

การเขียนโปรแกรมภาษา

JAVA

เรียบเรียงโดย นายจิราวัฒน์ นาม

ธันวาคม 2557



เอกสารประกอบการเรียน วิชาเทคโนโลยีสารสนเทศ 9 (ว33102)
โรงเรียนปทุมเทพวิทยาคาร

เนื้อหาจากหนังสือเล่มนี้เรียบเรียงจาก
เอกสารการเรียนการสอน วิชา ว30201 การเขียนโปรแกรมเชิงวัตถุ
โรงเรียนมหิดลวิทยานุสรณ์

ใช้สำหรับการวิจัยเท่านั้น



PTKSchool

โรงเรียนปทุมเทพวิทยาคาร

1

รู้จักกับภาษาจาวา (JAVA)

ความคิดรวบยอด

 Video USSEาย

Video 0 – Introduction

Video 1 – รู้จักกับภาษาจาวา

1

ประวัติความเป็นมา

ภาษาจาวาถือกำเนิดขึ้นจากความต้องการที่จะสร้างคอมพิวเตอร์ภาษาคอมไพเตอร์ซึ่งเป็นอิสระจาก Hardware รุ่นใดรุ่นหนึ่ง โดยการนำของ James Gosling แห่งบริษัท Sun Microsystems

2

เริ่มต้นเขียนภาษาจาวา

Netbeans เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมคอมพิวเตอร์ ซึ่งสนับสนุนภาษาคอมไพเตอร์หลายภาษา รวมทั้ง JAVA ในบทนี้จะกล่าวถึงการใช้ Netbeans สำหรับการพัฒนาโปรแกรมภาษา JAVA



goo.gl/Q5ktXo



ประวัติความเป็นมา

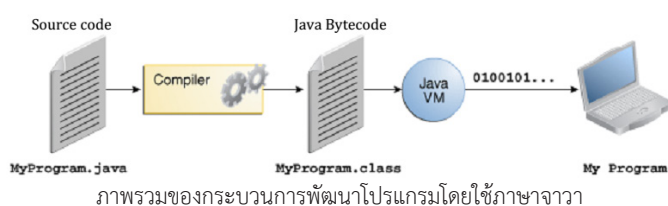
ภาษาจาวาถือกำเนิดขึ้นจากความต้องการที่จะสร้างคอมไพเลอร์ภาษาคอมพิวเตอร์ซึ่งเป็นอิสระจาก Hardware รุ่นใดรุ่นหนึ่ง หรือยี่ห้อใดยี่ห้อหนึ่ง โดยจุดประสงค์แรกนั้นต้องการที่จะใช้เขียนโปรแกรมเพื่อควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าและอุปกรณ์อิเล็กทรอนิกส์ต่างๆ เช่น เตารีด เครื่องซักผ้า โทรศัพท์มือถือ Set Top Box ของเคเบิลทีวี ฯลฯ ด้วยเหตุนี้ โดยการนำของ James Gosling หัวหน้ากลุ่ม Green Group แห่งบริษัท Sun Microsystems จึงได้เริ่มโครงการพัฒนาภาษาดังกล่าวอย่างจริงจังในปี 1991 โดยขั้นแรกชื่อว่า ภาษา Oak แต่หลังจากที่ไม่ประสบความสำเร็จในการนำไปใช้งานตามความคิดริเริ่มดังกล่าว ประกอบกับบริษัท Sun Microsystems เริ่มมองเห็นความจำเป็นที่ต้องมีภาษาที่สร้างโปรแกรมบนเครื่องหนึ่งแต่สามารถนำไปใช้งานบนเครื่องใดๆ ก็ได้ (Write Once Run Anywhere) บริษัทจึงได้นำเอา ภาษา Oak มาพัฒนาต่อให้เป็นภาษาเชิงวัตถุ (Object Oriented Language) จนได้ภาษาจาวาขึ้นในปี 1995



James Gosling
ที่มา: Wikipedia.org

การทำงานของภาษาจาวา

การพัฒนาโปรแกรมด้วยภาษาจาวาจะเริ่มจากการเขียนโค้ดโปรแกรมจนได้ไฟล์ Source code ที่มีนามสกุลเป็น .java ซึ่งก็คือไฟล์โค้ดที่เราเขียนขึ้น โดยไฟล์ Source code นี้จะถูกแปล (compile) ด้วย Javac compiler ได้เป็น Java Bytecode ซึ่งจะอยู่ในรูปของไฟล์ที่มีนามสกุลเป็น .class



เมื่อโปรแกรมถูกเรียกใช้งานบนคอมพิวเตอร์หรืออุปกรณ์ใดๆ ก็ตาม Java Bytecode นี้จะถูกแปลด้วย Interpreter ซึ่งในที่นี้คือ Java Virtual Machine หรือ JVM อีกครั้งหนึ่ง เพื่อให้เป็นภาษาเครื่องเฉพาะอุปกรณ์ชนิดนั้นๆ เพื่อให้โปรแกรมสามารถทำงานบนอุปกรณ์นั้นๆ ได้

เนื่องจาก Java VM สามารถทำงานบนระบบปฏิบัติการต่างๆ ได้ ทำให้ไฟล์นามสกุล .class (Java Bytecode) ไฟล์เดียวกันสามารถทำงานบนระบบปฏิบัติการที่หลากหลาย เช่น ระบบปฏิบัติการ Microsoft Windows, Solaris, Linux หรือ Mac OS เป็นต้น ทำให้โปรแกรมที่พัฒนาด้วยภาษาจาวานั้นจะไม่ยึดติดกับแพลตฟอร์ม

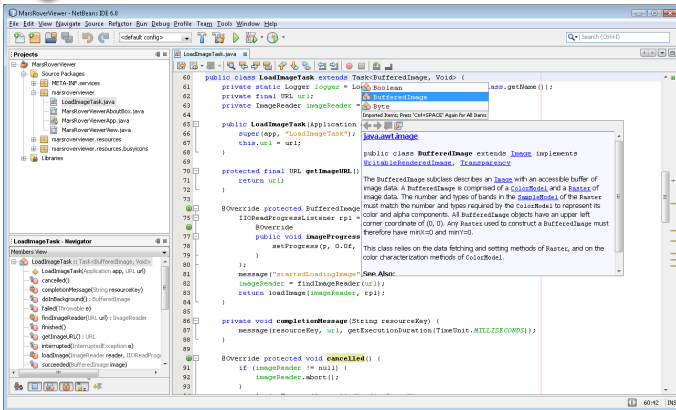
? แพลตฟอร์มคืออะไร

แพลตฟอร์ม (Platform) คือสภาวะแวดล้อมที่ประกอบด้วยฮาร์ดแวร์และซอฟต์แวร์ของระบบคอมพิวเตอร์ระบบหนึ่ง ซึ่งคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการต่างกัน ก็จะมี Platform ที่ต่างกันไปด้วย

ใดๆ (Platform Independent) ทั้งนี้คอมพิวเตอร์หรืออุปกรณ์ที่จะรันโปรแกรมที่เขียนด้วยภาษาจาวา จะต้องติดตั้ง Java Runtime Environment หรือ JRE ก่อนเสมอ ซึ่งภายใน JRE นั้นจะมี Java Virtual Machine อยู่ด้วย

เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาโปรแกรมด้วยภาษาจาวาจะมีเครื่องมือที่รองรับการพัฒนามากมายหลายตัว ซึ่งเครื่องมือแต่ละตัวก็จะมีคุณลักษณะที่แตกต่างกันไป

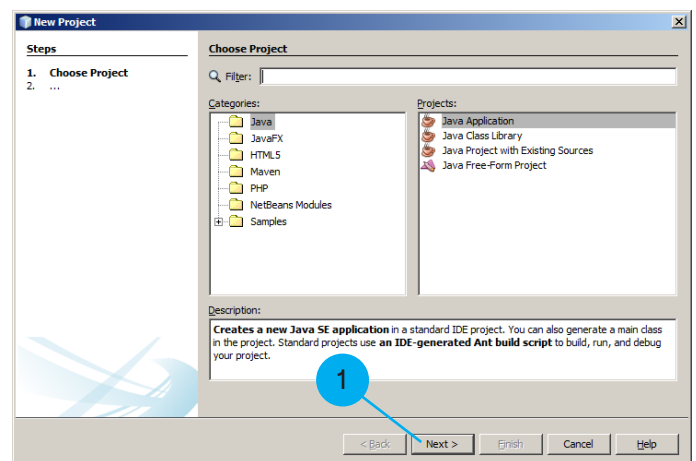


หน้าต่างโปรแกรม Netbeans

NetBeans เป็นเครื่องมือที่ใช้สำหรับพัฒนาแอปพลิเคชันด้วยภาษาจาวา ได้รับการสนับสนุนจากบริษัท Sun Microsystems ซึ่งเป็นซอฟต์แวร์ที่สามารถใช้งานได้โดยไม่มีค่าใช้จ่ายใดๆ

