เอกสารประกอบการสอน

3132207 การโปรแกรมคอมพิวเตอร์

(3132207 Computer Programming)

อมาตย์ สูหลง

สาขาวิชาการจัดการธุรกิจดิจิทัลและนวัตกรรม

คณะวิทยาการจัดการ

มหาวิทยาลัยราชภัฏยะลา

2564

แผนบริหารการสอนประจำวิชา

รหัสวิชา 3132207

ชื่อวิชา การโปรแกรมคอมพิวเตอร์ 3(2-2-5)
(Computer Programming)

เวลาเรียน 15 สัปดาห์ 45 ชั่วโมง/ภาคเรียน

คำอธิบายรายวิชา

หลักการออกแบบและเขียนโปรแกรม ผังงานโปรแกรม รูปแบบไวยากรณ์ ภาษา องค์ประกอบของภาษาคอมพิวเตอร์ คำสั่งเข้าข้อมูล การประมวลผลและแสดงผลลัพธ์ การจัดรูปแบบของข้อมูล ตัวดำเนินการ และนิพจน์ คำสั่งเงื่อนไข คำสั่งการวนซ้ำ ตัวแปรชุด ตัวชี้ ฟังก์ชัน ข้อมูลชนิดโครงสร้าง แฟ้มข้อมูล และ การจัดการแฟ้มข้อมูล

วัตถุประสงค์ทั่วไป

1. เพื่อให้นักศึกษาเข้าใจหลักการเขียนผังงาน
2. เพื่อให้นักศึกษาเข้าใจประเภทข้อมูลต่างๆ ค่าคงที่ ตัวแปร นิพจน์ และตัวกระทำการ
3. เพื่อให้นักศึกษาเข้าใจคำสั่งควบคุม ลำดำขั้นตอนการทำงาน
4. เพื่อให้นักศึกษามีทักษะในการเขียนโปรแกรมเบื้องต้นด้วยภาษาซี หรือ ภาษา Java
5. เพื่อให้มีทัศนคติที่ดีต่อการเรียนวิชาทางการโปรแกรมภาษาจาวา

6 เพื่อให้สามารถนำความรูพื้นฐานเกี่ยวกับหลักการเขียนโปรแกรมภาษาจาวามาประยุกต์ใช้ประโยชนและต่อยอดในระดับสูงขึ้นในอนาคตได้

วิธีสอนและกิจกรรมการเรียนการสอน

วิธีสอน

1. วิธีสอนโดยใช้รูปแบบของการบรรยาย
2. วิธีสอนโดยใช้การฝึกปฏิบัติการเขียนโปรแกรม
3. วิธีสอนโดยใช้การค้นคว้าสื่อข้อมูลโปรแกรมภาษา Java
4. วิธีสอนโดยใช้การสาธิต กรณีศึกษา และอภิปรายกลุ่มย่อยการใช้โปรแกรมภาษา Java

5. วิธีสอนโดยใช้การฝึกทักษะการเขียนโปรแกรมรูปแบบของกิจกรรมกลุ่มและการถอดบทเรียนหลังกิจกรรม

กิจกรรมการเรียนการสอน

1. ทำแบบฝึกหัดท้ายบทในเอกสารประกอบการเรียนการสอน
2. วิเคราะห์โปรแกรมให้สอดคล้องกับความรู้ที่ได้รับในชั้นเรียน
3. จำลองโค้ดตัวอย่างของการเขียนโปรแกรมตามเนื้อหาที่กำหนดในแต่ละบทเรียน

การวัดและประเมินผล

1. การวัดผล

1.1 คะแนนระหว่างภาคเรียน 70%

1.1.1	ทดสอบกลางภาค	20%
1.1.2	ความตั้งใจในการเข้าเรียน	5%
1.1.3	แบบฝึกหัด/Lab Java Programming	10%
1.1.4	ใบงาน	20%
1.1.5	ศึกษาค้นคว้าและรายงานรายงานกลุ่มหรือโปรเจคกลุ่ม	10%
1.1.6	นำเสนอ	5%

1.2 คะแนนสอบปลายภาคเรียน 30%

การประเมินผล

เกณฑ์การประเมินผลรายวิชา ดังนี้

คะแนน	80 - 100	ระดับ	A
คะแนน	75 - 79	ระดับ	B+
คะแนน	70 - 74	ระดับ	B
คะแนน	65 - 69	ระดับ	C+
คะแนน	60 - 64	ระดับ	C
คะแนน	55 - 59	ระดับ	D+
คะแนน	50 - 54	ระดับ	D
คะแนน	0 - 49	ระดับ	E

คำนำ

เอกสารประกอบการสอนรายวิชา 3132207 การโปรแกรมคอมพิวเตอร์ ได้จัดทำขึ้นโดยมีจุดมุ่งหมายเพื่อใช้ประกอบการสอนให้แก่นักศึกษาระดับปริญญาตรีโปรแกรมคอมพิวเตอร์ธุรกิจและสาขาวิชาอื่น ๆ ที่สนใจรายวิชานี้ ซึ่งเอกสารประกอบการสอนนี้ประกอบด้วยเนื้อหาทั้งหมด 6 บท ดังนี้ บทที่ 1 หลักการเขียนโปรแกรมคอมพิวเตอร์ บทที่ 2 ชนิดข้อมูลและตัวแปร บทที่ 3 เครื่องหมายดำเนินการ บทที่ 4 Java and Loop บทที่ 5 Java and Condition/Control-Flow Statement บทที่ 6 Class and Method เพื่อให้ผู้เรียนได้เกิดความรู้ความเข้าใจและทักษะพื้นฐานในเนื้อหารายวิชาการเขียนโปรแกรมได้ดียิ่งขึ้นจึงมีแบบฝึกหัดทบทวนไว้ด้วยในแต่ละบท

เอกสารประกอบการสอนเล่มนี้ คาดว่าจะเป็นประโยชน์สำหรับผู้สอนในการใช้เป็นแนวทางในการสอนสำหรับรายวิชานี้และเป็นประโยชน์ต่อผู้เรียนในการใช้อ่านทบทวนเพื่อให้เกิดความรู้ความเข้าใจในรายวิชาในนี้มากยิ่งขึ้นต่อไป

อมาตย์ สูหลง

19 กันยา 2564

สารบัญ

บทที่ 1
หลักการเขียนโปรแกรมคอมพิวเตอร์
(Principle of Computer Programming)

วัตถุประสงค์ของหน่วยการเรียนรู้

1. เกิดทักษะการเขียนโปรแกรมคอมพิวเตอร์
2. อธิบายความแตกต่างหรือสัมพันธ์ระหว่างโปรแกรมภาษาจาวามีกับภาษาแบบโครงสร้างได
3. บอกขั้นตอนการทำงานของโปรแกรมภาษาจาวาได
4. เขียนโปรแกรมภาษาจาวาตามรูปแบบภาษาได้

1.1 การเขียนโปรแกรมคอมพิวเตอร์

การเขียนโปรแกรมคอมพิวเตอร์ (Computer Programming) มีแนวคิดมาจากมนุษย์มีความต้องการที่นำงานที่เป็นลักษณะงานประจำ งานที่เป็นลักษณะงานซ้ำๆ ที่มีขั้นตอนทำงานที่แน่นอน มีระบบการตัดสินใจที่ชัดเจนแน่นอน มาให้ระบบคอมพิวเตอร์ทำงานแทนมนุษย์ ตามวัตถุประสงค์ในการทำงานของผู้คิดค้นโปรแกรมคอมพิวเตอร์ ดังนั้นการเขียนโปรแกรมคอมพิวเตอร์ จึงเป็นพื้นฐานที่จำเป็นต่อการทำงานของระบบคอมพิวเตอร์ ชนิดของโปรแกรมคอมพิวเตอร์สามารถแบตามระดับความลึกของการเขียนโปรแกรมควบคุมอุปกรณ์คอมพิวเตอร์ได้เป็น 2 ชนิด คือ **ชนิดแรก**เป็นโปรแกรมระบบปฏิบัติการ (Operating System) ทำหน้าที่ในการควบคุมอุปกรณ์พื้นฐานที่เป็นสวนประกอบคอมพิวเตอร์ทั้ง 4 หน่วย(Unit) ดังนี้คือ 1) หน่วยอินพุท (Input Unit) เช่น แป้นป้อนข้อมูล (Key Board) เมาส์ (Mouse) เป็นต้น 2) หน่วยเอาท์พุท (Output Unit) เช่น จอแสดงผล (Monitor) ลำโพง (Speaker) เป็นต้น 3) หน่วยความจำ (Memory Unit) เช่น หน่วยความจำชั่วคราว (RAM) หน่วยความจำภายนอก (External Memory) เช่น ฮารดดิสก (Hard Disk) และหน่วยสุดท้ายหน่วย 4) หน่วยประมวลผลกลาง (CPU: Central Processing Unit) ด้วยเพราะเทคโนโลยีของอุปกรณ์พื้นฐานของคอมพิวเตอร์มีการแข่งขันสูงส่งผลทำให้โปรแกรมที่ทำหน้าที่ในการควบคุมอุปกรณ์คอมพิวเตอร์นี้ต้องมีการพัฒนาอย่างต่อเนื่อง ดังนั้นโปรแกรมระบบปฏิบัติการจึงเป็นโปรแกรมที่สำคัญ

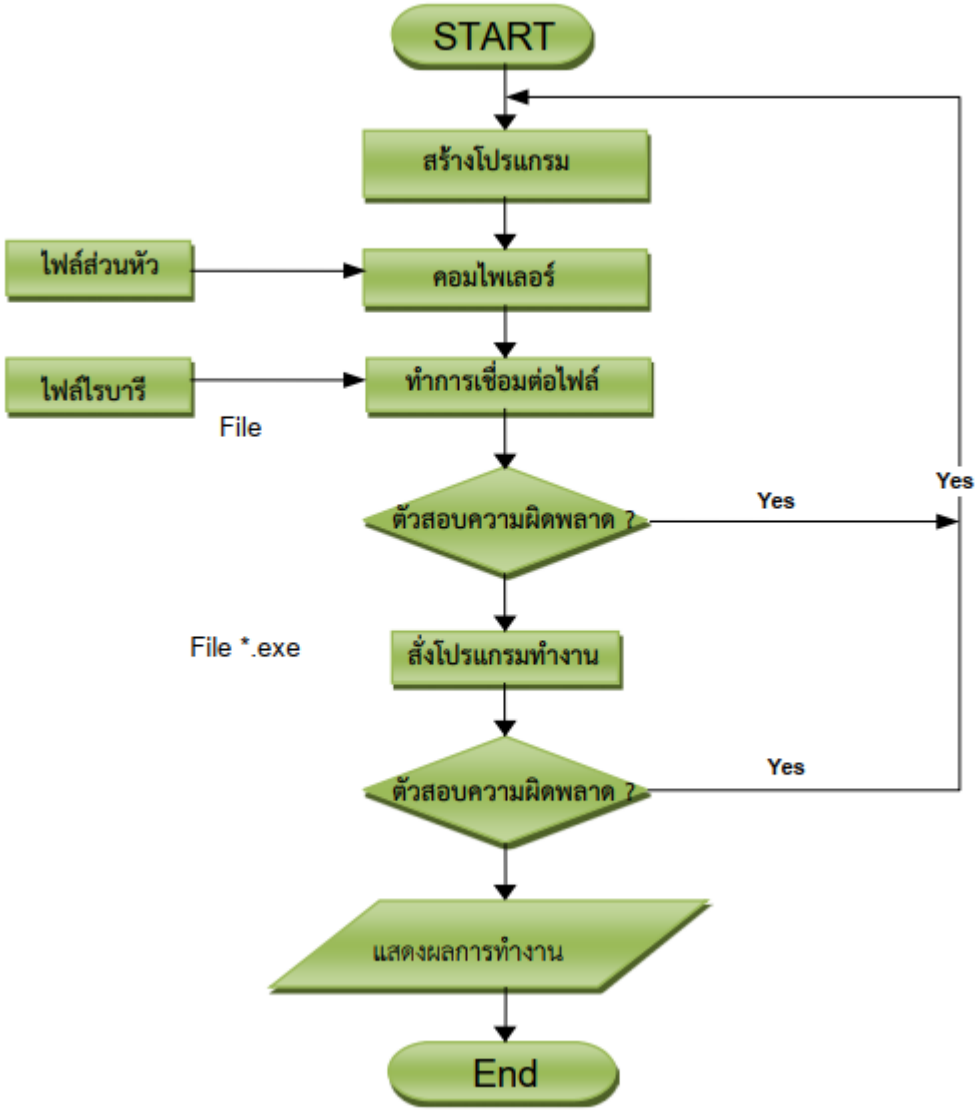
ต่อการ เจริญก้าวหน้าของบริษัทที่ทำหน้าที่ในการผลิตเครื่องคอมพิวเตอร์ โปรแกรมชนิดนี้จึงเป็นเทคโนโลยีปิด นักเขียนโปรแกรมคอมพิวเตอร์จึงมีระดับการเข้าถึงในการกำหนดค่าการควบคุมอุปกรณ์ต่างๆ มีอย่างจำกัดตามที่บริษัทที่พัฒนาโปรแกรมระบบปฏิบัติการอนุญาตให้เข้าถึงได้เท่านั้น โปรแกรมดังกล่าว เช่น โปรแกรมระบบปฏิบัติการวินโดว (Windows Operating System) โปรแกรมระบบปฏิบัติการลินุกส (Linux Operating System) เป็นต้น **ชนิดที่สอง**เป็นโปรแกรมภาษาคอมพิวเตอร์ ที่นักพัฒนาโปรแกรมคอมพิวเตอร์ สามารถเข้าถึงในการเขียนโปรกรมควบคุมอุปกรณ์ต่างๆได้อย่างกว้างขวางโปรแกรมชนิดภาษาคอมพิวเตอร์นี้ จึงเป็นโปรแกรมที่นักพัฒนาโปรแกรมสามารถเรียนรูและสร้างสรรค์งานไดอย่างต่อเนื่องตลอดเวลา ดังนั้นภาษาคอมพิวเตอร์จึงสามารถ

