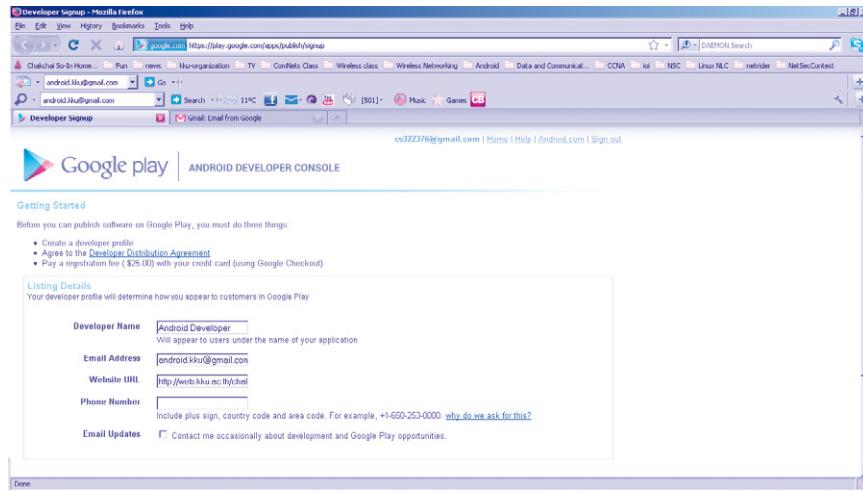
รอประมาณ 10-30 นาที แล้วลองเข้า Android Market ผู้อ่านก็จะเห็นเวอร์ชันของแอพพลิเคชันที่มีการเปลี่ยนแปลงดังรูป



## NOTE



สำหรับการใช้งานในอนาคต Google จะปรับเปลี่ยนจาก Market เป็น Google Play ซึ่งก็จะมีขั้นตอนการอัพโหลดที่คล้ายคลึงกัน เพียงแต่เปลี่ยนเป็นเว็บไซต์ <https://play.google.com/apps/publish/> โดยที่ผู้มีครั้งต้องเสียค่าบริการ \$25 ดังรูป



## สรุปก้ายบ

ในบทนี้เป็นการแนะนำขั้นตอนการอัพโหลดแอพพลิเคชันที่เราพัฒนาขึ้นเข้าไปใน Android Market เพื่อเผยแพร่แอพพลิเคชันในตลาดออนไลน์นั่นเอง

## แบบฝึกหัดก้ายบ

หลังจากผู้อ่านอัพโหลดแอพพลิเคชันเข้า Android Market เรียบร้อยแล้ว แบบฝึกหัดในส่วนนี้ให้ผู้อ่านทดลองอัพโหลดไฟล์ .apk เพิ่มเข้าไปใหม่เพื่อปรับเวอร์ชันเป็น 1.2



## การติดตั้งแอพพลิเคชันบนเครื่องโทรศัพท์มือถือ

ถึงแม้ว่าการพัฒนาโปรแกรมผ่าน Emulator จะมีความสะดวกและรวดเร็ว แต่สำหรับการสร้างแอพพลิเคชันบนประเภทนั้น การทดสอบโปรแกรมกับ Emulator ก็อาจจะทำให้เสียเวลาในการรันแต่ละครั้งเพิ่มขึ้น เนื่องจาก Emulator นั้นยังมีข้อจำกัดในหลายส่วน เช่น การติดต่อสื่อสารผ่าน Bluetooth และการใช้งานระบบ Sensor ดังนั้น อีกทางเลือกหนึ่งก็คือ การพัฒนาแอพพลิเคชันโดยตรงจากเครื่องโทรศัพท์เคลื่อนที่จริง ดังนั้น ในบทนี้จึงเป็นส่วนขยายเพิ่มเติม เพื่อช่วยให้ผู้อ่านมองเห็นถึงภาพการใช้งานจริง โดยสามารถนำแอพพลิเคชันที่ได้พัฒนาขึ้นในบทนี้แล้ว มาติดตั้งกับเครื่องโทรศัพท์เคลื่อนที่จริง โดยเป็นแนวทางให้กับผู้อ่านที่จะติดตั้งแอพพลิเคชันอื่นๆ ต่อไป

เนื้อหาในบทนี้ผู้เขียนจะอธิบายลักษณะการทำงานและการติดตั้ง โดยอ้างอิงกับเครื่องโทรศัพท์เคลื่อนที่ยี่ห้อ Samsung รุ่น Galaxy Nexus ที่มีระบบปฏิบัติการ Android 4.0.3 ติดตั้งมากับเครื่องอย่างไรก็ตามสามารถเลือกใช้โทรศัพท์รุ่นอื่นๆ ได้ ซึ่งก็จะมีขั้นตอนการติดตั้งโดยเฉพาะส่วนของโปรแกรมไดรเวอร์ (Driver) ที่มีการซื้อมต่อเข้ากับพอร์ต USB ที่แตกต่างกัน ทั้งนี้ผู้เขียนจะเน้นวิธีการนำเข้าหรือส่งออกไฟล์ .apk ไปยังเครื่องโทรศัพท์ผ่านทางสาย USB โดยแบ่งออกเป็น 3 ส่วนคือ

- การติดตั้งไดรเวอร์ของเครื่องโทรศัพท์ (Driver Installation)
- การส่งออกไฟล์ .apk
- การพัฒนาโปรแกรมผ่านทางเครื่องโทรศัพท์เคลื่อนที่จริง (Android Application Development via Mobile Device)

# การติดตั้งไดรเวอร์ของเครื่องโทรศัพท์ (Driver Installation)

การติดตั้งโปรแกรมไดรเวอร์ของโทรศัพท์เคลื่อนที่ยี่ห้อ Samsung Galaxy Nexus มีขั้นตอนดังๆ ดังต่อไปนี้

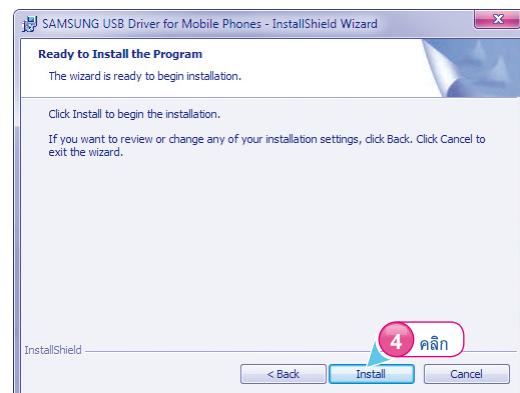


▲ โทรศัพท์เคลื่อนที่ชั้มชูงรุ่น Galaxy Nexus

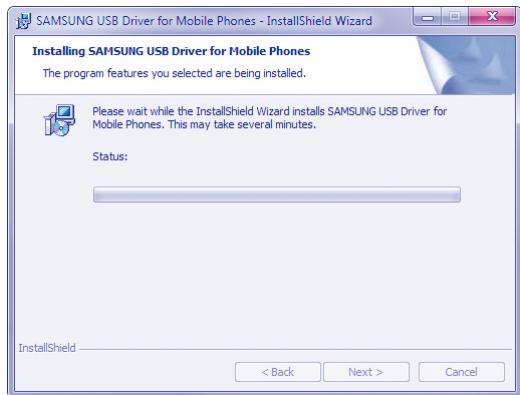
1. ดาวน์โหลดซอฟต์แวร์ Samsung\_Mobile\_USB\_Driver\_V1.3.800.0.msi ที่ใช้สำหรับการเชื่อมต่อโทรศัพท์เคลื่อนที่ยี่ห้อ Samsung Galaxy Nexus (หรือจาก CD ท้ายเล่ม)
2. ดับเบิลคลิกเพื่อติดตั้งซอฟต์แวร์
3. คลิกปุ่ม Next >



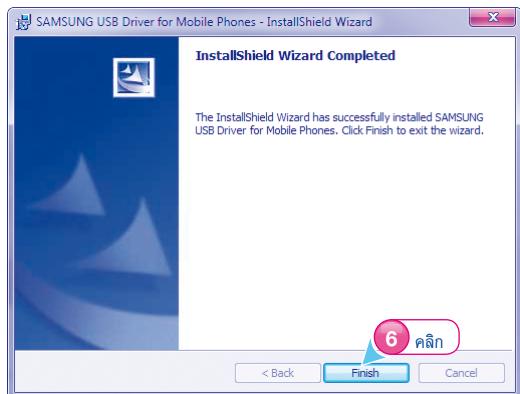
4. จะปรากฏหน้าจอสำหรับการติดตั้งโปรแกรม ให้คลิกปุ่ม Install



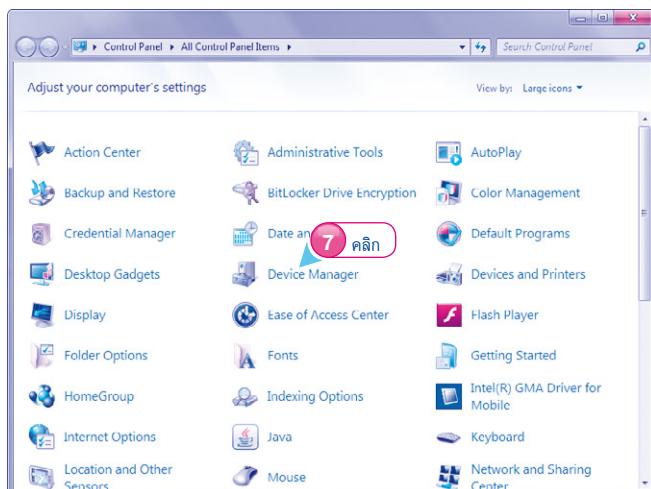
## 5. ดำเนินการติดตั้งโปรแกรม



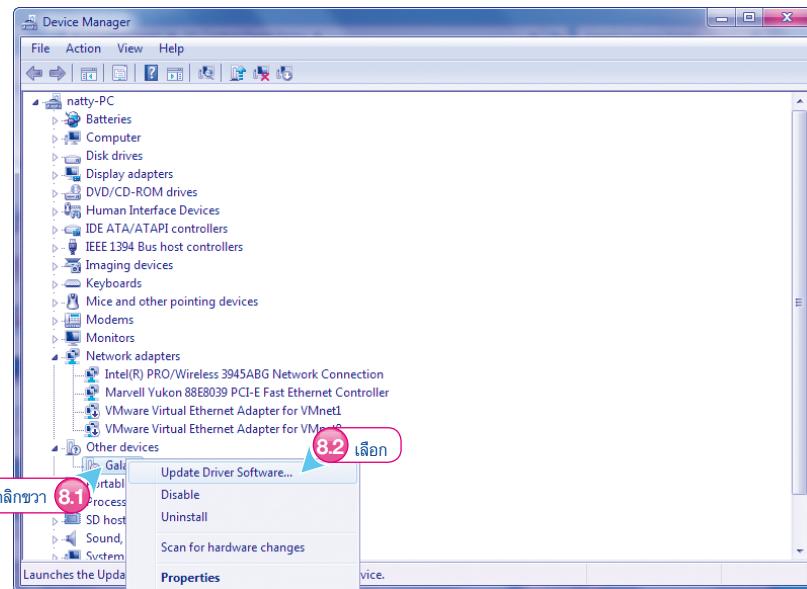
## 6. เมื่อติดตั้งเสร็จสมบูรณ์ ให้คลิกปุ่ม Finish



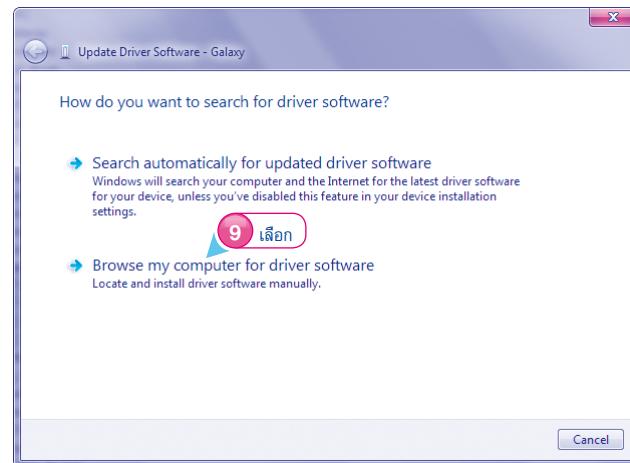
## 7. ให้ผู้อ่านเข้าไปที่ Control Panel และคลิกไอคอน Device Manager