เริ่มต้นเขียนโปรแกรมภาษาจาวา

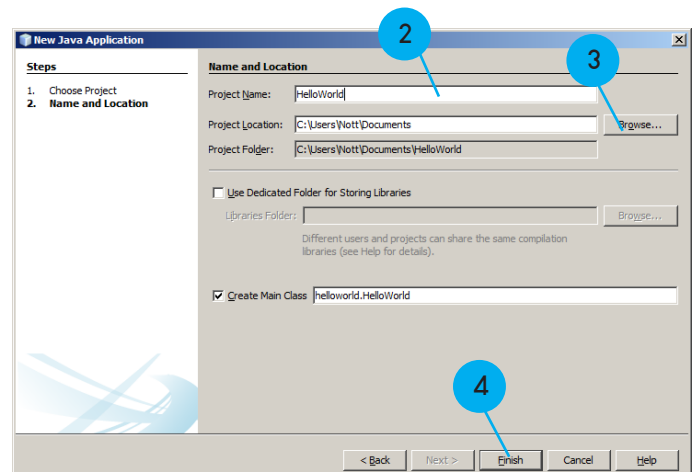
- 1 เริ่มต้นสร้างโปรเจกต์โดยการกดเมนู File > New Project ที่ Category เลือกเป็น Java และ Project เลือกเป็น Java Application จากนั้นกด Next



- 2 ตั้งชื่อโปรเจกต์ โดยจะต้องเป็นภาษาอังกฤษ และไม่มีช่องว่าง

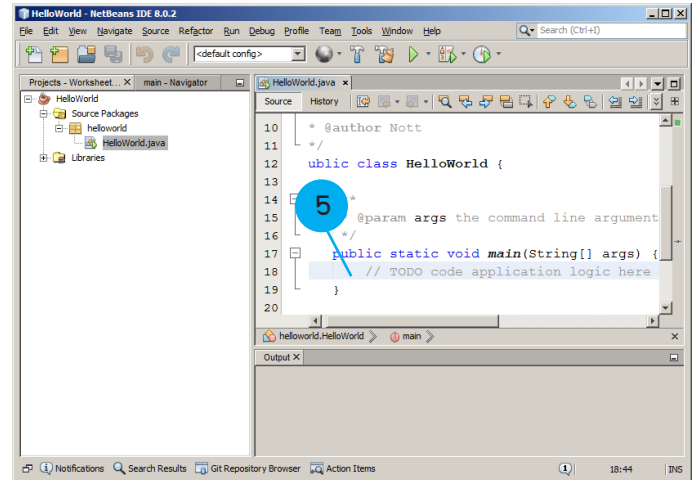
- 3 เลือกที่เก็บโปรเจกต์ โดยการกด Browse ชื่อโฟลเดอร์เก็บงานจะต้องเป็นภาษาอังกฤษ

- 4 กด Finish เพื่อเสร็จสิ้นขั้นตอน





5 เริ่มเขียนโค้ดภายในเมธอด main()



โปรแกรมแรก – Hello Word

ภาษาจาวามีรูปแบบการเขียนโปรแกรมที่ประกอบด้วย คลาสอย่างน้อยหนึ่งคลาสที่มีเมธอด main() ซึ่งเปรียบเสมือนจุดเริ่มต้นของโปรแกรม โดยมีการเขียนโปรแกรมดังนี้

```
// HelloWorld.java
1 public class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello World");
4     }
5 }
```

- บรรทัดที่ 1 เป็นการประกาศคลาสขึ้นในชื่อ HelloWorld ซึ่งขอให้สังเกตว่าชื่อไฟล์จะต้องเป็นชื่อเดียวกันกับชื่อคลาส นั่นคือต้องบันทึกไฟล์ชื่อ HelloWorld.java

- มีวงเล็บปีกกาเปิดในบรรทัดที่ 1 หมายถึง จุดเริ่มต้นคลาส HelloWorld และวงเล็บปีกกาปิดในบรรทัดที่ 5 หมายถึง จุดสิ้นสุดคลาส HelloWorld

- คลาส HelloWorld มีเมธอด main() ซึ่งเมื่อโปรแกรมประมวลผลก็จะทำการค้นหาเมธอดชื่อ main() เพื่อเริ่มทำงานตามชุดคำสั่งที่เขียนไว้ในเครื่องหมายวงเล็บปีกกา { } ของเมธอด main()

- เมธอด main() จะต้องมียุคิ่วรีด public static void รวมไปถึงพารามิเตอร์ที่เป็นตัวแปรชุดของ

ข้อมูลประเภท String

- ชุดคำสั่งต่างๆที่ต้องการให้ทำงาน จะต้องเขียนไว้ภายในเครื่องหมายวงเล็บปีกกา { } ของเมธอด main()

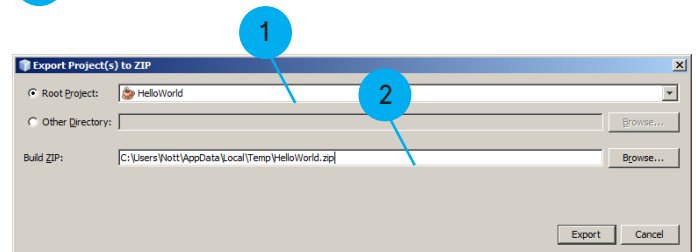
- บรรทัดที่ 3 เป็นคำสั่งแสดงข้อความ “Hello World” สังเกตว่าข้อความนั้นจะอยู่ในเครื่องหมาย “ ” เสมอ

- ทุกคำสั่งต้องปิดท้ายหรือจบด้วยเครื่องหมาย semicolon หรือ ; เสมอ

การส่งออกงาน (Export)

1 เลือกโปรเจกต์ที่ต้องการส่งออก (กรณีเปิดหลายโปรเจกต์พร้อมกัน)

2 เลือกที่เก็บไฟล์ พิมพ์นามสกุลต่อท้ายด้วย .zip



การส่งงาน

ที่ตำแหน่งเก็บไฟล์ จะพบไฟล์นามสกุล .zip ให้นำไฟล์นั้นแนบเข้ากับระบบส่งงาน



2

การแสดงผลและรับข้อมูล

ความคิดรวบยอด

 Video USSยาย

Video 2 - การรับและแสดงผลข้อมูล

1

การแสดงผล

ภาษาจาวามีคำสั่งที่ใช้พิมพ์ข้อความออกทางหน้าจอ คือ
`System.out.print` และ `System.out.println`



goo.gl/Q5ktXo

2

การรับข้อมูล

ภาษาจาวามีคลาสที่ใช้รับข้อมูลเข้าทางคีย์บอร์ด คือ คลาส `Scanner`
การใช้งานนักเรียนควรที่จะรู้วิธีการเขียนโปรแกรมเชิงวัตถุ
(Object-oriented Programming) ด้วย



Class Library คือแหล่งรวบรวมเมธอดหรือโปรแกรมต่างๆ ของ Java ที่เขียนไว้แล้วและสามารถเรียกใช้ได้ทันที เรียกอีกอย่างว่า Java package โดยต้องใช้คำสั่ง import ตามด้วยชื่อ package และ class ตัวอย่างเช่น import java.io.*; ซึ่งเป็นการเรียกใช้เมธอดต่างๆ ที่อยู่ใน Java package ที่ชื่อว่า java.io และ * คือนำเข้าเมธอดมาใช้ได้ทุก class ที่อยู่ใน java.io

java.lang เป็น package พื้นฐานของ Java ที่รวบรวม class ที่สำคัญและเรียกใช้ได้ทันทีโดยไม่ต้อง import เพราะคอมไพเลอร์จะรวบรวม class ที่อยู่ใน java.lang ให้เองอัตโนมัติ ซึ่งเมธอดที่ใช้ในการแสดงผลข้อมูลออกทางจอภาพที่จะกล่าวถึงในบทนี้ก็อยู่ใน java.lang ด้วยเช่นเดียวกัน

การแสดงผลข้อมูล

การแสดงผลข้อมูลในภาษาจาวามีอยู่ด้วยกันหลายวิธี ในที่นี้จะกล่าวถึงเมธอดที่ใช้ในการแสดงผลข้อมูลออกทางจอภาพซึ่งอยู่ใน java.lang package ได้แก่

คำสั่ง System.out.print()

