



# การเขียนโปรแกรมด้วย

# Java

## สำหรับผู้เริ่มต้น



เริ่มต้นการเขียนโปรแกรมจาก  
ขั้นพื้นฐานด้วย JDK



เขียนโค้ดบนเครื่องมือที่ชาญฉลาด  
โดยใช้ IntelliJ IDEA



ใช้เครื่องมือที่เป็นมาตรฐานของ  
จาวาอย่าง NetBeans

inspiration starts here

พร้อมตัวอย่าง แบบฝึกหัด  
และเวิร์คช็อปที่หลากหลาย

DOWNLOAD

ดาวน์โหลดโค้ดหนังสือเล่มนี้ได้ที่  
<http://www.developerthai.com>

**บัญชา ปะสีละเตสัง**

ผู้เขียนหนังสือขายดีระดับ Best Seller  
ด้าน Programming



# การเขียนโปรแกรมด้วย Java สำหรับผู้เริ่มต้น

โดย บัญชา ปะสีละเตลิ่ง

สงวนลิขสิทธิ์ตามกฎหมาย โดย บัญชา ปะสีละเตลิ่ง © พ.ศ. 2561

ห้ามคัดลอก ลอกเลียน ดัดแปลง ทำซ้ำ จัดพิมพ์ หรือกระทำการอื่นใด โดยวิธีการใดๆ ในรูปแบบใดๆ ไม่ว่าส่วนหนึ่งส่วนใดของหนังสือเล่มนี้ เพื่อเผยแพร่ในสื่อทุกประเภท หรือเพื่อวัตถุประสงค์ใดๆ นอกจากจะได้รับอนุญาต

## ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ

บัญชา ปะสีละเตลิ่ง.

การเขียนโปรแกรมด้วย Java สำหรับผู้เริ่มต้น.--กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2561.  
536 หน้า.

1. จาวา (ภาษาคอมพิวเตอร์). 2. การเขียนโปรแกรม (คอมพิวเตอร์).

I. ชื่อเรื่อง.

005.133

Barcode (e-book) : 9786160841677

# SE-ED

inspiration starts here

ผลิตและจัดจำหน่ายโดย



**บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)**  
SE-EDUCATION PUBLIC COMPANY LIMITED

เลขที่ 1858/87-90 ถนนเทพรัตน แขวงบางนาใต้ เขตบางนา กรุงเทพฯ 10260  
โทรศัพท์ 0-2826-8000

[หากมีคำแนะนำหรือติชม สามารถติดต่อได้ที่ [comment@se-ed.com](mailto:comment@se-ed.com)]

# คำนำ

Java เป็นภาษาคอมพิวเตอร์ที่มีผู้ใช้งานมากที่สุดในปัจจุบัน เนื่องจากมีโครงสร้างที่สะอาดสวย เรียนรู้ง่าย และโดยเฉพาะอย่างยิ่งการทำงานข้ามแพลตฟอร์มได้ ส่งผลให้ Java ถูกนำไปใช้อย่างแพร่หลายทั้งในภาคธุรกิจ อุตสาหกรรม เทคโนโลยีสื่อสารต่างๆ ตลอดจนการพัฒนาแอปพลิเคชันสำหรับระบบ Android อีกด้วย นอกจากนี้ภาษา Java ยังถูกเลือกใช้ในการเรียนการสอนของสถาบันการศึกษาทั่วโลก ดังนั้นการเขียนโปรแกรมด้วย Java จึงถือเป็นสิ่งสำคัญสำหรับผู้ต้องการก้าวสู่สายงานด้านการพัฒนาซอฟต์แวร์ระดับมืออาชีพที่ควรต้องเรียนรู้เอาไว้

ในหนังสือเล่มนี้จึงได้รวบรวมเนื้อหาที่จำเป็นทั้งหมดในขั้นพื้นฐาน แล้วยังมีตัวอย่าง และแบบฝึกหัดท้ายบท ตลอดจนเวิร์คช็อปที่หลากหลาย โดยใช้เครื่องมือยอดนิยมอย่าง NetBeans และ IntelliJ เพื่อให้ผู้เริ่มต้นการเขียนโปรแกรมด้วย Java ทุกคน สามารถใช้เป็นแนวทางในการศึกษาเพิ่มเติมต่อไปในอนาคตได้

Inspiration starts here

บัญชา ปะสีละเตลัง

[banchar\\_pa@yahoo.com](mailto:banchar_pa@yahoo.com)

[facebook.com/DeveloperThai](https://facebook.com/DeveloperThai)

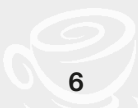


**SE-ED**

inspiration starts here

# สารบัญ

<b>บทที่ 1</b>	<b>ภาษาจาวาและเครื่องมือในการเขียนโค้ด.....</b>	<b>19</b>
1-1:	ประวัติความเป็นมาของภาษาจาวา.....	19
1-2:	ลักษณะเด่นของภาษาจาวา.....	20
1-3:	หลักการทำงานของภาษาจาวา.....	21
1-4:	รุ่นของภาษาจาวาและการติดตั้ง.....	22
1-5:	การติดตั้ง NetBeans.....	24
1-6:	การสร้างโปรเจกต์ด้วย NetBeans.....	26
1-7:	การกำหนดพอนต์สำหรับ NetBeans.....	31
1-8:	การติดตั้ง IntelliJ IDEA.....	34
1-9:	พื้นฐานการเขียนจาวาด้วย IntelliJ IDEA.....	36
1-10:	การกำหนดพอนต์สำหรับ IntelliJ IDEA.....	41
<b>บทที่ 2</b>	<b>องค์ประกอบพื้นฐานและตัวแปร .....</b>	<b>43</b>
2-1:	คีย์เวิร์ดของภาษาจาวา.....	43
2-2:	การกำหนดรูปแบบของคีย์เวิร์ดในโปรแกรม IDE.....	44
	● กรณีใช้ NetBeans.....	44
	● กรณีใช้ IntelliJ.....	45
2-3:	ลักษณะของคลาสที่ควรรู้จักในเบื้องต้น.....	46
2-4:	เมธอด main().....	49
	● กรณีใช้ NetBeans.....	49
	● กรณีใช้ IntelliJ.....	50



2-5: การกำหนดบล็อกคำสั่งด้วยเครื่องหมาย { ... }	51
● กรณีใช้ NetBeans	51
● กรณีใช้ IntelliJ	52
2-6: Statement และเครื่องหมายสิ้นสุดคำสั่ง (;)	53
2-7: การเขียนคำอธิบายโค้ด	54
● Line Comment	54
● Block Comment	55
● Doc Comment	56
2-8: การจัดรูปแบบคำอธิบายโค้ดในโปรแกรม IDE	56
● กรณีใช้ NetBeans	56
● กรณีใช้ IntelliJ	57
2-9: การรันโปรแกรม	58
● กรณีใช้ NetBeans	58
● กรณีใช้ IntelliJ	58
2-10: ข้อผิดพลาดในการเขียนโค้ด	59
2-11: ชนิดข้อมูลที่ควรรู้จักในเบื้องต้น	60
2-12: ลักษณะของตัวแปร	60
2-13: การประกาศตัวแปร	61
2-14: การกำหนดค่าให้กับตัวแปร	62
● หลักพื้นฐานของการกำหนดค่าตัวแปร	62
● การกำหนดข้อมูลชนิดตัวเลข (int และ double)	63
● การกำหนดข้อมูลชนิดสตริง (String)	63
2-15: ค่าคงที่	64

SE-ED  
inspiration starts here

### unit 3

<b>การรับและแสดงผลข้อมูล</b>	<b>65</b>
3-1: การแสดงผลด้วยเมธอด print()	65
3-2: การแสดงผลด้วยเมธอด println()	67
3-3: การขึ้นบรรทัดใหม่ด้วยอักขระ "\n"	68
3-4: การเว้นระยะแท็บด้วยอักขระ "\t"	68
3-5: การแสดงผลด้วยเมธอด printf()	70
3-6: การรับข้อมูลทางคีย์บอร์ดด้วยวิธีต่างๆ	71
● การรับข้อมูลด้วยคลาสด้าน I/O	71
● การรับข้อมูลด้วยคลาส Scanner	72
● การรับข้อมูลด้วยคลาส Console	74

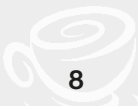
3-7: วิธีแก้ปัญหา Output Encoding ของ NetBeans.....	74
3-8: การรับข้อมูลชนิดสตริงทางคีย์บอร์ด.....	76
3-9: การรับข้อมูลชนิดตัวเลขทางคีย์บอร์ด.....	78
3-10: การแสดงผลด้วย Message Dialog.....	78
3-11: การรับข้อมูลด้วย Input Dialog.....	80
3-12: การปรับแต่งฟอนต์ของ Dialog.....	81
3-13: แบบฝึกหัด.....	82

## unที่ 4 ชนิดข้อมูลพื้นฐานและสตริง..... 85

4-1: ชนิดข้อมูลพื้นฐานของจาวา.....	85
4-2: ข้อมูลชนิดเลขจำนวนเต็ม.....	87
4-3: ข้อมูลชนิดเลขทศนิยม.....	89
4-4: ข้อมูลชนิดบูลีน.....	91
4-5: ข้อมูลชนิดชนิดอักขระ.....	92
4-6: การตรวจสอบอักขระด้วยคลาส Character.....	94
4-7: อักขระพิเศษ.....	95
4-8: ลักษณะเพิ่มเติมของข้อมูลชนิดสตริง.....	96
4-9: การเชื่อมต่อสตริง (String Concatenation).....	97
4-10: สตริงที่มีค่า empty และ null.....	98
4-11: เมธอดของคลาส String ที่ควรรู้จักในเบื้องต้น.....	99
4-12: การกำหนดข้อมูลแบบลุ่ม.....	101

## unที่ 5 โอเปอเรเตอร์และการแปลงข้อมูล..... 105

5-1: โอเปอเรเตอร์ประเภทต่างๆ.....	105
5-2: โอเปอเรเตอร์สำหรับการกำหนดค่า.....	106
5-3: โอเปอเรเตอร์ทางคณิตศาสตร์.....	107
5-4: โอเปอเรเตอร์สำหรับเพิ่มและลดค่า.....	110
5-5: โอเปอเรเตอร์สำหรับการคำนวณและกำหนดค่า.....	112
5-6: โอเปอเรเตอร์สำหรับการเปรียบเทียบ.....	114
5-7: การเปรียบเทียบสตริง.....	117
5-8: โอเปอเรเตอร์ทางตรรกะ.....	117
5-9: โอเปอเรเตอร์แบบ Ternary.....	119
5-10: ลำดับการประมวลผลของโอเปอเรเตอร์.....	121
5-11: การกำหนดค่าแบบต่อเนื่อง.....	125



5-12: การแปลงชนิดข้อมูล.....	125
● <i>Implicit Conversion</i> .....	126
● <i>Explicit Conversion</i> .....	126
5-13: กฎ 4 ข้อของการคำนวณตัวเลข.....	128
5-14: การใช้ Primitive Wrapper Class.....	130
5-15: การแปลงสตริงให้เป็นชนิดข้อมูลพื้นฐาน.....	130
5-16: Autoboxing และ Unboxing.....	133
5-17: แบบฝึกหัด.....	134

## บทที่ 6 การกำหนดเงื่อนไข ..... 135

6-1: ลักษณะพื้นฐานของคำสั่ง if.....	135
6-2: การกำหนดเงื่อนไขด้วย Comparison Operator.....	137
6-3: การกำหนดเงื่อนไขจากเมธอดที่คืนค่า boolean.....	140
6-4: การใช้ if ร่วมกับโอเปอเรเตอร์ Logical NOT (!).....	142
6-5: การใช้คำสั่ง if-else.....	143
6-6: การใช้คำสั่ง if-else if.....	145
6-7: การใช้คำสั่ง if-else if-else.....	149
6-8: การใช้คำสั่ง if ซ้อนกัน.....	150
6-9: การกำหนดหลายเงื่อนไขด้วย Logical Operator.....	153
6-10: การกำหนดค่าตัวแปรตามเงื่อนไขโดยไม่ใช่ if.....	155
6-11: การใช้คำสั่ง switch-case.....	157
6-12: แบบฝึกหัด.....	161

## บทที่ 7 การทำซ้ำแบบวงรอบ ..... 165

7-1: การใช้ลูปแบบ for ชั้นพื้นฐาน.....	165
7-2: การใช้คำสั่ง break ร่วมกับ for.....	170
7-3: การใช้คำสั่ง continue ร่วมกับ for.....	172
7-4: การกำหนดลูป for ซ้อนกัน.....	173
7-5: ลักษณะที่ควรรู้เพิ่มเติมเกี่ยวกับลูป for.....	175
7-6: การกำหนดลูป for แบบ infinite.....	177
7-7: การใช้ลูปแบบ while.....	178
7-8: การใช้ลูปแบบ do-while.....	183
7-9: แบบฝึกหัด.....	187

SE-ED

inspiration starts here



<b>บทที่ 8</b>	<b>เมธอดและฟิลด์ .....</b>	<b>189</b>
8-1:	ลักษณะของเมธอด .....	189
8-2:	เมธอดแบบไม่ส่งค่ากลับและพารามิเตอร์.....	191
	● การสร้างเมธอดแบบไม่ส่งค่ากลับ.....	191
	● ลักษณะพื้นฐานเกี่ยวกับพารามิเตอร์.....	192
	● การเรียกใช้เมธอดแบบไม่ส่งค่ากลับ.....	193
	● การกำหนดอาร์กิวเมนต์.....	194
8-3:	เมธอดแบบส่งค่ากลับ.....	196
	● การสร้างเมธอดแบบส่งค่ากลับ.....	197
	● การเรียกใช้เมธอดแบบส่งค่ากลับ.....	199
8-4:	วิธียุทธการทำงานของเมธอด.....	203
	● การออกจากเมธอดแบบไม่ส่งค่ากลับ.....	203
	● การออกจากเมธอดแบบส่งค่ากลับ.....	203
8-5:	เมธอดแบบ Overload .....	207
8-6:	ตัวแปรฟิลด์.....	209
8-7:	ขอบเขตของตัวแปรฟิลด์และตัวแปรแบบ Local.....	211
8-8:	แบบฝึกหัด.....	212
<b>บทที่ 9</b>	<b>อาร์เรย์และอ็อบเจกต์.....</b>	<b>215</b>
9-1:	ลักษณะการจัดเก็บข้อมูลแบบอาร์เรย์.....	215
9-2:	การสร้างอาร์เรย์.....	216
9-3:	การกำหนดข้อมูลให้แก่อาร์เรย์.....	218
	● การกำหนดข้อมูลให้สมาชิกทีละตัว.....	218
	● การกำหนดข้อมูลแบบค่าเริ่มต้น.....	219
	● การนำข้อมูลมาจากเมธอดของคลาสอื่นๆ.....	219
9-4:	การอ่านข้อมูลและใช้รูป for ร่วมกับอาร์เรย์.....	220
9-5:	ลักษณะที่ควรรู้เพิ่มเติมเกี่ยวกับอาร์เรย์.....	222
9-6:	การคัดลอกและเติมค่าลงในอาร์เรย์.....	224
9-7:	การส่งข้อมูลแบบอาร์เรย์ออกจากเมธอด.....	224
9-8:	การรับพารามิเตอร์แบบอาร์เรย์เข้ามาในเมธอด.....	226
9-9:	พารามิเตอร์แบบ Variadic (Variable-Length).....	226
9-10:	การส่งค่าพารามิเตอร์แบบอาร์เรย์จะเป็นแบบ Reference.....	227
9-11:	การเรียงลำดับข้อมูลในอาร์เรย์.....	229
9-12:	การค้นหาข้อมูลในอาร์เรย์.....	229