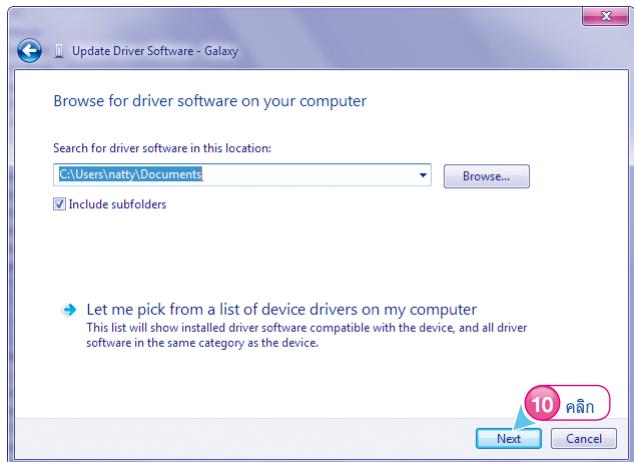
8. จะปรากฏหน้าต่าง Device Manager ขึ้นมา ให้คลิกขวาที่ Other devices > Galaxy > Update Driver Software... ดังรูป



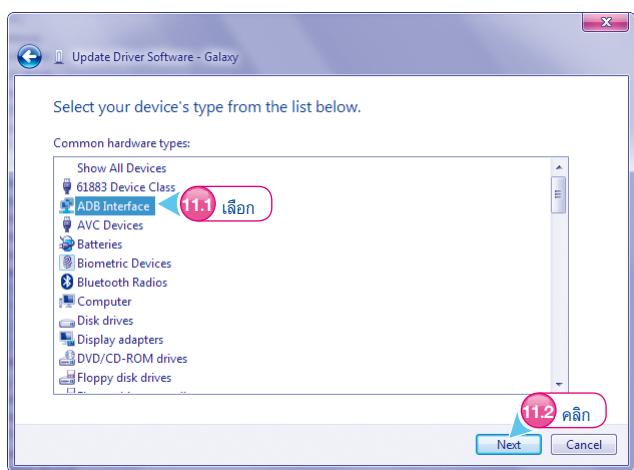
9. จะปรากฏหน้าต่าง Update Driver Software - Galaxy ให้คลิกที่ Browse my computer for driver software



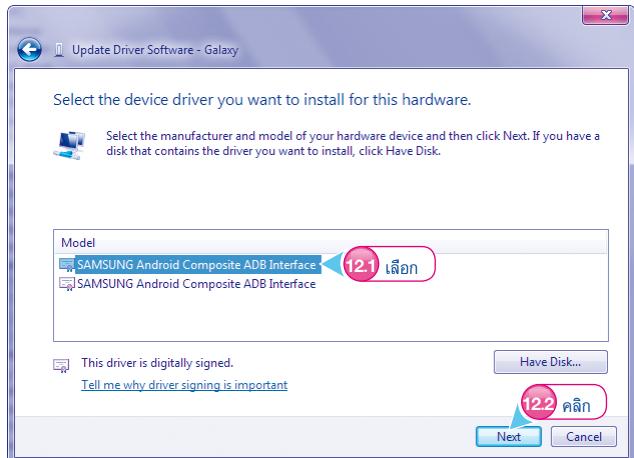
10. จะปรากฏหน้าต่างให้เลือกพาธในการติดตั้ง Driver Software ในที่นี่เลือกเป็นค่า Default แล้วให้คลิกปุ่ม **Next**



11. จะปรากฏหน้าต่างให้เลือกประเภทในการติดตั้ง Driver Software ในที่นี่เลือก ADB Interface และคลิกปุ่ม **Next**



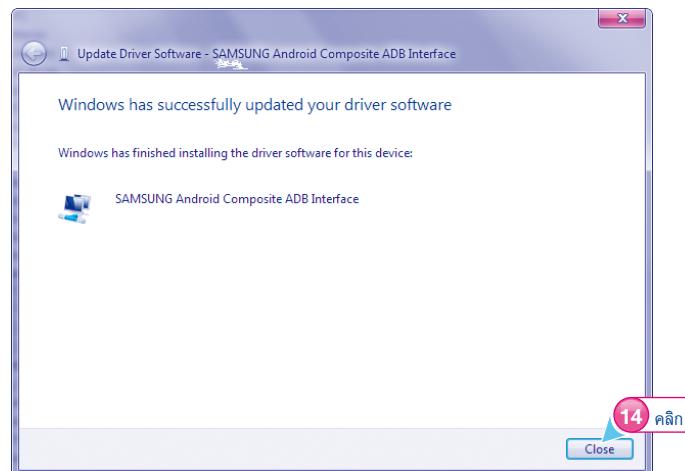
12. จะปรากฏหน้าต่างดังรูปให้คลิกเลือก SAMSUNG Android Composite ADB Interface และคลิกปุ่ม **Next**



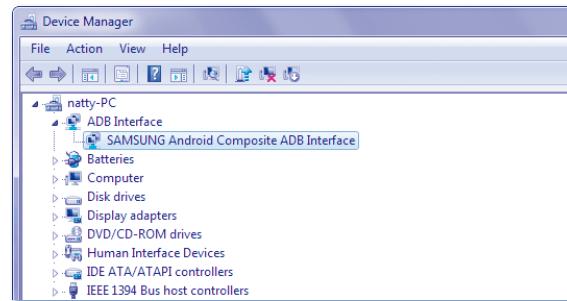
13. รอการดำเนินการติดตั้งโปรแกรมลักษณะ จะปรากฏหน้าต่างดังรูป แล้วคลิกปุ่ม Yes



14. เมื่อติดตั้งเสร็จสมบูรณ์ ให้คลิกปุ่ม Close



15. เมื่อติดตั้งโปรแกรมเสร็จสมบูรณ์ ผู้อ่านควรจะตรวจสอบการติดตั้งอีกครั้ง โดยดูที่หน้าต่าง Device Manager จะปรากฏ SAMSUNG Android Composite ADB Interface ขึ้นมา ถือว่าเป็นการเสร็จสมบูรณ์ในการติดตั้งไดรเวอร์ของโทรศัพท์เคลื่อนที่ Samsung Galaxy Nexus



# การส่งออกไฟล์ .apk

เนื้อหาในหัวข้อนี้จะเป็นการอธิบายถึงวิธีการส่งออก (Export) ไฟล์ .apk ซึ่งได้มาหลังจากที่ผู้อ่านคอมpile เอกพัลิเคชันเรียบร้อยแล้ว ทั้งนี้ไฟล์ .apk นั้นเปรียบเสมือนกับไฟล์ .exe บนระบบปฏิบัติการ Windows ซึ่งสามารถนำไปรันที่ Emulator หรือบนเครื่องโทรศัพท์เคลื่อนที่จริงก็ได้

ดังนั้น หัวข้อนี้จะเป็นการอธิบายขั้นตอนต่างๆ สำหรับการรันแอพพลิเคชันบน Emulator อีกทั้ง ซึ่งเนื้อหาในหัวข้อนี้จะแบ่งออกเป็น 3 ส่วนคือ

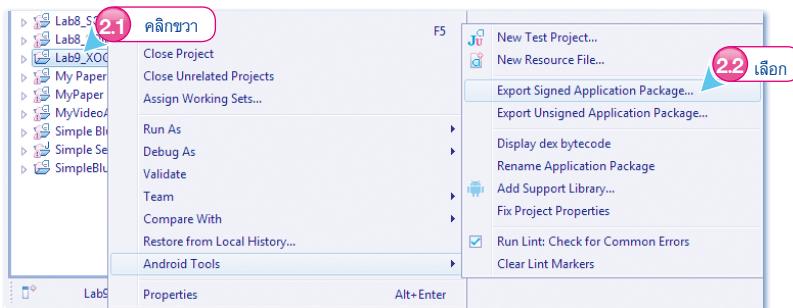
- การเตรียมการส่งออกไฟล์ .apk
- การนำไฟล์ .apk ไปติดตั้งบน Emulator
- การนำไฟล์ .apk ไปติดตั้งบนโทรศัพท์เคลื่อนที่จริง

## การเตรียมการส่งออกไฟล์ .apk

วิธีที่สำคัญที่สุดสำหรับการส่งออก (Export) ไฟล์ .apk ก็คือ การใช้เครื่องมือ ADT ซึ่งได้ผูกกับ Eclipse เรียบร้อยแล้ว โดยนำมาช่วยในการสร้างและส่งออกไฟล์ .apk ทั้งนี้ Eclipse จะเป็นส่วนที่ช่วยสร้าง Digital Certificate Signing ซึ่งเป็นส่วนเพิ่มเติมข้อมูลที่ระบุถึงผู้สร้างแอพพลิเคชันนั้นๆ

ในการนี้ที่ไม่มีการสร้าง Digital Certificate Signing ผู้อ่านก็จะไม่สามารถนำไฟล์ .apk ไปติดตั้งลงบนเครื่องอื่นๆ ได้ เพราะแอนดรอยด์จะปฏิเสธการติดตั้งแอพพลิเคชันที่ไม่ผ่านการรับรอง โดยที่มีขั้นตอนการรับรองดังต่อไปนี้

1. เปิดโปรแกรม Eclipse
2. คลิกขวาที่โปรเจกต์ แล้วเลือกคำสั่ง Android Tools > Export Signed Application Package... ในตัวอย่างนี้จะเป็นการส่งออกของแอพพลิเคชันเกม XO ที่ได้ถูกพัฒนาขึ้นในบทที่ 12

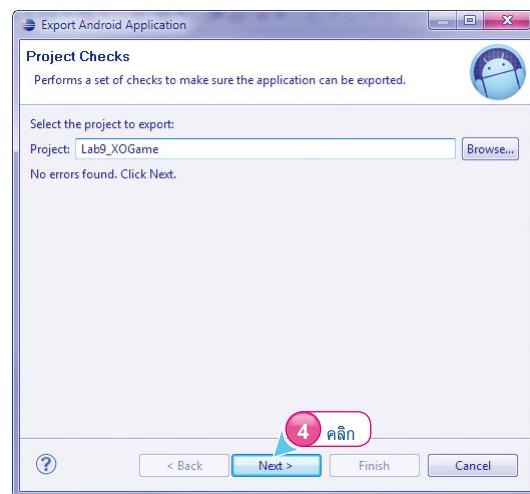




3. จะปรากฏ dialogue ของเลือกเบื้องต้น Export Android Application

4. ตรวจสอบชื่อของโปรเจกต์ว่าถูกต้องหรือไม่ จากนั้นคลิกปุ่ม

Next >

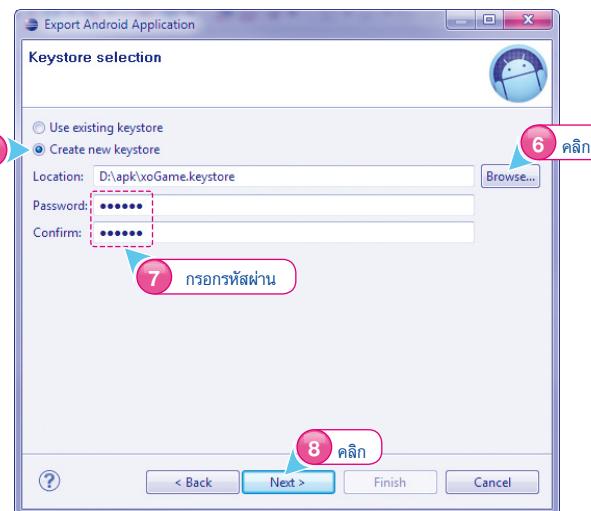


5. คลิกเลือก Create new keystore

6. คลิกปุ่ม **Browse...** เพื่อเลือกไดเร็คทอรีจัดการเก็บไฟล์ Keystore

7. ในส่วนของ Password: ให้ใส่ Password ที่ต้องการ และกรอกรหัสผ่านซ้ำในส่วนของ Confirm:  
โดยที่ทั้งสองส่วนจะต้องตรงกัน