เป็นการเรียกใช้เมธอด print() ในคลาส System เพื่อแสดงข้อความทางจอภาพในบรรทัดที่เคอร์เซอร์ (cursor) อยู่โดยไม่มี การขึ้นบรรทัดใหม่ มีรูปแบบการใช้งาน ดังนี้

```
System.out.print(argument_1 + argument_2 +
... + argument_n);
```

โดยที่ argument_1, argument_2 และ argument_n คือ ข้อมูลที่ต้องการแสดงผล ซึ่งอาจเป็นข้อความ ค่าคงที่ ตัวแปร หรือนิพจน์ก็ได้

ลองกันเลย!

```
//ShowText1.java
1 public class ShowText1 {
2     public static void main(String[] arg) {
3         System.out.print("I am print");
4         System.out.print(" command");
5         System.out.println("I am println command");
6     }
7 }
8 }
```

? OOP คืออะไร

OOP ย่อมาจาก Object-Oriented Programming คือรูปแบบการเขียนโปรแกรมที่ไม่ต้องสร้างโปรแกรมขึ้นมาเองทั้งหมด เพราะมีการสร้างสิ่งเหล่านั้นขึ้นมาก่อนแล้วนั่นเอง หรือเรียกได้ว่า OOP สามารถใช้ซ้ำได้ โดยเราจะมองสิ่งที่ถูกสร้างไว้แล้วเป็นวัตถุ แต่เราต้องรู้จักว่าวัตถุนั้นคืออะไร มีหน้าที่อะไร นักเรียนจะได้ยินคำว่า Class, Object หรือ Method บ่อยๆ หากพูดถึงการเขียนโปรแกรมแบบ OOP

คำสั่ง System.out.println()

เป็นการเรียกใช้เมธอด println() ในคลาส System เพื่อแสดงข้อความทางจอภาพแล้วขึ้นบรรทัดใหม่ มีรูปแบบการใช้งาน ดังนี้

```
System.out.println(argument_1 + argument_2
+ ... + argument_n);
```

โดยที่ argument_1, argument_2 และ argument_n คือ ข้อมูลที่ต้องการแสดงผล ซึ่งอาจเป็นข้อความ ค่าคงที่ ตัวแปร หรือนิพจน์ก็ได้

i คำสั่ง print จะพิมพ์ข้อความโดยไม่เริ่มบรรทัดใหม่ ส่วนคำสั่ง println จะพิมพ์ข้อความแล้วเริ่มบรรทัดใหม่อัตโนมัติ

i Escape Sequence คือ อักขระพิเศษที่ใช้สำหรับแทนหน้าที่การทำงานพิเศษของปุ่มบนแป้นพิมพ์ เช่น การขึ้นบรรทัดใหม่จะใช้ \n หรือปุ่ม tab จะใช้ \t เป็นต้น

Output:
I am print command
I am println command



การรับข้อมูล

การรับข้อมูลเข้าด้วยคลาส Scanner ซึ่งอยู่ใน java.util package ดังนั้นจึงต้องใช้คำสั่ง import java.util.Scanner; เพื่อนำเข้าคลาส Scanner ก่อนเสมอ ขั้นตอนการเรียกใช้งานเมธอดของคลาส Scanner เพื่อรับข้อมูลเข้า มีดังนี้

- 1) นำเข้าคลาส Scanner โดยใช้คำสั่ง

```
import java.util.Scanner;
```

- 2) สร้างออบเจกต์ของคลาส Scanner สำหรับรับข้อมูลเข้าจากแป้นพิมพ์ (keyboard) โดยจะสร้างเพียงครั้งเดียวเท่านั้น และจะใช้งานออบเจกต์นี้ตลอดทั้งโปรแกรม เช่น

```
Scanner kb = new Scanner(System.in);
```

- 3) เรียกใช้งานเมธอดเพื่อรับข้อมูลเข้าจากแป้นพิมพ์ ตัวอย่างเมธอดที่ใช้งานบ่อยๆ มีดังนี้

ที่	Method	Return Type	คำอธิบาย
1	nextInt()	int	รับจำนวนเต็มประเภท int
2	nextLong()	long	รับจำนวนเต็มประเภท long
3	nextFloat()	float	รับจำนวนทศนิยมประเภท float
4	nextDouble()	double	รับจำนวนทศนิยมประเภท double
5	nextLine()	String	รับข้อความทีละบรรทัด

* Return Type คือชนิดของข้อมูล รายละเอียดจะอยู่ในบทที่ 3

```
//InputData.java
1 import java.util.Scanner;
2 public class InputData {
3     public static void main(String[] arg) {
4         Scanner kb = new Scanner(System.in);
5         int num;
6         num = kb.nextInt();
7         System.out.println(num);
8     }
9 }
```

ตัวอย่างที่ 3 การรับข้อมูลเข้าเป็นจำนวนเต็มด้วยเมธอด nextInt()

บรรทัดที่ 1 คือ การนำเข้าคลาส Scanner

บรรทัดที่ 3 คือ การสร้างออบเจกต์ของคลาส Scanner ชื่อ kb หรือการกำหนดให้ตัวแปร kb อ้างอิงไปยังออบเจกต์ของคลาส Scanner ที่ถูกสร้างขึ้นสำหรับรับข้อมูลเข้าทางแป้นพิมพ์

บรรทัดที่ 6 คือ การเรียกใช้เมธอด nextInt() เพื่อรับจำนวนเต็มเก็บไว้ในตัวแปร num

ในบางครั้งเราอาจจะต้องการรับข้อมูลที่เป็นข้อความยาวๆ เช่น การรับข้อมูลชื่อ หรือนามสกุลเข้าโปรแกรมของเรา เราจะต้องเลือกใช้เมธอด next() ในการรับข้อมูลได้ ดังนี้

```
//InputString.java
1 import java.util.Scanner;
2 public class InputData {
3     public static void main(String[] arg) {
4         String name, surname;
5         Scanner in = new Scanner(System.in);
6         System.out.print("Your name? ");
7         name = in.next();
8         System.out.print("Your surname? ");
9         surname = in.next();
10        System.out.println("Hello" + name+
11            " " + surname);
12    }
13 }
```

บรรทัดที่ 3 คือ การสร้างตัวแปรชื่อ name และ surname เป็นชนิดข้อความ (String)

บรรทัดที่ 6 และ 8 คือ การพิมพ์ข้อความออกหน้าจอ

บรรทัดที่ 7 และ 9 คือ การเรียกใช้เมธอด next() ซึ่งมีหน้าที่รับข้อมูลทางคีย์บอร์ดที่เป็นชนิดข้อความ

บรรทัดที่ 10 และ 11 คือ การพิมพ์ข้อความออกทางหน้าจอ โดยข้อความที่ส่งออกจะมีทั้งชนิดข้อความ (ภายในเครื่องหมายคำพูด) และตัวแปร name และ surname การพิมพ์ข้อความออกหน้าจอในลักษณะนี้จะต้องมีตัวเชื่อม คือ เครื่องหมาย + นั่นเอง

i เครื่องหมาย + ในคำสั่ง print, println ทำหน้าที่เชื่อมข้อความ (Concatenate) เข้าด้วยกัน



3

ตัวแปร นิพจน์และตัวดำเนินการ

ความคิดรวบยอด

 Video USSยาย

Video 3 – ตัวแปรและตัวดำเนินการ

1

ตัวแปร

ตัวแปร (Variable) เปรียบเสมือนหน่วยความจำของโปรแกรม หน่วยความจำนี้จะมีหลายชนิด ทั้งที่เก็บเลขจำนวนเต็ม เก็บเลขทศนิยม เก็บอักขระ หรือข้อความ เป็นต้น



goo.gl/Q5ktXo

2

นิพจน์และตัวดำเนินการ

เป็นการกระทำกับตัวแปรในรูปแบบต่างๆ เช่น การนำตัวแปรมาบวก ลบ คูณ หรือหาร เรียกว่าเป็นการดำเนินการทางคณิตศาสตร์ ยังมีการดำเนินการอื่นๆ อีก ที่นักเรียนจะได้เรียนรู้ในบทนี้



ข้อมูล (Data) ไม่ว่าจะอยู่ในรูปแบบข้อความ ตัวเลข หรือรูปแบบใดๆก็ตาม เราจะใช้ข้อมูลเหล่านั้นมาประมวลผลในโปรแกรม เช่น บวก ลบ คูณ หาร เป็นต้น และเพื่อให้ข้อมูลเหล่านั้นง่ายต่อการจัดการ เราจึงเก็บข้อมูลไว้ในพื้นที่ของหน่วยความจำ และเรียกการแทนค่าข้อมูลในหน่วยความจำว่า ตัวแปร (Variable) โดยจะต้องมีการระบุชนิดข้อมูลที่จะจัดเก็บไว้ในตัวแปรชัดเจน

