

## CHAPTER

## 09

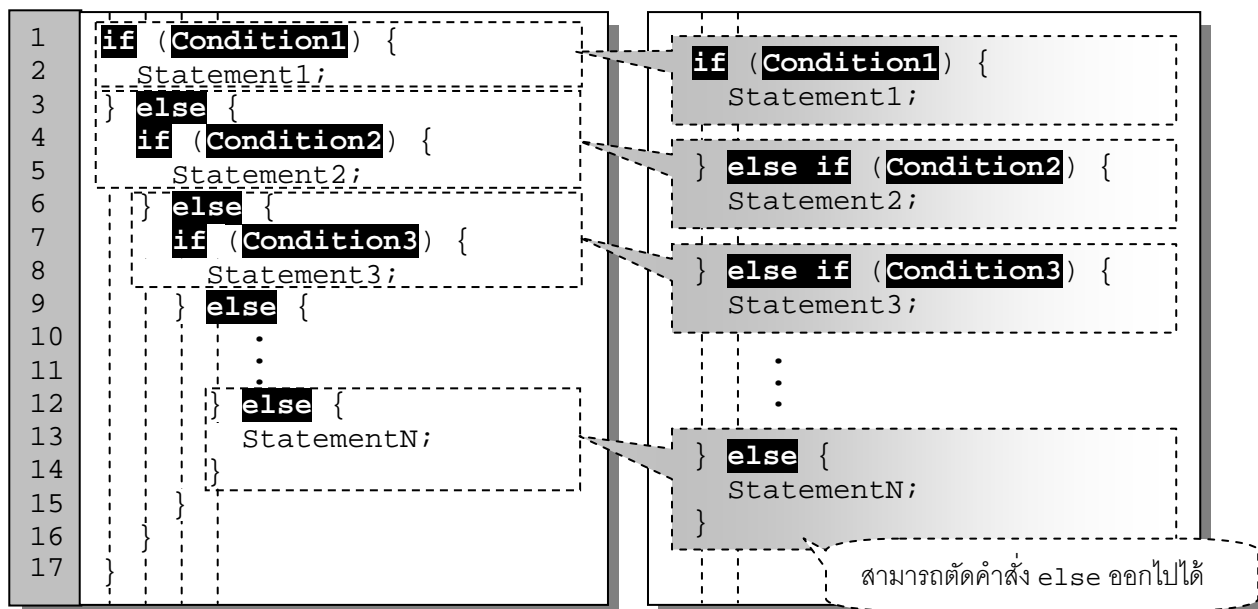
## คำสั่งแบบซับซ้อนและการประยุกต์ (Advanced Statements and Applications)

### 1. คำสั่งตัดสินใจแบบซับซ้อน (Advanced Decision Statements)

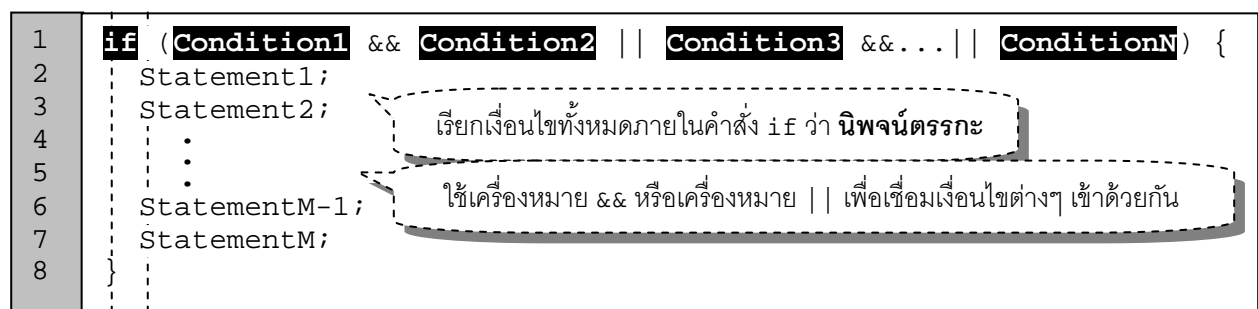
1. รูปแบบของคำสั่งตัดสินใจแบบซับซ้อน ประกอบไปด้วย 2 รูปแบบคือ

- 1) คำสั่งตัดสินใจที่ซ้อนกันหลายชั้น ในบทนี้ได้แก่อำสั่ง if-else ซ้อน if-else (Nested If-Else) ซึ่งได้กล่าวไปแล้วในบทที่ 3 และมีรายละเอียดเพิ่มเติมตามหัวข้อที่ 2
- 2) คำสั่งตัดสินใจแบบหลายเงื่อนไข ในบทนี้ได้แก่อำสั่ง if-else ที่มีเงื่อนไขหลายๆ เงื่อนไขเชื่อมเข้ากัน ด้วยตัวดำเนินการตรรกะ (Logical Operators) โดยมีรายละเอียดตามหัวข้อที่ 3

### 2. คำสั่ง if-else ซ้อน if-else (Nested If-Else)



### 3. คำสั่ง if-else แบบหลายเงื่อนไข



โจทย์ข้อที่ 1 [ระดับง่าย] จงเขียนนิพจน์ตรรกะตามหลักของภาษาจาวาจากเงื่อนไขต่อไปนี้ (10 คะแนน)

- 1) ตัวแปร n เป็นจำนวนเต็มบวกคู่ที่ไม่เกิน 3 หลัก แต่ไม่เท่ากับ 12 และ 112 (2 คะแนน)

- 2) ตัวแปร `n` เป็นจำนวนเต็มบวกที่มี 3 เป็นตัวร่วมหรือมี 5 เป็นตัวร่วม แต่ค่าไม่เกิน 100 (2 คะแนน)