แบ่งชนิดตามลักษณะการมองปัญหาของโจทย์ ที่ต้องการให้การเขียนโปรแกรม ทำการแก้ปัญหาได้เป็น 2 แบบคือ แบบแรกเป็นการเขียนโปรแกรมแบบโครงสร้าง (Structure Programming) ตัวอย่างภาษาที่ใช้เขียนโปรแกรม เช่น โปรแกรม ภาษาซี (C Language Program) โปรแกรมภาษาปาสคาล (Pascal Language Program) โปรแกรม ภาษาเบสิค (Basic Language Program) เป็นต้น ส่วนแบบที่สองเป็นการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming : OOP) ตัวอย่างภาษาที่ใช้เขียนโปรแกรม เช่น โปรแกรม C++ (C++ Language Program) โปรแกรมภาษาซีชารป (C# Language Program) โปรแกรมพีเอสพี (PHP Program) โปรแกรมวิชวลเบสิค (Visual Basic Program) โปรแกรมภาษาจาวา (Java Language Program) เปนตน

1.2 กระบวนการทำงานการเขียนโปรแกรมแบบโครงสร้าง

ในการพัฒนาโปรแกรมแบบโครงสรางหรือการเขียนโปรแกรมแบบโครง ซึ่งมีกระบวนการในการสร้าง คือ ตัวอย่างเช่น การโปรแกรมภาษาซี เริ่มต้นด้วยการจากทำการสร้างโปรแกรม (source program หรือ source code) ซึ่งต้องเขียนตามลักษณะโครงสร้างของภาษาซี ทำการบันทึก (save หรือ save as) เป็นไฟล์นามสกุล .c นำโคดได้ทำการเขียน ทำการคอมไพล ทำการตรวจสอบข้อผิดผลาดว่ามีหรือไม ถ้าไม่มีให้กลับไปแกไขที่โปรแกรม แต่ถ้ามีข้อผิดพลาดตัวระบบภาษาซีจะสร้างไฟล์นามสกุล .obj แล้วทำการเชื่อมต่อไฟล์จากไฟล์ไรบารีและระบบจะทำการสร้างไฟล์นามสกุล .obj และทำการตรวจสอบความผิดพลาดในการเชื่อมต่อ ถ้าไม่มีข้อผิดพลาดจะไดไฟล์นามสกุล .exe ซึ่งเป็นไฟล์ที่พรอมนำไปสั่งงานให้แสดงผลการทำงานตามที่กำหนดของโปรแกรม ไฟล์ที่เป็นนามสกุล .exe นี้ ไมสามารถทำการแกไขไดด้วยเพราะเป็นไฟล์ที่เป็นรหัสเครื่องคอมพิวเตอร์ทำงาน ดังนั้นถ้าต้องการ

ปรับปรุงหรือแกไขโปรแกรมผู้เขียนโปรแกรมต้องมีไฟล์ที่เปนนามสกุล .c มาเขากระบวนการประมวลผลภาษาซีใหม่ตั้งแต่เริ่มต้น แสดงใน**ภาพที่ 1.1**



ภาพที่ 1.1 แสดงกระบวนการประมวลผลของภาษาซี

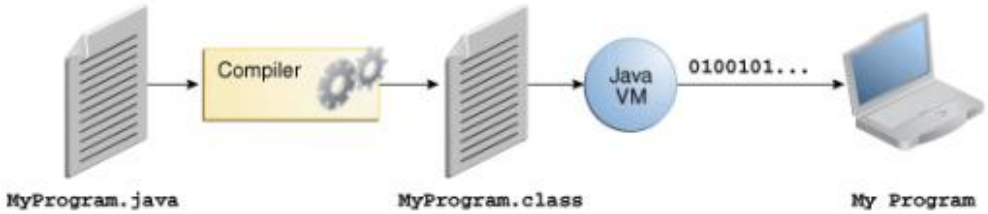
1.3 ความความเป็นมาภาษาจาวา

การพัฒนาภาษาจาวาถูกพัฒนาโดย ซัน ไมโครซิสเต็มท บจก. (sun microsystem co., ltd) โดยขณะที่ทีมงานของบริษัททำการพัฒนาภาษามักนิยมดื่มกาแฟกันเปนประจำ และกาแฟที่นิยมดื่มมีชื่อวา จาวา ทีมพัฒนาภาษาจาวาตนแบบจึงตั้งชื่อภาษาที่พัฒนาวา จาวา จึงเปนที่มาเทคโนโลยีของบริษัท ซัน ไมโครซิสเต็มท จำกัด เปนตนมา ซึ่งทำให บริษัท ซัน ฯลฯ เปนผูกำกับทิศทางไมใหมีการนำ

ภาษาจาวาไปดัดแปลงประยุกตใชในทางที่เบี่ยงเบนออกจากจุดประสงคเดิม การกำหนดทิศทางโดย ซันเปนไปเพื่อใหเกิดความชัดเจนในทิศทางการพัฒนาภาษาจาวา โดยภาษาจาวามีคำสั่งพื้นฐานคลายภาษาซี พลัส พลัสเปนอยางมาก ทำใหนักเขียนโปรแกรมภาษาซี พลัส พลัส ที่มีอยูมากที่สุดแลวในขณะนั้นสามารถสรางความคุนเคยไดอยางรวดเร็ว จึงทำใหภาษาจาวาเปนที่ยอมรับอยางกวางขวางอยางรวดเร็ว และไดรับการจัดเปนภาษาคอมพิวเตอรเชิงวัตถุ เชนเดียวกับภาษาซี พลัส พลัส ดวย แตสิ่งที่ทั้งสองภาษาตางกัน คือ โปรแกรมภาษาจาวาตองเขียนเปนแบบเชิงวัตถุเทานั้น ในขณะที่ภาษาซี พลัส พลัส สามารถเขียนไดทั้งแบบเชิงวัตถุ และแบบโครงสรางก็ได ที่เปนเชนนี้ไดเนื่องจากภาษาซี พลัส พลัส มีตนกำเนิดมาจากภาษาซี ซึ่งเปนภาษาแบบโครงสราง ดังนั้นภาษาซี พลัส พลัส จึงตองสนับสนุนการเขียนโปรแกรมแบบโครงสรางดวยเพื่อใหเขากันไดกับภาษาซี อยางไรก็ตามปจจุบันโปรแกรมประยุกตลวนแตเขียนดวยภาษาเชิงวัตถุทั้งสิ้น

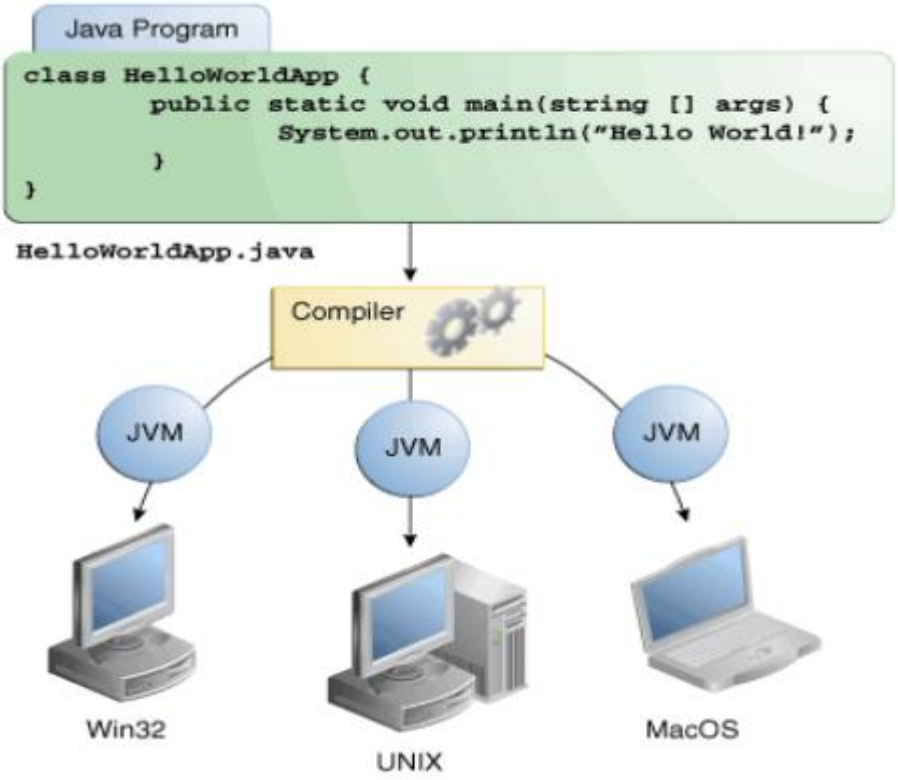
1.3.1 กระบวนการทำงานของโปรแกรมภาษาจาวา

การทำงานโปรแกรมภาษาจาวา การสรางไฟลตนฉบับ (source file) หรือ โคดตนฉบับ (source code) แลวทำการบันทึกไฟลดังกลาวโดยบันทึก นามสกุล .java เก็บไวใน folder ที่ตองการแลวทำ การ compiler ดวย java compiler ในระบบปฏิบัติการ DOS เพื่อเปลี่ยนคำสั่งภาษาจาวาเปนภาษาเฉพาะอยางหนึ่งเพื่อใหจาวาเวอรชัวรแมทชีนเขาใจ ภาษาเฉพาะที่จาวาเวอรชัวนแมทชีนเขาใจนี้เรา เรียกวาจาวาไบตโคด เปนไฟลที่มีนามสกุล .class แลวนำไฟล ดังกลาวไปรันบนเครื่องคอมพิวเตอรที่มีจาวาเวอรชัวรแมทชีน สรุปขั้นตอนการพัฒนาโปรแกรมดวยภาษาจาไดแสดงดัง**ภาพ 1.2**



ภาพที่ 1.2 กระบวนการพัฒนาภาษาจาวา

ดวยจาวาเวอรชัวนแมทชีน มีความสามารถนำมาใชกับระบบปฏิบัติการที่แตกตางกันได โดยใชคลาสที่คอมไพลตัวเดียวกัน นำมาทำงานบนระบบของไมโครซอฟทวินโดว (Microsoft Windows) ระบบโซลาริส (Solaris) หรือ ระบบยูนิกส (Unix) รวมทั้งระบบของแม็ค (Mac) ได ดังแสดงใน**ภาพที่ 1.3**