9-13: อาร์เรย์ 2 มิติ และ Ragged Array .....	230
● ลักษณะของอาร์เรย์ 2 มิติ.....	230
● ลักษณะของ Ragged Array.....	233
● การหาขนาดของอาร์เรย์.....	234
● การเข้าถึงสมาชิกในอาร์เรย์โดยใช้ลูป.....	235
9-14: การจัดเก็บข้อมูลด้วย ArrayList .....	237
9-15: การสร้างรายการแบบ Enum .....	241
● การใช้รายการของ Enum .....	243
● การแปลงสตริงให้เป็นรายการของ Enum.....	244
● การเข้าถึงรายการของ Enum ด้วยลูป for.....	245
9-16: แบบฝึกหัด.....	246

## unที่ 10 การใช้งานเพิ่มเติมเกี่ยวกับสตริง ..... 249

10-1: คลาส String .....	249
10-2: การตัดช่องว่างออกจากสตริง.....	250
10-3: การอ่านและตรวจสอบส่วนของสตริง.....	250
10-4: การค้นหาตำแหน่งของสตริง.....	254
10-5: การแทนที่สตริง.....	254
10-6: การแยกสตริงเป็นอาร์เรย์และรวมอาร์เรย์เป็นสตริง.....	255
10-7: คลาส StringBuilder และ StringBuffer.....	257
10-8: การกำหนดแพตเทิร์นของ Regular Expression.....	258
10-9: การใช้ Regular Expression ร่วมกับ Java.....	261
● การใช้ RegEx เพื่อค้นหาสตริง.....	261
● การใช้ RegEx เพื่อตัดแยกสตริง.....	261
● การใช้ RegEx เพื่อแทนที่สตริง.....	262
10-10: แบบฝึกหัด.....	263

## unที่ 11 ตัวเลขและวันเวลา..... 267

11-1: คลาส Math สำหรับการคำนวณ.....	267
11-2: การสร้างเลขสุ่มด้วย Math.random().....	268
11-3: เมธอดเพิ่มเติมของคลาส Random.....	269
11-4: การจัดรูปแบบตัวเลขด้วย NumberFormat .....	270
11-5: การจัดรูปแบบตัวเลขด้วย DecimalFormat .....	272
11-6: การใช้คลาส GregorianCalendar .....	274

11-7: การจัดการวันเวลาแบบใหม่ใน Java 8+	277
11-8: การกำหนดวันเวลาอ้างอิง	277
11-9: การอ่านค่าของวันเวลา	278
● การใช้เมธอด <code>getXxx()</code>	278
● การใช้เมธอด <code>get()</code> ร่วมกับ <code>ChronoField</code>	279
11-10: การเปลี่ยนแปลงและแก้ไขวันเวลา	280
● การเปลี่ยนแปลงบางส่วนของวันเวลาด้วย <code>withXxx()</code>	280
● การเพิ่มหรือลดวันเวลาด้วย <code>plusXxx()</code> และ <code>minusXxx()</code>	281
● การใช้ <code>plus()</code> และ <code>minus()</code> ร่วมกับ <code>ChronoUnit</code>	281
11-11: การหาระยะห่างของวันเวลา	283
● การใช้เมธอด <code>ChronoUnit.between()</code>	283
● การใช้เมธอด <code>LocalXxx.until()</code>	284
11-12: การตรวจสอบและเปรียบเทียบวันเวลา	286
11-13: การใช้คลาสสำหรับปี พ.ศ. ของไทย	286
11-14: การจัดรูปแบบวันเวลา	288
● การใช้รูปแบบ <code>Predefined Formatter</code>	288
● การใช้ <code>User-define Pattern</code>	289
● การจัดรูปแบบวันเวลาด้วยวิธีดั้งเดิม	290
● การแปลงสตริงเป็นวันเวลาอ้างอิงด้วย <code>parse()</code>	291
11-15: แบบฝึกหัด	291
<b>un ที่ 12</b> คลาสและออบเจ็กต์	293
12-1: แนวคิดเกี่ยวกับคลาสและออบเจ็กต์	293
12-2: การสร้างคลาสเพิ่มเติม	294
● การสร้างไฟล์ของคลาสใหม่	295
● ลักษณะของคลาสที่สร้างเพิ่มเติม	296
● การเซต <code>Main Class</code> ของแอปพลิเคชัน	297
12-3: ออบเจ็กต์หรืออินสแตนซ์ของคลาส	297
12-4: โมดิไฟเออร์แบบ <code>public</code> และ <code>private</code>	300
● โมดิไฟเออร์แบบ <code>public</code>	300
● โมดิไฟเออร์แบบ <code>private</code>	301
● การไม่ระบุโมดิไฟเออร์ใดๆ	302
12-5: การเข้าถึงตัวแปรฟิลด์จากคลาสอื่น	303
● หลักการกำหนดโมดิไฟเออร์ของตัวแปรฟิลด์	303



● การเข้าถึงตัวแปรฟิลด์ด้วย Accessor Method.....	304
● การสร้างฟิลด์แบบ Read Only.....	306
12-6: คอนสตรักเตอร์ของคลาส.....	307
● ลักษณะของคอนสตรักเตอร์.....	307
● การเรียกใช้คอนสตรักเตอร์.....	309
● การเรียกคอนสตรักเตอร์ด้วยตัวเอง.....	310
● การตรวจสอบความถูกต้องของข้อมูลที่ได้รับผ่านพารามิเตอร์.....	311
12-7: โมดิไฟเออร์แบบ static.....	313
12-8: คลาสแบบ Singleton.....	316
12-9: พารามิเตอร์แบบ By Reference.....	317
12-10: Method Chaining.....	319

## unที่ 13 การสืบทอดและแพ็คเกจ ..... 321

13-1: หลักการสืบทอด.....	321
● การสืบทอดระหว่างคลาส.....	321
● การเพิ่มเติมความสามารถให้กับ Subclass.....	323
13-2: คอนสตรักเตอร์กับการสืบทอด.....	324
● คอนสตรักเตอร์ของ Superclass ที่ถูกเรียกโดยอัตโนมัติ.....	325
● การอ้างถึงคอนสตรักเตอร์ของ Superclass ด้วยคีย์เวิร์ด super.....	326
● การเกิด Constructor Chaining.....	327
13-3: การ Override สมาชิกของ Superclass.....	328
● หลักการทำ Shadow ตัวแปรฟิลด์.....	329
● หลักการ Override เมธอด.....	330
● การใช้แอนโนเทชัน @Override.....	331
13-4: การอ้างถึงสมาชิกของ Superclass ด้วย super.....	333
13-5: โมดิไฟเออร์แบบ protected.....	334
13-6: โมดิไฟเออร์แบบ final.....	335
13-7: Anonymous Class.....	336
13-8: Upcast และ Dynamic Binding.....	337
13-9: การเกิด Polymorphism.....	340
13-10: Downcast และโอเปอเรเตอร์ instanceof.....	341
13-11: คลาสแบบ Abstract.....	344
● ข้อกำหนดของคลาสที่เป็น Abstract.....	345
● คลาสที่สืบทอดมาจากแอสแตรคคลาส.....	346



13-12: การสร้างแพ็คเกจ.....	348
● ตำแหน่งของโปรเจกต์และแพ็คเกจ.....	348
● การสร้างแพ็คเกจบน NetBeans.....	348
● การสร้างแพ็คเกจบน IntelliJ.....	350
13-13: การใช้คลาสระหว่างแพ็คเกจด้วยคำสั่ง import.....	351
● การกำหนดตำแหน่งคลาสด้วยคำสั่ง import.....	351
● โมดิไฟเออร์กับการเข้าถึงคลาสในแพ็คเกจ.....	353
● การเพิ่มคำสั่ง import แบบอัตโนมัติ (เฉพาะ IntelliJ).....	353

## บทที่ 14 การจัดการข้อผิดพลาดและไฟล์..... 355

14-1: ลักษณะของ Exception Handling .....	355
14-2: การกำหนดบล็อก try-catch .....	356
14-3: ชนิดของ Exception .....	359
14-4: การดัก Exception หลายชนิด.....	360
14-5: การกำหนดบล็อก finally.....	361
14-6: การใช้ try-with-resources ใน Java 7+ .....	363
14-7: การส่งต่อข้อผิดพลาดด้วย throw และ throws.....	364
● การใช้คำสั่ง throw.....	365
● การใช้คำสั่ง throws.....	366
14-8: Checked และ Unchecked Exception .....	367
14-9: การเขียนไฟล์.....	369
14-10: การเพิ่มข้อมูลลงในไฟล์.....	372
14-11: การอ่านไฟล์.....	374
14-12: การจัดการไฟล์และไดเรกทอรีด้วยคลาส File.....	375
14-13: แบบฝึกหัด.....	377

## บทที่ 15 อินเทอร์เฟซและแลมบ์ดา..... 379

15-1: การสร้างอินเทอร์เฟซ .....	379
15-2: การสืบทอดระหว่างอินเทอร์เฟซ .....	382
15-3: การขยายการทำงานของอินเทอร์เฟซ .....	382
● การขยายการทำงานจากหลายอินเทอร์เฟซ.....	383
● การใช้ทั้งการสืบทอดและอิมพลีเมนต์ร่วมกัน.....	383
15-4: การใช้อินเทอร์เฟซแบบ Polymorphism .....	384
15-5: ข้อกำหนดใหม่ของอินเทอร์เฟซใน Java 8+.....	386



15-6: Functional Interface ใน Java 8+.....	387
15-7: ขยายอินเทอร์เฟซด้วยรูปแบบ Anonymous Class .....	388
15-8: Lambda Expression ใน Java 8+.....	389
● รูปแบบพื้นฐานของแลมบ์ดา.....	389
● เขียนแลมบ์ดาแบบ Implicit Type และ Explicit Type.....	390
● เทคนิคเพิ่มเติมของการเขียน Lambda Expression.....	391
15-9: การใช้ Lambda ร่วมกับ Functional Interface.....	392
● การใช้ Lambda แทน Anonymous Class.....	392
● การผ่าน Lambda Expression ให้กับพารามิเตอร์.....	393

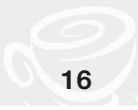
## บทที่ 16 Java Swing (ตอนที่ 1)..... 397

16-1: การสร้าง GUI ในยุคต่างๆ .....	397
16-2: พื้นฐานการสร้าง GUI ด้วย Java Swing.....	398
● การสร้าง Frame .....	398
● การเพิ่มคอมโพเนนต์ลงบนเฟรม.....	400
16-3: การกำหนดพอนต์ของคอมโพเนนต์.....	401
16-4: การจัด Layout แบบต่างๆ.....	403
● การจัดโครงร่างแบบ Flow Layout.....	403
● การจัดโครงร่างแบบ Border Layout.....	404
● การจัดโครงร่างแบบ Grid Layout.....	404
16-5: การสร้าง Subcontainer ด้วย JPanel.....	406
16-6: การจัดการอีเวนต์ของ Swing .....	407
● หลักการและชนิดของอีเวนต์.....	408
● จัดการอีเวนต์ด้วยวิธี Implement Interface.....	410
● จัดการอีเวนต์ด้วยวิธี Anonymous (Inner) Class.....	411
● จัดการอีเวนต์โดยใช้ Adapter.....	412
● จัดการอีเวนต์โดยใช้ Lambda (สำหรับ Java 8+).....	413
● การตรวจสอบข้อมูลจากอีเวนต์ที่เกิดขึ้น.....	415
● การใช้ตัวจัดการอีเวนต์ร่วมกัน.....	416
16-7: เมธอดพื้นฐานของคอมโพเนนต์ชนิดต่างๆ .....	416
16-8: คอมโพเนนต์ JButton และ JLabel .....	417
● การใช้ JButton .....	417
● การใช้ JLabel.....	417
16-9: คอมโพเนนต์ JTextField และ JTextArea.....	420

● การใช้ JTextField และ JPasswordField.....	420
● การใช้ JTextArea.....	421
16-10: คอมโพเนนต์ JCheckBox และ JRadioButton.....	424
● การใช้ JCheckBox.....	424
● การใช้ JRadioButton.....	425
16-11: คอมโพเนนต์ JComboBox และ JList.....	428
● การใช้ JComboBox.....	428
● การใช้ JList.....	429
16-12: แบบฝึกหัด.....	432

## บทที่ 17 Java Swing (ตอนที่ 2)..... 433

17-1: เทคนิคการใช้ JOptionPane เพิ่มเติม.....	433
● Message Dialog.....	433
● Confirm Dialog.....	434
● Input Dialog.....	435
17-2: การกำหนดสีของคอมโพเนนต์.....	435
17-3: คอมโพเนนต์ JColorChooser.....	437
17-4: คอมโพเนนต์ JFileChooser.....	438
17-5: การใช้ BorderLayout.....	441
17-6: คอมโพเนนต์ JTable.....	444
17-7: คอมโพเนนต์ JTabbedPane.....	445
17-8: การใช้ GridBag Layout.....	448
17-9: คอมโพเนนต์ ImageIcon.....	452
● การจัดเก็บภาพใน NetBeans.....	452
● การจัดเก็บภาพใน IntelliJ.....	452
● การนำภาพมาใช้กับคอมโพเนนต์.....	453
17-10: การสร้างเมนู.....	453
● พื้นฐานการสร้างรายการเมนู.....	454
● การใส่รูปภาพให้กับเมนู.....	455
● การกำหนด Mnemonic และ Shortcut.....	455
● การสร้าง Submenu.....	456
17-11: การสร้างทูลบาร์.....	456
17-12: แบบฝึกหัด.....	459