- 3) ตัวแปร `m` เป็นหมายเลขเดือนที่ลงท้ายด้วยคำว่า *ยน* (2 คะแนน)

- 4) ตัวแปร `m` เป็นหมายเลขเดือนที่ไม่ใช่เดือนกุมภาพันธ์ (2 คะแนน)

- 5) ตัวแปร `s` เป็นคำที่เขียนว่า "yes" หรือ "y" โดยไม่สนใจอักษรตัวพิมพ์ใหญ่และพิมพ์เล็ก (2 คะแนน)

โจทย์ข้อที่ 2 [ระดับปานกลาง] จงเขียนเมทอด `getDaysOfMonth(...)` ที่สมบูรณ์เพื่อรับเดือนและปี ค.ศ. เข้ามาทางพารามิเตอร์ โดยให้คำนวณและคืนค่าจำนวนวันจากหมายเลขเดือนและปี ค.ศ. ที่ระบุ เช่น เดือนที่ 1 คืนค่า 31 เดือนที่ 6 คืนค่า 30 เป็นต้น แต่ถ้าหมายเลขเดือนไม่ถูกต้องให้คืนค่า 0 โดยในการคำนวณจะต้องตรวจสอบปีอธิกสุรทิน (Leap Year) ที่เดือนกุมภาพันธ์มี 29 วันด้วย ซึ่งปีอธิกสุรทินคือปี ค.ศ. ที่หาร 4 ลงตัวแต่หาร 100 ไม่ลงตัว หรือปี ค.ศ. ที่หาร 400 ลงตัว (10 คะแนน)

โจทย์ข้อที่ 3 [ระดับปานกลาง] จงเขียนเมทอด `getSize(...)` ที่รับตัวเลขจำนวนเต็มจากพารามิเตอร์ในชื่อ `n` แล้วทำการตรวจสอบค่า `n` ดังต่อไปนี้ (10 คะแนน)

- 1) ถ้า `n >= 0` ให้หาค่ารากที่สองของ `n` แล้วเก็บไว้ในตัวแปร `m` และนำค่า `m` มาจัดกลุ่มดังนี้
  - (1) ถ้า `0 <= m <= 5` ให้แสดงค่า `m` และหาค่า 2 เท่าของ `m` แล้วพิจารณาดังนี้
    - ถ้ามีค่ามากกว่า 7 ให้คืนค่า "Little"
    - นอกเหนือจากนั้นให้คืนค่า "Undefined"
  - (2) ถ้า `5 < m <= 10` ให้แสดงค่า `m` และหาค่า 3 เท่าของ `m` แล้วพิจารณาดังนี้
    - ถ้ามีค่ามากกว่า 22 ให้คืนค่า "Medium"
    - นอกเหนือจากนั้นให้คืนค่า "Undefined"
  - (3) ถ้า `10 < m <= 25` ให้แสดงค่า `m` และหาค่า 4 เท่าของ `m` แล้วพิจารณาดังนี้
    - ถ้ามีค่ามากกว่า 80 ให้คืนค่า "Very Big"
    - ถ้ามีค่ามากกว่า 60 แต่ไม่เกิน 80 ให้คืนค่า "Big"
    - นอกเหนือจากนั้นให้คืนค่า "Undefined"
  - (4) ถ้า `25 < m` ให้แสดงค่า `m` และคืนค่า "Giant"
- 2) ถ้า `n < 0` ให้แสดงค่า `n` และคืนค่า "Negative"

โจทย์ข้อที่ 4 [ระดับปานกลาง] จงเขียนเมทอด `isEngStudent(...)` ที่รับสตริงที่เป็นเลขประจำตัวนิสิต ยาว 10 หลักเท่านั้น และตัวเลขจำนวนเต็มที่เป็นรหัสวิชาใดวิชาหนึ่งยาว 7 หลักเท่านั้น เข้ามาทางพารามิเตอร์ เพื่อใช้ในการตรวจสอบว่าใช้นิสิตชั้นปีที่ 1 ระดับปริญญาตรี คณะวิศวกรรมศาสตร์หรือไม่ และเป็นนิสิตที่ลงทะเบียนเรียนในรายวิชาที่เปิดสอนสำหรับชั้นปีที่ 1 สอนโดยคณะวิศวกรรมศาสตร์หรือไม่ โดยมีหลักเกณฑ์ในการตรวจสอบดังนี้ (10 คะแนน)

1	2	3						9	10		1	2		5					
5	4	3	0	8	1	9	5	2	1			2	1	1	0	1	9	1	
เลขประจำตัวนิสิต										รหัสวิชา									

- 1) ตัวเลขหลักที่ 1-2 ของเลขประจำตัวนิสิต หมายถึง ปีการศึกษาที่เริ่มเข้าเรียน
- 1) ตัวเลขหลักที่ 3 ของเลขประจำตัวนิสิต หมายถึง ระดับการศึกษา โดยเลข 3 คือระดับปริญญาตรี
- 2) ตัวเลขหลักที่ 9-10 ของเลขประจำตัวนิสิต หมายถึง รหัสคณะ โดยรหัส 21 คือคณะวิศวกรรมศาสตร์
- 3) ตัวเลขหลักที่ 1-2 ของรหัสวิชา หมายถึง รหัสคณะ โดยรหัส 21 คือคณะวิศวกรรมศาสตร์
- 4) ตัวเลขหลักที่ 5 ของรหัสวิชา หมายถึง ชั้นปี โดยเลข 1 คือชั้นปีที่ 1