ตัวแปร

ชนิดของข้อมูล

ชนิดข้อมูลแต่ละประเภทจะมีขนาดและช่วงค่าของข้อมูลแตกต่างกัน และสามารถสรุปประเภทชนิดข้อมูล ขนาดและช่วงค่าของข้อมูล ได้ดังนี้

ที่	ประเภท	ชนิดของข้อมูล	ขนาด (bit)	ช่วงของข้อมูล
1	จำนวนเต็ม	byte	8	-2^7 ถึง 2^7-1
		short	16	-2^{15} ถึง $2^{15}-1$
		int	32	-2^{31} ถึง $2^{31}-1$
		long	64	-2^{63} ถึง $2^{63}-1$
2	ทศนิยม	float	32	
		double	64	
3	อักขระ	char	16	0 ถึง 65,535
4	ทศนิยม	boolean	1	true หรือ false

การสร้างตัวแปร

ตัวแปร คือ ชื่อที่กำหนดขึ้นเพื่อการอ้างอิงตำแหน่งหน่วยความจำที่ใช้เก็บข้อมูล ซึ่งสามารถเก็บข้อมูลที่มีชนิดข้อมูลแตกต่างกันได้ มีรูปแบบการประกาศดังนี้

[modifier] dataType VariableName;

โดยที่ **[modifier]** คือ คีย์เวิร์ดกำหนดการเข้าถึงข้อมูล (มีหรือไม่มีก็ได้)

dataType คือ ชนิดของข้อมูล

VariableName คือ ชื่อตัวแปร

ตัวอย่าง

```
int x;
float data1, data2, sum;
```

จากตัวอย่าง เป็นการประกาศตัวแปรชื่อ x ที่เป็นชนิดจำนวนเต็ม int นอกจากนี้การประกาศตัวแปรยังสามารถประกาศพร้อมกันหลายตัวได้ ดังตัวอย่างเป็นการประกาศตัวแปรชื่อ data1, data2 และ sum พร้อมกัน โดยตัวแปรทั้ง 3 ตัวเป็นชนิดทศนิยม (float) เป็นต้น

กฎเหล็กในการสร้างตัวแปร

ชื่อต่างๆที่ใช้ในโปรแกรม เช่น ชื่อตัวแปร ชื่อค่าคงที่ ชื่อเมธอด ชื่อคลาส เป็นต้น ซึ่งจะต้องตั้งชื่อให้ถูกต้องตามหลักการตั้งชื่อ ดังนี้

- 1) ชื่อที่เป็นตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่มีความแตกต่างกัน หรืออาจกล่าวได้ว่าชื่อในภาษาจาวาเป็น case-sensitive
- 2) ชื่อจะต้องขึ้นต้นด้วยตัวอักษรภาษาอังกฤษ หรือ _ หรือ \$ เท่านั้น
- 3) ชื่อจะต้องประกอบด้วยตัวอักษรภาษาอังกฤษ หรือตัวเลข หรือ underscore (_) หรือ dollarsign (\$) เท่านั้น และห้ามมีช่องว่าง
- 4) ห้ามตั้งชื่อซ้ำกับคีย์เวิร์ดหรือคำสงวน ดังนี้

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

ข้อแนะนำในการตั้งชื่อตัวแปร (Variable) และเมธอด (Method)

- ขึ้นต้นชื่อด้วยตัวอักษรภาษาอังกฤษตัวพิมพ์เล็ก (a ถึง z)
- หากชื่อประกอบด้วยคำหลายคำ แนะนำให้ใช้ตัวอักษรตัวแรกของคำต่อไปเป็นตัวพิมพ์ใหญ่ หรือ underscore เพื่อให้อ่านง่าย
- ตัวอย่างชื่อตัวแปร เช่น name, studentRecord
- ตัวอย่างชื่อ Method เช่น getName, isLetter

i CamelCase คือ วิธีการเขียนคำในภาษาอังกฤษ โดยไม่ให้มีช่องว่าง ทำได้โดยการเขียนพยัญชนะตัวแรกของคำเป็นตัวพิมพ์ใหญ่ Camel แปลว่าอูฐ การเขียนลักษณะนี้จึงเหมือนกับหลังอูฐ





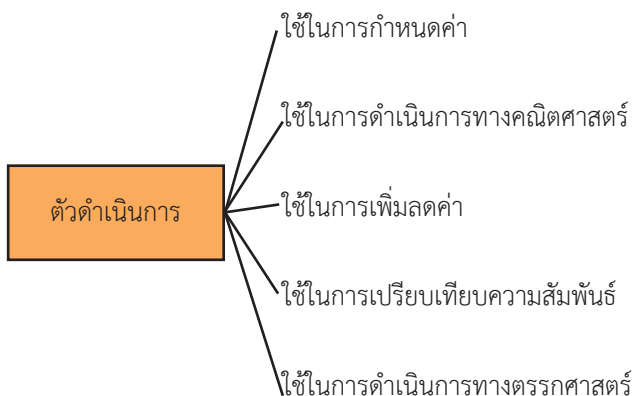
นิพจน์และตัวดำเนินการ

นิพจน์ คือรูปแบบการเขียนคำสั่งระหว่างตัวดำเนินการ (Operator) และตัวถูกกระทำ (Operand) เช่น การกำหนดค่าให้กับตัวแปร สามารถเขียนนิพจน์ได้ดังนี้

ตัวถูกกระทำ
int x = 2 + 3;
ตัวดำเนินการ
นิพจน์

เป็นการเขียนนิพจน์ 2+3 ซึ่งมีเครื่องหมาย + เป็นตัวดำเนินการ และตัวถูกกระทำคือ 2 และ 3 โดยเมื่อคำนวณแล้วจะได้ผลลัพธ์เท่ากับ 5 ซึ่งถูกกำหนดค่าให้กับตัวแปร x

ตัวดำเนินการ (Operators) สามารถแบ่งได้เป็น



1. ตัวดำเนินการกำหนดค่า (Assignment Operators)

ตัวดำเนินการกำหนดค่า คือเครื่องหมายที่ใช้ในการกำหนดค่าข้อมูลให้กับตัวแปรต่างๆ โดยใช้เครื่องหมายเท่ากับ (=) เช่น
int x = 20; เป็นการกำหนดค่าให้กับตัวแปร x เก็บข้อมูลจำนวนเต็ม มีค่าเท่ากับ 20

2. ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)

ใช้สำหรับการคำนวณทางคณิตศาสตร์ ประกอบด้วย

เครื่องหมายทางคณิตศาสตร์ ได้แก่ เครื่องหมาย +, -, *, / และ %

ทั้งนี้ เมื่อมีการคำนวณ ในภาษาจาวาจะมีการแปลงชนิดข้อมูลที่มีขนาดเล็กไปเป็นชนิดข้อมูลขนาดใหญ่โดยอัตโนมัติ ซึ่งจะมีการแปลงชนิดข้อมูลก็ต่อเมื่อค่าในตัวแปรทางขวามือของเครื่องหมายเท่ากับ (=) มีขนาดเล็กกว่าค่าทางซ้าย และเมื่อข้อมูลทางขวามือมีการคำนวณด้วยชนิดข้อมูลตัวเลขแบบต่างๆ ผลลัพธ์จากการคำนวณจะถูกแปลงให้เป็นชนิดข้อมูลที่มีขนาดใหญ่มากที่สุดตามลำดับ ดังนี้ double, float, long, int, char, short, byte

ตัวดำเนินการเครื่องหมาย / จะให้ผลลัพธ์เป็นชนิดข้อมูลจำนวนทศนิยม ก็ต่อเมื่อตัวถูกกระทำตัวใดตัวหนึ่งเป็นชนิดข้อมูลจำนวนทศนิยม เช่น 20 / 3.0 จะมีค่าเท่ากับ 6.66667 แต่ถ้าตัวถูกกระทำเป็นจำนวนเต็มทั้งหมดจะให้ผลลัพธ์เป็นจำนวนเต็ม เช่น 20 / 3 ซึ่งผลลัพธ์มีค่าเท่ากับ 6

ตัวดำเนินการเครื่องหมาย % จะให้ผลลัพธ์เป็นตัวเลขเศษซึ่งเกิดจากการหารของตัวเลข 2 จำนวนดังตัวอย่าง เช่น