8. คลิกปุ่ม **Next >**



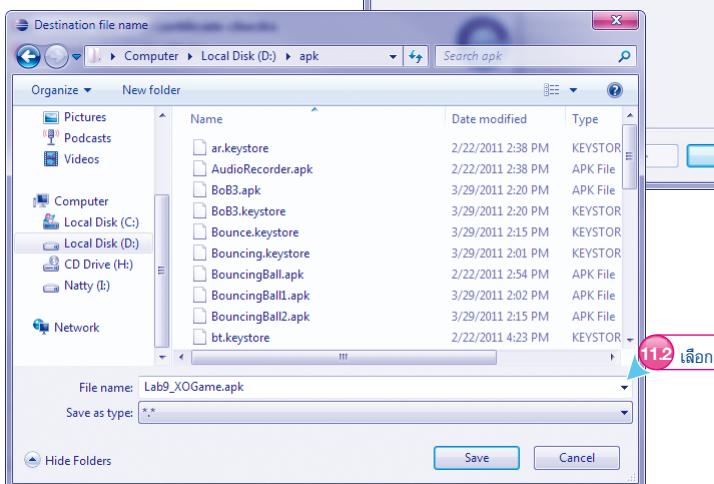
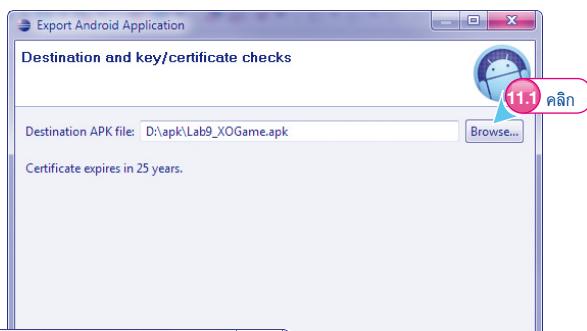
9. ให้ผู้อ่านใส่รายละเอียดต่างๆ ของผู้พัฒนาโปรแกรม ซึ่งจะถูกจัดเก็บอยู่ในไฟล์ Keystore ในที่นี้  
ให้กรอกรายละเอียดดังนี้

10. คลิกปุ่ม **Next >**



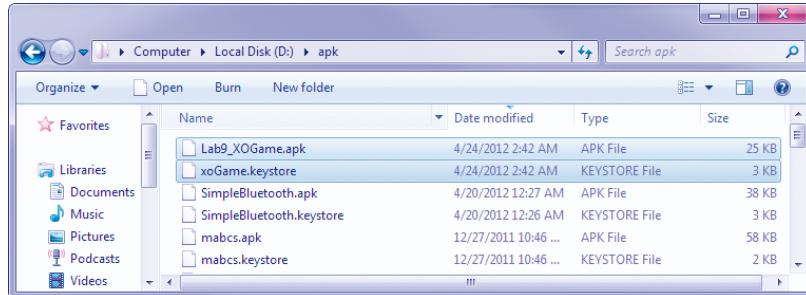
11. คลิกปุ่ม **Browse...** เพื่อเลือกไฟล์ .apk ที่จะส่งออก ในกรณีนี้เลือกไฟล์ Lab9\_XOGame.apk โดยให้บันทึกไว้ในที่นี่คือ D:\apk

12. คลิกปุ่ม **Finish**



ในกรณีนี้การรับรองจะสามารถใช้ได้เป็นระยะเวลา 25 ปี

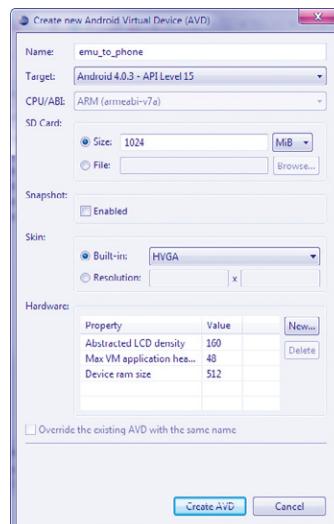
13. หลังจากที่ผู้อ่านได้รับไฟล์ Keystore และไฟล์ .apk เรียบร้อยแล้ว ผู้อ่านสามารถเข้าไปตรวจสอบในไดเรกทอรีที่ระบุคุ้มไว้ในข้อ 6 โดยลั่งเกตได้ว่าจะปรากฏไฟล์ 2 ไฟล์คือ ไฟล์ xoGame.keystore และไฟล์ Lab9\_XOGame.apk ซึ่งผู้อ่านสามารถนำไฟล์ .apk นี้ไปติดตั้งบน Emulator อื่นๆ หรือเครื่องโทรศัพท์เคลื่อนที่จริงได้ และสามารถอัปโหลดเข้า Android Market ได้เช่นกัน (วิธีการอัปโหลดผู้อ่านสามารถกลับไปทบทวนได้ในบทที่ 15)



## การนำไฟล์ .apk ไปติดตั้งบนเครื่องโทรศัพท์จำลอง หรือ อุปกรณ์เคลื่อนที่

หลังจากที่ผู้อ่านได้ไฟล์ .apk เรียบร้อยแล้ว ผู้อ่านสามารถนำไฟล์นี้ไปรันบน Emulator หรือโทรศัพท์เคลื่อนที่จริงได้ ดังนั้น ห้ามขอนี้จะกล่าวถึงการนำไฟล์ .apk ไปติดตั้งบน Emulator ซึ่งมีขั้นตอนดังนี้

1. เปิดเครื่อง Emulator หรือสร้างขึ้นใหม่ ในที่นี่สร้างโทรศัพท์จำลองใหม่โดยมีชื่อว่า emu\_to\_phone ซึ่งมีคุณสมบัติดังนี้



2. เปิด Command Prompt
3. ให้ผู้อ่านเข้าไปที่ไดเรกทอรี tools ซึ่งโดยปกติแล้วจะอยู่ในส่วนของ Android SDK ที่ไดติดตั้งไว้ในบทที่ 2

- พิมพ์คำสั่ง “adb install <.apk file>” ชี้ในกรณีนี้ <.apk file> คือ D:\apk ซึ่งเป็นที่อยู่ที่เลือกไว้ในหัวข้อที่ผ่านมา
- เมื่อติดตั้งเรียบร้อยแล้ว จะปรากฏผลลัพธ์ดังรูป

```

Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\natty>cd c:\android-sdk-windows\platform-tools
C:\android-sdk-windows\platform-tools>adb install D:\apk\Lab9_XOGame.apk
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
334 KB/s (24646 bytes in 0.072s)
pkg: /data/local/tmp/Lab9_XOGame.apk
Success

```

- จากนั้นผู้อ่านสามารถตรวจสอบการติดตั้งโปรแกรมที่ Emulator ได้ โดยเปิดหน้าจอแอพพลิเคชันรวม ซึ่งจะแสดงได้ว่ามีแอพพลิเคชันเกม XO ที่ได้ติดตั้งไว้ปรากฏขึ้นมา



## การนำไฟล์ .apk ไปติดตั้งบนโทรศัพท์เคลื่อนที่จริง

นอกจากไฟล์ .apk ที่สามารถนำไปติดตั้งยัง Emulator ได้แล้ว ยังสามารถนำไปติดตั้งบนโทรศัพท์เคลื่อนที่จริงได้อีกด้วย ซึ่งขั้นตอนการติดตั้งมีลักษณะเดียวกันกับการติดตั้งไฟล์ .apk บน Emulator ซึ่งมีขั้นตอนดังนี้

- เชื่อมต่อโทรศัพท์จริงเข้ากับเครื่องคอมพิวเตอร์ผ่านสาย USB ก่อน
- พิมพ์คำสั่ง “adb install <.apk file>” ใน Command Prompt เช่นเดียวกับการติดตั้งไฟล์ .apk บน Emulator ดังรูป

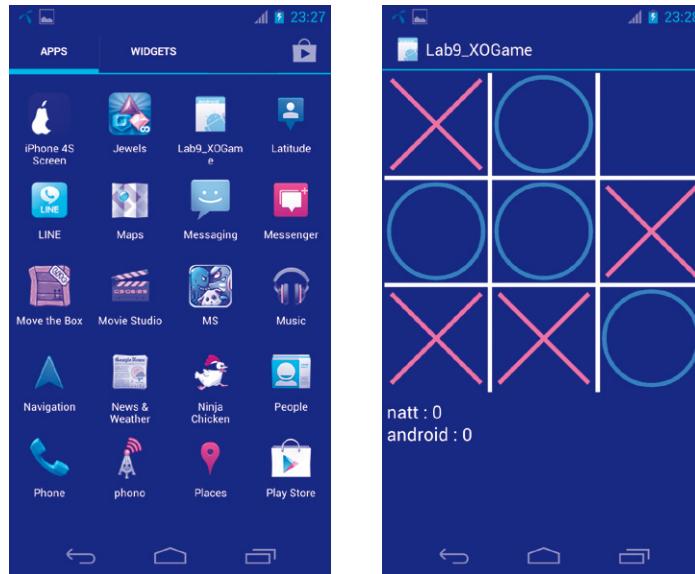
```

Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\natty>cd c:\android-sdk-windows\platform-tools
C:\android-sdk-windows\platform-tools>adb install D:\apk\Lab9_XOGame.apk
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
334 KB/s (24646 bytes in 0.072s)
pkg: /data/local/tmp/Lab9_XOGame.apk
Success

```

3. ตรวจสอบที่เครื่องโทรศัพท์เคลื่อนที่จริง ว่าจะปรากฏแอพพลิเคชันเกม XO และดูขึ้นมา



## สรุปก้ายบท

ในบทนี้เป็นการแนะนำขั้นตอนการติดตั้งแอพพลิเคชันที่พัฒนาขึ้นเข้าไปในโทรศัพท์เคลื่อนที่/Tablet Android เพื่อให้สามารถใช้งานแอพพลิเคชันได้จริงนั่นเอง

## แบบฝึกหัดก้ายบท

หลังจากผู้อ่านติดตั้งโปรแกรมเกม XO บนเครื่องโทรศัพท์เคลื่อนที่จริงแล้ว แบบฝึกหัดในส่วนนี้ให้ผู้อ่านทดลองติดตั้งแอพพลิเคชันอื่นในบทที่ผ่านมาแล้ว ลงบนเครื่องโทรศัพท์เคลื่อนที่จริง เช่น

- โปรแกรม Hello World
- โปรแกรมติดต่อกับระบบกำหนดตำแหน่ง GPS รวมไปถึงการอ้างอิงกับระบบแผนที่ของ Google Map
- โปรแกรมเพื่อติดต่อกับเครือข่าย หรือ Simple Android Chat จากนั้นทดสอบการใช้งานจริงผ่าน WiFi

# Appendix

## ความรู้เบื้องต้นการเขียนโปรแกรมภาษาจาวา

เนื้อหาในส่วนนี้ จะออกล่าวถึงเฉพาะความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรมภาษาจาวาโดยสรุปเท่านั้น เพื่อเป็นพื้นฐานให้ผู้อ่านได้ทบทวนและนำไปใช้ประโยชน์ในการพัฒนาแอพพลิเคชันบนระบบปฏิบัติการ Android ซึ่งมีเนื้อหาดังนี้

### ชนิดของข้อมูล (Data Type)

ข้อมูลที่กำหนดไว้ในภาษา Java ทั้งแบบอักขระ จำนวนเต็ม ทศนิยม และค่าความจริง ผู้อ่านจะต้องเลือกใช้นิยมของข้อมูลให้เหมาะสม เพื่อให้การทำงานของโปรแกรมไม่เกิดข้อผิดพลาด ซึ่งอาจจะสรุปได้ดังนี้