## 2. คำสั่งวนซ้ำแบบซับซ้อน (Advanced Iteration Statements)

1. รูปแบบของคำสั่งวนซ้ำแบบซับซ้อน ประกอบไปด้วย 2 รูปแบบคือ
  - 1) คำสั่งวนซ้ำที่ซ้อนกันหลายชั้น ในบทนี้ได้แก่อำสั่ง `for` ซ้อน `for` (Nested For) หรือคำสั่ง `while` ซ้อน `while` (Nested While) หรือคำสั่ง `for` ซ้อน `while` (Nested Loops) เป็นต้น ซึ่งได้กล่าวไปบ้างแล้วในบทที่ 4 และมีรายละเอียดเพิ่มเติมตามหัวข้อที่ 2



โจทย์ข้อที่ 6 [ระดับปานกลาง] จงแสดงผลลัพธ์จากโปรแกรมพร้อมทั้งตอบคำถามต่อไปนี้ (8 คะแนน)

```

1 public static void loopXYZ(int x, int y, int z) {
2     for (int i = x; i < 5; i++)
3         for (int j = y; j > 2 * i; --j)
4             for (int k = z; k <= 1; k++)
5                 System.out.println(i + "," + j + "," + k);
6     System.out.println("i = " + i);
7     System.out.println("j = " + j);
8     System.out.println("k = " + k);
9 }

```

```
loopXYZ(2, 7, 1);
```

จงหาจำนวนรอบทั้งหมดในการทำงาน

โจทย์ข้อที่ 7 [ระดับปานกลาง] จงเขียนเมทอด `formulaAtoB(...)` ที่สมบูรณ์เพื่อใช้ในการพิมพ์สูตรคูณ ตั้งแต่ `a` จนถึง `b` ขึ้นบนจอภาพ เช่น `formulaAtoB(4, 9)` จะพิมพ์สูตรคูณตั้งแต่ `a` 4 จนถึง `b` 9 บนจอภาพ เป็นต้น พร้อมทั้งระบุเส้นแบ่งของแต่ละแม่ด้วย (10 คะแนน)

```

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
...
4 x 11 = 44
4 x 12 = 48
-----
5 x 1 = 5
5 x 2 = 10
...

```

โจทย์ข้อที่ 8 [ระดับปานกลาง] จงเขียนเมทอด `showBarGraph(...)` ที่สมบูรณ์เพื่อวนวิ่งตัวเลขตั้งแต่ 1 จนถึงค่า `x` และตั้งแต่ค่า `y` จนถึง 1 โดยที่ตัวแปร `x` และ `y` จะรับเข้ามาทางพารามิเตอร์ ซึ่ง ณ ขณะที่ได้ค่า `x` และ `y` สองค่าใดๆ ให้แสดงกราฟแท่งที่มีความยาวเท่ากับผลคูณของสองตัวเลขนั้น ดังตัวอย่างต่อไปนี้ พร้อมทั้งเรียกใช้งานเมทอดเพื่อแสดงกราฟแท่งตามตัวอย่างทั้ง 3 แบบที่ให้มา (15 คะแนน)

```

1,2 || 2
1,1 || 1
2,2 || || 4
2,1 || | 2
3,2 || || || 6
3,1 || | 3
4,2 || || || || 8
4,1 || | 4
5,2 || || || || || 10
5,1 || || || || || 5

```

`x = 5`  
`y = 2`

```

1,4 || || || 4
1,3 || || 3
1,2 || | 2
1,1 || 1
2,4 || || || || 8
2,3 || || || 6
2,2 || || || 4
2,1 || | 2
3,4 || || || || || 12
3,3 || || || || 9
3,2 || || || 6
3,1 || | 3

```

`x = 3`  
`y = 4`

```

1,6 || || || || || 6
1,5 || || || || 5
1,4 || || || 4
1,3 || || 3
1,2 || | 2
1,1 || 1
2,6 || || || || || || 12
2,5 || || || || || 10
2,4 || || || || 8
2,3 || || || 6
2,2 || || 4
2,1 || | 2