ภาพที่ 1.3 แสดงระบบจาวาเวอรชัวนแมทชีนที่สามารถใชไดทุกระบบปฏิบัติการ

1.3.2 องค์ประกอบของเทคโนโลยีจาวา

เทคโนโลยีภาษาจาวามีองคประกอบที่สำคัญในการทำงาน ประกอบดวย 3 สวน คือ

1. จาวาเวอรชัวนแมทชีน (Java Virtual Machine: JVM) เปนสวนที่ทำหนาที่เปน ตัวอินเตอร - พรีเตอร (interpreter) คือ จะทำการแปลจาวาไบตโคด ใหเปนภาษาที่เครื่องเขาใจ ซึ่งจาวาไบตโคดจะสามารถรันไดหลายแพลตฟอรม ถาแพลตฟอรมนั้นมี จาวาเวอรชัวนแมทชีน

2. จาวารันไทลอินวิโรเมนต (Java Runtime Environment: JRE) เปนสวนที่ใช ในการรันโปรแกรมภาษาจาวา เปนสวนที่ใชในการรันโปรแกรม โดยจะทำงาน 3 ขั้นตอน ดังนี้ ขั้นตอนแรกโหลดไบตโคดโดยใช Class loader คือการโหลดคลาสทุกคลาสที่เกี่ยวของในการรันโปรแกรม ขั้นตอนที่สองตรวจสอบไบตโคดโดยใชการตรวจสอบไบทโคด (Byte code Verifier) คือการตรวจสอบ วาโปรแกรมจะตองไมมีคำสั่งที่ทำใหเกิดความผิดพลาดกับระบบ เชน การแปลงขอมูลที่ผิดพลาด หรือมีการแทรกแซงเขาสูระบบภายในเปนตน ขั้นตอนที่สุดทาย รันไบตโคด โดยใชตัวแปลรันไทล (Runtime

Interpreter)

3. ชุดพัฒนาภาษาจาวา (Java 2 Software Developer Kit: J2SDK) เปนชุดพัฒนาโปรแกรมภาษาจาวา ประกอบไปดวยโปรแกรมตางๆ แตไมมีโปรแกรมที่ทำหนาที่สรางโคด (Editor) รวมอยูดวย แตมีโปรแกรมคอมไพเลอร (javac.exe) และโปรแกรมอินเตอรพรีเตอร (java.exe)

1.3.3 แพลตฟอร์มของจาวา

แพลตฟอรมของจาวา หมายถึงลักษณะการนำภาษาจาวาไปสรางเปนโปรแกรมเพื่อวัตถุประสงคใด วัตถุประสงคหนึ่ง บนการประมวลผลของคอมพิวเตอร อาจเปนการทำการประมวลผล โดยไมมีการเชื่อมตอกับเครื่องคอมพิวเตอรเครื่องอื่นๆ หรือตองการนำโปรแกรมภาษาจาวาไปใชกับระบบเครือขายอินเทอรเน็ต เปนตน ซึ่งแพลตฟอรมของจาวาจึงมีการแบงรูปแบบเปน 3 รูปแบบ คือ

1. แพลตฟอรมจาวามาตรฐาน (Standard Edition: Java SE) เปนแพลตฟอรมที่ใชในการพัฒนาโปรแกรมภาษาจาวากับเครื่องพีซีทั่วไป ซึ่งสามารถแยกยอยออกเปน 2 แบบ คือ Java Application เปนโปรแกรมที่ใชพัฒนาโปรแกรมประยุกตทั่วไป และ Java Applet เปนโปรแกรมจาวาที่ใชพัฒนาโปรแกรมเพื่อรันบนเว็บบราวเซอร

2. แพลตฟอรมจาวาอินเตอรไพรส (Enterprise Edition: Java EE) เปนแพลตฟอรมที่มุงเนนในการพัฒนาโปรแกรมเครือขาย สำหรับใชงานในองคกร โดยใชโปรแกรม Application Server หรือ Web Server ใชสำหรับพัฒนาโปรแกรมที่มีขนาดใหญ

3. แพลตฟอรมจาวาไมโคร (Micro Edition: Java ME) เปนแพลตฟอรมที่ใชพัฒนาโปรแกรมเพื่อใชงานกับอุปกรณอิเล็กทรอนิกส ที่มีทรัพยากรจำกัด เชน โทรศัพทมือถือ เปนตน

1.4 การติดตั้ง Editor และ SDK โปรแกรมภาษาจาวา

การพัฒนาโปรแกรมดวยภาษาจาวา สำหรับคอมพิวเตอรสวนบุคคล เครื่องคอมพิวเตอรจำเป็นตองอาศัยชุดพัฒนา J2SDK แตดวยชุดพัฒนาไมมีโปรแกรมที่ทำหนาที่สราง source code ซึ่งในระบบปฏิบัติการวินโดวสสามารถใชโปรแกรมโนตแพด (notepad) ทำการสรางโคดภาษาจาวา เพื่อทำการเก็บโคดดวยการบันทึกเปนนามสกุล และสวนขยาย .java ซึ่งชุดพัฒนา J2SDK สวนตัวคอมไพลและตัวรันโปรแกรม สามารถทำการดาวนโหลดและติดตั้ง J2SDK ลงบนเครื่องคอมพิวเตอรระบบปฏิบัติการวินโดวสไดที่ *http://java.sun.com/javase/downloads/index.jsp*

การติดตั้ง Editor โปรแกรมภาษาจาวาประกอบด้วยดังนี้

- การติดตั้ง Java SDK และการปรับแต่งค่าพื้นฐาน (Install Java SDK and Config)

  http://www.oracle.com/technetwork/java/javase/downloads/index.html

- การติดตั้ง Eclipse สำหรับเขียน Java (Install Eclipse IDE for Java)

  http://www.eclipse.org/downloads/

- การติดตั้ง Netbeans สำหรับเขียน Java (Install Netbeans IDE for Java)

  http://netbeans.org/downloads/

- การติดตั้ง IntelliJ IDEA

  https://www.jetbrains.com/idea/

- การติดตั้งโปรแกรม Edit Plus

  http://www.editplus.com/download.html

1.5 หลักการเขียนโปรแกรมภาษาจาวา

การเขียนโปรแกรมจาวาตามที่กลาวมาขางตน ตัวซอรสโคดสามารถทำการเขียนดวยอิดิเตอรใดก็ได Notepad, IntelliJ หรือ Edit Plus สิ่งสำคัญในการบันทึกชื่อของซอรสโคด ตองเปนชื่อเดียวกับชื่อ classดวยเพราะ เมื่อซอรสโคดถูกคอมไพลตัวคอมไพเลอรจะทำการสรางคลาสที่เปนไบทโคดขึ้นมาโดยใชชื่อของแฟมขอมูลที่บันทึกจากนามสกุล .java มาเปนนามสกุล .class การสรางโคดภาษาจาวา มีความคลายกับการสรางโคดใหกับภาษาซี หรือ ซีพลัสพลัส คือตองมี เมธอด main() เปนจุดเริ่มตนการทำงาน และตองถูกกำหนดใหอยูในคลาสใดคลาสหนึ่ง ของโคดจาวาเมธอด main( ) จะตองปรากฏไดเพียงครั้งเดียวในหนึ่งโคดโปรแกรมภาษาจาวา ดวยเพราะเปนเมธอดหลัก และถาโคดใดไมมีสวนของเมธอด main() โคดนั้นจะไมสมบูรณ หมายถึงไมสามารทำงานไดเมธอด main () ตามกฎของโปรแกรมภาษาจาวา ตองมีสวนประกอบประโยคดังนี้ public static void main(String args[]){ } การกำหนดใหเปนเมธอด main() เปน public เพื่อตองการใหโคดที่เขียน สามารถเขาถึงหรือเรียกใชงานไดขณะเมื่อสังใหทำการการรัน (interpreter) สวนการกำหนดใหคลาสเปน static ดวย มีผลทำใหเมธอด

main() สามารถเรียกใชไดโดยไมตองสราง instance ของคลาส ดังนั้นทุกครั้งที่มีการสรางโคดโปรแกรมภาษาจาวา ตองมีเมธอด main() ที่ประกอบไปดวย public static void และสวน Argument ตองเปน String Array เสมอ ซึ่งจะมีการกำหนดคาในอารเรยอารกูเมนต หรือไมกำหนดก็สามารถสรางโคดได ตัวอยางโปรแกรมที่ 1.1 ชื่อ first.java ตัวอยางโปรแกรมที่ 1.1 เปนตัวอยางแรก ซึ่งมีโครงสรางพื้นฐานที่จำเปนในการสรางโปรแกรมครั้งแรก ดังนั้นการพิมพคำสั่งทุกคำสั่งตองถูกตอง โดยเฉพาะอยางยิ่งตัวอักษรพิมพเล็ก พิมพใหญ ตองถูกตองทั้งหมด บันทึกเปนชื่อ first.java (โดยใหทำการสำเนาชื่อแฟมขอมูล ที่อยูหลัง public class เปนชื่อของแฟมขอมูลที่ตองการบันทึก เปนการลดปญหาการผิดพลาดจากการใชชื่อที่ประกอบดวยตัวอักษรพิมพเล็กพิมพใหญ)

```
public class first{
public static void main(String args[]){
        System.out.println("Hello World!!!");
        }
}
```

โปรแกรมที่ 1.1

ตัวอยางโปรแกรมภาษาจาวา ที่มีการกำหนดรับคาใหกับอารเรยอารกูเมนต ผานทางการสั่ง Run ดังตัวอยาง**โปรแกรมที่ 1.2** ชื่อ FirstArgs.java

```
public class FirstArgs{
public static void main(String args[])
    {
       System.out.println("Created By>>>"+args[0]);
       System.out.print("Welcome to java, ");
       System.out.println("Thank you for looking. ");
       System.out.print("-..-");
    }
}
```

โปรแกรม 1.2

จากโปรแกรม 1.2 สวนของคำสั่ง System.out.println("Created By>>>"+args[0]); เปนการเรียกเมธอด println อยูในคลาส out ในแพ็คเก็ต System มาทำการแสดงผลออกทางจอหนา โดย

เมธอด println เปนการสั่งใหขอความหรือคาของตัวแปรที่อยูในอารกูเมนตลิสต (Created By>>>) ออกสูหนาจอ และในสวนของอารกูเมนตลิสต คาที่สอง (args[0]) เปนการรับคาจากแปนพิมพขณะทำการสั่งให Run โปรแกรม ดังภาพที่ 1.22 ไดยกตัวอยางคาที่สงเปนขอความ SKUL ซึ่งสามารถเปลี่ยนคาไดจากเมธอด println มีผลการแสดงคาในอารกูเมนตลิสตเสร็จแลวระบบจะทำการขึ้นบรรทัดใหมรอคำสั่งถัดไป สวนเมธอด print ของ System.out.print("Welcome to java, "); เปนการสั่งใหแสดงคา Welcome to java, โดยแสดงเสร็จแลว เคอรเซอรยังคงอยูที่บรรทัดเดิม เมื่อมีคำสั่งตอมาเปนเรียกใชเมธอด println จะทำการแสดงขอความหรือคาของตัวแปรที่อยูในอารกูเมนตลิสตออกทางหนาจอ

คำถามท้ายบท

1. การเขียนโปรแกรมภาษาจาวามีความแตกตาง หรือสัมพันธกับภาษาแบบโครงสรางอยางไร
2. จงอธิบายความแตกตางระหวางโปรแกรม Notepad กับ EditPlus
3. จงอธิบายความสำคัญของการแปลภาษา (Compiler) วาสำคัญอยางไร
4. จงอธิบายความสำคัญของการสั่งทำงาน (Run) วาสำคัญอยางไร
5. ในการสั่งจบคำสั่งของโปรแกรมภาษาจาวาในแตละบรรทัด ใชเครื่องหมายใด

บทที่ 2 ชนิดข้อมูลและตัวแปร (Java Data types and Variable)

วัตถุประสงค์ของหน่วยการเรียนรู้

1. เกิดทักษะการเขียนโปรแกรมคอมพิวเตอร์
2. อธิบายการกำหนดตัวแปรและชนิดข้อมูล
3. บอกขั้นตอนการทำงานของโปรแกรมภาษาจาวาได
4. เขียนโปรแกรมภาษาจาวาตามรูปแบบภาษาได้