ชนิดของข้อมูล	Type	ขนาด (bit)	ขอบเขตข้อมูล
อักขระ	char	16	อักขระตามมาตรฐาน 16-bit Unicode
จำนวนเต็ม	byte	8	-128 ถึง 127 ( $-2^7$ ถึง $2^7 - 1$ )
	short	16	-32,768 ถึง 32,767 ( $-2^{15}$ ถึง $2^{15} - 1$ )
	int	32	-2,147,483,648 ถึง 2,147,483,647 ( $-2^{31}$ ถึง $2^{31} - 1$ )
	long	64	-9,223,372,036,845,775,808 ถึง 9,223,372,036,845,775,807 ( $-2^{63}$ ถึง $2^{63} - 1$ )
ทศนิยม	float	32	$1.4 \times 10^{-45}$ ถึง $3.4028235 \times 10^{38}$
	double	64	$4.9 \times 10^{-324}$ ถึง $1.7876931348623157 \times 10^{308}$
ค่าความจริง	boolean	1	true กับ false

# การประกาศตัวแปร (Declaration)

ก่อนการเขียนโปรแกรมให้ทำงานตามที่ต้องการ ผู้อ่านต้องประกาศตัวแปรและกำหนดชนิดข้อมูลของตัวแปรให้ตรงกับข้อมูลที่เราต้องการทำงานเลียก่อน ซึ่งมีรูปแบบการประกาศตัวแปรและกำหนดชนิดข้อมูลให้กับตัวแปรดังนี้

```
type varName [= Value];
```

โดยที่ **type** เป็นชนิดของข้อมูล  
**varName** เป็นชื่อตัวแปร  
**Value** เป็นค่าข้อมูลของตัวแปร  
 ตัวอย่างเช่น

```
boolean boolTest = true;  
int intAge;
```

## หลักการตั้งชื่อตัวแปร

การตั้งชื่อตัวแปรเพื่อใช้งานในโปรแกรมนั้น ผู้อ่านจะต้องตั้งให้ลึกความหมาย เมื่อมาดูโปรแกรมในภายหลัง จะทำให้สามารถจำได้ว่าตัวแปรดังกล่าวใช้ในส่วนใดของโปรแกรม ซึ่งมีกฎการตั้งชื่อดังนี้

- ชื่อตัวแปรต้องเริ่มต้นด้วย a-z, A-Z, เครื่องหมาย \$ และเครื่องหมาย \_ (Underscore) เท่านั้น
- ภายในชื่อตัวแปรประกอบด้วย a-z, A-Z, 0-9, เครื่องหมาย \$ และเครื่องหมาย \_ (Underscore) เท่านั้น
- ตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่จะแตกต่างกัน
- ห้ามตั้งชื่อซ้ำกับคำส่วน (Reserved Word) ซึ่งมีดังนี้

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

## ตัวแปรบีดบ๊อคваบ

ในความจริงนั้นตัวแปรชนิดข้อความไม่มีการกำหนดไว้ในภาษา Java แต่ผู้อ่านสามารถใช้ตัวแปรชนิดข้อความในรูปแบบของคลาสได้ดังนี้

```
String strName = value;
```

โดยที่ **strName** เป็นชื่อตัวแปร  
**value** เป็นค่าข้อมูลของตัวแปร

ตัวอย่างเช่น

```
String strProvince = "Bangkok";
String strMonth = "May";
```

## การประกาศค่าคงที่

การประกาศค่าคงที่เป็นการประกาศใช้งานตัวแปรแบบหนึ่ง แต่ไม่สามารถเปลี่ยนแปลงค่าของตัวแปรได้ ซึ่งมีรูปแบบการประกาศค่าดังนี้

```
final type varName = Value;
```

โดยที่ type เป็นชนิดของข้อมูล

varName เป็นชื่อตัวแปร

Value เป็นค่าข้อมูลของตัวแปร

ตัวอย่างเช่น

```
final String Car = "Honda";
final float Pi = 3.14;
```

## ตัวดำเนินการ (Operators)

การเขียนโปรแกรมเพื่อทำงานอย่างโดยย่างหนึ่ง ผู้อ่านจำเป็นต้องใช้งานตัวดำเนินการเสมอ ทั้งการกำหนดค่า การเปรียบเทียบค่าข้อมูล และอื่นๆ อีกมากมาย ซึ่งตัวดำเนินการในภาษา Java ที่ควรรู้มีดังนี้

- ตัวดำเนินการกำหนดค่า (Assignment Operators)
- ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)
- ตัวดำเนินการเปรียบเทียบ (Comparison Operators)
- ตัวดำเนินการทางตรรกศาสตร์ (Logical Operators)
- ตัวดำเนินการยูนารี (Unary Operators)
- ตัวดำเนินการระดับบิต (Bitwise Operators)

ประเภทตัวดำเนินการ	ตัวดำเนินการ	ความหมาย	ตัวอย่าง
กำหนดค่า	=	กำหนดค่า	A = 5;
	+=	เพิ่มค่าแล้วกำหนดค่า	A += B;
	-=	ลดค่าแล้วกำหนดค่า	A -= B;
	*=	คูณแล้วกำหนดค่า	A *= B;
	/=	หารแล้วกำหนดค่า	A /= B;
	%=	การหารเอาเศษแล้วกำหนดค่า	A %= B;
ทางคณิตศาสตร์	+	การบวก	A + B;
	-	การลบ	A - B;
	*	การคูณ	A * B;
	/	การหาร	A / B;
	%	การหารเอาเศษ	A % B;
เปรียบเทียบ	==	เท่ากับ	A == B
	!=	ไม่เท่ากัน	A != B
	<	น้อยกว่า	A < B
	<=	น้อยกว่าหรือเท่ากับ	A <= B
	>	มากกว่า	A > B
	>=	มากกว่าหรือเท่ากับ	A >= B
ทางตรรกศาสตร์	&&	และ	A && B
		หรือ	A    B
	!	ไม่	!A
ยุนารี	++	เพิ่มค่าก่อนใช้งาน	++A
		เพิ่มค่าหลังใช้งาน	A++
	--	ลดค่าก่อนใช้งาน	--A
		ลดค่าหลังใช้งาน	A--
ระดับบิต	&	AND	A & B
		OR	A   B
	^	Exclusive OR	A ^ B
	~	ตรงกันข้าม	~A
	>>	เลื่อนบิตไปทางขวา	A >> 1
	<<	เลื่อนบิตไปทางซ้าย	A << 1
	>>>	เลื่อนบิตไปทางขวาแล้วเติม 0	A >>> 2

## ลำดับความสำคัญของตัวดำเนินการ

การใช้งานตัวดำเนินการ บางครั้งมีการใช้งานตัวดำเนินการมากกว่าหนึ่งตัวในนิพจน์เดียวกัน หากไม่เข้าใจลำดับการทำงานของตัวดำเนินการ อาจจะทำให้ผลลัพธ์ผิดพลาด ตัวอย่างเช่น  $A = B + C / D + A$  ซึ่งโปรแกรมจะประมวลผลตัวดำเนินการได้ก่อน เป็นต้น

ตัวดำเนินการ	กรณีมีลำดับความสำคัญเท่ากัน	ลำดับความสำคัญ
( ), [ ]	ทำการซ้ายไปขวา	
+, -, ++, --, !, ~	ทำการขวาไปซ้าย	
*, /, %	ทำการซ้ายไปขวา	
+, -	ทำการซ้ายไปขวา	
<<, >>, >>>	ทำการซ้ายไปขวา	
<, <=, >, >=	ทำการซ้ายไปขวา	
==, !=	ทำการซ้ายไปขวา	
&	ทำการซ้ายไปขวา	
^	ทำการซ้ายไปขวา	
	ทำการซ้ายไปขวา	
&&	ทำการซ้ายไปขวา	
	ทำการซ้ายไปขวา	
? :	ทำการขวาไปซ้าย	
=, +=, -=, *=, /=, <<=, >>=, &=,  =, ^=	ทำการขวาไปซ้าย	

## การแสดงผลอักขระพิเศษ

ในการแสดงผลข้อมูลในภาษา Java มีการแสดงผลอักขระพิเศษดังนี้

อักขระพิเศษ	ความหมาย
\n	เลื่อนเคอร์เซอร์โดยหลังไป 1 ตัวอักษร
\f	ขีนหน้าใหม่
\t	ขีนบรรทัดใหม่
\v	เลื่อนเคอร์เซอร์ไปทางซ้ายสุด
\t	แสดงแท็บตามแนวโนน
'	แสดงเครื่องหมาย '
"	แสดงเครื่องหมาย "
\\"	แสดงเครื่องหมาย \
\xxx	แสดงตัวอักษรรหัส ASCII เช่น \065 เป็นการแสดงตัวอักษร A

อักษรเป็นเศษ	ความหมาย
\nxxxx	แสดงตัวอักษรรหัส Unicode เช่น \n0008 เป็นการขึ้นบรรทัดใหม่

## การจัดรูปแบบแสดงผลตัวเลข

ในหัวข้อนี้จะกล่าวถึงการจัดการรูปแบบแสดงผลข้อมูลในลักษณะจำนวนรูปแบบต่างๆ เช่น จำนวนเต็ม ทศนิยม เป็นต้น ซึ่งต้องเรียกใช้งานไลบรารี `java.text` เข้ามาใช้งานในโปรแกรมก่อนเสมอ โดยเขียนโค้ดดังนี้

```
import java.text.*;
```

จะพบว่ารูปแบบการจัดการแสดงผลตัวเลขประกอบด้วยเครื่องหมาย #, 0, \$ และ , โดยมีรูปแบบดังนี้

```
DecimalFormat ValName = new DecimalFormat(pattern);
```

โดยที่ ValName เป็นชื่อของเจกต์

pattern เป็นรูปแบบที่ต้องการกำหนด

ตัวอย่าง

```
DecimalFormat ValName = new DecimalFormat("#,###.##");
System.out.println(ValName.format(7314.527));
```

ได้ผลลัพธ์เป็น 7,314.53

## การแปลงชนิดข้อมูล

ในการเขียนโปรแกรมบางครั้ง จะต้องมีการแปลงชนิดข้อมูลที่มีอยู่ให้ตรงตามความต้องการของระบบ เช่น แปลงข้อมูลชนิดข้อความเป็นข้อมูลชนิดจำนวนเต็ม แปลงข้อมูลชนิดจำนวนเต็มเป็นข้อมูลชนิดทศนิยม เป็นต้น ซึ่งภาษา Java มีวิธีการแปลงชนิดข้อมูล 2 วิธีคือ Implicit type conversion และ Explicit type conversion

### การแปลงชนิดข้อมูลแบบ Implicit type conversion

การแปลงชนิดข้อมูลแบบ Implicit type conversion เป็นการแปลงชนิดข้อมูลโดยใช้เมธอดของคลาส ซึ่งผู้อ่านสามารถศึกษารูปแบบการใช้งานได้จากตัวอย่างต่อไปนี้

```
intTest = Integer.parseInt(str);
boolTest = Boolean.parseBoolean(str);
fltTest = Float.parseFloat(str);
```

ตัวอย่าง การแปลงชนิดข้อมูลแบบ Implicit type conversion

```
package javaapplication1;

public class Main {
    public static void main(String[] args) {
        int intTest;
        String str;
```

```

        str = "37";
        intTest = Integer.parseInt(str);
        System.out.print(intTest);
    }
}

```

## การแปลงชนิดข้อมูลแบบ Explicit type conversion

การแปลงชนิดข้อมูลแบบ Explicit type conversion เป็นการแปลงชนิดข้อมูลโดยใช้ตัวดำเนินการ cast ซึ่งมีรูปแบบการใช้งานดังนี้

```
valNameResult = (dataType) valName
```