```

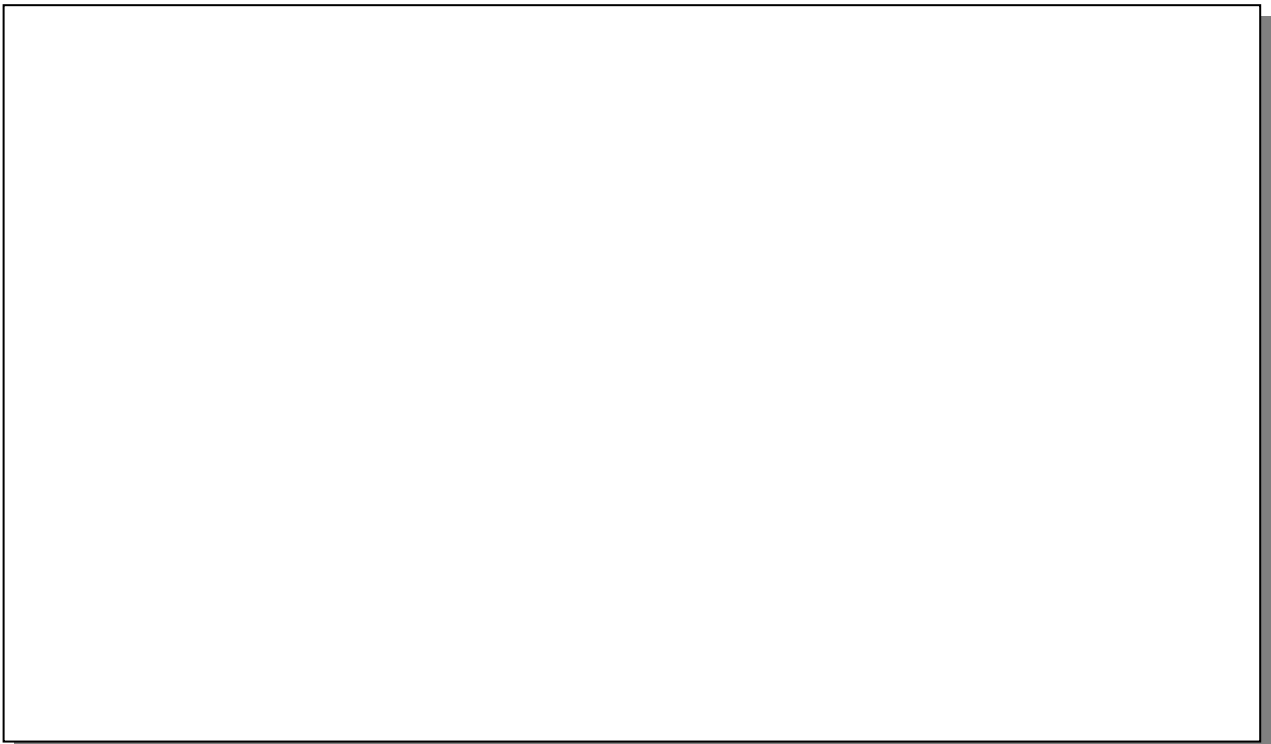
`x = 2`  
`y = 6`

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



```
} //End of main
```

```
//showBarGraph(...)
```



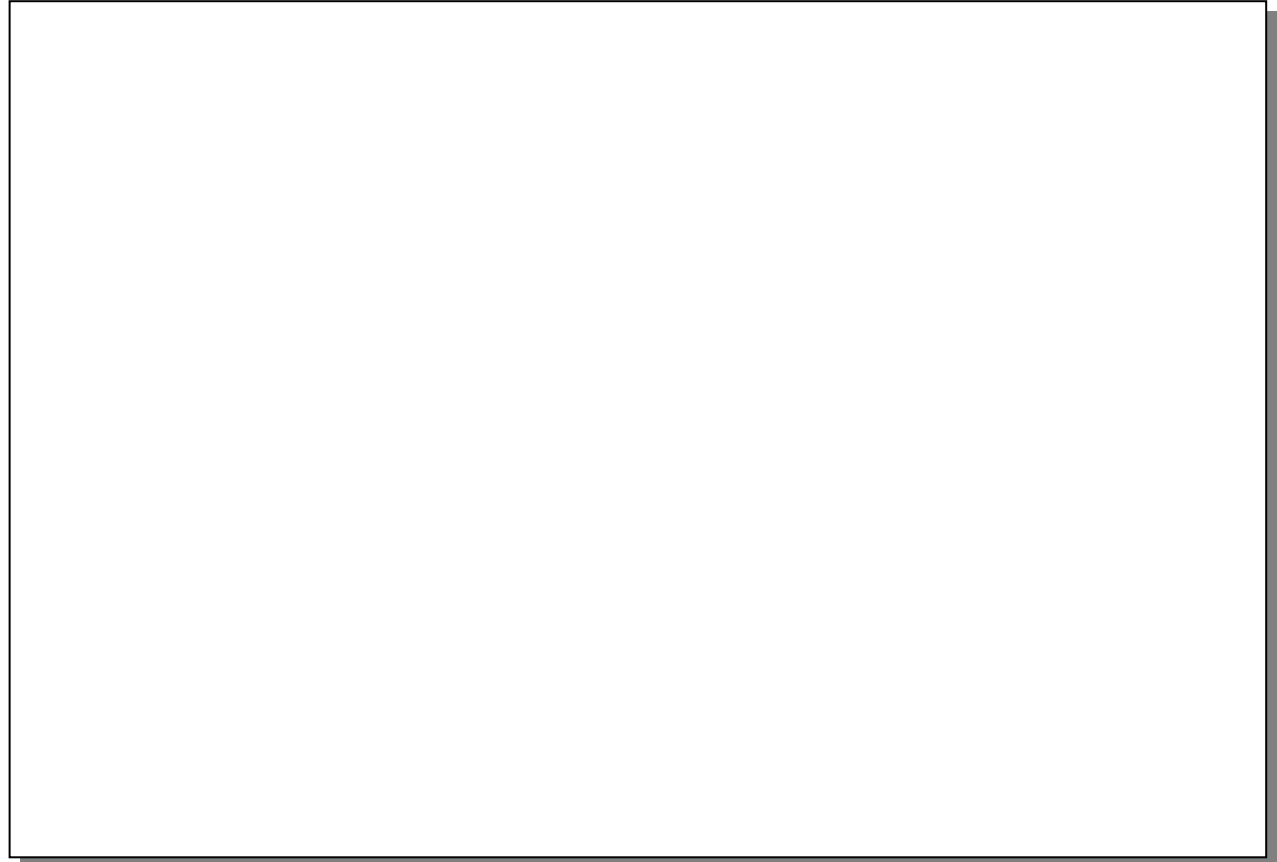
```
} //End of class
```

**โจทย์ข้อที่ 9 [ระดับยาก]** จงเขียนเมทอด `calEquation(...)` ที่สมบูรณ์เพื่อใช้สำหรับคำนวณหาคำตอบของสมการ  $s = 17x^9yz^8 + 15x^8y^2z^7 + 13x^7y^3z^6 + 11x^6y^4z^5 + \dots + xy^9z^0$  แล้วคืนค่า `s` ที่เป็นคำตอบในทุกๆ รอบกลับมา โดยมีข้อกำหนดดังต่อไปนี้