Java Variable and Data types

Java Variable and Data types ในการสร้างตัวแปรในภาษา Java ก่อนการเก็บค่าตัวแปร (Variable) ต่าง ๆ ในภาษา Java บังคับจะต้องให้ทำการประกาศชื่อตัวแปรและชนิดของตัวแปรก่อนเสมอ และภาษา Java ค่อนข้างจะมีโครงสร้างเกี่ยวกับ Data type ค่อนข้างแข็งแรงมาก และให้ความสำคัญกับชนิดของตัวแปร เช่น การบวก ลบ คูณ หาร และส่งค่า Parameters ต่าง ๆ ไปยังส่วนของ ๆ ของ class , method , properties จะต้องมีการ Assign ชนิดของตัวแปรให้ถูกต้อง รวมทั้งจะต้องส่ง ชนิดของตัวแปรให้ถูกต้องด้วยเช่นเดียวกัน แต่ข้อดีอย่างหนึ่งของภาษา Java ก็คือ เราสามารถทำการ Casting ค่าตัวแปรให้อยู่ในรูปแบบชนิดต่าง ๆ ที่ต้องการ ก่อนที่จะนำไปใช้ได้ และในส่วนของ Compiler ยังมีการตรวจสอบความถูกต้องก่อนที่จะทำการ Run โปรแกรมด้วย เช่น ถ้ายังมีข้อผิดพลาด ตัว Java Compiler จะทำแจ้ง Error นั้น ๆ ให้เราทำการแก้ไขถูกต้องเสียก่อน

สำหรับตัวแปรในภาษา **Java** จะมีรูปแบบตัวแปรเหมือนกับภาษา C หรือ C# ยิ่งถ้าเคย .Net แบบ C# อยู่แล้ว แทบจะเรียกได้ว่า พวกตัวแปรไม่ต่างกันเลย และในภาษา Java สามารถแยกชนิดของตัวแปรออกเป็น 4 ชนิดหลัก ๆ ที่เราจะได้ใช้เป็นประจำคือ

- “จำนวนเต็ม ตัวเลข (ไม่มีทศนิยม)” ได้แก่ **byte , short ,int ,long**

- “จำนวนจริง ตัวเลข (ทศนิยม)” ได้แก่ float , double
- “ตัวอักษรข้อความ” ได้แก่ **char**(ตัวอักษรตัวเดียว) ,**String** (หลายตัวอักษรหรือเป็นคำ)
- “บูลีน ตรรกะ (จริงหรือเท็จ)” ได้แก่ **boolean**

สำหรับ 4 กลุ่มนี้คือตัวแปรพื้นฐานทั่ว ๆ ไปที่เราจะใช้อยู่ประจำ ในการสร้างไว้สำหรับจัดเก็บค่าต่าง ๆ ของโปรแกรม และนอกจากนี้ยังมีพวกตัวแปรที่อยู่ในรูปแบบของ **Object** ที่จัดเก็บชนิดของ **Object หรือ Class library** ต่าง ๆ ที่เราสร้างขึ้น

<u>กฎพื้นฐานในการตั้งชื่อตัวแปรด้วยภาษา Java</u>

- ชื่อตัวแปร จะประกาศได้จะต้องประกาศชนิดของตัวแปรซะก่อน
- ชื่อต้องประกอบด้วย ตัวอักษร ตัวเลข $ หรือ _
- ชื่อตัวแปรห้ามใช้ตัวเลขขึ้นก่อน ห้ามใช้ space หรือพวกอักขระพิเศษต่าง ๆ
- ตัวพิมพ์เล็กและใหญ่มีค่าความหมายและชื่อต่างกัน
- จะต้องไม่ตรงกับคำสวงนต่าง ๆ ที่อยู่ในภาษา Java

<u>ธรรมเนียมนิยมในการตั้งชื่อตัวแปรด้วยภาษา Java</u>

- ชื่อคลาสขึ้นต้นด้วยตัวอักษรภาษาอังกฤษตัวใหญ่
- ชื่อตัวแปรขึ้นต้นด้วยตัวอักษรภาษาอังกฎษตัวเล็ก
- ตั้งชื่อให้สื่อความหมาย เช่น ถ้า String ก็อาจจะใช้ str*Name*

คำสงวน(reserve word) ในภาษาจาวามีดังนี้

abstract	do	import	public	throws	boolean	
double	instanceof		return	transient	break	else
int	short	try	byte	extends	interface	
static	void	case	final	long	strictfp	
vovatile	catch	finally	native	super	while	

char	float	new	switch	null	class
for	package	synchronized		true	continue if
private	this	false	default	implements	protected
throw					

```
// int
int num1 = 5;
int num2;
num2 = 10;
int num3 = 20 , num4 = 30;

// String
String name = "Win";
String firstname, lastname;
```

```
package com.java.myapp;

public class MyClass {

  int aVar = 0; // ใช้ได้ทั้งหมดภายใน MyClass

     public static void main(String[] args) {

           int bVar = 0; // ใช้เฉพาะใน method main

     }

     public static void method1() {

           int cVar = 0; // ใช้เฉพาะใน method method1

     }

}
```

Java String (data types)

Java String (data types) เป็นตัวแปร String ในภาษา Java ไว้สำหรับจัดเก็บพวกข้อความต่าง ๆ

Syntax String varname;

```
package com.java.myapp;

public class MyClass {

     public static void main(String[] args) {

          String myName = "Weerachai Nukitram";
         System.out.println("Hello " + myName);
```

```
        System.out.println("String Length = " + myName.length()
);

    }

}
```

Output

```
Hello Weerachai Nukitram
String Length = 18
```

Java byte (data types)

Java byte (data types) สำหรับ byte ตัวแปรในภาษา Java จะใช้จัดเก็บจำนวนเต็ม (integers) ที่มีขนาดอยู่ในช่วง -128 ถึง 127 ใช้ขนาด 1 (byte)

Syntax `byte varname;`

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

         byte aVar = 59;
         byte bVar = -35;

        System.out.println("aVar = " + aVar);
        System.out.println("bVar = " + bVar);

    }

}
```

Output

```
aVar = 59
bVar = -35
```

Java short (data types)

Java short (data types) สำหรับ short ตัวแปรในภาษา Java จะใช้จัดเก็บจำนวนเต็ม (integers) ที่มีขนาดอยู่ในช่วง -32768 ถึง 32767 ใช้ขนาด (2 byte)

Java int (data types)

Java int (data types) สำหรับ int ตัวแปรในภาษา Java จะใช้จัดเก็บจำนวนเต็ม (integers) ที่มีขนาดอยู่ในช่วง -214783648 ถึง 2147483647 ใช้ขนาด (4 byte)

Syntax `int varname;`

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

          int aVar;
          aVar = 1;
          int bVar = 2;
          int cVar = 3 , dVar = 40000;

         System.out.println("aVar = " + aVar);
         System.out.println("bVar = " + bVar);
         System.out.println("cVar = " + cVar);
         System.out.println("dVar = " + dVar);

    }

}
```

Output

```
aVar = 1
bVar = 2
cVar = 3
dVar = 40000
```

Java long (data types)

Java long (data types) สำหรับ long ตัวแปรในภาษา Java จะใช้จัดเก็บจำนวนเต็ม (integers) ที่มีขนาดอยู่ในช่วง -223372036854775808 ถึง 9223372036854775807 ใช้ขนาด (8 byte)

Syntax

```
long varname;
```

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

          long aVar = 6553665536;

         System.out.println("aVar = " + aVar);

    }

}
```

Output

```
aVar = 6553665536
```

Java float (data types)

Java float (data types) สำหรับ float ตัวแปรในภาษา Java จะใช้จัดเก็บตัวเลขในรูปแบบของทศนิยม ที่มีขนาดอยู๊ในช่วง -1.4 x 10-45 ถึง 3.40282 x 1038 ใช้ขนาด (4 byte)

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

         float aVar = 15.253f;
        System.out.println("aVar = " + aVar);

        String fm;
        fm = String.format("aVar = %.2f", aVar);
        System.out.println(fm);

    }

}
```

Output

```
aVar = 15.253
aVar = 15.25
```

Java double (data types)

Java double (data types) สำหรับ double ตัวแปรในภาษา Java จะใช้จัดเก็บตัวเลขในรูปแบบของทศนิยม ที่มีขนาดอยู๊ในช่วง -4.9 x 10-324 ถึง 1.79769 x 10308 ใช้ขนาด (8 byte)

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

         double aVar = 9.55321;
        System.out.println("aVar = " + aVar);

         double bVar = 3.1234;
        System.out.println("bVar = " + bVar);

        String fm;
```

```
        fm = String.format("aVar + bVar = %.2f", aVar + bVar);
        System.out.println(fm);

    }

}
```

Output

```
aVar = 9.55321
bVar = 3.1234
aVar + bVar = 12.68
```

Java boolean (data types)

Java boolean (data types) สำหรับ boolean ตัวแปรในภาษา Java จะใช้จัดเก็บตัวแปรตรรกที่มีค่าเป็น จริงและเท็จ (true/false)

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

         boolean aVar = true;
         boolean bVar = false;
         boolean cVar;
         cVar = false;

        System.out.println("aVar = " + aVar);
        System.out.println("bVar = " + bVar);
        System.out.println("cVar = " + cVar);

    }

}
```

Output

```
aVar = true
bVar = false
cVar = false
```

Java char (data types)

Java char (data types) สำหรับ char ตัวแปรในภาษา Java จะใช้จัดเก็บตัวแปรชนิดอักขระ (characters) ได้แก่ ตัวอักษร , ตัวเลข , เครื่องหมาย , และสัญลักษณ์ต่างๆ เราใช้เครื่องหมายคำพูดเดี่ยว ' (single quote) ล้อมตัวอักษรนั้นเวลาเขียน เช่น 'a' , 'A' , '0' , '$' เป็นต้น

Java Constants and final

Java Constants and final สำหรับ Constant หรือ final เป็นการกำหนดค่าคงที่ในตัวแปรภาษา Java ซึ่งสามารถกำหนดได้กับตัวแปรเกือบทุกชนิด และเมื่อประกาศ final แล้วตัวแปรนั้นจะไม่สามารถทำการเปลี่ยนแปลงค่าได้อีก

Syntax

```
final type varname;
```

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

         int aVar = 10;
         final int bVar = 20;

         aVar = 100;
         // bVar = 200; // cannot new assign value

        System.out.println("aVar = " + aVar);
        System.out.println("bVar = " + bVar);

    }

}
```

Output

```
aVar = 100
bVar = 20
```

Java StringBuffer

Java StringBuffer สำหรับ StringBuffer เป็น Class ไว้สำหรับการจัดการกับ String และ Object เช่นสามารถทำการรวม edit หรือสร้าง String ขึ้นจากหลายๆ object โดยการ append เข้าด้วยกัน โดยไม่จำเป็นจะต้องเป็น String ท่านั้น

```
StringBuffer sb=new StringBuffer();
sb.append("String1");
sb.append("String2");
sb.append("String3");
```

```
package com.java.myapp;

public class MyClass {
    public static void main(String[] args) {

          StringBuffer aVar = new StringBuffer("String  1");
          aVar.append(" String  2");
          aVar.append(" String  3");
          aVar.append(" String  4");
          System.out.println(aVar);

          System.out.println("Length = " + aVar.length());

          aVar.reverse();
          System.out.println(aVar);

    }

}
```

Output

```
String 1 String 2 String 3 String 4
Length = 39
4 gnirtS 3 gnirtS 2 gnirtS 1 gnirtS
```

Java StringBuilder

Java StringBuilder สำหรับ StringBuilder เป็น Class ไว้สำหรับการจัดการกับ String เช่นสามารถทำการรวม edit หรือสร้าง String ขึ้นจากหลายๆ ชุด โดยการ append เข้าด้วยกัน

```
StringBuilder sb=new StringBuilder();
sb.append("String1");
sb.append("String2");
sb.append("String3");
```

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

          StringBuilder aVar = new StringBuilder();
          aVar.append(" String  1");
          aVar.append(" String  2");
          aVar.append(" String  3");
          System.out.println(aVar);

          System.out.println("Length = " + aVar.length());

          aVar.reverse();
          System.out.println(aVar);

    }
```

```
}
```

Output

```
String 1 String 2 String 3
Length = 30
3 gnirtS 2 gnirtS 1 gnirtS
```

Java char (data types)