<b>unที่ 18</b>	<b>JavaFX เบื้องต้น.....</b>	<b>461</b>
18-1:	ความเป็นมาของ JavaFx .....	461
18-2:	พื้นฐานการสร้างแอปพลิเคชันด้วย JavaFx .....	462
	● องค์ประกอบพื้นฐานของ JavaFx Application.....	462
	● เมธอดพื้นฐานของ Stage และแพ็คเกจ javafx .....	464
18-3:	การใช้ Pane ชนิดต่างๆ.....	464
	● เมธอดของ Pane และการกำหนดค่าที่ควรรู้จักเบื้องต้น.....	464
	● การใช้ FlowPane .....	466
	● การใช้ BorderPane .....	466
	● การใช้ GridPane .....	467
	● การใช้ HBox และ VBox .....	469
	● การใช้ Pane หลายชนิดร่วมกัน.....	470
18-4:	การจัดการอีเวนต์ .....	470
	● ชนิดและหลักการของอีเวนต์.....	471
	● จัดการอีเวนต์ด้วยวิธี Lambda ใน Java 8+.....	471
18-5:	เมธอดพื้นฐานของคอนโทรล .....	472
18-6:	การใช้ Button และ Label .....	473
18-7:	การใช้ TextField และ PasswordField.....	474
18-8:	การใช้ TextArea.....	475
18-9:	การใช้ CheckBox และ RadioButton.....	477
18-10:	การใช้ ComboBox.....	478
18-11:	การใช้ Alert สำหรับแสดงไดอะล็อก .....	481
18-12:	แบบฝึกหัด.....	482

<b>unที่ 19</b>	<b>รวมเวิร์คช็อป ชุดที่ 1 .....</b>	<b>483</b>
19-1:	อัตราการผ่อนชำระงวดรถ .....	483
	● หลักการคำนวณงวดรถ.....	483
	● ลักษณะการจัดวางคอมพิวเตอร์.....	484
	● ขั้นตอนเขียนโค้ด .....	485
19-2:	โปรแกรมคำนวณโอกาสตั้งครรภ์.....	489
	● หลักการตรวจสอบวันที่มีโอกาสตั้งครรภ์.....	489
	● ลักษณะการจัดวางคอมพิวเตอร์.....	489
	● ขั้นตอนเขียนโค้ด .....	490



19-3: การแปลงหน่วยชนิดต่างๆ.....	494
● หลักการแปลงหน่วยระยะทาง .....	494
● หลักการแปลงหน่วยขนาดไฟล์.....	496
● การแปลงหน่วยอื่นๆ.....	497
● ลักษณะการจัดวางคอมพิวเตอร์.....	498
● คลาสสำหรับแปลงหน่วยของระยะทาง .....	499
● คลาสสำหรับแปลงหน่วยของขนาดไฟล์.....	500
● ลักษณะโค้ดคลาสหลัก.....	501

## บทที่ 20 รวมเวิร์คช็อป ชุดที่ 2..... 507

20-1: ระบบข้อมูลคลังสินค้าเบื้องต้น.....	507
● ลักษณะการจัดการข้อมูลคลังสินค้า .....	508
● การจัดวางคอมพิวเตอร์และจัดเตรียมไฟล์ .....	509
● ขั้นตอนการเขียนโค้ด .....	510
20-2: เครื่องคิดเลข .....	518
● หลักการโดยสังเขปของเครื่องคิดเลข .....	518
● การจัดวางคอมพิวเตอร์.....	519
● ตัวแปรและเมธอดส่วนกลาง.....	521
● หลักการของปุ่ม C และ DEL.....	522
● หลักการของปุ่มตัวเลขและทศนิยม.....	524
● หลักการของปุ่มเครื่องหมายและการคำนวณ.....	526
● หลักการของปุ่ม Memory.....	529
20-3: ปฏิทินไทย.....	530
● ลักษณะการจัดโครงสร้าง .....	530
● หลักการแสดงวันที่ของปฏิทิน.....	531
● การเลื่อนระหว่างเดือนต่างๆ.....	532
● ขั้นตอนการเขียนโค้ด .....	533





**SE-ED**

inspiration starts here

# ภาษาจาวาและเครื่องมือ ในการเขียนโค้ด

# 1

Java เป็นภาษาคอมพิวเตอร์ที่ถูกนำไปใช้พัฒนาซอฟต์แวร์กันอย่างแพร่หลาย ไม่ว่าจะเป็นทางด้านธุรกิจ อุตสาหกรรม หรือเทคโนโลยีสารสนเทศ และเป็นภาษาที่ใช้ในการเรียนการสอนของสถาบันการศึกษาทั่วโลก ดังนั้นการเรียนรู้ภาษา Java จึงมีความสำคัญต่อบุคคลที่จะก้าวเข้าสู่สายงานด้านไอทีต่อไปในอนาคต โดยในบทนี้จะกล่าวถึงสิ่งที่เราควรรู้ตั้งแต่เริ่มแรก เช่น การติดตั้งเครื่องมือในการเขียนโปรแกรมทั้ง NetBeans และ IntelliJ รวมถึงการเซตค่าที่จำเป็นต่างๆ เพื่อเป็นพื้นฐานสำหรับการใช้งานในบทต่อไป

## 1-1: ประวัติความเป็นมาของภาษาจาวา

จาวา เป็นภาษาที่สร้างขึ้นโดยบริษัท Sun Microsystems เพื่อแก้ไขข้อบกพร่องต่างๆ ของภาษา C/C++ และกลายมาเป็นภาษายอดนิยมที่ใช้กันอย่างแพร่หลายทั่วโลก ซึ่งมีประวัติความเป็นมาและวิวัฒนาการโดยสังเขปดังต่อไปนี้

- ทีมนักพัฒนาของบริษัท Sun Microsystems พบข้อบกพร่องหลายอย่างเมื่อใช้ภาษา C/C++ ในการเขียนโปรแกรมเพื่อควบคุมการทำงานของอุปกรณ์ไฟฟ้าและอิเล็กทรอนิกส์ จึงได้ริเริ่มแนวคิดที่จะสร้างภาษาใหม่ขึ้นมาใช้ทดแทน
- ภาษาใหม่ที่สร้างขึ้น จะต้องมึลักษณะที่สำคัญในเบื้องต้นดังนี้คือ
  - มีโครงสร้างที่ไม่ซับซ้อน เรียนรู้ได้ง่าย
  - มีขนาดเล็ก ใช้หน่วยความจำไม่มาก
  - แก้ไขข้อบกพร่องและใช้ทดแทนภาษา C/C++ ได้



- สามารถทำงานบนอุปกรณ์อิเล็กทรอนิกส์ที่ไม่มีระบบปฏิบัติการได้
- นำไปใช้งานกับอุปกรณ์ที่มีหน่วยประมวลผลแตกต่างกันได้
- โครงการได้เริ่มต้นขึ้นในปี ค.ศ.1991 ด้วยการจัดตั้งทีมงานที่ชื่อ Green ซึ่งมี James Gosling เป็นแกนนำ
- หลังจากใช้เวลาในการพัฒนานาน 18 เดือน จึงได้ภาษาใหม่เกิดขึ้น โดยในตอนแรกนั้นได้ตั้งชื่อภาษานี้ว่า Oak ตามชื่อต้นไม้ที่อยู่ใกล้ๆ กับสำนักงานของบริษัท
- ภาษา Oak ถูกนำไปใช้เป็นเครื่องมือในการพัฒนาอุปกรณ์ด้านอิเล็กทรอนิกส์หลายโครงการ แต่ทั้งหมดนั้นไม่ประสบความสำเร็จด้านการตลาดเท่าที่ควร
- ในปี ค.ศ. 1995 ภาษา Oak มีปัญหาเรื่องสิทธิบัตร เนื่องจากชื่อนี้ไปซ้ำกับผลิตภัณฑ์ของบริษัทอื่น ดังนั้นจึงได้เปลี่ยนชื่อเป็น Java ในเวลาต่อมา พร้อมกับใช้รูปถ้วยกาแฟเป็นสัญลักษณ์ เนื่องจากคำว่า Java ในภาษาอังกฤษแบบอเมริกันเป็น Slang ที่หมายถึง กาแฟ (แหล่งปลูกกาแฟบนเกาะชวาในประเทศอินโดนีเซีย)
- ในช่วงแรกๆ นั้นภาษา Java ต้องพบกับความยากลำบาก เพราะแทบไม่มีใครสนใจนำไปใช้งาน แม้บริษัท Sun จะใช้ความพยายามอย่างหนักก็ตาม
- จุดพลิกผันที่ทำให้จาวาเป็นที่รู้จักไปทั่วโลกคือ การที่บริษัท Netscape Communication ได้สนับสนุน Java Applet ซึ่งเป็นการสร้างโปรแกรมขนาดเล็กที่สามารถทำงานบนเว็บเบราว์เซอร์ได้ และนับเป็นจุดเริ่มต้นที่ทำให้ภาษาจาวาถูกนำไปใช้อย่างแพร่หลาย
- บริษัท Netscape ได้นำหลักการบางส่วนของจาวา มาเป็นต้นแบบในการพัฒนาภาษาสคริปต์ เพื่อขยายขีดจำกัดของ HTML ให้ทำงานแบบ Dynamic ได้ จึงเรียกภาษาดังกล่าวว่า JavaScript
- ปี ค.ศ. 2010 บริษัท Sun Microsystems ถูกซื้อกิจการโดยบริษัท Oracle ทำให้ลิขสิทธิ์ภาษาจาวาตกอยู่ภายใต้การควบคุมของ Oracle ไปด้วย

inspiration starts here

## 1-2: ลักษณะเด่นของภาษาจาวา

ปัจจุบันจาวาได้กลายเป็นภาษาคอมพิวเตอร์ที่มีผู้ใช้งานมากที่สุดในอีกภาษาหนึ่งของโลก ทั้งในภาคธุรกิจและอุตสาหกรรมต่างๆ เช่น การเงินการธนาคาร, สื่อสารโทรคมนาคม, ไอที, อิเล็กทรอนิกส์, มัลติมีเดีย, เกม ฯลฯ ซึ่งจุดเด่นบางประการที่ทำให้จาวาได้รับความนิยมอย่างกว้างขวาง มีดังนี้คือ

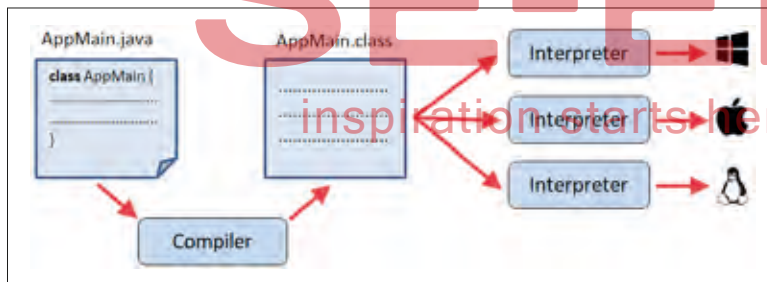
- มีรูปแบบที่สละสลวยและง่ายต่อการเรียนรู้ จึงสามารถศึกษาให้เข้าใจได้ในระยะเวลาอันสั้น
- จาวาจะทำงานอยู่บน Java Virtual Machine (JVM) ซึ่งสามารถนำไปติดตั้งในระบบหรืออุปกรณ์ที่หลากหลาย ดังนั้นจาวาจึงสามารถทำงานข้ามแพลตฟอร์มได้ (Platform Independent) ถ้าอุปกรณ์เหล่านั้นมี JVM ติดตั้งอยู่ ดังคำกล่าวที่ว่า **Write Once Run Anywhere**

- จาวามีความคงทนและความปลอดภัยในการทำงานมากกว่า เพราะได้แก้ไขข้อบกพร่องหลายอย่างในภาษา C++ เช่น Pointer, Garbage Collection, การจัดการหน่วยความจำ รวมถึงหลักการด้าน OOP ของ C++ ที่อาจก่อให้เกิดปัญหา และมีกลไกในการป้องกันข้อผิดพลาดต่างๆ ที่อาจเป็นสาเหตุให้แอปพลิเคชันต้องหยุดการทำงานลงกลางครันได้
- จาวาเป็นภาษาที่ยึดหลักการเขียนโปรแกรมแบบ OOP อย่างเข้มงวด ดังนั้นเราจึงสามารถใช้ประโยชน์จากข้อกำหนดต่างๆ ของ OOP เพื่อพัฒนาแอปพลิเคชันได้อย่างเต็มที่
- จาวารองรับการทำงานพร้อมๆ กันของหลายแอปพลิเคชัน จึงสามารถใช้หน่วยประมวลผลได้อย่างเต็มประสิทธิภาพ
- จาวาทำงานโดยไม่ขึ้นกับสถาปัตยกรรมของระบบใดๆ

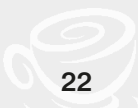
และสิ่งที่สำคัญอีกประการหนึ่งคือ จาวาถูกเลือกให้เป็นภาษาในการสร้างแอปพลิเคชันบนระบบ Android ดังนั้นผู้ที่ต้องการจะไปสู่การเป็นนักพัฒนาแอปพลิเคชันสำหรับแอนดรอยด์ ก็จำเป็นต้องมีพื้นฐานการเขียนจาวามาก่อนด้วย

### 1-3: หลักการทำงานของภาษาจาวา

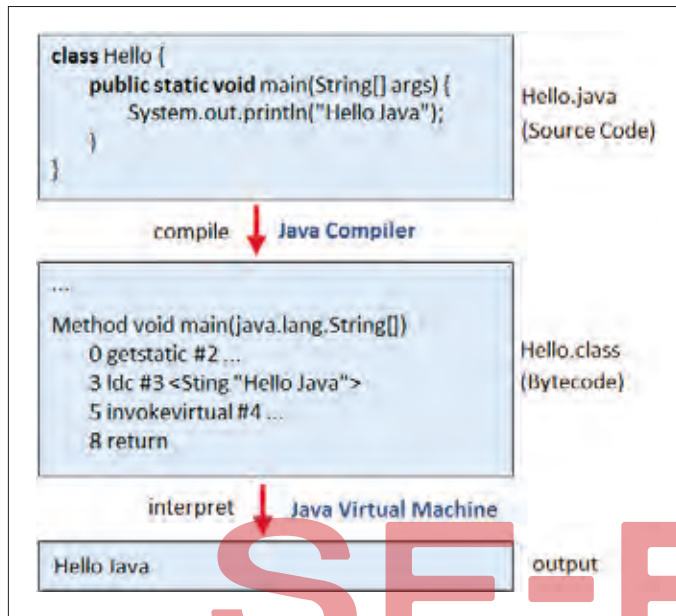
โปรแกรมที่พัฒนาด้วยภาษาจาวา ไม่ว่าจะเขียนบนแพลตฟอร์มใด เช่น Windows, MacOS, Linux ก็สามารถนำไปใช้งานบนแพลตฟอร์มอื่นๆ ได้ทันที แต่การที่จะทำงานแบบข้ามแพลตฟอร์มในลักษณะดังกล่าวได้ ต้องอาศัยการประมวลผลคำสั่งใน 2 รูปแบบร่วมกันคือ