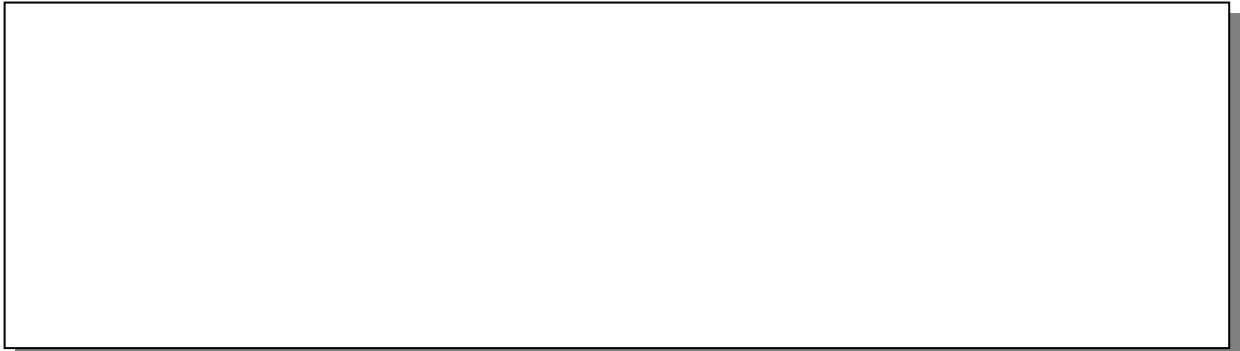
- 1) ค่า `x` เริ่มต้นที่ 1 จนถึง `a` โดยค่าจะเพิ่มขึ้นครั้งละ 1 โดยที่ `a` เป็นค่าที่รับมาจากพารามิเตอร์
- 2) ค่า `y` เริ่มต้นที่ 5 จนถึง `b` โดยค่าจะเพิ่มขึ้นครั้งละ 5 โดยที่ `b` เป็นค่าที่รับมาจากพารามิเตอร์
- 3) ค่า `z` เริ่มต้นที่ 12 จนถึง `c` โดยค่าจะลดลงครั้งละ 2 โดยที่ `c` เป็นค่าที่รับมาจากพารามิเตอร์

ในการประมวลผล กำหนดให้ `x` ถูกพิจารณาก่อน `y` และ `y` ถูกพิจารณาก่อน `z` โดยจะวนทำงานซ้ำกันเป็นชั้นๆ และให้เรียกใช้เมทอดนี้ที่เมทอด `main(...)` โดยให้ค่าของพารามิเตอร์ทุกตัวเป็น 10 พร้อมทั้งแสดงผลลัพธ์ที่ได้ขึ้นบนจอภาพให้สวยงาม (15 คะแนน)

```
import java.util.Scanner;
public class TheEquation {
    //calEquation(...)
```



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



```
} //End of main
} //End of class
```

### 3. คำสั่งวนซ้ำแบบหลายตัวควบคุม

#### 1) คำสั่ง for แบบหลายตัวควบคุมและหลายเงื่อนไข

```
1  for (Initial1, Initial2, Initial3, ...;
2      Condition1 && Condition2 || Condition3 && ...;
3      Update1, Update2, Update3, ...) {
4      :
5      :
6  }
```

เรียก Initial1 ว่าตัวควบคุมตัวที่ 1

Initial แต่ละตัวคั่นด้วย Comma, Condition แต่ละตัวเชื่อมด้วย Logical Operators, Update แต่ละตัวคั่นด้วย Comma



## 2) คำสั่ง while แบบหลายตัวควบคุมและหลายเงื่อนไข

```

1  Initial1, Initial2, Initial3, ...;
2  while (Condition1 && Condition2 || Condition3 && ...) {
3      :
4      :
5      Update1; Update2; Update3; ...;
6  }

```

Initial แต่ละตัวคั่นด้วย Comma, Condition แต่ละตัวเชื่อมด้วย Logical Operators, Update แต่ละตัวคั่นด้วย Semicolon

โจทย์ข้อที่ 10 [ระดับง่าย] จงหาผลลัพธ์จากส่วนของโปรแกรมและตอบคำถามต่อไปนี้ (8 คะแนน)

```

1  public static void iterABC(int a, int b, int c) {
2      while (a <= 5 && b > 3 || c >= 0) {
3          System.out.println(a + "\t" + b + "\t" + c);
4          a++;
5          b--;
6          if ((a + b) % 2 == 0) c--;
7      }
8  }

```

```
iterABC(1, 15, 2);
```

จงหาจำนวนรอบทั้งหมดในการทำงาน

โจทย์ข้อที่ 11 [ระดับง่าย] จงหาผลลัพธ์จากส่วนของโปรแกรมและตอบคำถามต่อไปนี้ (8 คะแนน)

```

1  for (int i = 0, j = 10; i < 10 && j > 0; i++, j--) {
2      if (i == j / 2) break;
3      System.out.println(i);
4      if (j % 2 != 0) continue;
5      System.out.println(j);
6      if (i * 4 == j)
7          System.out.println(i + "," + j);
8  }