โดยที่ valNameResult เป็นชื่อตัวแปรที่รับค่าจากการแปลงข้อมูล ซึ่งจะต้องมีชนิดข้อมูลเดียวกันกับ

dataType

dataType เป็นชนิดข้อมูลที่ต้องการแปลงให้เป็นไปตามต้องการ

valName เป็นชื่อตัวแปรที่ต้องการแปลงชนิดข้อมูล

ซึ่งผู้อ่านสามารถศึกษารูปแบบการใช้งานได้จากตัวอย่างต่อไปนี้

```

dbl = (double)intTest;
intTest = (int) dbl;
flt = (int) intTest;

```

### ตัวอย่าง การแปลงชนิดข้อมูลแบบ Explicit type conversion

```

package javaapplication1;

public class Main {
    public static void main(String[] args) {
        int intTest;
        double dbl;;
        intTest = 37;
        dbl = (double)intTest;
        System.out.print(dbl);
    }
}

```

**WARN**



#### ข้อควรระวังในการใช้ cast

1. การแปลงข้อมูลจากชนิดข้อมูลที่นิยมไปเป็นชนิดข้อมูลจำนวนเต็ม จะทำให้ค่าที่นิยมหายไป
2. การแปลงข้อมูลจากชนิดข้อมูลที่มีขนาดใหญ่กว่าไปเป็นชนิดข้อมูลที่มีขนาดเล็กกว่า อาจจะทำให้ค่าข้อมูลผิดพลาด เช่น การแปลงข้อมูลจาก long ไปเป็น int เป็นต้น

# การเขียนคำอธิบายโปรแกรม

การเขียนโปรแกรมภาษา Java สามารถเขียนคำอธิบายโปรแกรมได้ 2 รูปแบบคือ

- การเขียนคำอธิบายโปรแกรมแบบบรรทัดเดียว โดยใช้เครื่องหมาย //
- การเขียนคำอธิบายโปรแกรมแบบหลายบรรทัด โดยใช้เครื่องหมาย /\* \*/

เพื่อให้ผู้อ่านเข้าใจการใช้งานการเขียนคำอธิบายโปรแกรมมากยิ่งขึ้น ผู้อ่านสามารถศึกษาได้จากตัวอย่างต่อไปนี้

```

6 public class AndroidJava extends Activity {
7     /** Called when the activity is first created. */
8     @Override
9
10    // เริ่มต้นเมื่อเปิดโปรแกรม
11    public void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.main);
14
15        /* ทดสอบการคอมpileที่ตั้งโปรแกรม
16         * Android_Programming*/
17    }
18
19 }
```

## คำสั่งตัดสินใจ

คำสั่งตัดลินใจ เป็นคำสั่งที่ใช้กำหนดพิศวงการทำงานของโปรแกรมเมื่อมีเงื่อนไขในการตัดลินใจ เช่น ถ้าเงื่อนไขที่กำหนดเป็นจริง ให้โปรแกรมทำงานในชุดคำสั่งที่ 1 แต่หากเงื่อนไขที่กำหนดเป็นเท็จ ให้โปรแกรมทำงานในชุดคำสั่งที่ 2 เป็นต้น ซึ่งสามารถสรุปคำสั่งตัดลินใจได้ดังนี้

รูปแบบการใช้งาน	การกำหนดของคำสั่ง	ตัวอย่างการใช้งาน
if (กำหนดเงื่อนไขที่ 1) { ชุดคำสั่งเมื่อเงื่อนไขเป็นจริง; }	● เป็นคำสั่งตัดลินใจเลือกทำงานในชุดคำสั่งที่กำหนดเมื่อเงื่อนไขที่กำหนดเป็นจริง	if (A > B) { A = A + B; }
if (กำหนดเงื่อนไขที่ 1) { ชุดคำสั่งเมื่อเงื่อนไขเป็นจริง; } else { ชุดคำสั่งเมื่อเงื่อนไขเป็นเท็จ; }	● เป็นคำสั่งตัดลินใจเลือกทำงานในชุดคำสั่งใดชุดคำสั่งหนึ่งจาก 2 ชุดคำสั่ง ซึ่งพิจารณาจากเงื่อนไขที่กำหนด ● เมื่อเงื่อนไขที่กำหนดเป็นจริงให้ทำงานในชุดคำสั่งหนึ่ง และถ้าเงื่อนไขเป็นเท็จให้ทำงานในอีกชุดคำสั่งหนึ่ง	if (A > B) { A = A + B; } else { B = A + B; }

รูปแบบการใช้งาน	การทำงานของคำสั่ง	ตัวอย่างการใช้งาน
<pre>if (กำหนดเงื่อนไขที่ 1) {     ชุดคำสั่งที่ 1; } else if (กำหนดเงื่อนไขที่ 2) {     ชุดคำสั่ง 2; } . . . else {     ชุดคำสั่งสุดท้าย; }</pre>	<ul style="list-style-type: none"> <li>เป็นคำสั่งตัดสินใจเลือกทำงานในชุดคำสั่งใดชุดคำสั่งหนึ่งจากหลาย ๆ ชุดคำสั่ง ซึ่งพิจารณาจากเงื่อนไขที่กำหนด</li> <li>เมื่อเงื่อนไขที่ 1 เป็นจริงให้ทำงานในชุดคำสั่งที่ 1 และถ้าเงื่อนไขเป็นเท็จให้พิจารณาเงื่อนไขที่ 2</li> <li>เมื่อเงื่อนไขที่ 2 เป็นจริงให้ทำงานในชุดคำสั่งที่ 2 และถ้าเงื่อนไขเป็นเท็จให้พิจารณาเงื่อนไขต่อไป</li> <li>ทำเช่นนี้ต่อไปหากพบว่าทุกเงื่อนไขเป็นเท็จ ให้ทำงานในชุดคำสั่งสุดท้าย</li> </ul>	<pre>if (A &gt; B) {     A = A + B; } else if (B &gt; C) {     A = A + C; } else if (C &gt; D) {     A = A + D; } else {     B = A; }</pre>
<pre>switch (ตัวแปรหรือนิพจน์) {     case ค่าที่ 1 :         ชุดคำสั่งที่ 1;         break;     case ค่าที่ 2 :         ชุดคำสั่งที่ 2;         break;     case ค่าที่ 3 :         ชุดคำสั่งที่ 3;         break;     default :         ชุดคำสั่งสุดท้าย; }</pre>	<ul style="list-style-type: none"> <li>เป็นคำสั่งตัดสินใจเลือกทำงานในชุดคำสั่งใดชุดคำสั่งหนึ่งจากหลาย ๆ ชุดคำสั่ง ซึ่งพิจารณาจากตัวแปรหรือนิพจน์ที่กำหนด</li> <li>หากตัวแปรหรือนิพจน์ที่กำหนดตรงกับค่าที่ 1 ให้ทำงานในชุดคำสั่งที่ 1 และถ้าไม่ตรงกันให้พิจารณาค่าที่ 2</li> <li>หากตัวแปรหรือนิพจน์ที่กำหนดตรงกับค่าที่ 2 ให้ทำงานในชุดคำสั่งที่ 2 และถ้าไม่ตรงกันให้พิจารณาค่าต่อไป</li> <li>ทำเช่นนี้ต่อไปหากพบว่าตัวแปรหรือนิพจน์ที่กำหนดไม่ตรงกับค่าใดๆ ให้ทำงานในชุดคำสั่งสุดท้าย</li> </ul>	<pre>switch (A) {     case 1 :         A = A + B;         break;     case 2 :         A = A + C;         break;     case 3 :         B = C;         break;     default :         A = C; }</pre>

# คำสั่งวนลูป

คำสั่งวนลูปเป็นคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรมแบบวนซ้ำๆ การทำงานเดิมตามเงื่อนไขที่กำหนด เช่น การวนซ้ำการทำงานเดิมเป็นจำนวน 10 รอบ ทำงานซ้ำๆ จนกว่าเงื่อนไขเป็นเท็จ เป็นต้น ซึ่งสามารถสรุปคำสั่งวนลูปได้ดังนี้

รูปแบบการใช้งาน	การทำงานของคำสั่ง	ตัวอย่างการใช้งาน
<pre>for(initial; condition; change) {     ชุดคำสั่ง; }</pre> <p>โดยที่</p> <ul style="list-style-type: none"> <li>initial = ค่าเริ่มต้นของตัวแปร</li> <li>condition = เงื่อนไขที่กำหนด</li> <li>change = การเปลี่ยนแปลงของตัวแปรในแต่ละรอบ</li> </ul>	<ul style="list-style-type: none"> <li>เป็นคำสั่งวนซ้ำการทำงานด้วยจำนวนรอบที่แน่นอน</li> <li>โปรแกรมจะตรวจสอบเงื่อนไขก่อนถ้าเงื่อนไขเป็นจริงให้ทำงานกับชุดคำสั่งภายในลูป</li> <li>เมื่อทำงานเสร็จจะเพิ่มหรือลดค่าตัวแปรและตรวจสอบเงื่อนไขใหม่อีกครั้ง</li> </ul>	<pre>for(A = 0; A &lt; 10; A++) {     Sum = Sum + A; }</pre>
<pre>while(เงื่อนไขที่กำหนด) {     ชุดคำสั่ง; }</pre>	<ul style="list-style-type: none"> <li>เป็นคำสั่งวนซ้ำการทำงาน</li> <li>โปรแกรมจะทำงานชุดคำสั่งภายในลูป เมื่อเงื่อนไขที่กำหนดเป็นจริง</li> <li>เมื่อทำงานเสร็จจะตรวจสอบเงื่อนไขใหม่อีกครั้ง</li> <li>โปรแกรมจะออกจากลูปการทำงานเมื่อเงื่อนไขที่กำหนดเป็นเท็จ</li> </ul>	<pre>while(intRow &lt; 10) {     A = A + B;     intRow++; }</pre>
<pre>do {     ชุดคำสั่ง; }while(เงื่อนไขที่กำหนด); </pre>	<ul style="list-style-type: none"> <li>เป็นคำสั่งวนซ้ำการทำงาน</li> <li>โปรแกรมจะทำงานชุดคำสั่งภายในลูป ก่อน 1 รอบ จึงตรวจสอบเงื่อนไขที่กำหนด</li> <li>ถ้าเงื่อนไขที่กำหนดเป็นจริง ให้กลับไปทำงานชุดคำสั่งภายในลูปอีกครั้ง และตรวจสอบเงื่อนไขที่กำหนดอีกครั้ง</li> <li>ถ้าเงื่อนไขที่กำหนดเป็นเท็จโปรแกรมจะออกจากลูปการทำงาน</li> </ul>	<pre>do {     A = A + B;     intRow++; }while(intRow &lt;= 5); </pre>

# ตัวแปรอาร์เรย์

ตัวแปรอาร์เรย์เปรียบเสมือนการนำตัวแปรมาเรียงต่อกันหลายๆ ตัว โดยที่ทุกตัวจะมีชนิดข้อมูลเดียวกัน มีชื่อตัวแปรเดียวกัน แต่สามารถอ้างถึงตำแหน่งข้อมูลแต่ละตัวที่เรียงต่อกันด้วยลำดับการจัดเรียง ซึ่งเรียกตำแหน่งข้อมูลแต่ละตัวว่า อินเด็กซ์ (Index)

ตัวแปรอาร์เรย์ที่มีให้ใช้งานในภาษา Java นั้นสามารถแยกได้ 2 แบบคือ ตัวแปรอาร์เรย์ 1 มิติ และตัวแปรอาร์เรย์หลายมิติ ซึ่งในที่นี้จะยกถึงตัวแปรอาร์เรย์ 1 มิติ ตัวแปรอาร์เรย์ 2 มิติ และตัวแปรอาร์เรย์ 3 มิติเท่านั้น

## ตัวแปรอาร์เรย์ 1 มิติ

ตัวแปรอาร์เรย์ 1 มิติเปรียบได้กับการนำตัวแปรมาเรียงต่อกันหลายๆ ตัวในลักษณะของเก้าข้อมูลซึ่งสามารถจำลองตัวอย่างตัวแปรอาร์เรย์ 1 มิติ ชื่อตัวแปร intEx1 เป็นตัวแปรชนิดจำนวนเต็มที่สามารถเก็บข้อมูลจำนวนเต็มได้ 6 ตัวยกตัวอย่างดังนี้