20 % 3	จะได้เศษของการหารเท่ากับ 2
51.5 % 2	จะได้เศษของการหารเท่ากับ 1.5

3. ตัวดำเนินการเพิ่มค่าและลดค่า

(Increment and Decrement Operators)

ตัวดำเนินการเพิ่มค่าและลดค่า ตัวดำเนินการเพิ่มค่า คือการเพิ่มค่าขึ้นครั้งละ 1 โดยการใช้เครื่องหมาย ++ วางไว้หน้าหรือหลังตัวแปร เช่น x++, ++x เป็นต้น ตัวดำเนินการลดค่า คือการลดค่าลงครั้งละ 1 โดยการใช้เครื่องหมาย -- วางไว้หน้าหรือหลังตัวแปร เช่น x--, --x เป็นต้น เช่น

x++	มีความหมายเดียวกันกับ x = x + 1 หรือ x += 1
++x	มีความหมายเดียวกันกับ x = x + 1 หรือ x += 1
x--	มีความหมายเดียวกันกับ x = x - 1 หรือ x -= 1
--x	มีความหมายเดียวกันกับ x = x - 1 หรือ x -= 1

นอกจากนี้ การวางเครื่องหมายเพิ่มค่าและลดค่าไว้หน้าหรือหลังตัวแปรก็จะทำให้ได้ผลลัพธ์แตกต่างกัน เช่น

ถ้า x = 10; แล้ว เมื่อกำหนดให้ y = x++; จะได้ค่าข้อมูลของ y เท่ากับ 10 และค่าข้อมูลของ x เท่ากับ 11 เนื่องจากว่าเครื่องหมาย ++ อยู่หลังตัวแปร x ซึ่งตัวแปร x จะกำหนดค่าให้กับ y ก่อนแล้วจึง เพิ่มค่าของตัวแปร x ที่หลัง

ถ้า x = 10; แล้ว เมื่อกำหนดให้ y = ++x; จะได้ค่าข้อมูลของ y เท่ากับ 11 และค่าข้อมูลของ x เท่ากับ 11 เนื่องจากว่าเครื่องหมาย ++ อยู่ข้างหน้าตัวแปร x ซึ่งตัวแปร x จะเพิ่มค่าก่อนแล้วจึงกำหนดค่าให้กับ y ที่หลัง

4. ตัวดำเนินการเปรียบเทียบความสัมพันธ์

(Equality and Relational Operators)

ตัวดำเนินการเปรียบเทียบความสัมพันธ์ใช้ในการเปรียบเทียบค่าของชนิดข้อมูลใดๆ สองค่า โดยผลลัพธ์ที่ได้จากการเปรียบเทียบความสัมพันธ์จะเป็นชนิดข้อมูลตรรกะ ได้แก่ จริง(true) หรือเท็จ(false) เครื่องหมายของตัวดำเนินการเปรียบเทียบความสัมพันธ์มีดังนี้



ตัวดำเนินการ	ความหมาย	ตัวอย่าง
==	เท่ากับ	a == b
!=	ไม่เท่ากับ	a != b
>	มากกว่า	a > b
>=	มากกว่าหรือเท่ากับ	a >= b
<	น้อยกว่า	a < b
<=	น้อยกว่าหรือเท่ากับ	a <= b

ลำดับความสำคัญ

ในนิพจน์ต่างๆ อาจมีตัวดำเนินการประกอบกันอยู่มากกว่าหนึ่งตัว ในการหาผลลัพธ์ของนิพจน์นั้นคอมไพเลอร์จะพิจารณาว่าจะทำตัวดำเนินการใดก่อนหลัง ในภาษาจาวามีการจัดลำดับการทำงานของตัวดำเนินการดังนี้

ลำดับที่	เครื่องหมาย	ลำดับการทำงาน
1	()	-
2	++, --, !	ซ้ายไปขวา
3	*, / , %	ซ้ายไปขวา
4	+, -	ซ้ายไปขวา
5	< , <= , > , >=	ซ้ายไปขวา
6	==, !=	ซ้ายไปขวา
7	&&	ซ้ายไปขวา
8		ซ้ายไปขวา
9	= , +=, -=, *=, /=, %=	ซ้ายไปขวา

5. ตัวดำเนินการทางตรรกะ (Logical Operators)

ตัวดำเนินการทางตรรกะ เป็นตัวดำเนินการเกี่ยวข้องกับนิพจน์ที่สามารถบอกค่าความจริงเป็นจริง (true) หรือเท็จ (false) ได้ เช่น ตัวแปรประเภท boolean ผลลัพธ์ที่ได้จากการกระทำจะได้ค่าคงที่ตรรกะเป็น true หรือ false ตัวดำเนินการทางตรรกะ ได้แก่

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
!	NOT (นิเสธ)	!(a < b)
&&	AND (และ)	(a > 0)&&(a < 10)
	OR (หรือ)	(a > 0) (a > 50)

ตัวดำเนินการทางตรรกะ

ลองกันเลย!

```
//App3_3.java
1 public class App3_3 {
2     public static void main(String[] args) {
3         float x = 1.5f;
4         double y = 200.0;
5         int a = 10;
6         float result1 = x + a;
7         double result2 = a+15/a-y*x;
8         System.out.println(124%a) ;
9         System.out.println(result1);
10        System.out.println(result2);
11    }
12 }
```

Output:
4
11.5
-289.0



4

คำสั่งควบคุมแบบมีทางเลือก

ความคิดรวบยอด

 Video USSยาย

Video 4 – คำสั่งควบคุมแบบมีทางเลือก

1

คำสั่ง IF

คำสั่ง if เป็นคำสั่งที่ใช้เป็นทางเลือกในการตัดสินใจทำงานอย่างใดอย่างหนึ่ง โดยมีการเขียนนิพจน์ที่เป็นเงื่อนไขทางตรรกศาสตร์ในคำสั่ง if และจะทำคำสั่งใน if ถ้าเงื่อนไขเป็นจริง



goo.gl/Q5ktXo

2

คำสั่ง SWITCH

การเขียนโปรแกรมที่ต้องการเลือกทำหลายทางเลือก เราสามารถนำคำสั่ง if มาซ้อนกันได้ แต่ถ้าเงื่อนไขที่ต้องตัดสินใจขึ้นกับตัวแปรเดียวหรือนิพจน์ใดๆ เราสามารถใช้คำสั่ง switch แทนได้



คำสั่งควบคุมแบบมีทางเลือก คือ คำสั่งที่ใช้ในการตัดสินใจหรือเลือกทำเหตุการณ์ใดเหตุการณ์หนึ่ง โดยที่มีการตรวจสอบเงื่อนไขของคำสั่งก่อนทำงาน เพื่อตัดสินใจเลือกทิศทางการทำงานของโปรแกรม ซึ่งคำสั่งควบคุมแบบมีทางเลือกนี้ มีอยู่ 2 คำสั่งหลักๆ ได้แก่ คำสั่ง if และ คำสั่ง switch

คำสั่ง if

คำสั่ง if เป็นคำสั่งสำหรับให้โปรแกรมเลือกทำ โดยคำสั่ง if มีอยู่ 3 รูปแบบ คือ

คำสั่ง if แบบทางเดียว เป็นคำสั่งที่ใช้เป็นทางเลือกในการตัดสินใจทำงานอย่างใดอย่างหนึ่ง โดยมีการเขียนนิพจน์ที่เป็นเงื่อนไขทางตรรกศาสตร์ในคำสั่ง if และจะทำคำสั่งใน if ถ้าเงื่อนไขเป็นจริง

```
if (condition) statement; หรือ if (condition){ statements; }
```

รูปแบบการเขียนคำสั่ง if ทางเดียว

คำสั่ง if สองทางเลือก โปรแกรมจะทำตามชุดคำสั่งใดคำสั่งหนึ่งจาก 2 ทางเลือก โดยตรวจสอบนิพจน์เงื่อนไขที่กำหนดว่าเป็นจริงหรือเท็จ ถ้านิพจน์เป็นจริง โปรแกรมจะทำงานตามชุดคำสั่งภายใต้ if แต่ถ้านิพจน์เป็นเท็จ โปรแกรมจะทำงานตามชุดคำสั่งที่อยู่ภายใต้ else