```

คำสั่ง continue เป็น  
คำสั่งที่วกกลับขึ้นไปยัง  
ต้น Loop

จงหาจำนวนรอบทั้งหมดในการทำงาน

โจทย์ข้อที่ 12 [ระดับยาก] จงเขียนเมทอด toStringCalculation(...) ที่สมบรูณ์เพื่อรับค่าตัวแปร a, b และ c ที่เป็นจำนวนเต็มเข้ามาทางพารามิเตอร์ เพื่อใช้เป็นค่าเริ่มต้นของตัวแปรควบคุมวงวน (Loop) x, y และ z ตามลำดับ พร้อมทั้งรับค่าตัวแปร p, q และ r อีกหนึ่งโดยการเรียกใช้เมทอด inputNumber(...) เพื่อนำมาเป็นค่าที่เพิ่มขึ้นในแต่ละรอบของตัวแปรควบคุมข้างต้นตามลำดับ โดยค่าทั้งหมดนี้ถูกใช้เพื่อคำนวณหาคำตอบของสมการ  $s = 10x^{10}yz^{10} + 9x^9y^2z^{12} + 8x^8y^3z^{14} + 7x^7y^4z^{16} + \dots + xy^{10}z^{28}$  และกำหนดให้ใช้คำสั่ง for แบบ 3 ตัวควบคุม x, y และ z โดยจะมีการหยุดรอบเมื่อ  $x + y + z$  มากกว่าหรือเท่ากับ 1000 และให้คืนค่าผลลัพธ์ของ x, y, z และ s ในแต่ละรอบให้ครบถ้วน (15 คะแนน)

```
import java.util.Scanner;
import java.util.io.*;
public class Calculation {
    //toStringCalculation(...)
```

```
public static int inputNumber(String s) {
    Scanner kb = new Scanner(System.in);
    System.out.print("Enter " + s + ": ");
    return kb.nextInt();
}
```

//เรียกใช้เมทอด toStringCalculation(...) ที่เมทอด main(...) ให้ถูกต้องสมบูรณ์โดยกำหนดให้ค่าของพารามิเตอร์ทั้งสามเป็น 10, 20 และ 30 ตามลำดับ

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

```
    } //End of main
} //End of class
```

### 3. การประยุกต์ใช้คำสั่งแบบซับซ้อน (Applications of Advanced Statements)

- 1) คำสั่งตัดสินใจแบบซับซ้อนและคำสั่งวนซ้ำแบบซับซ้อนสามารถทำงานร่วมกันได้ เช่น คำสั่ง for ซ้อน for หรือ for ซ้อน while ที่ภายในอาจมีคำสั่ง if-else ซ้อน if-else แทรกอยู่ เป็นต้น

```

1  for (Initial1; Condition1; Update1) {
2      for (Initial2; Condition2; Update2) {
3          if (Condition3) Statement;
4          else if (Condition4) Statement;
5          else Statement;
6      }
7  }

```

มีคำสั่ง for ซ้อน for ทั้งหมด 2 ชั้น  
ซึ่งภายในคำสั่ง for ชั้นในสุดมีคำสั่ง  
if-else ซ้อน if-else แทรกอยู่  
อีกชั้นหนึ่ง

- 2) โปรแกรมประยุกต์ส่วนใหญ่หลีกเลี่ยงไม่พ้นที่จะใช้คำสั่งเหล่านี้ในการประมวลผล ดังนั้นเราจึงควรฝึกเขียนคำสั่งแบบซับซ้อนเหล่านี้ให้เกิดความชำนาญให้มากที่สุด

**โจทย์ข้อที่ 13** [ระดับง่าย - ระดับยาก] จงเขียนเมทอด `starxx(...)` ให้สมบุรณ์โดยรับจำนวนเต็มหนึ่งค่า มาทางพารามิเตอร์เพื่อแสดงรูปเลขาชนิดต่างๆ ตามที่โจทย์กำหนด โดยอนุญาตให้ใช้เพียงคำสั่งต่อไปนี้เท่านั้นในการแสดงผล ได้แก่คำสั่ง `system.out.print("*");` คำสั่ง `system.out.println("*");` คำสั่ง `system.out.print(" ");` และคำสั่ง `system.out.println();` (80 คะแนน : ข้อละ 8 คะแนน)

```

public class Shape {
    public static void starA(int n) { [ระดับง่าย]

```

```

starA(11)
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

```

```

} //End of method

```

```

public static void starB(int n) { [ระดับง่าย]

```

```

starB(11)
*
**
***
****
*****
*****
*****
*****
*****
*****
*****

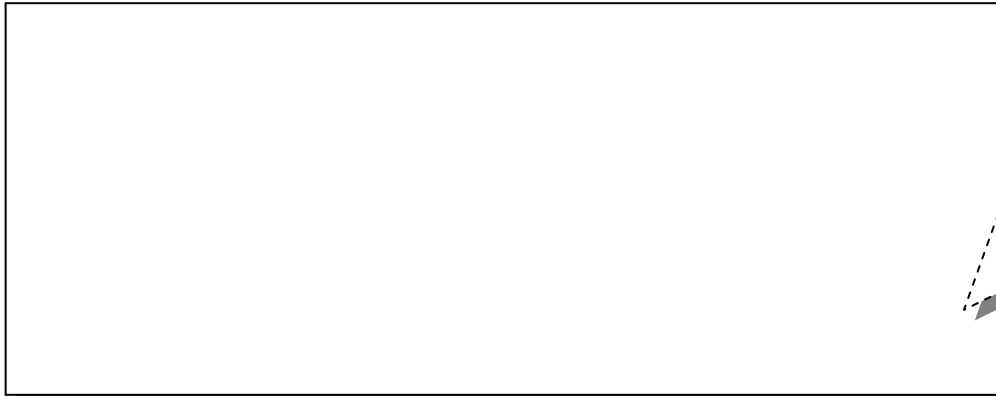
```

```

} //End of method

```

```
public static void starC(int n) {   [ระดับง่าย]
```



```
starC(11)
*****
*****
*****
*****
*****
*****
*****
****
***
**
*
```

```
} //End of method
```

```
public static void starD(int n) {   [ระดับปานกลาง]
```



```
starD(11)
      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****
*****
```

```
} //End of method
```

```
public static void starE(int n) {   [ระดับปานกลาง]
```



```
starE(11)
*****
*****
*****
*****
*****
*****
*****
****
***
**
*
```

```
} //End of method
```