- **การคอมไพล์ (Compile)** โดยโค้ดที่เขียนเป็นภาษาจาวา (มีส่วนขยายเป็น `.java`) จะถูกคอมไพล์ออกมาเป็น **Java Bytecode** (มีส่วนขยายเป็น `.class`) ซึ่งเป็นโค้ดมาตรฐานในการทำงานข้ามแพลตฟอร์มของจาวา
- **การแปลคำสั่ง (Interpret)** เมื่อนำ Java Bytecode ไปรันบนแพลตฟอร์มใดก็ตามที่มี Java Virtual Machine สำหรับแพลตฟอร์มนั้นอยู่ Java Bytecode นั้นจะถูกแปลคำสั่ง (Interpret) ให้ทำงานตามที่กำหนด



สำหรับในส่วนของการเขียนโปรแกรม เราต้องเริ่มจากการเขียนโค้ดจาวาแล้วบันทึกไฟล์นั้นเป็น .java (เช่น Hello.java) หลังจากนั้นเมื่อสั่งรัน (Run) โค้ดจะถูกคอมไพล์ไปเป็น Bytecode ซึ่งเราจะได้ไฟล์ .class ที่เป็นชื่อเดียวกับไฟล์ .java แต่มีส่วนขยายเป็น .class เพิ่มขึ้นมา (เช่น Hello.class) แล้วไฟล์ .class ที่ได้นี้จะถูกแปลคำสั่ง (Interpret) โดย Java Virtual Machine เพื่อทำงานตามโค้ดที่เขียนเอาไว้ในไฟล์ .java ซึ่งหน้าที่ของเราก็คือ การเขียนโค้ดแล้วสั่ง Run จากนั้นกระบวนการทั้งหมด (Compile/Interpret) จะดำเนินไปเองโดยอัตโนมัติ ตามขั้นตอนในภาพ



SE-ED  
inspiration starts here

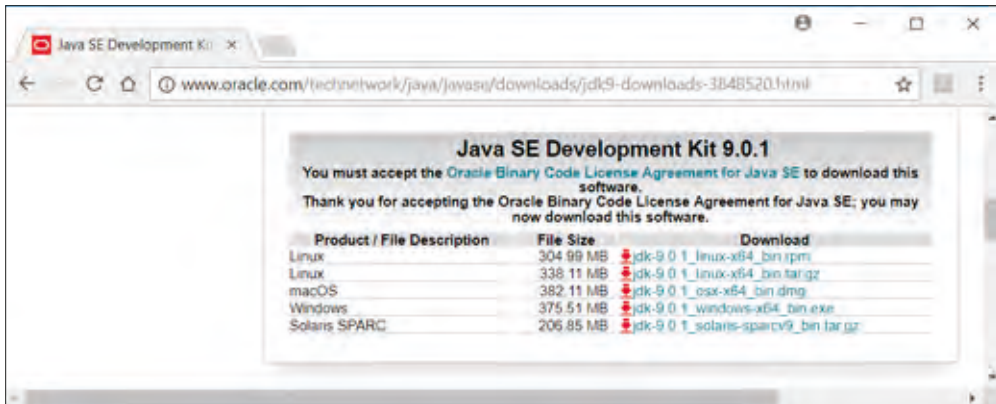
## 1-4: รุ่นของภาษาจาวาและการติดตั้ง

ภาษาจาวาแบ่งออกเป็นรุ่นย่อยๆ ตามระดับการใช้งานที่แตกต่างกัน ดังนี้คือ

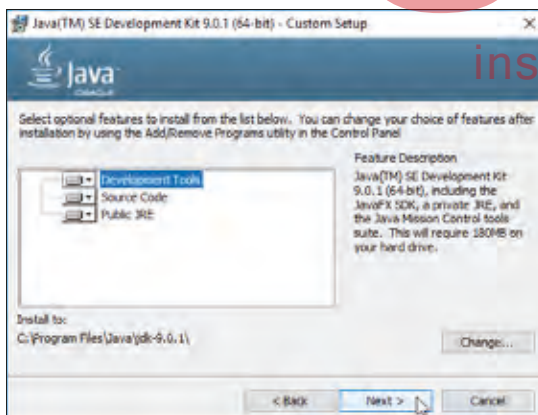
- **Java Standard Edition (JSE)** เป็นรุ่นมาตรฐานที่ประกอบด้วยเครื่องมือในการพัฒนาแอปพลิเคชันสำหรับใช้งานทั่วไป
- **Java Enterprise Edition (JEE)** เป็นรุ่นที่ประกอบด้วยเครื่องมือและบริการสำหรับธุรกิจหรือองค์กรขนาดใหญ่ รวมถึงการทำงานทางฝั่งเซิร์ฟเวอร์ (Server-Side) ได้แก่ Servlets, JavaServer Pages (JSP) และ JavaServer Faces (JSF)
- **Java Micro Edition (JME)** เป็นรุ่นที่ถูกลดขนาดให้เล็กลง สำหรับใช้ในอุปกรณ์เคลื่อนที่ (Mobile Device) เช่น โทรศัพท์มือถือ หรืออุปกรณ์อิเล็กทรอนิกส์บางชนิด

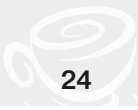
แม้ว่า JEE จะมีความสามารถที่ครอบคลุมรุ่นอื่นๆ ทั้งหมด แต่ก็มีความใหญ่เกินไป จึงไม่เหมาะที่จะใช้เป็นเครื่องมือในการศึกษาสำหรับผู้เริ่มต้น ดังนั้นในหนังสือเล่มนี้จึงเลือกกลุ่ม JSE มาประกอบการเรียนรู้ทั้งหมด ซึ่งเราต้องนำมาติดตั้งลงในระบบก่อนจึงจะใช้งานได้ โดยชุดเครื่องมือในการพัฒนาจาวาจะเรียกว่า Java Development Kit (JDK) และเป็นสิ่งแรกที่เรามาติดตั้งลงไป ดังขั้นตอนต่อไปนี้

1. สามารถดาวน์โหลด JDK ได้ที่เว็บไซต์ <http://www.oracle.com/technetwork/java/javase/downloads/> โดยเลือกกลุ่ม Standard Edition (JSE) และเลือกชนิดไฟล์ติดตั้งให้ตรงกับระบบปฏิบัติการที่เราใช้งานอยู่ ซึ่งเวอร์ชันล่าสุดในขณะนี้เขียนหนังสือเล่มนี้คือ Java 9 (JDK 9)



2. เมื่อดาวน์โหลดเสร็จให้ดับเบิลคลิกที่ไฟล์ติดตั้ง หลังจากนั้นก็คลิกปุ่ม Next ไปเรื่อยๆ จนกว่ากระบวนการจะเสร็จสิ้น ซึ่งตำแหน่งการติดตั้ง สำหรับระบบ Windows ตามปกติจะอยู่ที่ C:\Program Files\Java\





หลังการติดตั้ง หากจะใช้ตามแนวทางของหนังสือเล่มนี้ ก็ไม่จำเป็นต้องเซตค่าใดๆ เพราะเราไม่ใช้งานผ่าน JDK โดยตรง แต่จะใช้ผ่านโปรแกรมประเภท IDE แทน ทั้งนี้เนื่องจาก JDK นั้นไม่มี Editor สำหรับเขียนโค้ดมาให้ ซึ่งเราอาจใช้โปรแกรมพวก Text Editor เช่น Notepad ในการเขียนโค้ด แต่การสั่งงานต่างๆ ก็ต้องไปเขียนคำสั่งแบบ Command Line จึงเป็นวิธีการที่ยุ่งยากจนเกินไป ดังนั้นในหนังสือเล่มนี้จึงจะเขียนโค้ดและทดสอบการทำงานของจาวาผ่านโปรแกรมประเภท IDE (Integrated Development Environment) เพราะมีเครื่องมือและตัวช่วยมาให้ใช้งานอย่างครบครัน โดยจะนำเสนอแนวทางการใช้ IDE จำนวน 2 ตัว ที่โดดเด่นและได้รับความนิยมสูงสุดในปัจจุบันคือ NetBeans และ IntelliJ ซึ่งผู้อ่านจะเลือกใช้ตัวใดก็ได้ สำหรับวิธีการติดตั้งและแนวทางการใช้งาน IDE เหล่านี้ในเบื้องต้น จะกล่าวถึงหัวข้อต่อไป

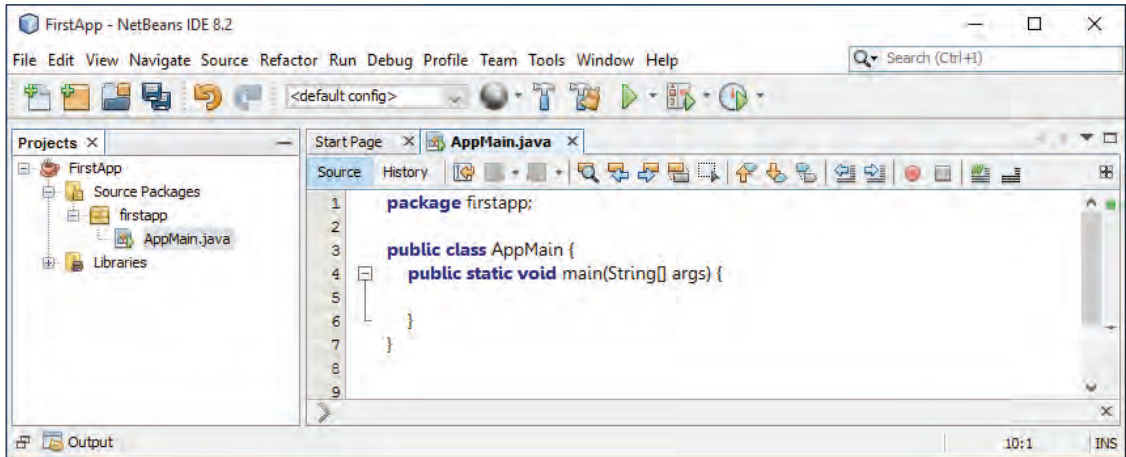
## 1-5: การติดตั้ง NetBeans

SE-ED  
Inspiration starts here

NetBeans เป็นโปรแกรมประเภท IDE ที่มีผู้ใช้งานเป็นจำนวนมาก ซึ่งจุดเด่นของ NetBeans มีดังนี้คือ

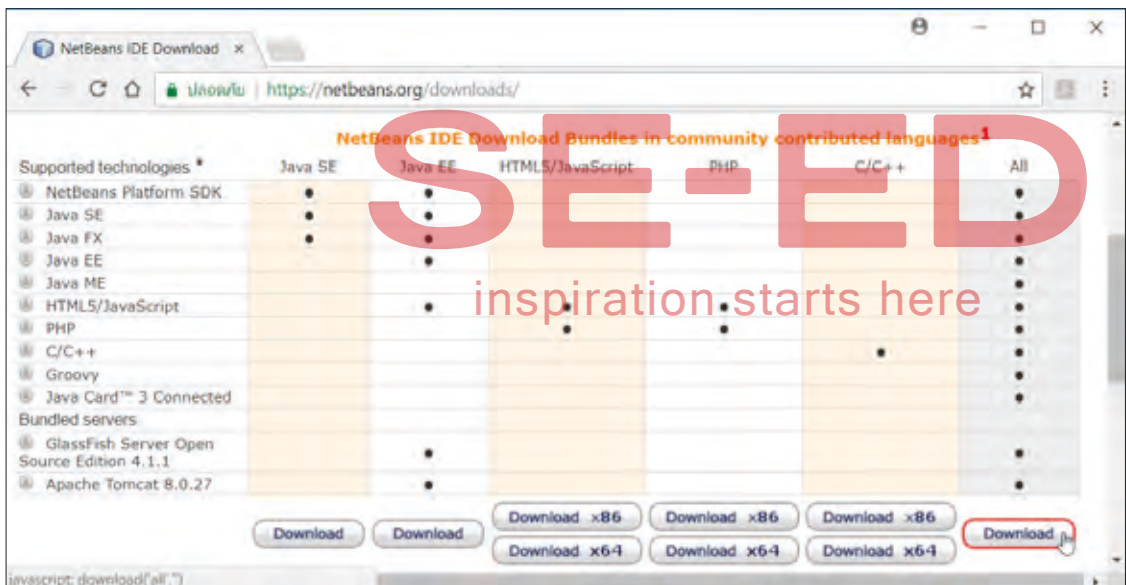
- เครื่องมือต่างๆ มีขั้นตอนการใช้งานที่ง่าย โดยไม่มีลักษณะที่ยุ่งยากหรือซับซ้อน
- มีเครื่องมือสำหรับพัฒนาแอปพลิเคชันด้วยจาวามาให้อย่างครบถ้วน
- ได้รับการสนับสนุนโดยตรงจากเจ้าของภาษาตั้งแต่ยุค Sun Microsystems มาจนถึง Oracle ดังนั้นจึงมีระดับความน่าเชื่อถือมากกว่า
- เป็นโปรแกรมที่แจกให้ใช้ฟรี โดยไม่มีรุ่น Commercial เหมือนกับ IDE บางตัว
- นอกจากจะใช้เขียนจาวาแล้ว ยังสามารถใช้เขียนโปรแกรมประเภทเว็บ เช่น PHP, HTML หรืออื่นๆ ได้อีกด้วย
- เหมาะอย่างยิ่งสำหรับผู้เริ่มต้น หรือผู้ที่ต้องการเน้นศึกษาจาวาเป็นหลัก

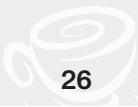




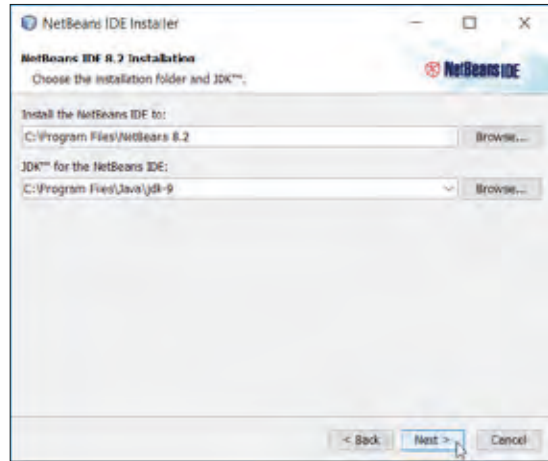
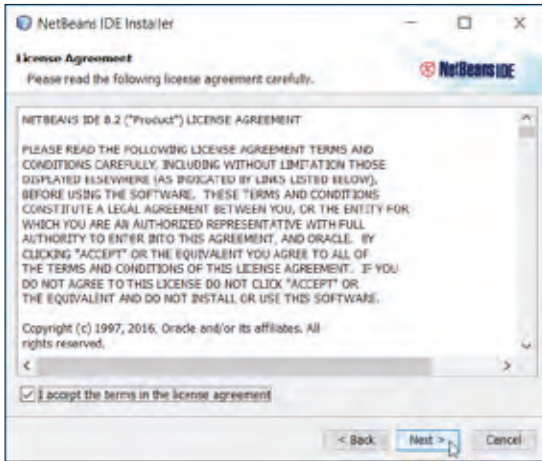
ถ้าเราจะนำ NetBeans มาเป็นเครื่องมือในการเขียนจาวา ก็ต้องติดตั้ง JDK ดังที่ได้กล่าวมาในหัวข้อที่แล้วก่อน หลังจากนั้นจึงค่อยติดตั้ง NetBeans ซึ่งมีขั้นตอนโดยสังเขปดังต่อไปนี้