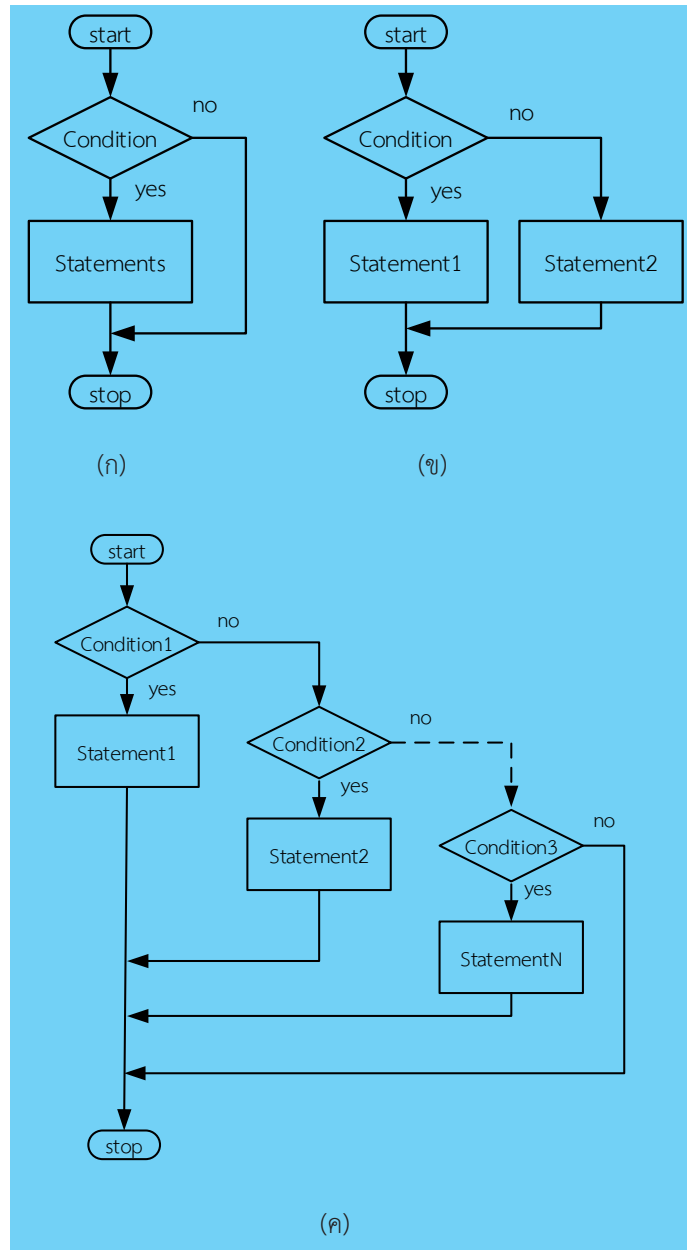
```
If(condition){ statement1; } else{ statement2; }
```

รูปแบบการเขียนคำสั่ง if สองทางเลือก

Nested-if เป็นคำสั่งควบคุมให้โปรแกรมเลือกทำงานชุดคำสั่งใดชุดคำสั่งหนึ่งจากหลายทางเลือก แต่ละทางเลือกจะมีการกำหนดนิพจน์เงื่อนไขของแต่ละทางไว้ด้วย โดยโปรแกรมจะตรวจสอบนิพจน์เงื่อนไขของแต่ละทางเลือก หากพบว่าทางเลือกไหนมีนิพจน์เป็นจริง ก็จะทำงานในทางเลือกนั้นโดยไม่พิจารณาทางเลือกอื่นที่ยังไม่ตรวจสอบอีก

```
if(condition1){ statements1; }else if (condition2){ statements2; } . . . else{ statementsN; }
```

รูปแบบการเขียนคำสั่ง if แบบหลายทางเลือก



Flow Chart การทำงานของคำสั่งควบคุมแบบมีทางเลือก (ก) คำสั่ง if แบบทางเดียว (ข) คำสั่ง if แบบสองทาง และ (ค) คำสั่ง if แบบหลายทาง (nested if)



คำสั่ง switch

การเขียนโปรแกรมที่ต้องการเลือกทำหลายทางเลือก เราสามารถนำคำสั่ง if มาซ้อนกันได้ แต่ถ้าเงื่อนไขที่ต้องตัดสินใจขึ้นกับตัวแปรเดียวหรือนิพจน์ใดๆ เราสามารถใช้คำสั่ง switch แทนได้

คำสั่ง switch เป็นคำสั่งที่จะตรวจสอบค่าของตัวแปรหรือนิพจน์ ถ้าตรงตาม case ใดก็จะทำตามคำสั่งนั้นๆ จนกว่าจะเจอคำสั่ง break ในกรณีที่ตรงกับ case ที่ระบุในโปรแกรม จะมาทำงานที่คำสั่ง default โดยอัตโนมัติ

```
switch(variable/expression){
    case constant1:  statements;
                    break;
    case constant2:  statements;
                    break;
    .
    .
    .
    case constantN:  statements;
                    break;
    default:         statements;
                    break;
}
```

รูปแบบการเขียนคำสั่ง switch

ตัวอย่างที่ 4-2

```
1 public class ScoreTest {
2     public static void main(String[] args) {
3         Scanner in = new Scanner(System.in);
4         float score = in.nextFloat();
5         if(score >= 50){
6             System.out.println("Passed");
7         }
8         else {
9             System.out.println("Failed");
10            System.out.println("You must take this
11                course again.");
12        }
13    }
14 }
```

Output:
Enter score: 10
Failed
You must take this course again.

จากตัวอย่างที่ 4-2 เป็นการใช้งานคำสั่ง if...else เพื่อตรวจสอบคะแนนสอบของนักเรียน โดยจะแสดงข้อความ Passed ออกทางจอภาพ ถ้าคะแนนสอบของนักเรียนมากกว่าหรือเท่ากับ 50 แต่ถ้าเงื่อนไขเป็นเท็จ (คะแนนสอบของนักเรียนน้อยกว่า 50 คะแนน) จะแสดงข้อความ Failed และ You must take this course again. จากตัวอย่างผู้ใช้ได้ป้อนคะแนนเข้ามา 10 คะแนน ทำให้เงื่อนไขภายในคำสั่ง if เป็นเท็จ โปรแกรมจึงทำงานภายในขอบเขตของคำสั่ง else

ตัวอย่างที่ 4-1

```
1 public static void main(String[] args) {
2     Scanner in = new Scanner(System.in);
3     System.out.print("Enter score: ")
4     float score = in.nextFloat();
5     if(score < 50) {
6         System.out.println("Failed");
7         System.out.println("You must take this
8             course again.");
9     }
```

Output:
Enter score: 10
Failed
You must take this course again.

จากตัวอย่างที่ 4-1 เป็นการใช้นำคำสั่ง if แบบทางเดียวเพื่อตรวจสอบค่าคะแนนที่ป้อนเข้ามา โดยเงื่อนไขในคำสั่ง if (บรรทัดที่ 5) เขียนไว้เพื่อตรวจสอบค่า score หากมีค่าน้อยกว่า 50 ก็จะทำงานภายในคำสั่ง if จากกรณีตัวอย่างผู้ใช้ป้อนค่าคะแนนเป็น 10 คำนี้นี้จะถูกเก็บไว้ในตัวแปร score ทำให้เงื่อนไขภายใน if จึงเป็นจริง โปรแกรมจึงพิมพ์ข้อความว่า Failed และ You must take this course again.



ตัวอย่างที่ 4-3

```
1 import java.util.Scanner;
2 public class GradeTest {
3     public static void main(String[] args) {
4         Scanner in = new Scanner(System.in);
5         System.out.print("Enter Score: ");
6         float score = in.nextFloat();
7         char grade;
8         if(score >= 80)
9             grade = 'A';
10        else if(score >= 70)
11            grade = 'B';
12        else if(score >= 60)
13            grade = 'C';
14        else if(score >= 50)
15            grade = 'D';
16        else
17            grade = 'F';
18        System.out.println(grade);
19    }
20 }
```

Output:
Enter score: 80
A

ตัวอย่างการใช้งาน nested-if ได้แก่โปรแกรมคำนวณเกรด แล้วแสดงผลออกทางจอภาพ โดยโปรแกรมจะรับค่าคะแนนจากผู้ใช้ใน ช่วง 0 – 100 แล้วแสดงเกรดในระบบ A – F ออกทางจอภาพ จากตัวอย่างผู้ใช้ได้ป้อนค่าคะแนนเข้ามา 80 คะแนน คำสั่งในเงื่อนไข if (บรรทัดที่ 8) จึงเป็นจริง ทำให้ตัวแปร grade มีค่าเป็นอีกจะ 'A' จากนั้นโปรแกรมจะกระโดดไปทำงานที่บรรทัดที่ 18 คือพิมพ์อักขระ A ออกมานั่นเอง