Java char (data types) สำหรับ char ตัวแปรในภาษา Java จะใช้จัดเก็บตัวแปรชนิดอักขระ (characters) ได้แก่ ตัวอักษร , ตัวเลข , เครื่องหมาย , และสัญลักษณ์ต่างๆ เราใช้เครื่องหมายคำพูดเดี่ยว ' (single quote) ล้อมตัวอักษรนั้นเวลาเขียน เช่น 'a' , 'A' , '0' , '$' เป็นต้น

```
char varname;
```

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

          char aVar = 'a';
          char bVar = 13;
          Character cVar = new Character('b');

        System.out.println("aVar = " + aVar);
        System.out.println("bVar = " + bVar);
        System.out.println("cVar = " + cVar);

    }

}
```

Output

```
aVar = a
bVar =

cVar = b
```

คำถามท้ายบท

1. จะอธิบายประโยชน์ของชนิดข้อมูลและการกำหนดตัวแปร
2. จงเรียงลำดับชนิดข้อมูลจากมากไปหาน้อยในการกำหนดตัวแปร
3. จงยกตัวอย่างชนิดข้อมูลแต่ละประเภทในการกำหนดตัวแปร
4. เหตุใดจึงต้องมีการกำหนดชนิดก่อนการประมวลผลโปรแกรม

5. จงยกตัวอย่างประกอบในการกำหนดตัวแปรและชนิดข้อมูล 3 ตัวอย่างในรูปภาษา Java

บทที่ 3 เครื่องหมายดำเนินการ

(Java and Operators)

วัตถุประสงค์ของหน่วยการเรียนรู้

1. เกิดทักษะการเขียนโปรแกรมคอมพิวเตอร์ เกี่ยวกับเครื่องหมายที่ใช้ในการเปรียบเทียบ ดำเนินการทางคณิตศาสตร์ เพิ่มค่าและลดค่า และเครื่องหมายตรรก
2. อธิบายความแตกต่างหรือสัมพันธ์ระหว่างโอเปอร์เรเตอร์หรือเครื่องหมายดำเนินการ
3. ศึกษาขั้นตอนการทำงานของโปรแกรมภาษาจาวาได
4. เขียนโปรแกรมภาษาจาวาตามรูปแบบภาษาได้

Java and Operators

Java and Operators โอเปอร์เรเตอร์ หรือ เครื่องหมายดำเนินการ เป็นการกระทำกับชุดของตัวแปรบนภาษา Java ในรูปแบบต่าง ๆ เพื่อให้ได้ผลลัพธ์ที่ต้องการ และสามารถแยกย่อยออกเป็นหลายส่วนขึ้นอยู่กับวัตถุประสงค์ของการใช้งาน เช่น การ บวก ลบ คูณ หาร เปรียบเทียบค่า เพิ่ม ลด และอื่น ๆ ซึ่งหลัก ๆ แล้วจะประกอบด้วยดังนี้

- เครื่องหมายดำเนินการทางคณิตศาสตร์ (Arithmatic Operator) เช่น + , - , * , / , %
- เครื่องหมายเพิ่มค่าและลดค่า ++ , --
- เครื่องหมายที่ใช้ในการเปรียบเทียบ (Compare Operator) เช่น < , > , <= , >= , != , ==
- เครื่องหมายตรรก (Logical Operator) เช่น ! , & , | , ^ , && , ||

Java and Operators และเราจะคุ้น ๆ กับการใช้งานอยู่แล้ว และจะไม่ขออธิบายอะไรมาก เพราะเมื่อเขียนโปรแกรมไปเรื่อย ๆ เราจะคุ้นเคยกับเครื่องหมายเหล่านี้ และอาจจะได้ใช้งานเฉพาะบางตัว

ตัวอย่างการใช้ เครื่องหมายดำเนินการทางคณิตศาสตร์ (Arithmetic Operator)

เครื่องหมาย	สัญลักษณ์แทน	ตัวอย่าง
เลขบวก	+	+5 +5.0 +a
เลขลบ	-	-2 -2.0 -a
บวก	+	1+2 1.0+2.0 a+b
ลบ	-	2-1 2.0-1.0 a-b
คูณ	*	4*5 4.0*5.0 a*b
หาร	/	10/2 10.0/2.0 a/b

ตัวอย่างการใช้ เครื่องหมายที่ใช้ในการเปรียบเทียบ (Compare Operator)

ชื่อเครื่องหมาย	สัญลักษณ์	ตัวอย่าง
น้อยกว่า	<	a < b
มากกว่า	>	a > b
น้อยกว่าหรือเท่ากับ	<=	a <= b
มากกว่าหรือเท่ากับ	>=	a >= b
ไม่เท่ากับ	!=	a != b
เท่ากับ	==	a == b

ตัวอย่างการใช้ เครื่องหมายตรรก (Logical Operator)

ชื่อเครื่องหมาย	สัญลักษณ์	รูปแบบคำสั่ง
NOT	!	!x
AND	&	x & y
OR	|	x | y
XOR	^	x ^ y
AND	&&	x && y
OR	||	x || y

สรุปเครื่องหมาย Operators ของ Java

Operators	Precedence
postfix	*expr*++ *expr*--
unary	++*expr* --*expr* +*expr* -*expr* ~ !
multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	|
logical AND	&&
logical OR	||
ternary	? :
assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

Java and Operators example

Java and Operators example บทความก่อนหน้านี้เราได้รู้จักกับ Operator ในภาษา Java มาคร่าว ๆ กันแล้ว และในบทความนี้จะมาลองทดสอบการใช้งาน Operator แบบง่าย ๆ เช่น การ บวก ลบ คูณ หาร การเปรียบเทียบ ค่า และการใช้งานในรูปแบบอื่น ๆ ที่น่าสนใจ

Example 1 การ บวก ลบ คูณ หาร แบบง่าย ๆ

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        // result is now 3
        int result = 1 + 2;
        System.out.println(result);

        // result is now 2
        result = result - 1;
        System.out.println(result);
```

```
        // result is now 4
        result = result * 2;
        System.out.println(result);

        // result is now 2
        result = result / 2;
        System.out.println(result);

        // result is now 10
        result = result + 8;
        // result is now 3
        result = result % 7;
        System.out.println(result);

    }

}
```

Output

```
3
2
4
2
3
```

Example 2 การรวมข้อความ หรือ Concat String

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        String firstString = "This is";
        String secondString = " a concatenated string.";
        String thirdString = firstString+secondString;
        System.out.println(thirdString);

    }

}
```

Output

```
This is a concatenated string.
```

Example 3 การ เพิ่ม ลด ค่า

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        // result is now 1
        int result = +1;
        System.out.println(result);
        // result is now 0
```

```
        result--;
        System.out.println(result);
        // result is now 1
        result++;
        System.out.println(result);
        // result is now -1
        result = -result;
        System.out.println(result);
        boolean success = false;
        // false
        System.out.println(success);
        // true
        System.out.println(!success);

    }

}
```

Output

```
1
0
1
-1
false
true
```

Example 4 การ เพิ่ม ลด ค่า ในตัวแปร

```
package com.java.myapp;


public class MyClass {


    public static void main(String[] args) {


        int i = 3;

        i++;

        // prints 4

        System.out.println(i);

        ++i;

        // prints 5

        System.out.println(i);

        // prints 6

        System.out.println(++i);

        // prints 6

        System.out.println(i++);
```

```
        // prints 7
        System.out.println(i);

    }

}
```

Output

```
4
5
6
6
7
```

Example 5 การเปรียบเทียบตัวแปรด้วย if

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        int value1 = 1;
        int value2 = 2;
        if(value1 == value2)
            System.out.println("value1 == value2");
        if(value1 != value2)
            System.out.println("value1 != value2");
        if(value1 > value2)
            System.out.println("value1 > value2");
        if(value1 < value2)
            System.out.println("value1 < value2");
        if(value1 <= value2)
            System.out.println("value1 <= value2");

    }

}
```

Output

```
value1 != value2
value1 < value2
value1 <= value2
```

Example 6 การใช้ if และ && , ||

```
package com.java.myapp;

public class MyClass {
```

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

            int value1 = 1;

            int value2 = 2;

            if((value1 == 1) && (value2 == 2))

                System.out.println("value1 is 1 AND value2 is 2");

            if((value1 == 1) || (value2 == 1))

                System.out.println("value1 is 1 OR value2 is 1");

        }

    }
```

Output

```
value1 is 1 AND value2 is 2
value1 is 1 OR value2 is 1
```

Example 7 การใช้และเปรียบเทียบค่า Boolean

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        int value1 = 1;
        int value2 = 2;
        int result;
        boolean someCondition = true;
        result = someCondition ? value1 : value2;

        System.out.println(result);

    }

}
```

Output

```
1
```

Example 8 การใช้งานในรูปแบบอื่น ๆ

```
package com.java.myapp;


class Parent {}

class Child extends Parent implements MyInterface {}

interface MyInterface {}


public class MyClass {


    public static void main(String[] args) {


        Parent obj1 = new Parent();

        Parent obj2 = new Child();


        System.out.println("obj1 instanceof Parent: "

            + (obj1 instanceof Parent));

        System.out.println("obj1 instanceof Child: "

            + (obj1 instanceof Child));

        System.out.println("obj1 instanceof MyInterface: "

            + (obj1 instanceof MyInterface));
```

```
        System.out.println("obj2 instanceof Parent: "
            + (obj2 instanceof Parent));
        System.out.println("obj2 instanceof Child: "
            + (obj2 instanceof Child));
        System.out.println("obj2 instanceof MyInterface: "
            + (obj2 instanceof MyInterface));
    }
}
```

Output

```
obj1 instanceof Parent: true
obj1 instanceof Child: false
obj1 instanceof MyInterface: false
obj2 instanceof Parent: true
obj2 instanceof Child: true
obj2 instanceof MyInterface: true
```

คำถามท้ายบท

1. จะอธิบายประโยชน์ของเครื่องหมายดำเนินการ(Operator)
2. เครื่องหมายดำเนินการ(Operator) มีกี่ประเภท อะไรบ้าง
3. จงยกตัวอย่างเครื่องหมายดำเนินการ(Operator) ในรูปของโปรแกรม Java
4. จงยกตัวอย่างประกอบในการกำหนดตัวดำเนินการ 3 ตัวอย่าง
5. ในรูปภาษา Java ตัวดำเนินการมีความสำคัญอย่างไร

บทที่ 4 Java and Loop

วัตถุประสงค์ของหน่วยการเรียนรู้

1. เกิดทักษะการเขียนโปรแกรมคอมพิวเตอร์ รูปแบบการทำซ้ำ ๆ

2. อธิบายความแตกต่างหรือสัมพันธ์ระหว่างโปรแกรมภาษาจาวามีกับภาษาแบบโครงสร้างได

3. บอกขั้นตอนการทำงานของโปรแกรมภาษาจาวาได

4. เขียนโปรแกรมภาษาจาวาตามรูปแบบภาษาได้

Java and Loop

ในภาษา Java เป็นรูปแบบการทำซ้ำ ๆ โดยสามารถกำหนด Scope การทำงานภายใน Loop และให้ Loop ทำงานคำสั่งนั้น ๆ จนตรงกับเงื่อนไขหรือสิ้นสุดการทำงาน แล้วโปรแกรมค่อยออกจาก Loop นั้น โดยสรุปแล้ว Loop ในภาษา Java จะประกอบอยู่ 3 ตัวด้วยกันคือ while , do while และ for และมี Statement ที่ทำหน้าที่ควบคุมลูปเช่น break และ continue

Loop ในภาษา Java จะมีรูปแบบการใช้งานเหมือนกับภาษา C , PHP และ C# และการใช้งานต้องบอกว่าไม่มีความซับซ้อนอะไร และอาจจะมีความง่ายต่อการใช้งานมากกว่าภาษาอื่นด้วยซ้ำ ลองดูตัวอย่างได้จากบทความเหล่านี้

Java while Loop

Java while Loop เป็นรูปแบบลูป (Loop) ในภาษา Java โดยที่ลูป while จะทำงานไปเรื่อย ๆ ตราบใดที่เงื่อนไขนั้นยังเป็นจริง โดยจะพิจารณาเงื่อนไขแล้วค่อยทำ

Syntax:

```
while (statement)
{
statement;
}
```

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        int i = 1;
```

```
            while( i <= 10 ) {
               System.out.println("Value i : " + i );
               i++;
            }

         }

      }
```

จากตัวอย่าง Loop จะทำงานตราบใดที่ i <= 10 และเมื่อ i = 11 ก็จะหยุดออกจาก Loop และไม่ทำงานใน Loop

Output

```
Value i : 1
Value i : 2
Value i : 3
Value i : 4
Value i : 5
Value i : 6
Value i : 7
Value i : 8
Value i : 9
Value i : 10
```

Java do...while Loop

Java do...while Loop เป็นรูปแบบลูป (Loop) ในภาษา Java โดยที่ลูป do...while จะเข้าไปทำงานใน Loop ซะก่อนและถ้ามีค่าเป็นเท็จค่อยออกจาก Loop โดย do...while จะทำก่อนแล้วค่อยพิจารณาเงื่อนไข

Syntax

```
do {
statement ;
} while (condition) ;
```

Example MyClass.java

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        int i = 1;
        do{
           System.out.println("Value i : " + i );
           i++;
        }while( i <= 10 );

    }

}
```

จากตัวอย่าง Loop จะทำงานตราบใดที่ i <= 10 และเมื่อ i = 11 ก็จะหยุดออกจาก Loop และไม่ทำงานใน Loop

Output

```
Value i : 1
Value i : 2
Value i : 3
Value i : 4
Value i : 5
Value i : 6
Value i : 7
Value i : 8
Value i : 9
Value i : 10
```

Java for Loop

Java for Loop เป็นรูปแบบลูป (Loop) ในภาษา Java โดยที่ลูป for จะวนและทำซ้ำตามจำนวนที่กำหนดขึ้น

Syntax

```
for (initial_value ; condition ; increment/decrement)
{
statement ;
}
```

```
package com.java.myapp;

public class MyClass {

    private static String[] sCountry = new String[] {
        "Belgium", "France", "Italy", "Germany", "Spain"
    };

    public static void main(String[] args) {

        for(int i = 0; i < sCountry.length; i++) {
            System.out.println("Value index["+ i +"] : " +
sCountry[i] );
         }

    }

}
```

ตัวอย่างการใช้ Loop for เพื่อแสดงค่าใน Array ออกมาแสดงผล

Output

```
Value index[0] : Belgium
Value index[1] : France
Value index[2] : Italy
```

```
Value index[3] : Germany
Value index[4] : Spain
```

Java foreach Loop

Java foreach Loop ในภาษา Java จะไม่มี foreach ใช้งาน เพราะปกติแล้วเราสามารถใช้ Loop for ทำการอ้างถึงข้อมูลที่อยู่ใน Object และแสดงค่าออกมาทีล่ะรายการได้เช่นเดียวกัน

Syntax

```
for (value:object)
{
statement ;
}
```

```
package com.java.myapp;

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

public class MyClass {

    public static void main(String[] args) {

           List<String> country = new ArrayList<String>();
           country.add("Belgium");
           country.add("France");
           country.add("Italy");
           country.add("Germany");
           country.add("Spain");
           for(String c : country){
            System.out.println("Value : " + c);
           }

    }

}
```

ตัวอย่างการใช้ Loop for เพื่อแสดงค่าใน ArrayList ออกมาแสดงผล

Output

```
Value : Belgium
Value : France
Value : Italy
Value : Germany
Value : Spain
```

Java break (Loop)

Java break (Loop) ในภาษา Java คำสั่ง break จะเป็นการหยุดการทำงานของ Loop สามารถใช้ได้กับ Loop ทุกตัว เช่น for , while , do...while

Syntax

```
for (value:object)
{
statement ;
break;
}

while (statement)
{
statement;
break;
}

do {
statement ;
break;
} while (condition) ;
```

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        String[] sCountry = new String[] {
                "Belgium", "France", "Italy", "Germany",
"Spain"
            };

          for(String country : sCountry) {
                  if(country == "Italy") { break; }
                  System.out.println("Value : " + country );
          }

          System.out.println("==== End =====");

    }

}
```

ตัวอย่างการใช้ break เพื่อออกจาก Loop โดยถ้า country = "Italy" จะออกจาก Loop ทันที

Output

Value : Belgium

Value : France

==== End =====

Java continue (Loop)

Java continue (Loop) ในภาษา Java คำสั่ง continue คล้าย ๆ กับ break (ออกจาก Loop ทันที) แค่ continue จะไม่ออกจาก Loop เพียงแต่ไม่ทำงานบรรทัดที่เหลือ แต่จะกลับไปเริ่มต้น Loop ใหม่ และทำงานจนเสร็จสิ้น

```
for (value:object)
{
continue;
statement ;
}

while (statement)
{
continue;
statement;
}

do {
continue;
statement ;
} while (condition) ;
```

Example

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        String[] sCountry = new String[] {
                "Belgium", "France", "Italy", "Germany", "Spain"
            };

          for(String country : sCountry) {
                  if(country == "Italy") { continue; } // go to loop
                  System.out.println("Value : " + country );
          }

          System.out.println("==== End =====");

    }

}
```

ตัวอย่างการใช้ continue โดยเมื่อ country = "Italy" ตัว Loop จะไม่ทำงานที่เหลือ แต่จะไปทำงานเริ่มต้นที่ Loop ใหม่

Output

Value : Belgium

Value : France

Value : Germany

Value : Spain

==== End =====

คำถามท้ายบท

1. จะอธิบายประโยชน์ของคำสั่งลูปต่างอย่างสรุป
2. คำสั่งลูปมีกี่ประเภท อะไรบ้าง
3. จงยกตัวอย่างคำสั่งลูปโดยให้มีการคำนวณทางคณิตศาสตร์
4. จงยกตัวอย่างประกอบในคำสั่งลูป 3 ตัวอย่างในรูปภาษา Java
5. จงอธิบายข้อแตกต่างในประเภทพอสังเขปและยกตัวอย่างประกอบ

บทที่ 5 Java and Condition/Control-Flow Statement

วัตถุประสงค์ของหน่วยการเรียนรู้

1. เกิดทักษะการเขียนโปรแกรมที่สร้างเงื่อนไข(Condition)
2. อธิบายความแตกต่างหรือสัมพันธ์การสร้างเงื่อนไขในภาษาจาวา
3. บอกขั้นตอนการทำงานของโปรแกรมเงื่อนไขในภาษาจาวาได
4. เขียนโปรแกรมภาษาจาวาตามรูปแบบภาษาได้

Java Condition Statement

Java Condition Statement เป็นรูปแบบคำสั่งในภาษา Java ที่สร้างเงื่อนไข Condition สายงานการตัดสินใจการทำงาน เมื่อเป็นจริง หรือจะทำงานอีกชุด เมื่อเงื่อนไขเป็นเท็จ หรือจะกำหนดทางเลือกหลาย ๆ ทางเลือกในการทำงาน โดยคำสั่งในกลุ่มนี้ จะเป็นพวก if.. , if...else , if...elseif...else ,Nested if...else และ switch

Java if

Java if เป็นคำสั่งในภาษา Java โดย if ใช้สร้างเงื่อนไข เมื่อเงื่อนไขเป็นจริง ค่อยทำงานใน block นั้น ๆ

Syntax

```
if(expression)
{
statement;
}
```

Example

```
package com.java.myapp;


public class MyClass {


    public static void main(String[] args) {


          int i = 10;


        if( i == 10 )
           System.out.println("true statement i = 10");




        if( i == 10 )
        {
          System.out.println("true statement i = 10");
           System.out.println("OK");
        }


    }


}
```

Output

```
true statement i = 10

true statement i = 10
OK
```

Java if..else

Java if..else เป็นคำสั่งในภาษา Java โดย if..else ใช้สร้างเงื่อนไข เมื่อเงื่อนไขเป็นจริงจะทำ block ที่เป็น true และเงื่อนไขเป็นเท็จจะทำ block ที่เป็น false

Syntax

```
if(expression)
{
true statement;
}
else
{
false statement;
}
```

Example 1

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

          int i = 5;

         if( i == 10 )
            System.out.println("true statement i = 5");


         if( i == 10 )
         {
          System.out.println("true statement i = 10");
         }
         else
         {
          System.out.println("false statement i <> 10");
         }

    }

}
```

Output

```
false statement i <> 10
```

Example 2

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        boolean condition = true;
        if (condition) {
             System.out.println("Condition is true.");
        }
        else {
             System.out.println("Condition is false.");
        }

    }

}
```

Output

```
Condition is true.
```

Java if...else if...else

Java if...else if...else เป็นคำสั่งในภาษา Java โดย if...else if...else ใช้สร้างเงื่อนไข โดยสามารถ else if กำหนดและสร้างทางเลือกได้หลายทาง และหลาย block โดยในแต่ละ block สามารถสร้างเงื่อนไขของตัวเองได้

Syntax

```
if(expression1)
{
  statement;
}
else if(expression2)
{
  statement;
}
else if(expression3)
{
  statement;
}
else
{
  statement;
}
```

Example 1

```
package com.java.myapp;

public class MyClass {
```

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

         int i = 2;

         if(i == 1){
                System.out.println("true statement i = 1");
         }else if(i == 2){
                System.out.println("true statement i = 2");
         }else if(i == 3){
                System.out.println("true statement i = 3");
         }else if(i == 4){
                System.out.println("true statement i = 4");
         }else{
                System.out.println("false statement i = " + i);
         }

    }

}
```

Output

```
true statement i = 2
```

Example 2

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        int testscore = 76;
        char grade;

        if (testscore >= 90) {
            grade = 'A';
        } else if (testscore >= 80) {
            grade = 'B';
        } else if (testscore >= 70) {
            grade = 'C';
        } else if (testscore >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }
        System.out.println("Grade = " + grade);

    }

}
```

Output

```
Grade = C
```

Java Nested if...else

Java Nested if...else เป็นคำสั่งในภาษา Java โดย Nested if...else ใช้สร้างเงื่อนไขซ้อนเงื่อนไข โดยเงื่อนไขทั้งสอง จะต้องเป็นจริง ถึงจะทำงานใน Statement ที่ต้องการ

Syntax

```
if(Boolean_expression 1){

   if(Boolean_expression 2){

      Statement;

   }

}
```

Example

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

           int i = 2;

           if (i > 0)       //Outer if
           {
             if ( i <= 5 )  // Inner if
               System.out.println( "i is between 0 and 5" ) ;
           }

           if (i > 0 && i <= 5)
           {
               System.out.println( "i is between 0 and 5" ) ;
           }

    }

}
```

Output

```
i is between 0 and 5
i is between 0 and 5
```

Java switch

Java switch เป็นคำสั่งในภาษา Java โดย switch ใช้สร้างเงื่อนไข และหลายทางเลือก สามารถใช้ได้เหมือนกับ if...elseif...elseif... แต่การเขียนสั้นกว่า และ สามารถเข้าใจได้ง่ายกกว่า

Syntax

```
switch(expression){
    case value :
       //Statements
       break; //optional
    case value :
       //Statements
       break; //optional
    default : //Optional
       //Statements
}
```

Example 1

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        int month = 8;
        String monthString;
        switch (month) {
            case 1:  monthString = "January";
                     break;
            case 2:  monthString = "February";
                     break;
            case 3:  monthString = "March";
                     break;
            case 4:  monthString = "April";
                     break;
            case 5:  monthString = "May";
                     break;
            case 6:  monthString = "June";
                     break;
            case 7:  monthString = "July";
                     break;
            case 8:  monthString = "August";
                     break;
            case 9:  monthString = "September";
                     break;
            case 10: monthString = "October";
                     break;
            case 11: monthString = "November";
                     break;
            case 12: monthString = "December";
                     break;
            default: monthString = "Invalid month";
                     break;
        }
        System.out.println("Month Name : " + monthString);

    }

}
```

Output

```
Month Name : August
```

Example 2

```
package com.java.myapp;

public class MyClass {

    public static void main(String[] args) {

        java.util.ArrayList<String> futureMonths =
                new java.util.ArrayList<String>();

            int month = 8;

            switch (month) {
                case 1:  futureMonths.add("January");
                case 2:  futureMonths.add("February");
                case 3:  futureMonths.add("March");
                case 4:  futureMonths.add("April");
                case 5:  futureMonths.add("May");
                case 6:  futureMonths.add("June");
                case 7:  futureMonths.add("July");
                case 8:  futureMonths.add("August");
                case 9:  futureMonths.add("September");
                case 10: futureMonths.add("October");
                case 11: futureMonths.add("November");
                case 12: futureMonths.add("December");
                         break;
                default: break;
            }

            if (futureMonths.isEmpty()) {
                System.out.println("Invalid month number");
            } else {
                for (String monthName : futureMonths) {
                   System.out.println(monthName);
                }
            }