1. สามารถดาวน์โหลด NetBeans ได้จากเว็บไซต์ <https://netbeans.org/downloads/>





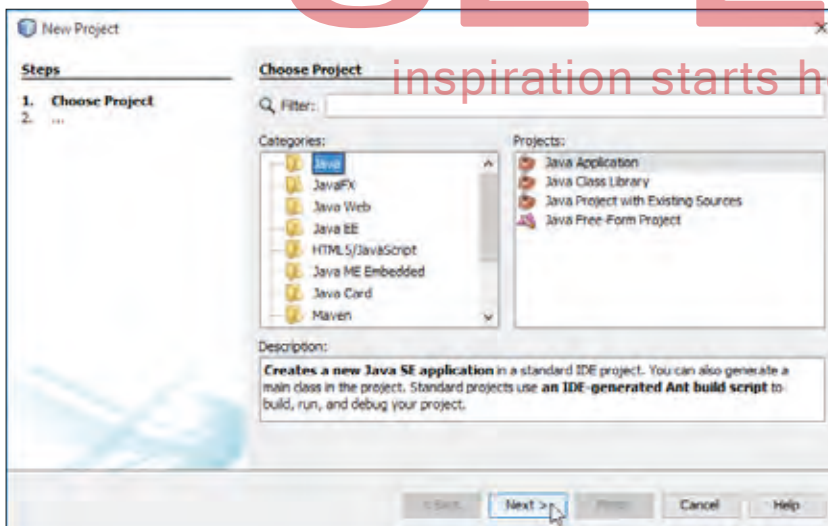
## 2. การติดตั้ง NetBeans ก็ทำเหมือนกับโปรแกรมอื่นๆ ไป โดยสิ่งที่ต้องกำหนดในบางขั้นตอนมีดังนี้



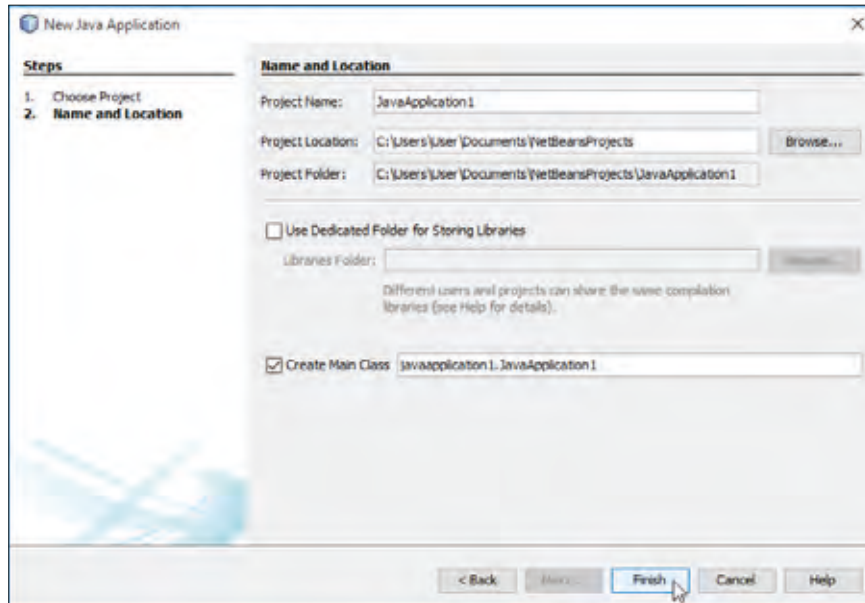
## 1-6: การสร้างโปรเจกต์ด้วย NetBeans

การเขียนจาวาด้วยโปรแกรมประเภท IDE ไม่ว่าตัวใดก็ตาม จะต้องเริ่มที่การสร้างโปรเจกต์ (Project) ใหม่เสมอ ซึ่งก็คือแอปพลิเคชันอย่างใดอย่างหนึ่งนั่นเอง โดยให้เปิดเข้าสู่ NetBeans แล้วทำตามขั้นตอนดังต่อไปนี้

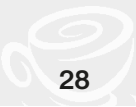
1. เลือกที่เมนู **File > New Project** หรือคลิกที่ไอคอน  บนทูลบาร์
2. ในหน้าจอถัดไป มีข้อกำหนดดังนี้คือ



- ตรงช่อง Categories ให้เลือก Java (ต้องการเขียนโปรแกรมด้วยภาษาจาวา)
  - ตรงช่อง Projects ให้เลือก Java Application (ต้องการสร้างเป็นแอปพลิเคชัน)
3. ในขั้นตอนถัดไป มีข้อกำหนดที่เราควรรู้เพิ่มเติมดังนี้คือ



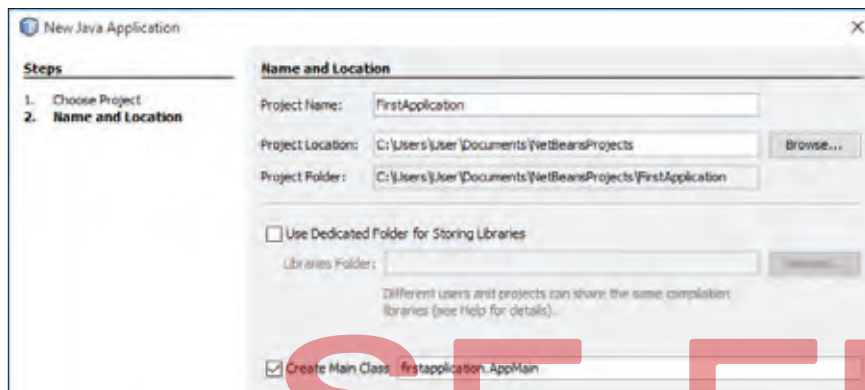
- **Project Name** คือชื่อโปรเจกต์ ซึ่งตามปกติ NetBeans จะตั้งชื่อเป็น JavaApplicationX มาให้ล่วงหน้า (X คือตัวเลข) ซึ่งเราสามารถเปลี่ยนเป็นชื่อใหม่ได้ตามต้องการ และถ้านำหลายๆ คำมารวมกัน ให้เขียนอักขระขึ้นต้นของแต่ละคำเป็นตัวพิมพ์ใหญ่ เช่น MyFirstApp, JavaExample, TestOfNetBeans เป็นต้น
- **Project Location** คือตำแหน่งการจัดเก็บโปรเจกต์ ซึ่งตามปกติ NetBeans จะเก็บไว้ที่ **C:\Users\ชื่อผู้ใช้\Documents\NetBeansProjects\** แม้เราจะเปลี่ยนไปเก็บไว้ที่ตำแหน่งอื่นได้ แต่เพื่อให้เป็นแนวทางเดียวกัน ผู้เขียนแนะนำว่าควรใช้ตำแหน่งตามที่ NetBeans กำหนดมาให้นี้จะดีกว่า
- **Project Folder** คือตำแหน่งและชื่อโฟลเดอร์ที่จะเก็บไฟล์ต่างๆ ของโปรเจกต์ โดยเราไม่ต้องใส่ค่าลงไป เพราะตามปกติแล้ว NetBeans จะสร้างโฟลเดอร์ตามชื่อโปรเจกต์ที่เรากำหนดให้โดยอัตโนมัติ



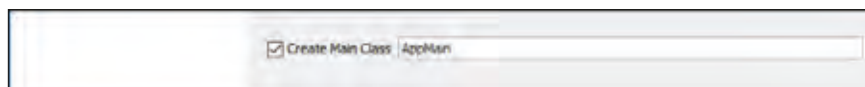
- **Use Dedicated ...** ในเบื้องต้น ยังไม่จำเป็นต้องใช้ตัวเลือกนี้
- **Create Main Class** เป็นการกำหนดว่า ต้องการสร้าง Main Class ซึ่งเป็นคลาสเริ่มแรกในการทำงานของโปรเจกต์ ซึ่งในเบื้องต้นนี้ ผู้เขียนแนะนำให้เช็คที่ตัวเลือกนี้ และสิ่งที่ต้องทำความเข้าใจคือการตั้งชื่อคลาส โดยถ้ายึดตามแนวทางของ NetBeans จะใช้รูปแบบ

### ชื่อแพ็คเกจ.ชื่อคลาส

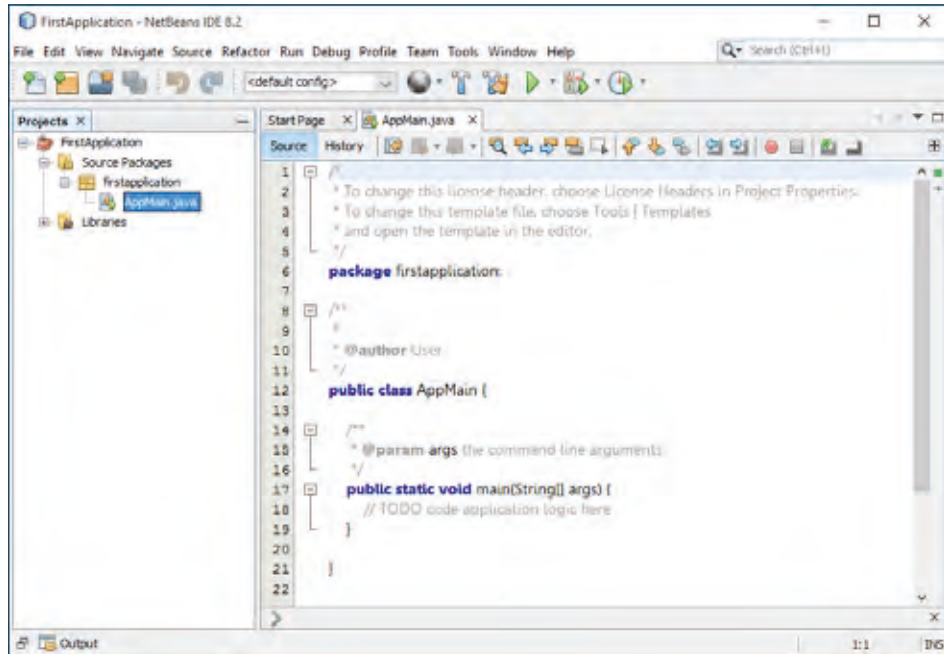
- ชื่อแพ็คเกจ ให้ใช้ชื่อเดียวกับชื่อโปรเจกต์ แต่เขียนเป็นตัวพิมพ์เล็กทั้งหมด
- ชื่อคลาส ให้ตัวเอง และถ้ามีหลายๆ คำมารวมกัน ให้เขียนอักษรขึ้นต้นของแต่ละคำเป็นตัวพิมพ์ใหญ่ เช่น สมมติว่าเราตั้งชื่อโปรเจกต์เป็น **FirstApplication** และต้องการให้คลาสชื่อ **AppMain** ก็กำหนดค่าลงในช่องนี้เป็น **firstapplication.AppMain** เป็นต้น



- อย่างไรก็ตาม ในช่อง **Create Main Class** เราอาจกำหนดเพียงแค่ชื่อคลาสลงไปอย่างเดียว โดยไม่ต้องกำหนดชื่อแพ็คเกจก็ได้ (ใช้ Default Package) เช่น ถ้าต้องการให้คลาสชื่อ AppMain ก็ระบุแค่ชื่อนี้ลงไป ดังภาพ

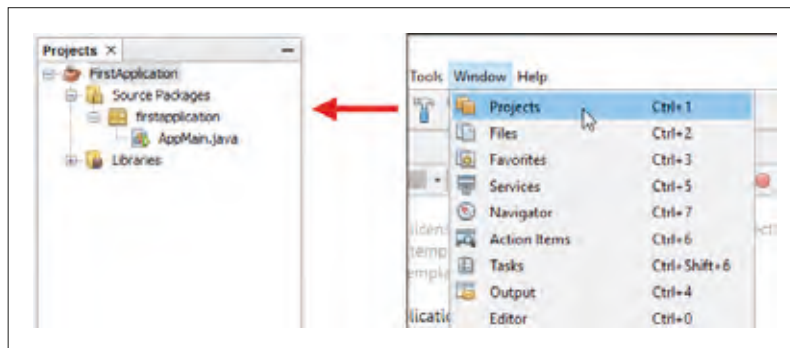


4. สมมติว่าจากขั้นตอนที่ผ่านมา เราตั้งชื่อโปรเจกต์เป็น FirstApplication และกำหนดชื่อคลาสเป็น AppMain เมื่อคลิกปุ่ม Finish ก็จะไปสู่หน้าจอหลักของ NetBeans ซึ่งมีลักษณะดังนี้



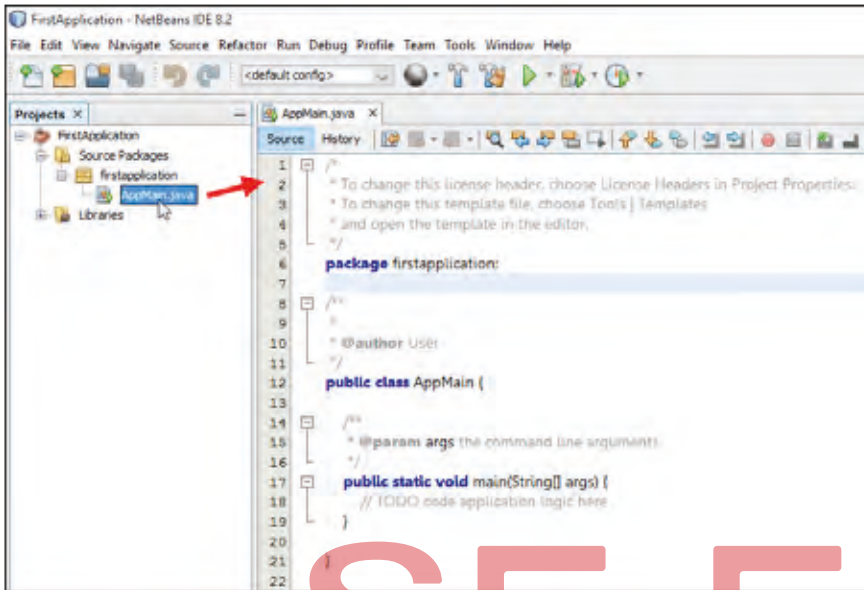
นอกจากนี้ยังมีกรณีพื้นฐานเกี่ยวกับโปรเจกต์ที่เราควรรู้เพิ่มเติม ดังนี้