ตัวอย่างที่ 4-4

```
1 public class SwitchDemo {
2     public static void main(String[] args) {
3         int day = 5;
4         String dayString;
5         switch (day) {
6             case 1: dayString = "Monday";
7                 break;
8             case 2: dayString = "Tuesday";
9                 break;
10            case 3: dayString = "Wednesday";
11                break;
12            case 4: dayString = "Thursday";
13                break;
14            case 5: dayString = "Friday";
15                break;
16            case 6: dayString = "Saturday";
17                break;
18            case 7: dayString = "Sunday";
19                break;
20            default: dayString = "Invalid day";
21        }
22        System.out.println(dayString);
23    }
24 }
```

Output:
Friday

จากตัวอย่างที่ 4-4 เป็นการใช้คำสั่ง switch เพื่อพิมพ์ชื่อวันออกทางหน้าจอ โดยใช้หมายเลขวันที่เก็บไว้ในตัวแปร day ในตัวอย่างค่าของ day คือ 5 เพราะฉะนั้นเมื่อโปรแกรมนำ day ไปตรวจสอบด้วยคำสั่ง switch จะตรงกับ case 5 โปรแกรมจึงเก็บข้อความ "Friday" ไว้ในตัวแปร dayString (ตัวแปรชนิด String คือตัวแปรใช้เก็บข้อความ) เมื่อโปรแกรมทำงานถึงคำสั่ง break; ก็จะกระโดดมาทำคำสั่ง System.out.print (บรรทัด 22) เพื่อพิมพ์ค่าของ dayString ออกทางหน้าจอ ผลลัพธ์ที่ได้คือโปรแกรมจะพิมพ์ค่า Friday ออกทางหน้าจอ



สถานการณ์ปัญหา 4-1

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมคิดเงินร้านค้า ปัญหาที่พบบ่อยๆ คือคิดเงินผิดจึงอยากจะได้โปรแกรมที่จะช่วยให้การคิดเงินถูกต้องและรวดเร็วขึ้น ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

Output:
Product price : 100
Cash : 150
Change : 50

ตัวอย่างผลลัพธ์ 4-1

สถานการณ์ปัญหา 4-2

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมตรวจสอบโรคอ้วน โดยค่าตรวจชนิมวลกาย ตั้งแต่ 30 ขึ้นไปเรียกว่าเป็นโรคอ้วน โดยความต้องการของผู้ใช้ต้องการให้มีการป้อนค่าน้ำหนักและส่วนสูงเข้าสู่โปรแกรม จากนั้นโปรแกรมจะทำการคำนวณและบอกว่าเป็นโรคอ้วนหรือไม่ ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

Output:
Enter weight(kg): 55
Enter height(cm): 165
BMI: 21.2 - Normal

ตัวอย่างผลลัพธ์ 4-2

สถานการณ์ปัญหา 4-3

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมตรวจสอบสุขภาพของนักเรียน ซึ่งปัญหาของนักเรียนในโรงเรียนส่วนใหญ่ก็คือโรคอ้วน หากนักเรียนในโรงเรียนได้รับข้อมูลการตรวจสอบสุขภาพของตนเองตั้งแต่เนิ่นๆ ก็จะเป็นผลดีต่อการกำหนดวิธีการดำเนินชีวิตประจำวัน เช่น การควบคุมอาหาร แต่เนื่องจากนักเรียนในโรงเรียนมีมากทำให้การดำเนินการล่าช้า โดยความต้องการของผู้ใช้คือโปรแกรมสามารถรายงานผลออกมา 4 ระดับ คือ

Output:
Enter weight(kg): 70
Enter height(cm): 165
BMI: 27.0 - Overweight

ตัวอย่างผลลัพธ์ 4-3

BMI <18.5 หมายถึง น้ำหนักต่ำกว่ามาตรฐาน (Underweight)

BMI 18.5–24.9 หมายถึง ปกติ (Normal weight)

BMI 25–29.9 หมายถึง น้ำหนักเกิน (Overweight)

BMI >30 หมายถึง อ้วน (Obesity)

ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

ภารกิจการเรียนรู้

1. นักเรียนวิเคราะห์ว่าจะใช้คำสั่งอะไรบ้างในการเขียนโปรแกรม
2. นักเรียนวิเคราะห์ว่าจะแก้ปัญหาดังกล่าวอย่างไร อธิบายเป็น Flow Chart อธิบายว่าปัญหาใหม่มีความเหมือนหรือแตกต่างกับปัญหาที่นักเรียนเคยแก้ได้อย่างไร
3. นักเรียนออกแบบและพัฒนาโปรแกรมเพื่อแก้ปัญหาดังกล่าว



5

คำสั่งควบคุมแบบวนซ้ำ

ความคิดรวบยอด

 Video USSาย

Video 5 – คำสั่งควบคุมแบบวนซ้ำ

1

คำสั่ง FOR

คำสั่ง for เป็นคำสั่งแบบวนซ้ำ (loop) ที่เหมาะสำหรับงานที่ต้องการ
การกำหนดจำนวนครั้งของการวนซ้ำที่ตายตัว

2

คำสั่ง WHILE

คำสั่ง while เป็นคำสั่งแบบวนซ้ำ (loop) ที่ตรวจสอบเงื่อนไขก่อนการ
ทำงานงานใน loop

3

คำสั่ง DO-WHILE

คำสั่ง do-while เป็นคำสั่งแบบวนซ้ำ (loop) ที่ตรวจสอบเงื่อนไขหลัง
การทำงานงานใน loop เพราะฉะนั้นคำสั่งแบบ do-while จะ
ทำงานใน loop อย่างน้อย 1 รอบเสมอ



goo.gl/Q5ktXo



การวนซ้ำ (Loop) หมายถึง กลุ่มของคำสั่งที่ต้องการประมวลผลซ้ำหลายครั้ง ทั้งนี้ขึ้นอยู่กับเงื่อนไขที่กำหนดไว้ ภาษาจาวามีคำสั่งวนซ้ำอยู่ 3 คำสั่ง ได้แก่ คำสั่ง for, คำสั่ง while และคำสั่ง do...while

คำสั่ง for

คำสั่ง for เป็นคำสั่งวนลูปที่มีส่วนประกอบอยู่ 3 ส่วน ได้แก่
1) ส่วนเริ่มต้น (initialization) 2) ส่วนเงื่อนไข (condition) และ
3) ส่วนปรับค่า (update)

```
for (initialization; condition; update){
    statement1;
    statement2;
}
```

คำสั่ง while

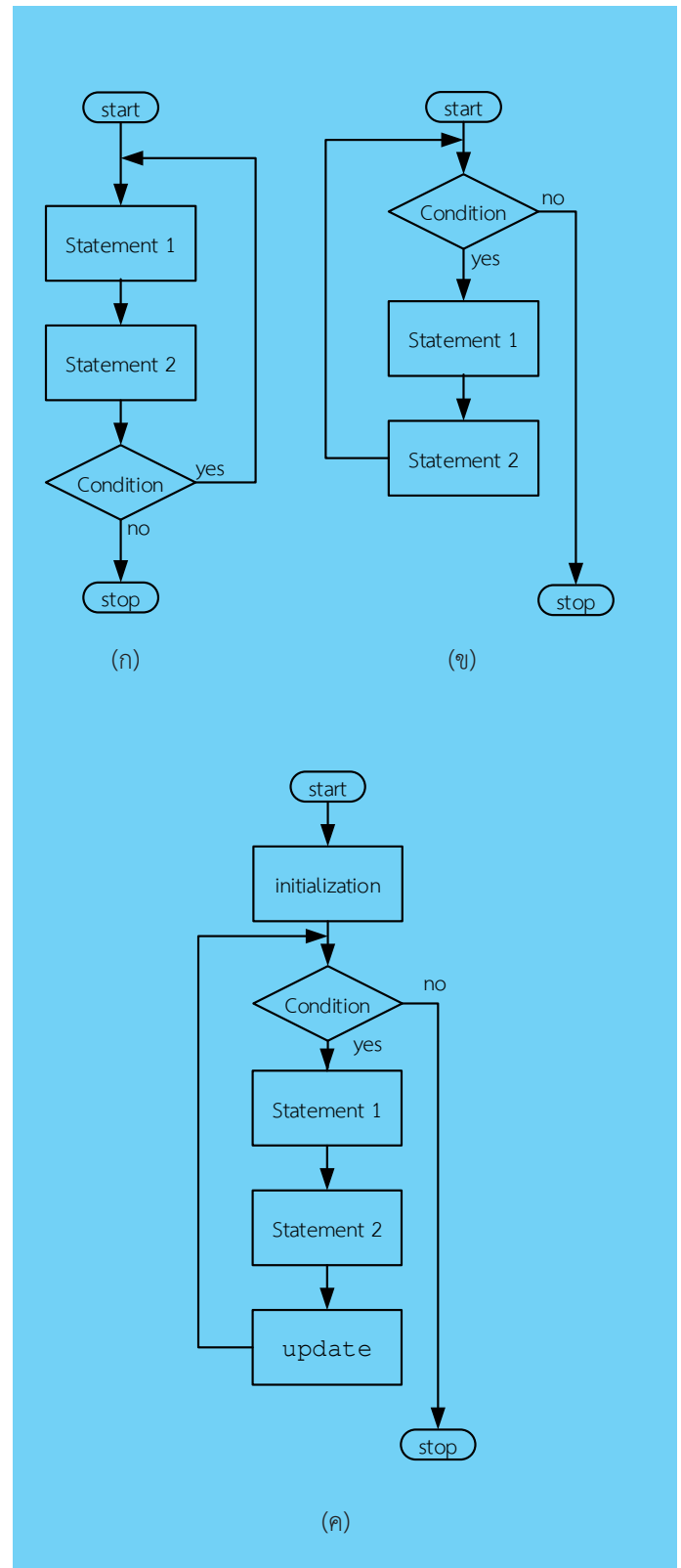
การทำงานของคำสั่ง while จะเริ่มต้นด้วยการตรวจสอบเงื่อนไข (condition) ถ้าผลการตรวจสอบเงื่อนไขเป็นจริง คำสั่ง (statement) ที่อยู่ภายใต้เครื่องหมาย { } ทุกคำสั่งจะได้รับการประมวลผลหนึ่งครั้ง แล้วโปรแกรมจะกลับไปตรวจสอบเงื่อนไข ถ้าผลการตรวจสอบยังคงเป็นจริงอยู่ โปรแกรมจะประมวลผลคำสั่งเดิมอีก 1 รอบ และจะทำซ้ำลักษณะนี้จนกระทั่งผลการตรวจสอบเงื่อนไขเป็นเท็จ จึงจะยุติการทำงานในลูป

```
while (condition){
    statement1;
    statement2;
}
```