    }

}
```

Output

```
August
September
October
November
December
```

Example 3

```
package com.java.myapp;

public class MyClass {

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

```
        String month = "August";

        int returnedMonthNumber = getMonthNumber(month);

        if (returnedMonthNumber == 0) {
            System.out.println("Invalid month");
        } else {
            System.out.println(returnedMonthNumber);
        }

    }

    public static int getMonthNumber(String month) {

        int monthNumber = 0;

        if (month == null) {
            return monthNumber;
        }

        switch (month.toLowerCase()) {
            case "january":
                monthNumber = 1;
                break;
            case "february":
                monthNumber = 2;
                break;
            case "march":
                monthNumber = 3;
                break;
            case "april":
                monthNumber = 4;
                break;
            case "may":
                monthNumber = 5;
                break;
            case "june":
                monthNumber = 6;
                break;
            case "july":
                monthNumber = 7;
                break;
            case "august":
                monthNumber = 8;
                break;
            case "september":
                monthNumber = 9;
                break;
            case "october":
                monthNumber = 10;
                break;
            case "november":
                monthNumber = 11;
                break;
            case "december":
                monthNumber = 12;
                break;
            default:
                monthNumber = 0;
                break;
```

```
            }
            return monthNumber;
        }
    }
```

Output

```
8
```

คำถามท้ายบท

1. จะอธิบายประโยชน์ของคำสั่งเงื่อนไขอย่างสรุป
2. คำสั่งเงื่อนไขกี่ประเภท อะไรบ้าง
3. จงยกตัวอย่างคำสั่งเงื่อนไขที่สามารถใช้ในการคำนวณทางคณิตศาสตร์
4. จงยกตัวอย่างประกอบคำสั่งเงื่อนไข 3 ตัวอย่างในรูปภาษา Java programming
5. จงอธิบายข้อแตกต่างคำสั้งเงื่อนไขพอสังเขปและยกตัวอย่างประกอบ

บทที่ 6 Java Class and Method

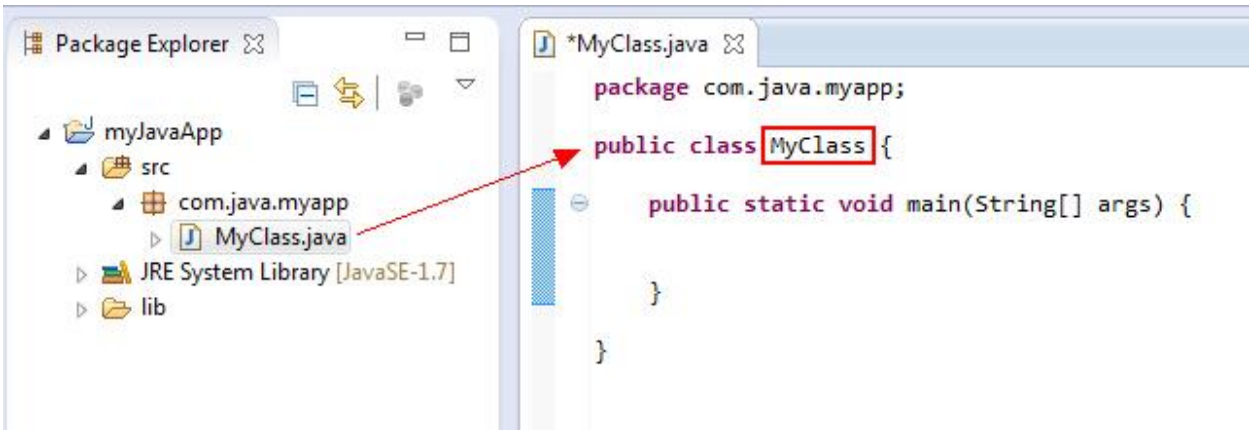
วัตถุประสงค์ของหน่วยการเรียนรู้

1. เกิดทักษะการเขียนโปรแกรม Java ในรูปแบบ Class และ Method
2. อธิบายความแตกต่างหรือสัมพันธ์ระหว่างโปรแกรมภาษาจาวามีกับภาษาแบบโครงสร้างได
3. บอกขั้นตอนการทำงานของโปรแกรมภาษาจาวาได
4. เขียนโปรแกรมภาษาจาวาตามรูปแบบภาษาได้

Java Class and Method

Java Class and Method พื้นฐานคลาสและเมธอด ในหัวข้อนี้ตั้งอธิบายความรู้สั้น ๆ เกี่ยวกับ Class และ Method ในภาษา Java ว่ามันมีรูปแบบการเขียนและเรียกใช้งานอย่างไร และเพราะโครงสร้าง Class ในภาษา Java และสำหรับพื้นฐานของการเขียน Class จะอธิบายหลักการความเข้าใจในระดับหนึ่งที่สามารถนำไปเขียนโปรแกรมและศึกษาต่อยอดในอนาคตได้ ซึ่งในบทนี้จะอธิบายขั้นตอนสั้น ๆ ดังนี้

และตามที่เราเข้าในแล้วว่าพื้นฐานการเขียนโปรแกรมด้วยภาษา Java จะเป็น OOP ค่อนข้างสมบูรณ์ นั้นหมายถึงตั้งแต่ขั้นตอนการสร้าง Project และ Class ใน Project ก็จะมีการสร้างค่า Default ที่อยู่ในรูปแบบของ OOP ให้ทันที ก่อนอื่นขอย้อนไปถึงหลักการตั้งชื่อ Class ก่อนว่า ธรรมเนียมการตั้งชื่อ Class จะขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่ และ ชื่อ Class กับชื่อไฟล์จะต้องเป็นชื่อเดียวกันเสมอ เช่น



จากภาพจะเห็นว่าเรามี Class ชื่อว่า MyClass โดยไฟล์ของ Class ชื่อว่า MyClass.java และอยู่ภายใต้ Package ของ com.java.myapp สำหรับใครยังไม่รู้และเข้าใจความหมายของ Package แนะนำให้อ่านบทความนี้ก่อน และจากตัวอย่างเราจะเห็นว่ามี Method อยู่ 1 ตัวชื่อว่า main (หรือในภาษาอื่น ๆ เราจะเรียกว่า Function ภายใน Class)

```
package com.java.myapp;
public class MyClass {
  public static void main(String[] args) {

    }
}
```

สำหรับ Method ชื่อว่า main นี้เป็น Method แรกสุดจะทำงานหลังจากที่ Class นี้เริ่มการทำงาน เช่น

```
package com.java.myapp;
public class MyClass {
  public static void main(String[] args) {
          System.out.println("Welcome to ThaiCreate.Com");
    }
}
```

Output

```
Welcome to ThaiCreation.Com
```

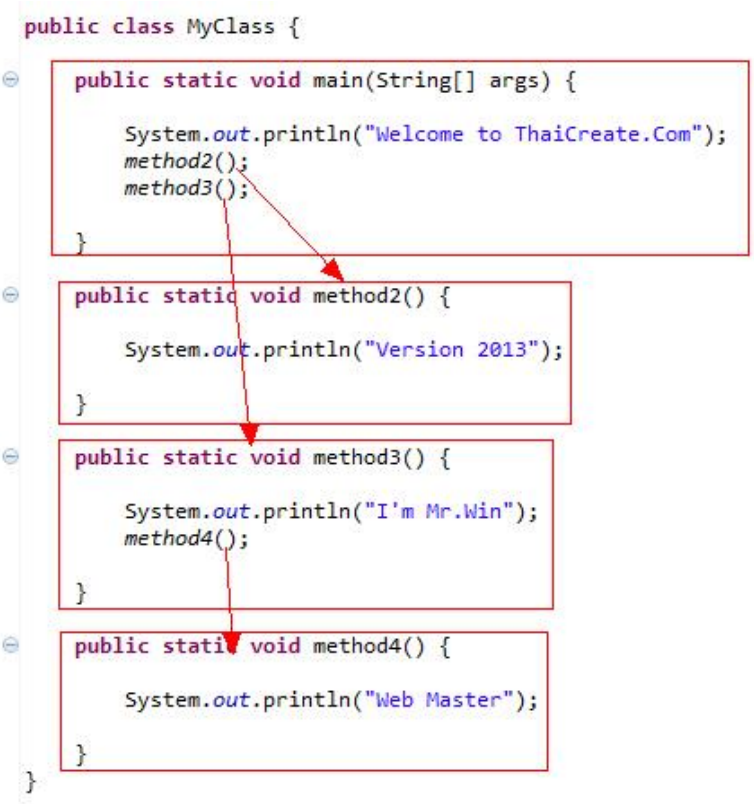
จากตัวอย่าง Class ของ MyClass จะทำงานที่ Method ของ main และแสดงผลลัพธ์ดังรูป และภายใน Class หนึ่ง ๆ นั้น ก็สามารถมีได้หลาย Method และแต่ล่ะ Method ก็สามารถเรียกใช้งานซึ่งกันและกันได้ โดยถ้า Method อยู่ใน Class เดียวกัน สามารถเรียกใช้ได้ทันที โดยที่ไม่ต้องทำการ new

Class and Multiple Method ภายใน Class ประกอบด้วยหลาย ๆ Method และภายไต้ Method ก็สามารถเรียกใช้งานซึ่งกันและกันได้

```
package com.java.myapp;
public class MyClass {
	public static void main(String[] args) {
		System.out.println("Welcome to ThaiCreate.Com");
		method2();
		method3();
    }



	public static void method2() {
		System.out.println("Version 2013");
    }
	public static void method3() {
		System.out.println("I'm Mr.Win");
		method4();
```

```
    }

        public static void method4() {

            System.out.println("Web Master");

    }

}
```

ภาพคำอธิบาย



รูปอธิบาย Class ที่ประกอยด้วยหลาย ๆ Method และการเรียกใช้งานระหว่าง Method

Output

```
Welcome to ThaiCreate.Com
Version 2013
I'm Mr.Win
Web Master
```

Class Method and Parameter/Argument นอกจากนี้ภายใน Method ก็ยังสามารถเปิดรับค่า Parameter ได้เหมือนกับการเขียน Function ในภาษาอื่นทั่ว ๆ ไป เช่น

```
package com.java.myapp;

public class MyClass {

  public static void main(String[] args) {

          System.out.println("Hello ");

          String name = "Weerachai";
          String surname = "Nukitram";
          displayName(name,surname);

     }

  public static void displayName(String strName,String
strSurname) {

          System.out.println(strName + " " + strSurname);

     }
}
```

ตัวอย่างการสร้าง Method และการกำหนด Parameters รวมทั้งการส่งค่า Parameters

Output

```
Hello
Weerachai Nukitram
```

Class and Variable Scope สำหรับหลักการประกาศตัวแปร คือตัวแปรที่ประกาศระดับ Class จะสามารถใช้ได้ทุก Method และตัวแปรที่ประกาศภายใน Method จะสามารถใช้ได้เฉพาะภายใน Method เท่านั้น

```
package com.java.myapp;

public class MyClass {

  static String name = "Weerachai";  // ใช้ได้ทุกส่วนของ Class
  static String surname = "Nukitram";  // ใช้ได้ทุกส่วนของ Class

  public static void main(String[] args) {

          String address = "Bangkok";  // ใช้ได้เฉพาะใน Class main

          System.out.println("Hello " + name + " " + surname + "
" + address);

     }

  public static void displayName(String strName,String
strSurname) {

          System.out.println(name + " " + surname);
```