- ถ้าต้องการออกจากโปรเจกต์ จะแยกพิจารณาเป็น 2 กรณีคือ
  - ถ้าต้องการออกจากโปรเจกต์พร้อมกับปิด Netbeans ให้เลือกที่เมนู **File > Exit**
  - ถ้าต้องการเพียงแคปิดโปรเจกต์ แล้วกลับเข้าสู่หน้าจอเริ่มต้นของ NetBeans ให้เลือกที่เมนู **File > Close Project** (ชื่อโปรเจกต์ที่เปิดอยู่)
- เมื่อเราเปิดเข้าสู่โปรแกรม NetBeans ครั้งต่อไป ตามปกติจะมีรายชื่อโปรเจกต์ที่เราสร้างเอาไว้อยู่ในวินโดว์ Projects ทางฝั่งซ้ายมือ ถ้าไม่ปรากฏวินโดว์นี้ ให้เลือกที่เมนู **Window > Projects**

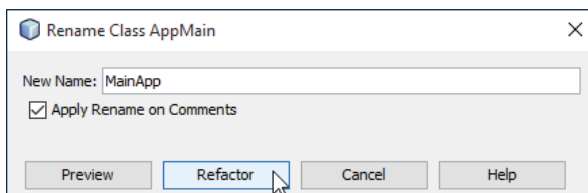
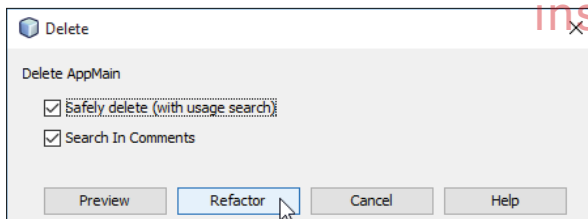




- ถ้าโปรเจกต์ที่เราจะเปิดไม่ปรากฏอยู่ในวินโดว์ Project ก็ให้เลือกที่เมนู **File > Open Project**
- หากเราจะเขียนโค้ดจาวา ก็ต้องเปิดไฟล์ของคลาสตามชื่อที่ได้กำหนดเอาไว้ตอนสร้างโปรเจกต์ โดยไปที่วินโดว์ Projects แล้วขยายโหนด **ชื่อโปรเจกต์ > Source Packages > ชื่อแพ็คเกจ** ต่อไปดับเบิลคลิกที่ไฟล์ของคลาส จากนั้นแท็บของโค้ดจะถูกเปิดขึ้น (ถ้าไม่ได้กำหนดแพ็คเกจตอนสร้างโปรเจกต์ ไฟล์ของคลาสอาจจะอยู่ในโฟลเดอร์ default package)



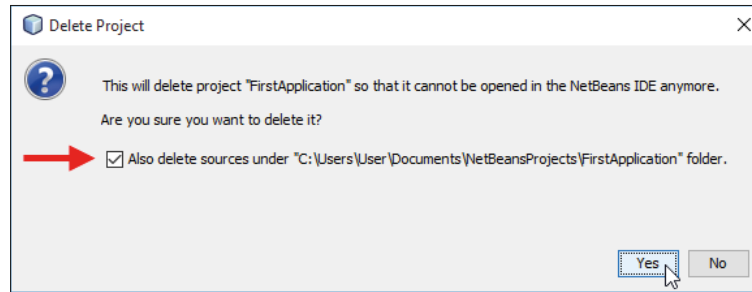
- หากต้องการลบหรือเปลี่ยนชื่อคลาส ก็ทำดังนี้คือ



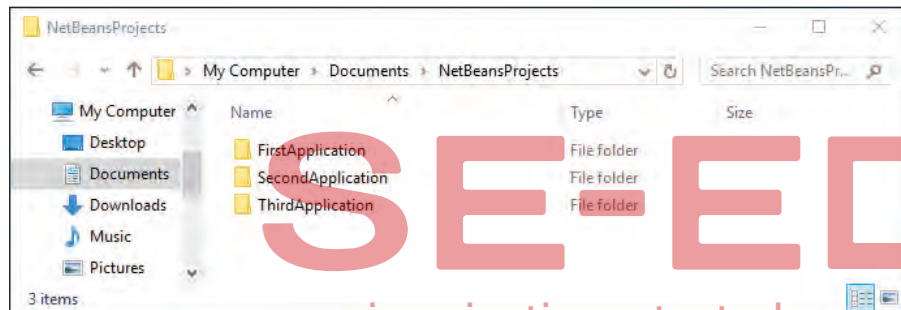
● **การลบคลาส** ให้คลิกขวาที่ชื่อคลาสในวินโดว์ Projects แล้วเลือก Delete เมื่อมีไดอะล็อกขึ้นมาให้เลือกอปชั่น ก็ควรเช็คทุกอันเพื่อให้คลาสถูกลบอย่างสมบูรณ์

● **การเปลี่ยนชื่อคลาส** จะเปลี่ยนที่โค้ดโดยตรง เลยไม่ได้ เพราะมีส่วนที่เกี่ยวข้องอีกหลายแห่งที่อ้างอิงชื่อคลาสที่เราได้สร้างขึ้น โดยวิธีที่ถูกต้องคือ ให้คลิกขวาที่ชื่อคลาสแล้วเลือก **Refactor > Rename** แล้วส่วนต่างๆ ที่อ้างถึงคลาสในชื่อเดิมจะถูกเปลี่ยนเป็นชื่อใหม่โดยอัตโนมัติ

- ถ้าต้องการลบโปรเจกต์ที่สร้างขึ้น หากปรากฏชื่ออยู่บนวินโดว์ Projects ก็ให้คลิกที่ชื่อโปรเจกต์นั้นแล้วกดคีย์บอร์ดปุ่ม Delete หรือจะคลิกขวาแล้วเลือก Delete ก็ได้ จากนั้นเมื่อมีไดอะล็อกขึ้นมาให้ยืนยัน ควรเช็คที่ตัวเลือก Also delete sources under ... ด้วย เพื่อให้โปรเจกต์ถูกลบอย่างสมบูรณ์



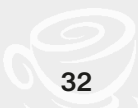
- แต่ถ้าโปรเจกต์ที่เราต้องการลบไม่ปรากฏในวินโดว์ Project ก็อาจเปิดเข้าไปดูยังตำแหน่งที่เราเก็บโปรเจกต์ของ NetBeans เอาไว้ ซึ่งหากใช้ตามค่าดีฟอลต์จะอยู่ที่ C:\Users\ชื่อผู้ใช้\Documents\NetBeansProjects\ จากนั้นลบไฟล์เดอร์ของโปรเจกต์นั้นทิ้งไปได้เลย



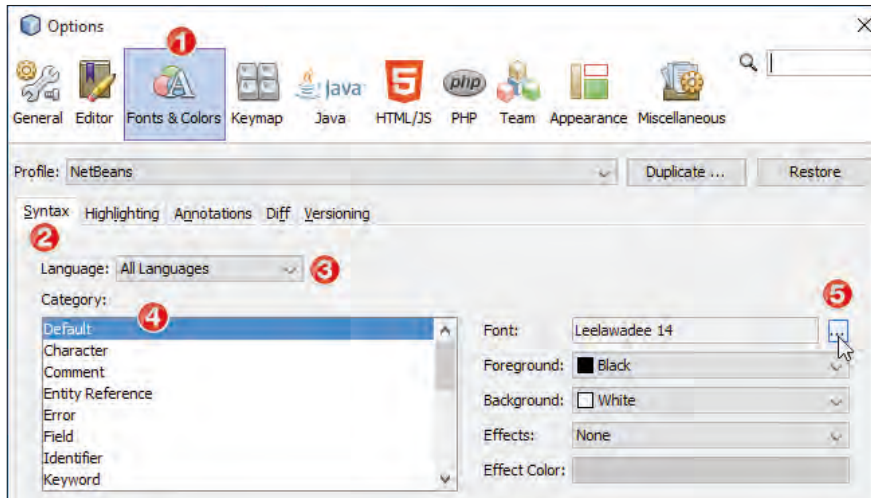
## 1-7: การกำหนดฟอนต์สำหรับ NetBeans

ลักษณะฟอนต์ในการแสดงโค้ดรวมถึงการแสดงผลลัพธ์ที่ NetBeans กำหนดมาให้ อาจจะไม่ถูกใจเรานัก เช่น แสดงอักษรภาษาไทยไม่ค่อยสวยงาม หรือมีขนาดที่ไม่พอดี หรืออยู่ชิดกันจนเกินไปทำให้อ่านโค้ดลำบาก เป็นต้น ซึ่งเราสามารถปรับเปลี่ยนฟอนต์ใหม่ได้ตั้งแนวทางต่อไปนี้

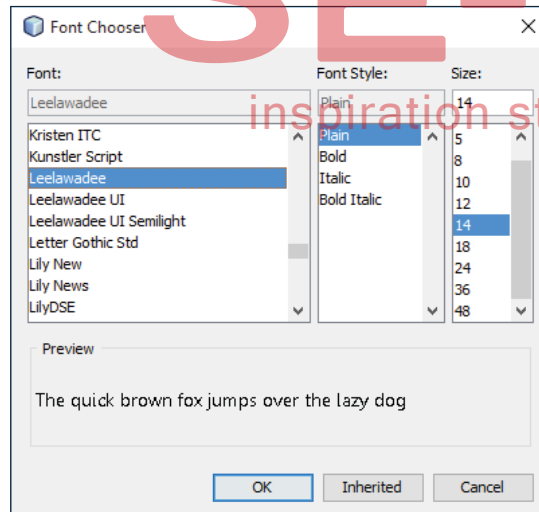
1. เลือกที่เมนู **Tools > Options**
2. ในหน้าจอถัดไป ให้เลือกตามลำดับหมายเลขในภาพคือ



- 1) Fonts & Colors
- 2) แท็บ Syntax
- 3) Language: All Language
- 4) Category: Default
- 5) Font: คลิกปุ่ม ...



3. จากนั้นจะปรากฏไดอะล็อกขึ้นมาให้เราเลือกชนิด รูปแบบ และขนาดของฟอนต์ที่ต้องการ โดยควรเลือกชนิดที่รองรับภาษาไทยและไม่ขัดกันเกินไป สำหรับในกรณีของผู้เขียนได้เลือกชนิด Leelawadee ขนาด 14 หลังการเลือก ให้คลิกปุ่ม Apply เพื่อให้ฟอนต์ถูกนำไปใช้



SE-ED  
inspiration starts here

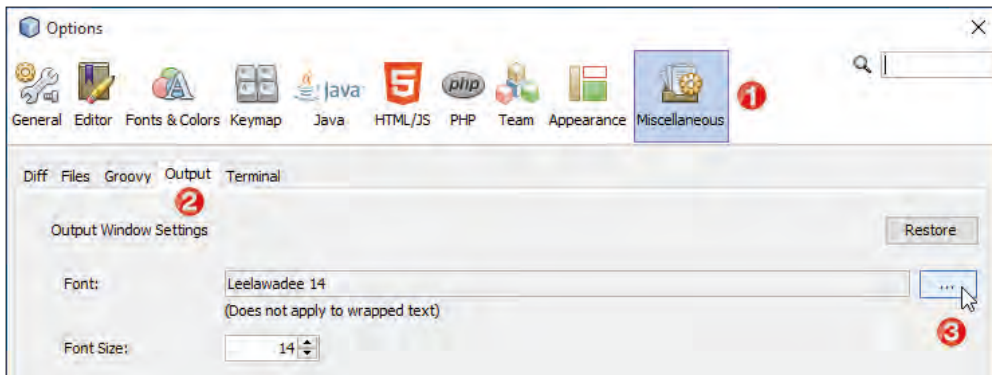


4. ถ้าต้องการปรับแต่งฟอนต์ของ Output (ส่วนแสดงผลพัทธ์หรือคอนโซล) ให้คลิกตามลำดับหมายเลข ในภาพถัดไป แล้วเลือกฟอนต์จากไดอะล็อกเหมือนเดิม สำหรับในกรณีของผู้เขียนได้เลือกชนิด Leelawadee ขนาด 14

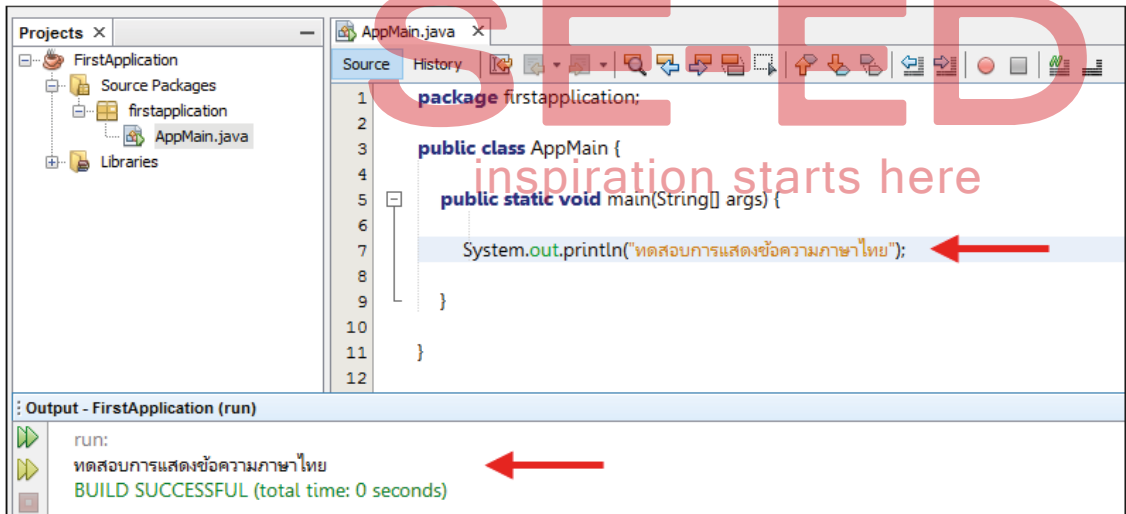
1) Miscellaneous

3) คลิกปุ่ม ...