จากตัวอย่างจะเห็นได้ว่า ตัวแปรอาร์เรย์สามารถเก็บข้อมูลได้ 6 ตัว โดยที่

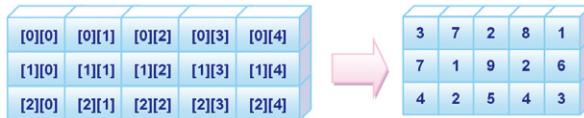
- ตัวแปรตัวแรกคือ ตำแหน่งอินเด็กซ์ที่ 0 มีค่าเท่ากับ 5 ซึ่งสามารถเขียนได้เป็น  $\text{intEx1}[0] = 5$
- ตัวแปรตัวที่ 2 คือ ตำแหน่งอินเด็กซ์ที่ 1 มีค่าเท่ากับ 8 ซึ่งสามารถเขียนได้เป็น  $\text{intEx1}[1] = 8$
- ตัวแปรตัวสุดท้ายคือ ตำแหน่งอินเด็กซ์ที่ 5 มีค่าเท่ากับ 2 ซึ่งสามารถเขียนได้เป็น  $\text{intEx1}[5] = 2$

## ตัวแปรอาร์เรย์หลายมิติ

ตัวแปรอาร์เรย์หลายมิติคือ ตัวแปรที่มีมุ่งมองการเข้าถึงข้อมูลของตัวแปรได้มากกว่า 1 ด้าน ซึ่งในที่นี้จะกล่าวถึงเฉพาะตัวแปรอาร์เรย์ 2 มิติ และตัวแปรอาร์เรย์ 3 มิติเท่านั้น

## ตัวแปรอาร์เรย์ 2 มิติ

ตัวแปรอาร์เรย์ 2 มิติเปรียบได้กับการนำตัวแปรมาเรียงต่อกันหลายๆ ตัวในลักษณะของตารางข้อมูลซึ่งสามารถจำลองตัวอย่างตัวแปรอาร์เรย์ 2 มิติ ชื่อตัวแปร intEx2 เป็นตัวแปรชนิดจำนวนเต็มที่สามารถเก็บข้อมูลจำนวนเต็มได้ 15 ตัวยกตัวอย่างดังนี้



จากตัวอย่างจะเห็นได้ว่า ตัวแปรอาร์เรย์ที่มีขนาด 3 คอลัมน์สามารถเก็บข้อมูลได้ 15 ตัว โดยที่

- ตัวแปรแถวที่ 1 คอลัมน์ที่ 1 มีค่าเท่ากับ 3 ซึ่งสามารถเขียนได้เป็น  $\text{intEx2}[0][0] = 3$
- ตัวแปรแถวที่ 1 คอลัมน์ที่ 2 มีค่าเท่ากับ 7 ซึ่งสามารถเขียนได้เป็น  $\text{intEx2}[0][1] = 7$
- ตัวแปรแถวที่ 3 คอลัมน์ที่ 2 มีค่าเท่ากับ 2 ซึ่งสามารถเขียนได้เป็น  $\text{intEx2}[2][1] = 2$
- ตัวแปรแถวที่ 3 คอลัมน์ที่ 3 มีค่าเท่ากับ 3 ซึ่งสามารถเขียนได้เป็น  $\text{intEx2}[2][4] = 3$

## ตัวแปรอาร์เรย์ 3 มิติ

ตัวแปรอาร์เรย์ 3 มิติเปรียบได้กับการนำตัวแปรมาเรียงต่อกันหลายๆ ตัวในลักษณะของกล่องข้อมูล หรือเรียกอีกอย่างว่า อาร์เรย์ของอาร์เรย์ 2 มิติ ซึ่งสามารถจัดลองตัวอย่างตัวแปรอาร์เรย์ 3 มิติ ชื่อตัวแปร intEx3 เป็นตัวแปรชนิดจำนวนเต็มที่สามารถเก็บข้อมูลจำนวนเต็มได้ 24 ตัวยกตัวอย่างดังนี้



จากตัวอย่างจะเห็นได้ว่า ตัวแปรอาร์เรย์ที่มีขนาด 2 บล็อก 3 แถว 4 คอลัมน์สามารถเก็บข้อมูลได้ 24 ตัว โดยที่

- ตัวแปรบล็อกที่ 1 และที่ 1 คอลัมน์ที่ 1 มีค่าเท่ากับ 9 เชียนได้เป็น intEx3[0][0][0] = 9
- ตัวแปรบล็อกที่ 1 และที่ 1 คอลัมน์ที่ 2 มีค่าเท่ากับ 2 เชียนได้เป็น intEx3[0][0][1] = 2
- ตัวแปรบล็อกที่ 1 และที่ 3 คอลัมน์ที่ 4 มีค่าเท่ากับ 7 เชียนได้เป็น intEx3[0][2][3] = 7
- ตัวแปรบล็อกที่ 2 และที่ 1 คอลัมน์ที่ 1 มีค่าเท่ากับ 7 เชียนได้เป็น intEx3[1][0][0] = 7
- ตัวแปรบล็อกที่ 2 และที่ 1 คอลัมน์ที่ 2 มีค่าเท่ากับ 1 เชียนได้เป็น intEx3[1][0][1] = 1
- ตัวแปรบล็อกที่ 2 และที่ 3 คอลัมน์ที่ 4 มีค่าเท่ากับ 2 เชียนได้เป็น intEx3[1][2][3] = 2

## การเขียนโปรแกรมกับตัวแปรอาร์เรย์

เมื่อผู้อ่านได้ทำความรู้จักกับลักษณะของตัวแปรอาร์เรย์ในประเภทต่างๆ กันไปแล้ว ในหัวข้อนี้จะเป็นการใช้งานตัวแปรอาร์เรย์ในการเขียนโปรแกรมของเรา โดยจะกล่าวถึงการประกาศใช้งานตัวแปร กำหนดค่าตัวแปร และการอ้างถึงข้อมูลตัวแปร

### การประกาศตัวแปรอาร์เรย์

ในหัวข้อที่ผ่านมาผู้อ่านได้เรียนรู้การประกาศตัวแปรชนิดต่างๆ กันไปแล้ว ซึ่งการประกาศตัวแปรอาร์เรย์นั้นมีกฎการตั้งชื่อและรูปแบบคำสั่งเหมือนกับการประกาศตัวแปรทั่วไป ต่างกันเพียงแต่การประกาศตัวแปรอาร์เรย์นั้นต้องมีการกำหนดขนาดของตัวแปรไว้เท่านั้นเอง ซึ่งมี 2 รูปแบบดังนี้

```
type varName[ ] = new type[n];
```

หรือ

```
type varName[ ];
varName = new type[n];
```

โดยที่ type เป็นชนิดของข้อมูล  
 varName เป็นชื่อตัวแปรอาร์เรย์  
 n เป็นขนาดของตัวแปรอาร์เรย์  
 มีตัวอย่างดังนี้

```
int intTest[] = new int[7];
char ch[] = new char[10];

double dblTest[];
dblTest = new double[3];
boolean bool[];
bool = new boolean[2];
```

จากตัวอย่างโค้ดจะเห็นได้ว่าจะมีการกำหนดขนาดตัวแปรไว้ด้วยเครื่องหมาย [ ] เช่น การประกาศตัวแปรชนิดจำนวนเต็มชื่อ intTest ขนาด 7 ตัว เป็นต้น

### การกำหนดค่าข้อมูลให้ตัวแปรอาร์เรย์

การกำหนดค่าให้กับตัวแปรแบบอาร์เรย์นั้น มีรูปแบบเหมือนกับการกำหนดค่าให้กับตัวแปรทั่วไป ต่างกันเพียงแต่การกำหนดค่าให้กับตัวแปรแบบอาร์เรย์นั้น จะต้องกำหนดตำแหน่งอินเด็กซ์เพื่อรับตำแหน่งของตัวแปรที่จะกำหนดค่าภายใต้เครื่องหมาย { } ยกตัวอย่างเช่น

```
int intNum[] = {5, 3, 4, 1, 7};
char chPassword[] = {'P', 'a', 's', 's'};
double fPrice[][] = {{12.5, 17.8, 8.1}, {45.9, 9.7, 86.5}};
int intScore[][][] = {{2, 4, 7, 6}, {9, 1, 3, 4}};
```

ตัวอย่างที่ผ่านมาเป็นการประกาศตัวแปรร่วมกับกำหนดค่า ซึ่งอาจจะกำหนดค่าภายหลังประกาศตัวแปรได้ เช่นกัน ซึ่งมีตัวอย่างดังนี้

```
intNum[1] = 6;
chPassword[0] = 'Z';
fPrice[2][1] = 0.08;
```

จากตัวอย่างจะเห็นได้ว่า เรากำหนดค่าให้ตัวแปร intNum ในลำดับที่ 2 อินเด็กซ์ที่ 1 มีค่าเท่ากับ 6 ซึ่งได้ผลลัพธ์ดังนี้



### การอ้างถึงข้อมูลในตัวแปรอาร์เรย์

เมื่อมีการกำหนดค่าให้กับตัวแปรแล้ว ต่อไปก็ต้องรู้วิธีการอ่านข้อมูลจากตัวแปรโดยการระบุตำแหน่งข้อมูลที่ต้องการลงไป หรือที่เรียกว่า อินเด็กซ์ (Index) ซึ่งมีตัวอย่างการใช้งานดังนี้

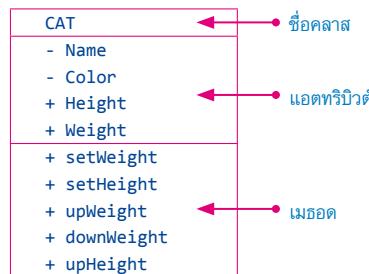
```
Ans = intScore[2] * 3;
fSum = fPrice[3][2] + fPrice[3][2];
intNum[0] = intNum[1] + intNum[2];
```

จากตัวอย่างจะเห็นได้ว่า มีการเรียกใช้งานตัวแปรอาร์เรย์โดยระบุตำแหน่งของข้อมูลภายในเครื่องหมาย [ ] เช่น Ans = intScore[2] \* 3; เป็นการทำหน้าที่ให้กับตัวแปร Ans โดยมีค่ามาจากผลคูณระหว่างค่าในตัวแปรอาร์เรย์ตำแหน่งที่ 3 กับ 3 เป็นต้น

## พื้นฐานการเขียนโปรแกรมเบิงวัตถุ

ในแนวคิดการเขียนโปรแกรมแบบ OOP นั้น ความรู้พื้นฐานที่ควรรู้เพื่อให้เข้าใจในหลักการเขียนโปรแกรมมีดังนี้

- **คลาส (Class)** เป็นต้นแบบของออบเจกต์ที่เป็นตัวกำหนดคุณสมบัติและความสามารถพื้นฐานที่ทุกๆ ออบเจกต์ต้องมี เช่น คลาสของแมว ซึ่งแมวแต่ละตัวจะต้องมีชื่อสายพันธุ์ มีลี มีความสูง และน้ำหนัก โดยที่แมวแต่ละตัวต้องสามารถเดินได้ วิ่งได้ และร้องได้ เป็นต้น
- **แอ็ตทริบิวต์ (Attribute)** เป็นคุณสมบัติต่างๆ ของคลาส
- **เมธอด (Method)** เป็นความสามารถต่างๆ ของคลาส
- **ออบเจกต์ (Object)** เป็นเหมือนตัวแทนแมวแต่ละตัวที่มีคุณสมบัติและความสามารถที่แตกต่างกัน เช่น แมวตัวที่ 1 เป็นพันธุ์ไทย, สีขาว, สูง 30 cm, น้ำหนัก 1.5 kg โดยที่เดินໄodic 100 m/h, วิ่งได้ 1 km/h, ร้องเลียงดัง และแมวตัวที่ 2 เป็นพันธุ์ไทย, สีเทา, สูง 35 cm, น้ำหนัก 1.9 kg โดยที่เดินໄodic 200 m/h, วิ่งได้ 1.7 km/h, ร้องเลียงดัง เป็นต้น สรุกด้วยว่าแมวแต่ละตัวมีโครงสร้างเดียวกัน แต่จะต่างกันเพียงแค่คุณสมบัติและความสามารถเท่านั้น