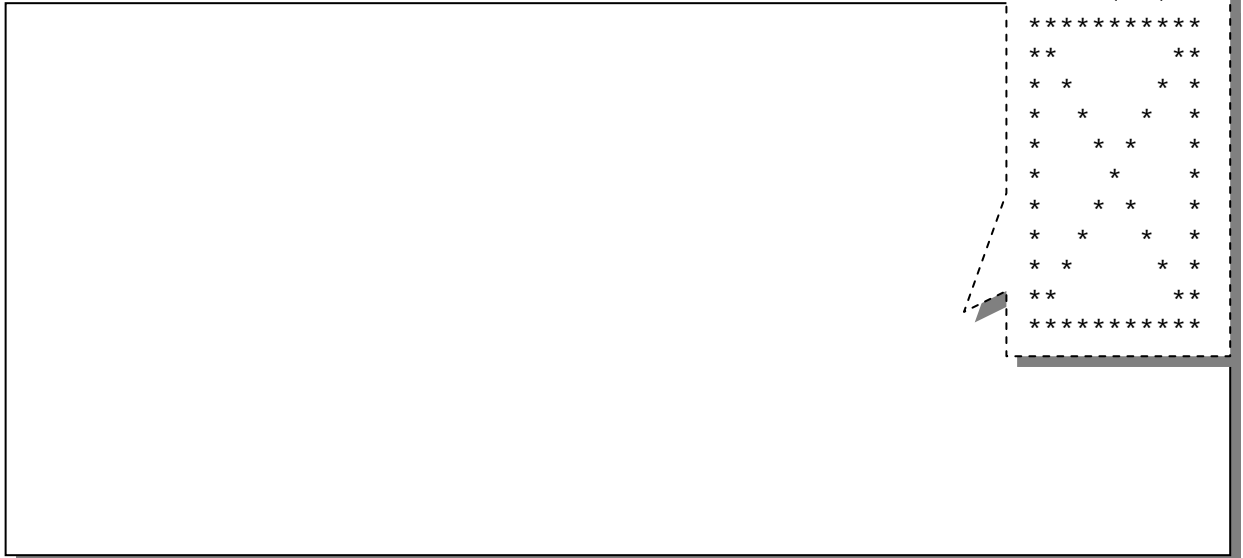
```
public static void starF(int n) {   [ระดับปานกลาง]
```



```
starF(11)
*****
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
*****
```

```
} //End of method
```

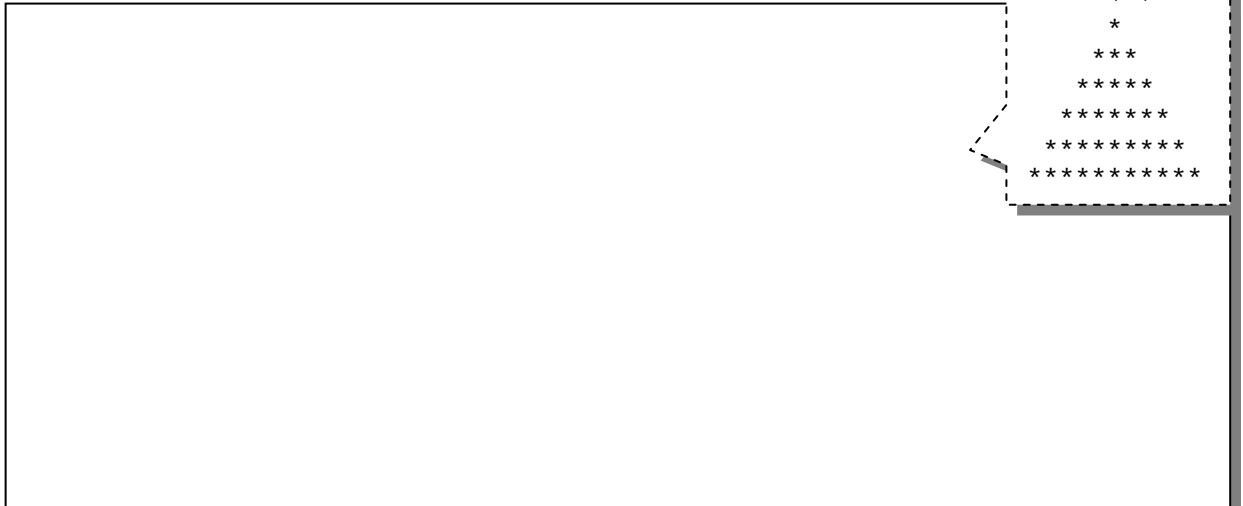
```
public static void starG(int n) {   [ระดับปานกลาง]
```



```
starG(11)
*****
**          **
* *        * *
*  *      *  *
*   *    *   *
*    *  *    *
*     * *     *
*      * *      *
*       * *       *
*        * *        *
*****
```

```
} //End of method
```

```
public static void starH(int n) {   [ระดับยาก]
```



```
starH(6)
  *
 ***
*****
*****
*****
*****
```

```
} //End of method
```

```
public static void starI(int n) {   [ระดับยาก]
```

```
starI(11)
*****
*****
*****
*****
***
*
***
*****
*****
*****
*****
```

```
} //End of method
```

```
public static void starJ(int n) {   [ระดับยาก]
```

```
starJ(11)
*           *
**          **
***         ***
****        ****
*****       *****
*****       *****
*****       *****
****        ****
***         ***
**          **
*           *
```

```
} //End of method
```