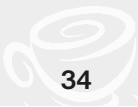
2) แท็บ Output



เราสามารถทดลองดูลักษณะของฟอนต์ที่เลือกในเบื้องต้นได้ โดยพิมพ์โค้ดดังภาพถัดไป แล้วกดคีย์บอร์ดปุ่ม F6 เพื่อรันโปรแกรม ดูว่าฟอนต์ที่เลือกมานั้นถูกใจเราหรือไม่ ถ้าไม่ชอบก็กลับไปเลือกใหม่



ส่วนการปรับแต่งในกรณีอื่นๆ จะนำมากล่าวเสริมต่อไปเรื่อยๆ ในหัวข้อที่เกี่ยวข้อง



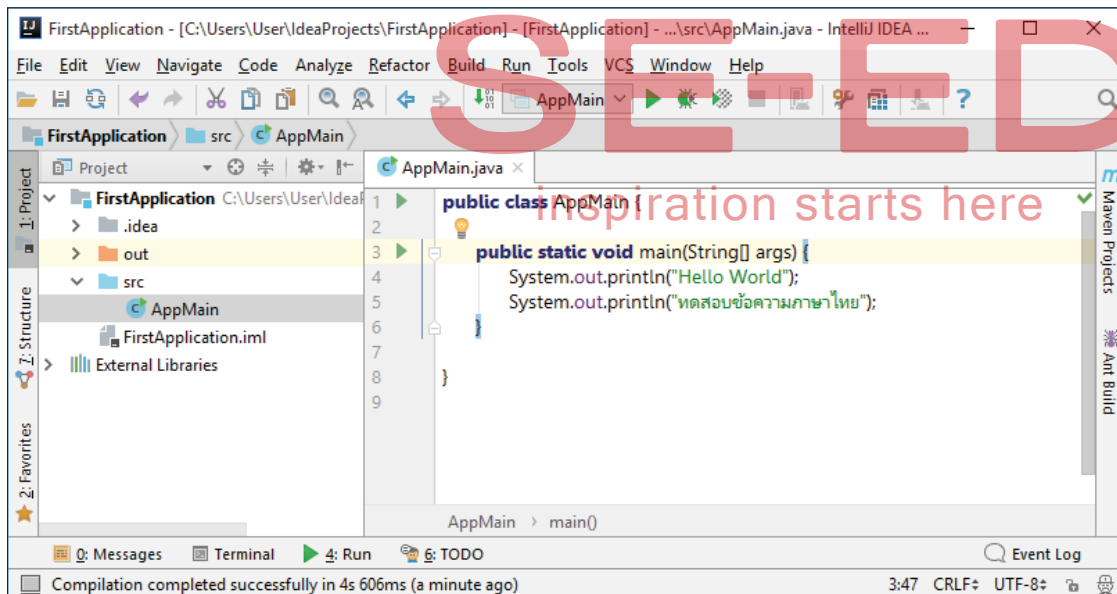
### ▼ หมายเหตุ

NetBeans อาจจะมีปัญหาเกี่ยวกับฟอนต์ไทยบางชนิด กล่าวคือ หลังจากที่เรเลือกฟอนต์เสร็จ เมื่อกลับมาดูที่ Code Editor อาจแสดงผลได้ถูกต้อง แต่เมื่อปิดโปรแกรม NetBeans แล้วเปิดขึ้นมาใหม่ ลักษณะของฟอนต์อาจผิดเพี้ยนไป ดังนั้นเพื่อให้แน่ใจว่าฟอนต์ที่เราเลือกสามารถใช้งานได้จริง หลังจากเลือกเสร็จแล้วให้ปิดโปรแกรม NetBeans แล้วเปิดขึ้นมาใหม่ ถ้าฟอนต์มีปัญหาให้เปลี่ยนเป็นชนิดอื่นแทน

## 1-8: การติดตั้ง IntelliJ IDEA

IntelliJ IDEA เป็นโปรแกรมประเภท IDE สำหรับการเขียนโค้ดจาวาที่กำลังได้รับความนิยมอย่างมากจากนักพัฒนาระดับมืออาชีพ เพราะมีลักษณะเด่นหลายประการ เช่น

- มีรูปร่างหน้าตาที่สวยงาม น่าใช้งานมากกว่า
- มีเทคนิคการทำงานในหลายๆ กรณีที่ชาญฉลาดกว่า IDE ตัวอื่นๆ
- มีรุ่นที่แจกให้ใช้งานฟรีแบบไม่มีเงื่อนไข
- เป็น IDE ที่ถูกนำไปพัฒนาเป็น Android Studio สำหรับการสร้างแอปบนแอนดรอยด์ ดังนั้นจึงเหมาะอย่างยิ่งกับผู้ที่ศึกษาจาวาเพื่อก้าวไปสู่การเขียนแอปแอนดรอยด์ในอนาคต

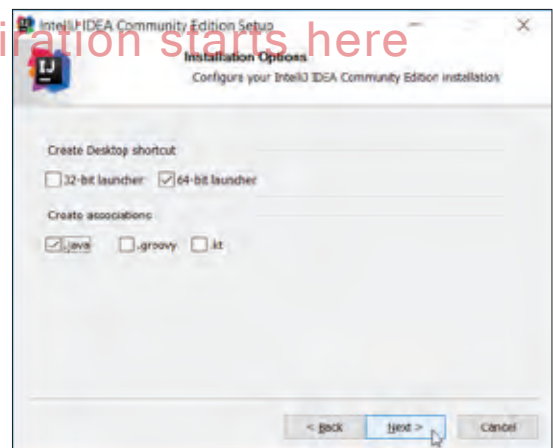
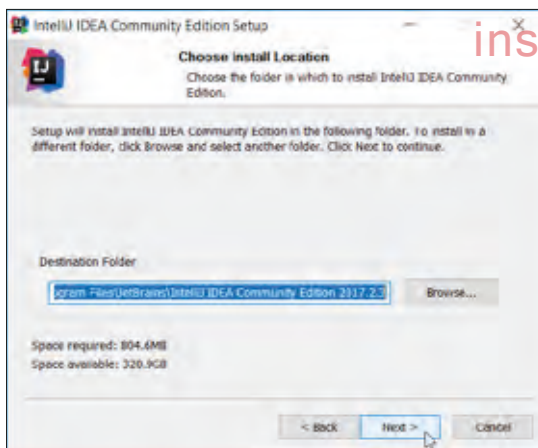


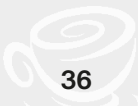
ถ้าเราจะนำ IntelliJ มาเป็นเครื่องมือในการเขียนจาวา ก็ต้องติดตั้ง Java (JDK) ดังที่ได้กล่าวมาในหัวข้อก่อนๆ นี้ หลังจากนั้นจึงค่อยติดตั้ง IntelliJ ซึ่งมีขั้นตอนโดยสังเขปดังต่อไปนี้

1. สามารถดาวน์โหลด IntelliJ ได้จากเว็บไซต์ <https://www.jetbrains.com/idea/download> โดยหาความต้องการใช้งานฟรีก็ให้เลือกรุ่น Community Edition



2. การติดตั้ง IntelliJ ก็ทำเหมือนกับโปรแกรมต่างๆ ไป โดยข้อกำหนดในบางขั้นตอนมีดังนี้





## 1-9: พื้นฐานการเขียนจาวาด้วย IntelliJ IDEA

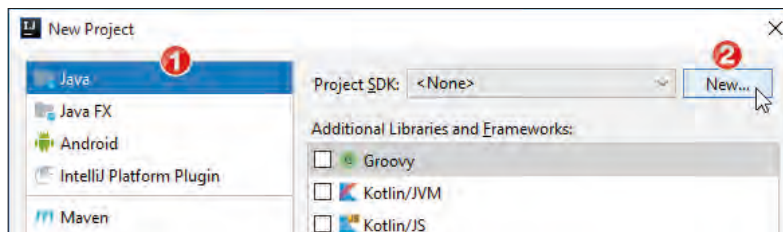
การเขียนจาวาด้วย IntelliJ จะต้องเริ่มที่การสร้างโปรเจกต์ (Project) ใหม่เสมอ โดยให้เปิดเข้าสู่ IntelliJ แล้วทำตามขั้นตอนดังต่อไปนี้



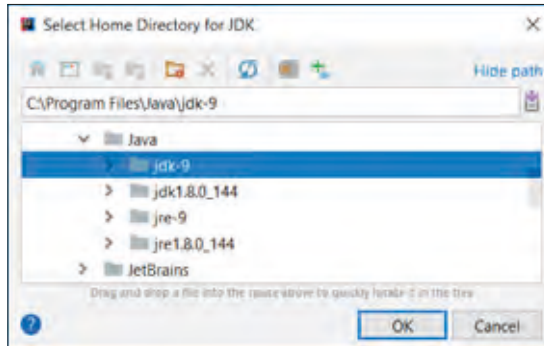
1. เมื่อปรากฏหน้าจอ Welcome to IntelliJ IDEA เลือกที่เมนู **Create New Project**
2. ถ้าเราเพิ่งจะเริ่มใช้ IntelliJ เป็นครั้งแรก และยังไม่ได้กำหนดตำแหน่งที่ติดตั้ง Java (JDK) เอาไว้ ก็ให้ทำดังนี้คือ
  - 1) เลือก Java
  - 2) คลิกปุ่ม New ...

SEED

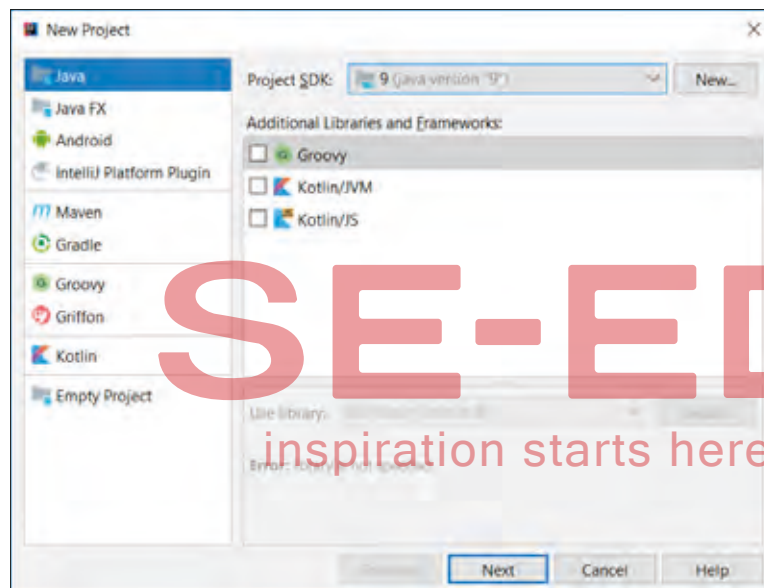
inspiration starts here



- 3) เลือกตำแหน่งที่เราติดตั้ง JDK เอาไว้ (ต้องเลือกโฟลเดอร์ jdk ไม่ใช่ jre)

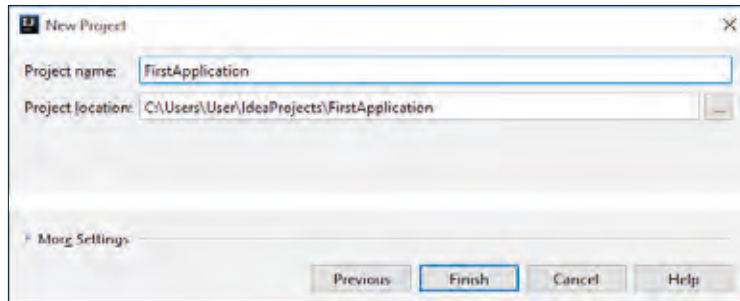


- 4) ตำแหน่งที่เราเลือก จะมาปรากฏในช่อง Project SDK

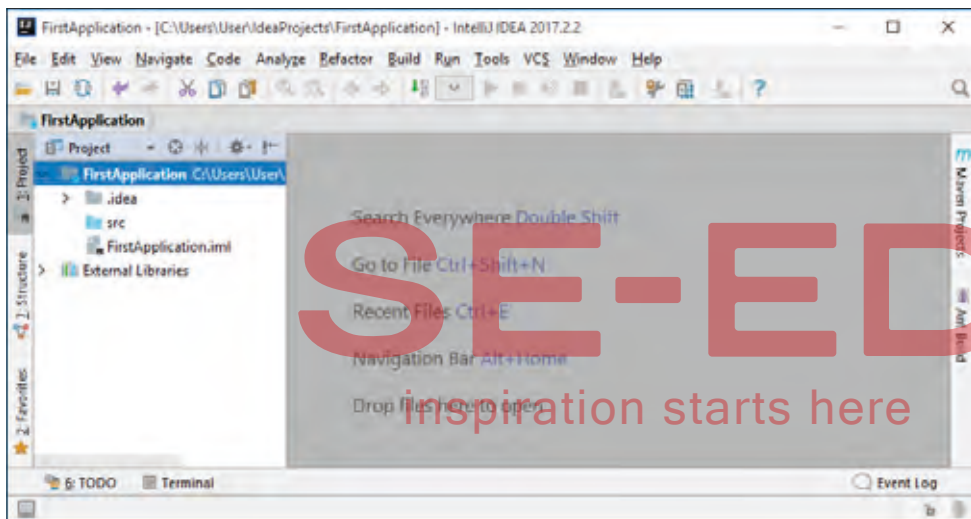


การเซตตำแหน่ง JDK นี้ จะทำเฉพาะเมื่อสร้างโปรเจกต์บน IntelliJ เป็นครั้งแรกเท่านั้น แล้วครั้งต่อไป เราไม่ต้องเซตค่านี้อีก เพราะโปรแกรมจะจดจำตำแหน่งนี้ไว้ให้เราโดยอัตโนมัติ ยกเว้นแต่เราเปลี่ยนตำแหน่งการติดตั้ง JDK ใหม่ หรืออัปเดตเป็นเวอร์ชันใหม่

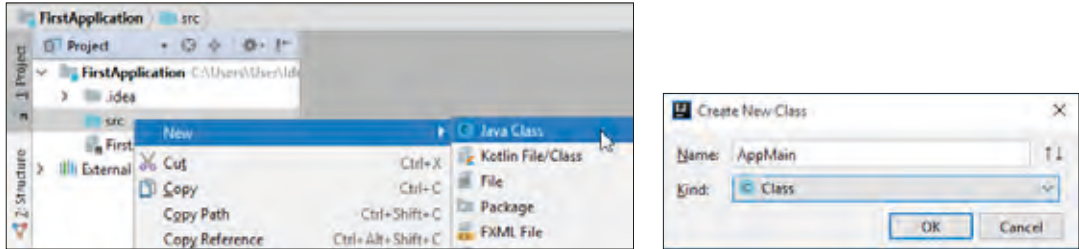
- คลิกปุ่ม Next ต่อไปจนถึงขั้นตอนการตั้งชื่อโปรเจกต์ ก็ให้กำหนดชื่อตามต้องการ (ใช้หลักการเดียวกับ NetBeans) ส่วนตำแหน่งการจัดเก็บโปรเจกต์ แนะนำให้ใช้ค่าดีฟอลต์ของ IntelliJ คือ **C:\Users\ชื่อผู้ใช้\IdeaProjects\**