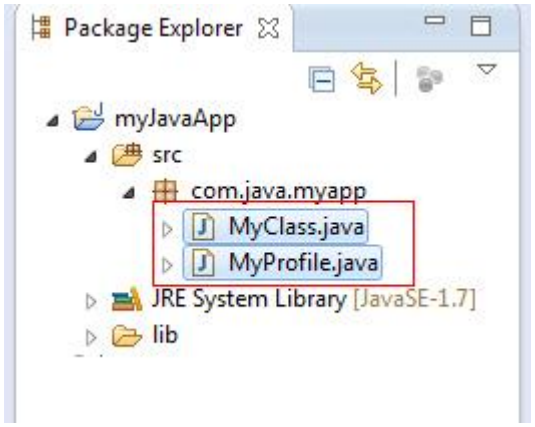
```
        }
    }
```

สำหรับพื้นฐานของ Class ก็มีง่าย ๆ เพียงเท่านี้ และในบทความถัดไป เราจะมารู้เกี่ยวกับ Constructor รวมทั้ง Multi-Class และการเรียกใช้งานระหว่าง Class

Java Class and Multi-Class

Java Class and Multi-Class สำหรับ Multi-Class คือการสร้าง Class หลาย ๆ Class ไว้สำหรับแยกการทำงานของโปรแกรมให้เป็นสัดส่วน ซึ่งปกติการเขียนโปรแกรมทุกภาษา เมื่อโปรแกรมใหญ่ขึ้น จำนวน Class ก็จะเยอะขึ้นตามไปด้วย และการออกแบบ Class ก็ไม่มีรูปแบบที่ตายตัว ขึ้นอยู่กับความสามารถของโปรแกรมเมอร์ในการออกแบบและจัดระเบียบของ Class รวมทั้งแนวคิดการออกแบบในรูปแบบของ Framework ต่าง ๆ

โดยพื้นฐานทั่วไปแล้ว Class ในภาษา Java แต่ล่ะ Class จะแยกไฟล์ออกเป็นของใครของมัน และแต่ล่ะ Class ก็สามารถที่จะอยู่ภายใต้ Package ต่าง ๆ แยกย่อยไปได้อีก ซึ่ง Class ที่อยู่ใน Package เดียวกันก็จะสามารถมองเห็นและเรียกใช้ Instance (เรียกใช้งาน) กันได้ทันที ส่วน Class ที่อยู่คนล่ะ Package ก็จะต้องทำการ อ้างถึงชื่อ Package ด้วย หรือจะใช้การ Import ตัว Package นั้นเข้ามา ก็สามารถเรียกใช้งาน Class ภายใต่ Package นั้น ๆ ทั้งหมดได้ ลองอ่านบทความนี้เพิ่มเติม



ตัวอย่างการสร้าง Class ในตัวอย่างนี้ประกอบด้วย Class ชื่อว่า MyClass และ MyProfile

MyClass.java

```
package com.java.myapp;

public class MyClass {

  public static void main(String[] args) {

          MyProfile profile = new MyProfile(); // ทำการเรียกใช้งาน Class
ของ MyProfile โดยได้ Instance ใหม่ชื่อว่า profile
          profile.displayName(); // ทำการเรียกใช้งาน Method ของ displayName
          profile.displaySurname();  // ทำการเรียกใช้งาน Method ของ
displaySurname
     }

}
```

MyProfile.java

```
package com.java.myapp;

public class MyProfile {

  public void displayName() {

          System.out.println("Weerachai");

     }

  public void displaySurname() {

          System.out.println("Nukitram");

     }

}
```

Output

```
Weerachai
Nukitram
```

Java Class and Constructor

Java Class and Constructor สำหรับ Constructor เป็น Method ที่มีชื่อเดียวกับ Class โดยจะใช้กำหนดค่าเริ่มต้นให้กับ Class ที่เป็น Instance ที่ผ่านการ new ของ Class ปลายทาง โดย Constructor จะถูกเรียกใช้งานอัตโนมัติ และเราจะต้องทำการ ใส่ค่า Parameters ตามจำนวน Argument ของ Constructor Class นั้น ๆ

MyClass.java

```
package com.java.myapp;

public class MyClass {

  public static void main(String[] args) {

          String name = "Weerachai";
          String surname = "Nukitram";

          MyProfile profile = new MyProfile(name,surname); //
Instance ใช้งาน MyProfile  และ กำหนดค่าเริ่มต้น
          profile.displayFullname(); // เรียกใช้งาน Method
displayFullname
     }

}
```

MyProfile.java

```
package com.java.myapp;

public class MyProfile {

  String strName = null;
  String strSurname = null;

  public MyProfile(String name, String surname) {
          strName = name;  // กำหนดค่าให้ตัวแปร strName
          strSurname = surname; // กำหนดค่าให้ตัวแปร strSurname
  }

  public void displayFullname() {

          System.out.println(strName + " " + strSurname); // แสดงค่า
strName และ strSurname

     }

}
```

Output

```
Weerachai Nukitram
```

Java Class and extends Class (Inheritant)

Java extends Class สำหรับการ extends Class (Inheritant) ในภาษา Java จะใช้ในกรณีที่ต้องการขยายขอบเขต (คล้าย ๆ กับการขยายอำนาจการเรียกใช้ ทันทีโดยไม่ต้อง new ตัว Instance) การทำงานของ Class ปัจจุบันไปยัง Class ปลายทางที่ถูก extends เช่น

```
public class MyClass extends MemberProfile
```

ซึ่งต่อไปนี้เราจะเรียก MemberProfile ว่า Superclass และ MyClass เรียกว่า Subclass นั่นหมายถึง MyClass สามารถเรียกใช้งาน Method รวมทั้งพวกตัวแปร ต่าง ๆ ที่อยู่ภายใน Class ของ MemberProfile ได้ทันที ลองมาดูตัวอย่าง

MemberProfile.java

```
package com.java.myapp;

public class MemberProfile {

  public static String address = "Bangkok Thailand";

  public static void displayFullname() {

          System.out.println("Weerachai Nukitram");

     }

}
```

MyClass.java

```
package com.java.myapp;

public class MyClass extends MemberProfile {

  public static void main(String[] args) {

          displayFullname();

          System.out.println(address);

     }

}
```

Output

```
Weerachai Nukitram
Bangkok Thailand
```

Java Override Method

Java Override Method ในการเขียนโปรแกรมภาษา Java บนพวก Android เราอาจจะพบเห็นพวก Method ที่ชื่อว่า Override Method สำหรับคำว่า Override ก็คือการเขียนทับ หรือประกาศใช้งาน Method ทับกับตัวที่มีอยู่แล้ว โดยตัวที่ที่อยู่แล้ว อาจจะเป็น Class ที่อยู่ใน Superclass ที่ผ่านการ extends (Inheritant) ซึ่งเมื่อมีการ extends หรือ inheritant หลาย ๆ ขั้น ตัวโปรแกรมจะเลือกใช้ Method ที่อยู่ใน Class ที่ถูกเรียกใช้ล่าสุด

MemberProfile.java (Superclass)

```
package com.java.myapp;

public class MemberProfile {

  public void displayFullname() {

          System.out.println("Surachai Sirisart");

     }

}
```

MyProfile.java (Subclass)

```
package com.java.myapp;

public class MyProfile extends MemberProfile {

  //Override displayFullname
  public void displayFullname() {

          System.out.println("Weerachai Nukitram");

     }

}
```

MyClass.java

```
package com.java.myapp;

public class MyClass {

  public static void main(String[] args) {

          MyProfile profile =  new MyProfile();
          profile.displayFullname();

     }

}
```

Output

```
Weerachai Nukitram
```

ตัวอย่างการสร้าง Class ที่เป็นแบบ Override Method คือ Subclass ชื่อเดียวกับ Superclass โดยโปรแกรมจะเลือกทำงาน Method ที่ถูกอ้างถึงล่าสุดคือ Method ที่อยู่ภายใน MyProfile

Java static method

Java static method สำหรับ static ในภาษา Java จะใช้ประกาศข้างหน้า Method หรือ Variable โดยเมื่อ Method หรือ Variable ที่ประกาศนำหน้าด้วย static แล้ว Method นั้น ๆ จะถูก

เรียกใช้งานด้วย Class อื่น ๆ โดยไม่ต้องผ่านการ new หรือประกาศ Instance (แต่ต้องดูพวก public / private / protected ด้วย)

การประกาศ static รวมถึงขอบเขตของ Method ภายใน Class เดียวกันด้วย คือถ้าไม่ประกาศ static จะไม่สามารถเรียกชื่อตรง ๆ ได้

MyProfile.java

```
package com.java.myapp;

public class MyProfile {

  static String address = "Bangkok Thailand";

  public static void displayFullname1() {
         System.out.println("Weerachai Nukitram");
    }

}
```

MyClass.java

```
package com.java.myapp;

public class MyClass {

  public static void main(String[] args) {
         MyProfile.displayFullname1();
         System.out.println(MyProfile.address);
    }

}
```

Output

```
Weerachai Nukitram
Bangkok Thailand
```

จากตัวอย่างจเห็นว่า MyClass สามารถเรียก Method และ Variable ที่อยู่ใน MyProfile โดยไม่ต้องประกาศ new หรือ Instance

Java void , public , private , protected

Java void , public , private , protected หัวข้อสุดท้ายในส่วนของ Class จะพื้นความจำเกี่ยวกับ void , public , private protected ว่าแต่ล่ะตัวมันคืออะไร และ ใช้งานอย่างไร โดยคำพวกนี้เราจะพบเจอมันในทุก ๆ ส่วนของโปรแกรม อาทิเช่น การประกาศตัวแปร Variable การประกาศ Class หรือการประกาศ Method ซึ่งโดยสรุปแล้ว

- void ใช้ประกาศ Method ที่ไม่ต้องการ return ค่ากลับ
- public ใช้ประกาศ Variable , Class และ Method ให้เป็นแบบ public คือสามารถใช้กับ Instance อื่น ๆ ได้
- private ใช้ประกาศ Variable , Method ที่สามารถเรียกใช้ได้เฉพาะใน Class นั้น ๆ
- protected ใช้ประกาศ Variable , Method ที่สามารถเรียกใช้ได้ใน Class นั้น ๆ และก็ Class ที่ extends (Inheritance)

Example

MyProfile.java

```
package com.java.myapp;

public class MyProfile {

  public void displayFullname1() {
         System.out.println("Weerachai Nukitram");
         displayFullname2();
    }

  private void displayFullname2() {
         System.out.println("Surachai Sirisart");
         displayFullname3();
    }

  protected void displayFullname3() {
         System.out.println("Adisorn Bunsong");
    }

}
```

MyClass.java

```
package com.java.myapp;

public class MyClass {

  public static void main(String[] args) {

         MyProfile profile = new MyProfile();
         profile.displayFullname1();

    }

}
```

Output

```
Weerachai Nukitram
Surachai Sirisart
Adisorn Bunsong
```

คำถามท้ายบท

1. จะอธิบายประโยชน์ของClass และ Method อย่างสรุป

2. Class และ Method ต่างจากการออกแบบ ฟังค์ชันที่เป็นTop down อย่างไร

3. จงยกตัวอย่างการใช้ Class และ Method ที่สามารถใช้ในการคำนวณทางคณิตศาสตร์

4. Class และ Method ในภาษา Java programming มีลักษณะเด่นอะไรบ้าง

5. จงอธิบายข้อแตกต่างการใช้คำสั่ง Class และ Functionและยกตัวอย่างประกอบ

เอกสารอ้างอิง

ทศพล ธนะทิพานนท์, วรเศรษฐ สวุ รรณกิ . เขียนโปรแกรม Java เบื้องต้น 2nd Edition. ซีเอด็ ยูเค ชัน, บมจ.

การเขียนโปรแกรม JAVA. [ระบบออนไลน์] . แหลง่ ที่มา http://www.thaicreate.com/java.html

สุรางคนา ระวังยศ. (2555). การเขียนโปรแกรมเชิงวัตถุ. คนเมื่อ 23 สิงหาคม 2555, จาก http://www.kmitl.ac.th/~s3010819/MyFile/My%20Ebook/JAVA/%BA%B7%B7%D5%E8%201%20%20%BA%B7%B9%D3.pdf

About the java technology. (2012). Retrieved August 28, 2012, from:http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

Anban Pillay (2007). Object Oriented Programming using Java. University of KwaZulu

บัญชา ปะสีละเตสัง(2561) การเขียนโปรแกรมด้วย Java สำหรับผู้เริ่มต้น. ซีเอด็ ยูเคชัน, บมจ.

อนรรฆนงค์ คุณมณี(2554), คู่มือเรียนเขียนโปรแกรมภาษา Java ฉบับสมบูรณ์, Dev Book.