### การสร้างคลาส

การสร้างคลาสเปรียบเสมือนการเริ่มต้นเขียนโปรแกรมแบบ OOP ซึ่งผู้อ่านควรศึกษาให้เข้าใจ โดยมีรูปแบบการสร้างคลาสดังนี้

```
[รูปแบบการเข้าถึง] class ชื่อคลาส
{
    [การประกาศแอ็ตทริบิวต์]
    [การประกาศเมธอด]
}
```

โดยที่การตั้งชื่อคลาส นิยมตั้งชื่อด้วยรูปแบบดังนี้

- ตั้งชื่อตามกฎการตั้งชื่อตัวแปรภาษา Java
- นิยมชื่นต้นคำแต่ละคำด้วยอักษรตัวพิมพ์ใหญ่
- ควรตั้งชื่อเป็นคำนาม

ตัวอย่างการสร้างคลาส เช่น

```
public class Cat  
{  
}
```

## การประกาศแอ็ตทริบิวต์

การประกาศแอ็ตทริบิวต์เป็นส่วนหนึ่งของการสร้างคลาส ซึ่งผู้อ่านควรศึกษาให้เข้าใจ โดยมีรูปแบบ การประกาศแอ็ตทริบิวต์ดังนี้

[รูปแบบการเข้าสู่] ชนิดข้อมูล ชื่อแอ็ตทริบิวต์;

โดยที่การตั้งชื่อแอ็ตทริบิวต์ นิยมตั้งชื่อด้วยรูปแบบดังนี้

- ตั้งชื่อตามกฎการตั้งชื่อตัวแปรภาษา Java
- นิยมชื่นต้นด้วยอักษรคำแรกเป็นอักษรตัวพิมพ์เล็ก แต่คำต่อไปจะชื่นต้นด้วยอักษรตัวพิมพ์ใหญ่
- ควรตั้งชื่อเป็นคำนาม

ตัวอย่างการประกาศแอ็ตทริบิวต์ เช่น

```
private String nameCat;  
private String colorCat;  
public double heightCat;  
public double weightCat;
```

## การประกาศเมธอด

การประกาศเมธอดเป็นส่วนหนึ่งของการสร้างคลาส ซึ่งผู้อ่านควรศึกษาให้เข้าใจ โดยมีรูปแบบการ ประกาศเมธอดดังนี้

[รูปแบบการเข้าสู่] ชนิดข้อมูลที่ส่งกลับ ชื่อเมธอด ([ข้อมูลที่รับเข้ามา])

```
{  
    ชุดคำสั่ง;  
}
```

โดยที่การตั้งชื่อเมธอด นิยมตั้งชื่อด้วยรูปแบบดังนี้

- ตั้งชื่อตามกฎการตั้งชื่อตัวแปรภาษา Java
- นิยมชื่นต้นด้วยอักษรคำแรกเป็นอักษรตัวพิมพ์เล็ก แต่คำต่อไปจะชื่นต้นด้วยอักษรตัวพิมพ์ใหญ่
- ควรตั้งชื่อเป็นคำกริยา



### ตัวอย่างการประกาศเมธอด เช่น

```
public void setWeight(double dblWeight)
{
    weightCat = dblWeight;
}

public void setHeight(double dblHeight)
{
    heightCat = dblHeight;
}

public double upWeight(double dblWeight)
{
    weightCat = weightCat + dblWeight;
    return weightCat;
}

public double downWeight(double dblWeight)
{
    weightCat = weightCat - dblWeight;
    return weightCat;
}

public double upHeight(double dblHeight)
{
    heightCat = heightCat + dblHeight;
    return heightCat;
}
```

จากตัวอย่างการสร้างคลาส การประกาศแอ็ตทริบิวต์ และการประกาศเมธอด ซึ่งจะได้คลาสดังนี้

```
public class Cat {
    private String nameCat;
    private String colorCat;
    public double heightCat;
    public double weightCat;

    public void setWeight(double dblWeight)
    {
        weightCat = dblWeight;
    }

    public void setHeight(double dblHeight)
    {
        heightCat = dblHeight;
    }

    public double upWeight(double dblWeight)
    {
        weightCat = weightCat + dblWeight;
        return weightCat;
    }

    public double downWeight(double dblWeight)
    {
```

```

        weightCat = weightCat - dblWeight;
        return weightCat;
    }

    public double upHeight(double dblHeight)
    {
        heightCat = heightCat + dblHeight;
        return heightCat;
    }
}

```

### การสร้างออบเจกต์

ในการเขียนโปรแกรมแบบ OOP ตอนเจก็ตจะถูกสร้างมาจากคลาส หมายความได้ว่า คลาสคือ แบบสำหรับการสร้างแมว และออบเจกต์คือ โมเดลของแมวที่สร้างขึ้นจากแบบเพื่อใช้งาน โดยที่สามารถสร้างโมเดลของแมวได้มากกว่า 1 ตัวจากแบบที่มีเพียงแบบเดียว ซึ่งมีรูปแบบการสร้างออบเจกต์ดังนี้

ชื่อคลาส ชื่อออบเจกต์;  
ชื่อออบเจกต์ = new ชื่อคลาส;

หรือ

ชื่อคลาส ชื่อออบเจกต์ = new ชื่อคลาส;

ตัวอย่างการสร้างออบเจกต์ เช่น

```

Cat catThai;
catThai = new Cat;

Cat catPersia = new Cat;

```

### การใช้งานออบเจกต์

เมื่อได้เรียนรู้การสร้างออบเจกต์มาแล้ว หัวข้อนี้จะได้ศึกษาการใช้งานออบเจกต์ที่ได้สร้างขึ้น ซึ่งมีตัวอย่างดังนี้

```

Cat catPersia = new Cat;
catPersia.setHeight(30.5);
catPersia.setWeight(2.3);

```



# ussutayut

## 1. Android Community [Access 04/05/2012]

Android Developer: <http://groups.google.com/group/android-developers/>

Source Code: <http://source.android.com/>

Thailand Android Community <http://www.thaiandroidphone.com/>

Droidsand Thailand Community <http://www.droidsans.com/>

## 2. Atreco, Inc., HelloAndroid – Android Resource [Access 04/05/2012]

<http://www.helloandroid.com/>

<http://www.helloandroid.com/tutorials>

## 3. Bogotobogo, “Android Tutorial,” [Access 04/05/2012] Available at <http://bogotobogo.com/android.html>

## 4. Brian Hall, “Beej’s Guide to Network Programming,” [Access 04/05/2012] Available at <http://beej.us/guide/bgnet/>

## 5. Chakchai So-In, “322 261, 322 262, 322749 Course Material,” Available at <http://web.kku.ac.th/chakso/>.

## 6. Dharma P. Agrawal and Qing-An Zeng, “Introduction to Wireless and Mobile Systems,” 608 pp., CL-Engineering, June 2010.

## 7. DroidNova [Access 04/05/2012]

<http://www.droidnova.com/android-3d-game-tutorial-part-i,312.html>

<http://www.droidnova.com/android-3d-game-tutorial-part-ii,328.html>

<http://www.droidnova.com/android-3d-game-tutorial-part-iii,348.html>

<http://www.droidnova.com/android-3d-game-tutorial-part-iv,358.html>

<http://www.droidnova.com/android-3d-game-tutorial-part-v,376.html>

<http://www.droidnova.com/android-3d-game-tutorial-part-vi,436.html>

## 8. Ed Burnette, “Hello, Android: Introducing Google’s Mobile Development Platform (Pragmatic Programmers)”, 250 pp., Pragmatic Bookshelf, October 2009.

## 9. Frank Ableson and Robi Sen, “Android in Action,” 592 pp., Manning Publications, February 2011.

## 10. Frank Ableson, “Introduction to Android development,” IBM Technical Report, May 2009. [Access 04/05/2012]

Available at <http://www.ibm.com/developerworksopensource/library/os-android-devel/>

## 11. Frank Ableson, Charlie Collins and Robi Sen, “Unlocking Android: A Developer’s Guide,” 416 pp., Manning Publications, June 2009.

## 12. Google Android Websites: [Access 04/05/2012]

Sample Codes, Available at <http://developer.android.com/resources/samples/index.html>

Tutorials, Available at <http://developer.android.com/resources/tutorials/hello-world.html>

Android Emulator, Available at <http://developer.android.com/guide/developing/tools/emulator.html>

13. James F. Kurose and Keith W. Ross, "Computer Networking: A Top-Down Approach," 864 pp., Addison Wesley, March 2009.
14. James Steele and Nelson To, "The Android Developer's Cookbook: Building Applications with the Android SDK," 400 pp., Addison-Wesley Professional, October 2010.
15. Kenneth L. Calvert and Michael J. Donahoo, "TCP/IP Sockets in Java, Second Edition: Practical Guide for Programmers," 192 pp., Morgan Kaufmann, February 2008.
16. Lars Vogel, "Java Tutorial," [Access 04/05/2012] Available at <http://www.vogella.de/java.html>
17. Lars Vogel, "Eclipse IDE Tutorial," [Access 04/05/2012] Available at <http://www.vogella.de/articles/Eclipse/article.html>
18. Lauren Darcey and Shane Coder, "Sams Teach Yourself Android Application Development in 24 Hours (Sams Teach Yourself – Hours)," 480 pp., Sams, June 2010.
19. Mark L Murphy, "Android Programming Tutorials," 434 pp., CommonsWare, LLC, September 2010.
20. Mark Murphy, "Beginning Android 2," 416 pp., Apress, March 2010.
21. Marko Gargenta, "Learning Android," 268 pp., O'Reilly Media, March 2011.
22. Rick Rogers, John Lombardo, Zigurd Mednieks and G. Blake Meike, "Android Application Development: Programming with the Google SDK," 336 pp., O'Reily Media, May 2009.
23. Sayed Hashimi, Satya Komatineni and Dave MacLean, "Pro Android 2," 736 pp., Apress, March 2010.
24. Shane Conder and Lauren Darcey, "Android Wireless Application Development," 792 pp., Addison-Wesley Professional, December 2010.
25. W. Richard Stevens, "UNIX Network Programming: Networking APIs: Sockets and XTI; Volume 1," 1009 pp., Prentice Hall PTR, January 1998.
26. Wiki, "Introduction to Android," [Access 04/05/2012] Available at [http://en.androidwiki.com/wiki/Introduction\\_to\\_Android](http://en.androidwiki.com/wiki/Introduction_to_Android)
27. William Stallings, "Data and Computer Communications," 888 pp., Prentice Halls, August 2010.
28. William Stallings, "Wireless Communications & Networks," 576 pp., Prentice Hall, November 2004.
29. จักรชัย โลอินทร์ ดร., พงษ์ธร จันทร์ยอย, "Basic Android APP Development," 264 หน้า, ไอเดีย พรีเมียร์, 2554.
30. อนันต์ ผลเพิ่ม, "แคนไวร์ลai," 192 หน้า, ซีเอ็ด, 2550.
31. อรุณพ ขันธิกุล, อำนาจ มีมงคล, "ออกแบบและติดตั้งระบบ Wireless LAN 2nd Edition," 704 หน้า, ไอเดีย พรีเมียร์, 2553.