คำสั่ง do-while

การทำงานของคำสั่ง do...while จะเริ่มต้นด้วยการทำคำสั่ง (statement) ที่อยู่ภายใต้เครื่องหมาย { } แล้วจึงค่อยตรวจสอบเงื่อนไข (condition) ถ้าผลการตรวจสอบเป็นจริง ก็จะประมวลผลคำสั่งที่อยู่ใต้เครื่องหมาย { } อีก และทำซ้ำในลักษณะนี้จนกระทั่งการตรวจสอบเงื่อนไขเป็นเท็จ จึงจะยุติการทำงานในลูป นั่นคือจะมีการทำคำสั่งใน loop อย่างน้อย 1 ครั้งเสมอ

```
do{
    statement1;
    statement2;
}while (condition);
```



Flow Chart การทำงานของคำสั่งแบบวนซ้ำ (ก) คำสั่ง do-while
(ข) คำสั่ง while และ (ค) คำสั่ง for



ตัวอย่างที่ 5-1

```
1 public class ForTest1 {  
2     public static void main(String[] args) {  
3         int digit;  
4         for(digit=0; digit<=9; digit++)  
5             System.out.print(digit);  
6     }  
7 }
```

Output:
0123456789

ตัวอย่างการใช้คำสั่ง *for*

ตัวอย่างที่ 5-2

```
1 public class WhileTest1 {  
2     public static void main(String[] args) {  
3         int digit = 0;  
4         while(digit <= 9) {  
5             System.out.print(digit);  
6             digit++;  
7         }  
8     }  
9 }
```

Output:
0123456789

ตัวอย่างการใช้คำสั่ง *while*

ตัวอย่างที่ 5-3

```
1 public class DoWhileTest1 {  
2     public static void main(String[] args) {  
3         int digit = 0;  
4         do {  
5             System.out.print(digit);  
6             digit++;  
7         } while(digit <= 9);  
8     }  
9 }
```

Output:
0123456789

ตัวอย่างการใช้คำสั่ง *do-while*

จากตัวอย่างที่ 5-1, 5-2 และ 5-3 เป็นตัวอย่างการใช้คำสั่งแบบวนซ้ำ นักเรียนจะสังเกตว่าผลลัพธ์ของทั้ง 3 โปรแกรมเหมือนกัน คือพิมพ์ตัวตั้งแต่ 0 ถึง 9 ออกทางหน้าจอ แต่ใช้คำสั่งแตกต่างกัน เป็นเพราะว่าทั้งคำสั่ง *for*, *while* และ *do-while* ต่างก็มีการทำงานคล้ายๆ กัน คือ ทำซ้ำคำสั่งจนกว่าเงื่อนไขการวนซ้ำจะเป็นเท็จ



สถานการณ์ปัญหา 5-1

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมจำลองการเคลื่อนที่ในแนวเส้นตรง ลูกค้านี้ให้เงื่อนไขมาว่าระยะทางที่เคลื่อนที่ไปได้จะเท่ากับความเร็วคูณกับเวลา ความต้องการของลูกค้าจะป้อนความเร็ว กับเวลาเข้าไปในโปรแกรม จากนั้นโปรแกรมจะคำนวณระยะทางที่เคลื่อนที่ไปได้ของแต่ละวินาที ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

สถานการณ์ปัญหา 5-2

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมเรียนรู้ภาษาอาเซียน โดยโปรแกรมรับจะชื่อประเทศสมาชิก AEC (กำหนดให้ชื่อประเทศจำต้องขึ้นต้นด้วยตัวพิมพ์ใหญ่ เช่น Thailand หรือ Loa เป็นต้น) แล้วแสดงคำทักทายของประเทศนั้นๆ ออกทางจอภาพ (สกดด้วยภาษาไทยหรือภาษาอังกฤษก็ได้) หากไม่ใช่ประเทศสมาชิก AEC ให้ขึ้นข้อความว่า “ประเทศนี้ไม่ได้เป็นสมาชิกอาเซียน” โดยโปรแกรมจะวนซ้ำเพื่อรับคำสั่ง จนกว่าจะกด q เพื่อจบการทำงาน ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

สถานการณ์ปัญหา 5-3

บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมโปรแกรมเพื่อคำนวณค่าบริการโทรศัพท์รายเดือน โดยให้ผู้ใช้ป้อนระยะเวลาที่ใช้งานไป และจำนวนของอินเทอร์เน็ตที่ใช้ไป เพื่อคำนวณออกมาเป็นค่าโทรศัพท์ โดยรายละเอียดโปรแกรมเป็นดังนี้

ค่าบริการรายเดือน 349 บาท

โทรฟรีทุกเครือข่าย 200 นาที

และใช้อินเทอร์เน็ตได้ 500 MB

ค่าโทรและจำนวนอินเทอร์เน็ตส่วนเกินคิดเป็น

ค่าโทร 1.27 บาท/นาที

อินเทอร์เน็ต 1.27 บาท/MB

เมื่อคำนวณเสร็จแล้วโปรแกรมจะวนซ้ำกลับไปรับค่าจากผู้ใช้นี้ใหม่ จนกว่าผู้ใช้จะกดปุ่ม q เพื่อออกจากการทำงานถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

```
Output:
**** DISTANCES CALCULATION***
Formular: S=v*t
Enter velocity(m/s): 5
Enter time(s): 3
Distances:
t = 0: S = 0;
t = 1: S = 5;
t = 2: S = 10;
t = 3: S = 15;
```

ตัวอย่างผลลัพธ์ 4-1

```
Output:
*****ASIAN GREETING WORDS*****
Enter the first charater: T
Greeting: SA-WAD-DEE

*****ASIAN GREETING WORDS*****
Enter the first charater: _
```

ตัวอย่างผลลัพธ์ 5-2

```
Output:
***PHONE FEE CALCULATION***
Press 'q' to quite
Enter call duration (minutes): 100
Enter internet (MB): 400
TOTAL (THB): 349

***PHONE FEE CALCULATION***
Press 'q' to quite
Enter call duration (minutes):_
```

ตัวอย่างผลลัพธ์ 5-3

ภารกิจการเรียนรู้

1. นักเรียนวิเคราะห์ว่าจะใช้คำสั่งอะไรบ้างในการเขียนโปรแกรม
2. นักเรียนวิเคราะห์ว่าจะแก้ปัญหาดังกล่าวอย่างไร อธิบายเป็น Flow Chart อธิบายว่าปัญหาใหม่มีความเหมือนหรือแตกต่างกับปัญหาที่นักเรียนเคยแก้ได้อย่างไร
3. นักเรียนออกแบบและพัฒนาโปรแกรมเพื่อแก้ปัญหาดังกล่าว