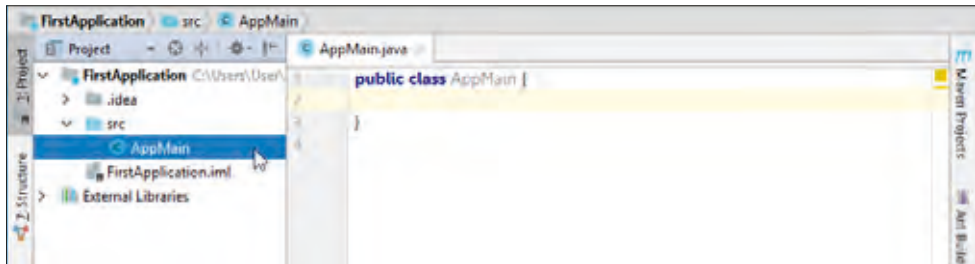
- เมื่อเข้าสู่โปรแกรม IntelliJ ก็จะมีลักษณะดังนี้



- เนื่องจาก IntelliJ ไม่ได้สร้างคลาสมาให้เราล่วงหน้า ดังนั้นสิ่งที่ต้องทำคือ การเพิ่มคลาสใหม่เข้ามาในโปรเจกต์ โดยที่วินโดว์ Project (หากไม่ปรากฏให้คลิกตรงแท็บด้านข้างทางซ้ายมือ หรือเลือกจากเมนู **View > Tool Windows > Project**) คลิกขยายโหนดชื่อโปรเจกต์ (กรณีของผู้เขียนตั้งชื่อเป็น FirstApplication) จากนั้นคลิกขวาที่ **โฟลเดอร์ src > New > Java Class** แล้วกำหนดชื่อคลาสลงไป โดยใช้หลักการเดียวกับที่สร้างใน NetBeans และในช่อง Kind ต้องเลือกชนิดเป็น Class ด้วย



- ไฟล์ของคลาสจะถูกสร้างขึ้นในโฟลเดอร์ src ซึ่งหากเราดับเบิ้ลคลิกที่ไฟล์ดังกล่าว ก็จะปรากฏแท็บของโค้ดดังนี้

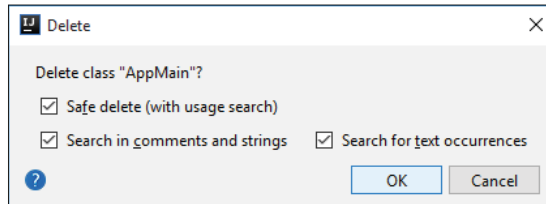


- ถ้าต้องการปิดโปรเจกต์ ให้เลือกที่เมนู **File > Close Project** ซึ่งก็จะกลับเข้าสู่หน้า Welcome แต่ถ้าต้องการปิดทั้งโปรเจกต์และปิด IntelliJ ด้วย ให้เลือกที่เมนู **File > Exit**
- เมื่อเราเปิดเข้าสู่ IntelliJ ในครั้งต่อไป จะปรากฏรายชื่อโปรเจกต์ที่เราเคยสร้างเอาไว้ที่ช่องด้านซ้ายมือของหน้าจอ Welcome แต่ถ้าไม่มีรายชื่อปรากฏในช่องนี้ให้คลิกที่เมนู Open แล้วเลือกตำแหน่งที่เราจัดเก็บโปรเจกต์เอาไว้ ซึ่งถ้าใช้ตามค่าดีฟอลต์ตั้งที่กล่าวมาก็คือ **C:\Users\ชื่อผู้ใช้\IdeaProjects\**

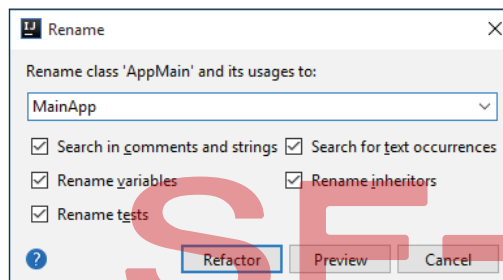
inspiration starts here



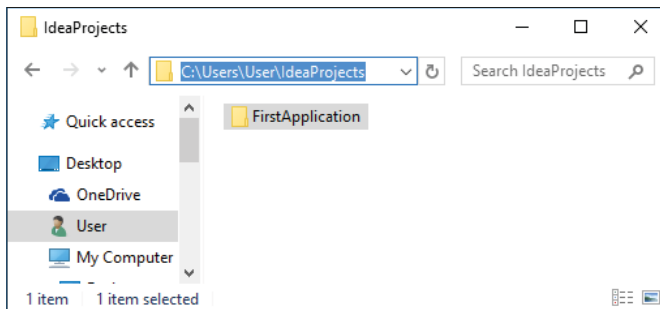
- หากต้องการลบหรือเปลี่ยนชื่อคลาส ก็ให้ทำดังนี้คือ
  - **การลบคลาส** ให้คลิกขวาที่ชื่อคลาสตรงไฟล์เดอร์ src แล้วเลือก Delete เมื่อมีโด้ะลือกขึ้นมาให้เลือกอปชั่น แนะนำให้เช็คทุกอันเพื่อให้คลาสถูกลบอย่างสมบูรณ์



- **การเปลี่ยนชื่อคลาส** จะเปลี่ยนที่โค้ดโดยตรงเลยไม่ได้ เพราะมีส่วนที่เกี่ยวข้องอีกหลายแห่งที่อ้างอิงชื่อคลาสที่เราได้สร้างขึ้น โดยวิธีที่ถูกต้องคือ ให้คลิกขวาที่ชื่อคลาสตรงไฟล์เดอร์ src แล้วเลือก > Refactor > Rename โดยแนะนำให้เช็คทุกอปชั่นเพื่อให้ส่วนต่างๆ ที่อ้างถึงคลาสในชื่อเดิม ถูกเปลี่ยนเป็นชื่อใหม่โดยอัตโนมัติ



- ถ้าต้องการลบโปรเจกต์ ไม่สามารถลบผ่าน IntelliJ ได้ ต้องเปิดเข้าไปยังตำแหน่งที่เราจัดเก็บโปรเจกต์เอาไว้คือ C:\Users\ชื่อผู้ใช้\IdeaProjects\ แล้วลบไฟล์เดอร์ของโปรเจกต์นั้นเหมือนกับไฟล์เดอร์ทั่วไป

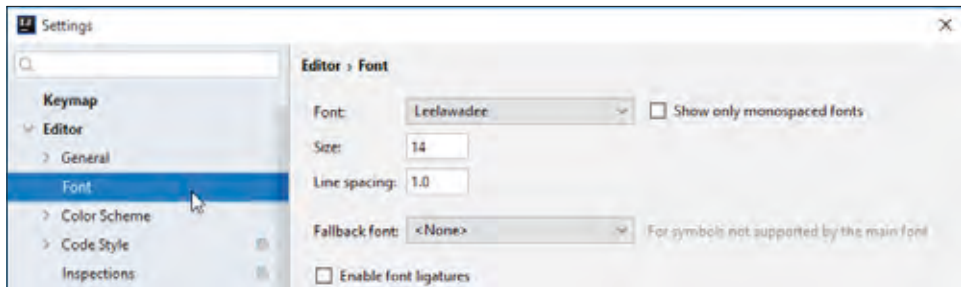




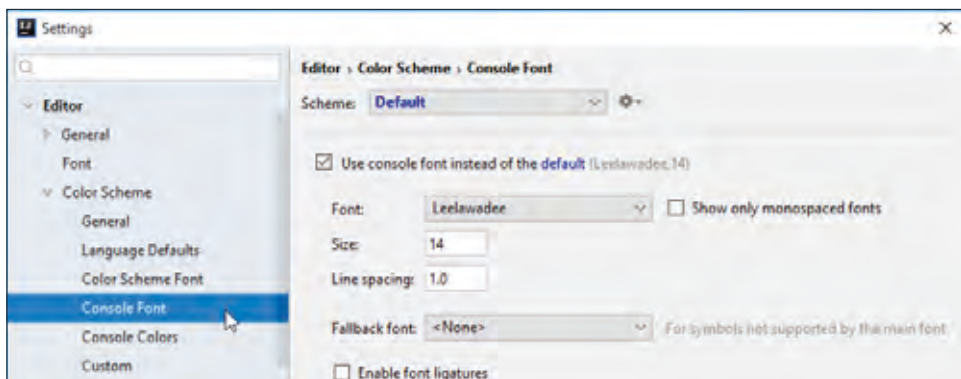
## 1-10: การกำหนดฟอนต์สำหรับ IntelliJ IDEA

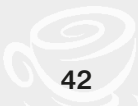
ฟอนต์ของโค้ดและส่วนผลลัพธ์ที่ IntelliJ กำหนดมาให้ล่วงหน้า อาจเกิดปัญหาในการแสดงภาษาไทย หรือ มีรูปแบบที่ไม่ค่อยถูกใจเรานัก ซึ่งก็สามารถเปลี่ยนแปลงใหม่ได้ ดังนี้


1. เลือกที่เมนู **File > Settings...**
2. ถ้าต้องการแก้ไขฟอนต์ของการเขียนโค้ด มีขั้นตอนดังนี้
  - เลือกที่ **Editor > Font**
  - เอาเช็คตรง **Show only monospaced fonts** ออก
  - เลือกชนิดฟอนต์และกำหนดขนาดตามต้องการ โดยในกรณีของผู้เขียน ได้เลือกชนิด Leelawadee ขนาด 14

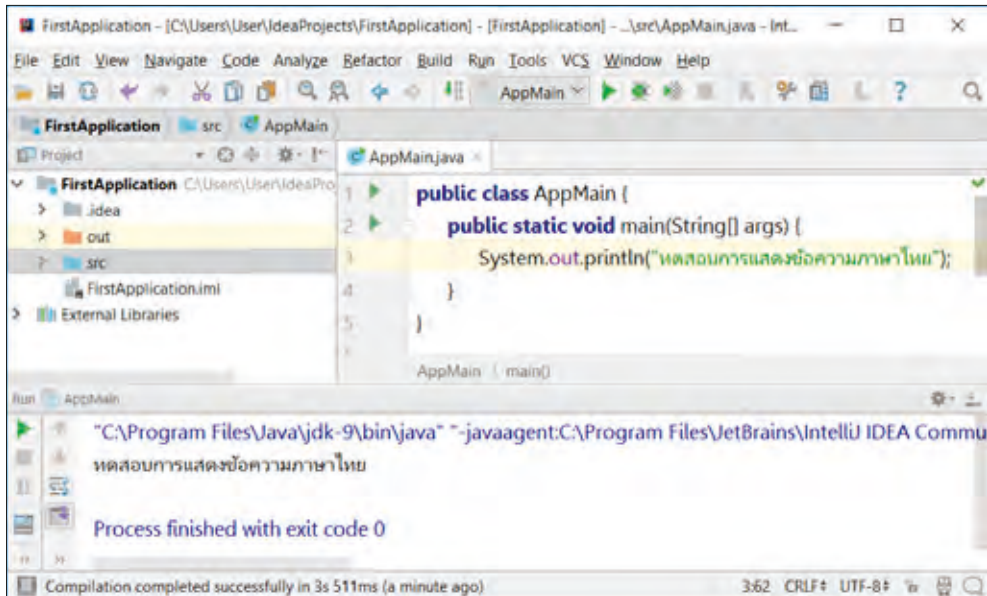
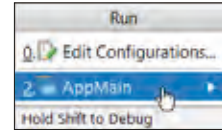


3. ถ้าต้องการกำหนดฟอนต์ของส่วน Console (แสดงผลลัพธ์) มีขั้นตอนดังนี้
  - เลือก **Editor > Color Scheme > Console Font**
  - เอาเช็คตรง **Show only monospaced fonts** ออก
  - เลือกชนิดฟอนต์และกำหนดขนาดตามต้องการ โดยในกรณีของผู้เขียนเลือกชนิด Leelawadee ขนาด 14





เราสามารถทดสอบการแสดงผลฟอนต์ที่เลือกมา โดยกำหนดโค้ดดังในภาพถัดไป แล้วการรันโปรแกรมครั้งแรกให้เลือกที่เมนู **Run > Run...** จากนั้นเลือกชื่อ **Main Class** (ในที่นี้คือ AppMain) แต่ในการรันครั้งต่อไปคลิกที่ไอคอน  บนทูลบาร์ได้เลย



**SE-ED**  
inspiration starts here

# การเขียนโปรแกรมด้วย Java สำหรับผู้เริ่มต้น

หนังสือเล่มนี้ได้รวบรวมเนื้อหาเกี่ยวกับ Java ที่จำเป็นต้องรู้ในเบื้องต้น พร้อมตัวอย่างและแบบฝึกหัดท้ายบท ตลอดจนเวิร์คช็อปที่หลากหลาย โดยใช้เครื่องมือยอดนิยมอย่าง NetBeans และ IntelliJ เพื่อให้ผู้ที่กำลังเริ่มต้นเขียนโปรแกรมด้วย Java ใช้ประกอบการเรียนรู้ขั้นพื้นฐานและเป็นแนวทางสำหรับการศึกษาเพิ่มเติมในระดับสูงขึ้นไปได้

## เนื้อหาในหนังสือประกอบด้วย :

- |   |                                       |
|---|---------------------------------------|
| บทที่ 1 : ภาษาจาวาและการติดตั้งเครื่องมือ   | บทที่ 11 : ตัวเลขและวันเวลา           |
| บทที่ 2 : องค์ประกอบพื้นฐานและตัวแปร        | บทที่ 12 : คลาสและออบเจกต์            |
| บทที่ 3 : การรับและแสดงผลข้อมูล             | บทที่ 13 : การสืบทอดและแพ็คเกจ        |
| บทที่ 4 : ชนิดข้อมูลพื้นฐานและสตริง         | บทที่ 14 : การจัดการข้อผิดพลาดและไฟล์ |
| บทที่ 5 : โอเปอเรเตอร์และการแปลงข้อมูล      | บทที่ 15 : อินเทอร์เฟซและแลมบ์ดา      |
| บทที่ 6 : การกำหนดเงื่อนไข                  | บทที่ 16 : Java Swing ตอนที่ 1        |
| บทที่ 7 : การทำงานแบบวงรอบ                  | บทที่ 17 : Java Swing ตอนที่ 2        |
| บทที่ 8 : เมธอดและฟิลด์                     | บทที่ 18 : JavaFx เบื้องต้น           |
| บทที่ 9 : อาร์เรย์และอินัม                  | บทที่ 19 : รวมเวิร์คช็อป ชุดที่ 1     |
| บทที่ 10 : การใช้งานเพิ่มเติมเกี่ยวกับสตริง | บทที่ 20 : รวมเวิร์คช็อป ชุดที่ 2     |



www.se-ed.com



sbc.fans



ISBN 978-616-08-3186-9



9 786160 831869

320 บาท