# INDEX

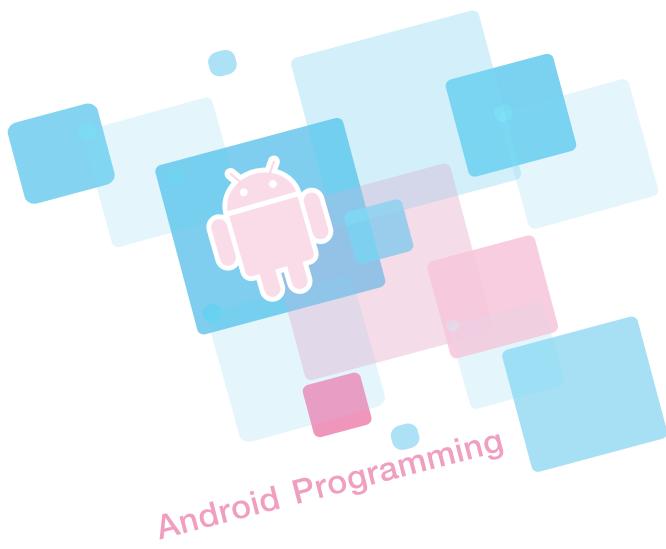
<b>ก</b>	
กราฟิก.....	241
การเขียนคำอธิบายโปรแกรม.....	334
การจัดรูปแบบ.....	332
การจัดวางหน้าจอ.....	150
การประมวลผลตัวแปร.....	328
การแปลงชนิดข้อมูล.....	332
การรับข้อความ.....	166
การส่งข้อความ.....	164
การส่งออกไฟล์ .apk .....	297, 321
เกม.....	241
แก้ไขข้อมูล.....	237
<b>ค</b>	
คลาส.....	340
ค่าคงที่.....	329
<b>ช</b>	
ชนิดของข้อมูล.....	327
<b>ซ</b>	
เซนเซอร์.....	271
<b>ต</b>	
ตัวดำเนินการ.....	329
<b>ห</b>	
เทเบิล เลย์เอาต์.....	119
<b>พ</b>	
พอร์ต.....	207
เพิ่มข้อมูล.....	235
<b>ฟ</b>	
เฟรม เลย์เอาต์.....	114
<b>ม</b>	
เมธอด.....	340
<b>ร</b>	
รีเลย์ฟ เลย์เอาต์.....	111
<b>ล</b>	
ลบข้อมูล.....	236
ล็อกแคช .....	289

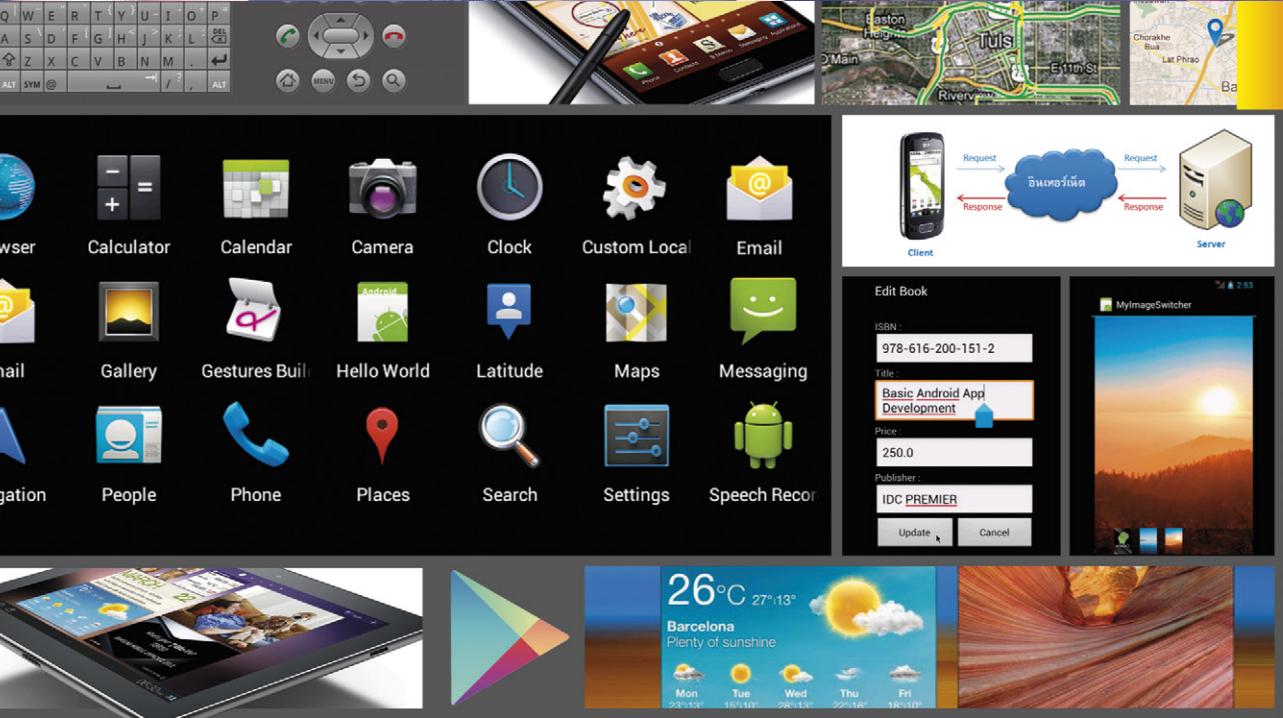
ลำดับความสำคัญของตัวดำเนินการ.....	331
ลีเนียร์ เลย์เอาต์ .....	107
<b>ว</b>	
วงรอบชีวิตของแอพพลิเคชัน.....	10
วิว.....	92
วิวกรุ๊ป .....	123
<b>ส</b>	
สถาปัตยกรรมของแอนดรอยด์.....	5
สร้างและปรับปรุงฐานข้อมูล.....	223
<b>ห</b>	
หน้าจอสัมผัส .....	263
หลักการตั้งชื่อตัวแปร .....	328
<b>อ</b>	
ອอบเจกต์ .....	340
อาร์เรย์ .....	337
อินเด็กซ์.....	337
อีเวนต์.....	102
แอคทิวิตี้.....	51
แอตทริบิวต์.....	340
แอนดรอยด์.....	1
แอนโพรูลต เลย์เอาต์ .....	116
<b>آ</b>	
Absolute Layout.....	116
ACCESS_FINE_LOCATION .....	176
Activity .....	51
ADT .....	27
Android .....	1
Android Architecture.....	5
Android Development Tool.....	27
Android Market .....	295
Android SDK .....	25
Android Software Development Kit .....	25
API Level.....	298
apk .....	297, 321
Application Component.....	9
Application Life Cycle.....	10
Attribute .....	340



<b>B</b>	Button.....	101
<b>C</b>	cast .....	333
	Class.....	340
<b>D</b>	Dalvik Debug Monitor Server.....	180, 285
	Data.....	287
	Database Adapter .....	216
	Database Management System.....	215
	Data Type .....	327
	DBMS.....	215
	DDMS.....	180, 285
	Declaration.....	328
	delete.....	223
	Devices .....	287
	Dialog .....	250
	do .....	336
<b>E</b>	Eclipse.....	18
	EditText.....	98
	Emulator .....	171
	Emulator Control.....	287
	Event.....	102
	Explicit type conversion.....	333
<b>F</b>	File Explorer .....	292
	for .....	336
	Frame Layout.....	114
<b>G</b>	getPrefValue.....	149
	Global Positioning System .....	173
	Google API Key .....	182
	Google Map .....	182
	GPS.....	173
	Graphic .....	241, 254
<b>I</b>	if.....	334
	ImageButton.....	104
	Image Switcher View.....	130
	ImageView.....	100
	Imperial Systems .....	75
	Imperial Units.....	75
	Implicit type conversion .....	332
	Index.....	337
	insert.....	223
<b>J</b>	Java Development Kit.....	14
	JDK .....	14
<b>L</b>	Latency .....	287
	Linear Layout .....	107
	ListView .....	123
	LogCat .....	289
<b>M</b>	Marker .....	191
	Method.....	340
<b>O</b>	Object.....	340
	ODBC .....	215
	onAccuracyChanged .....	277
	onCreate.....	11
	onDestroy.....	11
	onPause.....	11
	onRestart.....	11
	onResume.....	11
	onSensorChanged.....	277
	onStart .....	11
	onStop .....	11
	OOP .....	340
	Open Database Connectivity .....	215
	Operators .....	329
	Options Menu.....	134
	Outline.....	57
<b>P</b>	Permission .....	158, 161
	Port .....	207
	Port of Entry .....	207
	Preferences Screen.....	145

Process.....	287
Properties.....	57
<b>Q</b>	
query.....	223
<b>R</b>	
Relative Layout.....	111
<b>S</b>	
Screen Orientation.....	150
ScrollView .....	126
Sensor.....	271
setContent .....	141
setIndicator.....	141
setOnPreferenceClickListener.....	149
Short Message Service .....	157
SMS .....	157
Socket.....	198
Speed.....	287
SQLite .....	215
switch.....	335
<b>T</b>	
Table Layout .....	119
Tablet .....	72
TabWidget.....	138
TCP .....	198
Telephony Status.....	287
TextView.....	95
TouchScreen.....	263
Transmission Control Protocol.....	198
<b>U</b>	
UI.....	91
update.....	223
User Interface.....	91
<b>V</b>	
versionCode .....	298, 309
versionName .....	298, 309
VIBRATE.....	273
View .....	92
ViewGroup .....	123
Voice.....	287
<b>W</b>	
WebView.....	141
while .....	336





# Android App Development

ฉบับสมบูรณ์

## เนื้อหาภายในประกอบด้วย

- แนะนำให้รู้จักระบบปฏิบัติการ Android
- แนะนำการติดตั้งเครื่องมือสำหรับพัฒนาแอพพลิเคชัน
- พัฒนาแอพพลิเคชันอย่างง่าย
- รู้จักและใช้งาน User Interface
- เรียนรู้การออกแบบหน้าจอด้วย Layout แบบต่างๆ
- พัฒนาแอพพลิเคชันสำหรับรับส่งข้อความ (SMS)
- พัฒนาแอพพลิเคชันสำหรับใช้งาน GPS
- พัฒนาแอพพลิเคชันติดต่อบนเครือข่าย
- พัฒนาแอพพลิเคชันที่ทำงานกับฐานข้อมูล
- พัฒนาเกมในระบบ 2 มิติ
- แนะนำการพัฒนาแอพพลิเคชันบนเครื่องโทรศัพท์มือถือ
- รู้จักการใช้งาน Sensor
- แนะนำเครื่องมือช่วยเหลือในการพัฒนาแอพพลิเคชัน
- แนะนำการอัปโหลดแอพพลิเคชันขึ้น Google Play
- แนะนำติดตั้งแอพพลิเคชันแบบเครื่องโทรศัพท์มือถือ

ดร.จักรชัย โซธินทร์, พงษ์ศธร จันทร์ย้อย, ณัฐณิชา วะระมงคลเสศ  
บรรณาธิการ กิตตินันท์ พลสวัสดิ์

## ประวัตินักเขียน

### ● ดร.จักรชัย โซธินทร์ (IEEE Member)

- ปริญญาโท/เอก Computer Engineering, Washington University in St. Louis.
- ปริญญาตรี/โท (เกียรตินิยม) วิศวกรรมคอมพิวเตอร์, มหาวิทยาลัยเกษตรศาสตร์
- อาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ (ANT Labs), มหาวิทยาลัยขอนแก่น
- CCNP/CCDP/MCP

### ● พงษ์ศธร จันทร์ย้อย

- ปริญญาตรี (เกียรตินิยม) วิทยาการคอมพิวเตอร์, มหาวิทยาลัยขอนแก่น
- นักวิจัย NECTEC (Human Language Technology Laboratory)

### ● ณัฐณิชา วะระมงคลเสศ

- ปริญญาตรี (เกียรตินิยม) วิทยาการคอมพิวเตอร์, มหาวิทยาลัยขอนแก่น
- นักศึกษาปริญญาโท ภาควิชาวิทยาการคอมพิวเตอร์ (ANT Labs), มหาวิทยาลัยขอนแก่น

E-Book  
VERSION

จัดทำโดย IDC  
ISBN 978-616-200-361-5  
ราคา 275 บาท