//เรียกใช้เมื่อกดต่างๆ ก่อนหน้านี้ โดยเลือกเพียง 4 เมื่อกดเท่านั้น (0 คะแนน ถ้าไม่ทำ -5 คะแนน)

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

```
} //End of main
```

```
} //End of class
```

**โจทย์ข้อที่ 14 [ระดับยาก]** จงเขียนเมทอด `revWords(...)` ที่สมบูรณ์เพื่อเปิดอ่านแฟ้มข้อมูลตามชื่อแฟ้มที่ระบุจากพารามิเตอร์ โดยจะอ่านประโยคทีละบรรทัดแล้วกลับตัวอักษรภายในคำ (Words) ใดๆ ทุกคำของประโยคนั้น โดยไม่ต้องสลับตำแหน่งของคำ เช่น "we love you" จะได้เป็น "ew evol uoy" เป็นต้น พร้อมทั้งแสดงผลลัพธ์ที่ได้ออกทางจอภาพ (15 คะแนน)

**message.txt**

```
I Love          Java Programming
This is My First Program
And It also is my  LAST Program
...
```

**ตัวอย่างการแสดงผลบนจอภาพ**

```
I evol          avaJ gnimmargorP
sihT si yM tsriF margorP
dnA tI osla si ym  TSAL margorP
...
```

**โจทย์ข้อที่ 15 [ระดับเทพ]** จงเขียนโปรแกรมภาษาจาวาให้สมบูรณ์ เพื่อสร้างเมทอดที่ใช้สำหรับประมวลผลและดำเนินการเกี่ยวกับอาร์เรย์ที่มีสมาชิกทุกตัวเป็นตัวเลขจำนวนเต็ม ตามรายละเอียดของเมทอดดังต่อไปนี้ โดยกำหนดให้เมทอดแต่ละเมทอดสามารถเรียกใช้งานซึ่งกันและกันได้ (50 คะแนน)

```
import java.util.Scanner;
public class OperationsOfArrays {
    //จงเขียนเมทอด removeDuplicatedMembers(...) เพื่อรับอาร์เรย์หนึ่งตัวใดๆ จากพารามิเตอร์แล้ว
    //ทำการตัดสมาชิกตัวที่ซ้ำกันออกจากอาร์เรย์ แล้วคืนค่าอาร์เรย์ตัวใหม่นั้น (10 คะแนน)
```

//จงเขียนเมทอด `unionArray(...)` เพื่อรับอาร์เรย์สองตัวใดๆ จากพารามิเตอร์แล้วทำการหาค่าสมาชิกที่ปรากฏอยู่ในอาร์เรย์ตัวที่หนึ่งหรืออาร์เรย์ตัวที่สอง (ปรากฏที่อาร์เรย์ใดอาร์เรย์หนึ่งหรือทั้งสองอาร์เรย์ก็ได้) เก็บไว้ในอาร์เรย์ตัวใหม่ แล้วคืนค่าอาร์เรย์ตัวใหม่นั้น (10 คะแนน)

//จงเขียนเมทอด `intersectArray(...)` เพื่อรับอาร์เรย์สองตัวใดๆ จากพารามิเตอร์แล้วทำการหาค่าสมาชิกที่ปรากฏอยู่ในอาร์เรย์ตัวที่หนึ่งและอาร์เรย์ตัวที่สอง (ต้องปรากฏอยู่ทั้งสองอาร์เรย์) เก็บไว้ในอาร์เรย์ตัวใหม่ แล้วคืนค่าอาร์เรย์ตัวใหม่นั้น (10 คะแนน)



//จงเขียนเมทอด complementArray(...) เพื่อรับอาร์เรย์สองตัวใดๆ จากพารามิเตอร์แล้วทำการหาค่าสมาชิกที่ปรากฏอยู่ในอาร์เรย์ตัวที่หนึ่งเท่านั้น (โดยจะไม่ปรากฏอยู่ในอาร์เรย์ตัวที่สอง) เก็บไว้ในอาร์เรย์ตัวใหม่ แล้วคืนค่าอาร์เรย์ตัวใหม่นั้น (10 คะแนน)

//จงเขียนเมทอด `printArray(...)` เพื่อรับอาร์เรย์หนึ่งตัวใดๆ จากพารามิเตอร์แล้วทำการแสดงค่าสมาชิกทุกตัวของอาร์เรย์นั้น โดยให้สมาชิกแต่ละตัวคั่นด้วยเครื่องหมายจุลภาค (Comma) และสมาชิกทุกตัวจะต้องอยู่ภายในเครื่องหมายปีกกาเปิด-ปิด ดังตัวอย่างต่อไปนี้ { 1, 2, 3, 4, 5 } หรือ { 5, 12, 7, 0 } เป็นต้น (5 คะแนน)

```
public static void main(String[] args) {
    int a[] = { 7, 7, 2, 4, 5, 6, 5, 3, 2, 3, 0, 10};
    int b[] = { 2, 3, 4, 2, 1, 1, 4, 5, 6, 5, 3, 2, 3 };
    //จงเรียกใช้เมทอดทุกเมทอดที่เขียนขึ้นตามรายละเอียดดังต่อไปนี้ (5 คะแนน)
```

- (1) แสดงสมาชิกของอาร์เรย์ a และอาร์เรย์ b ก่อนการประมวลผล
- (2) แสดงสมาชิกของอาร์เรย์ a และอาร์เรย์ b หลังจากตัดสมาชิกตัวที่ซ้ำออกไปแล้ว
- (3) แสดงผลลัพธ์จากการ Union ของอาร์เรย์ a และอาร์เรย์ b
- (4) แสดงผลลัพธ์จากการ Intersect ของอาร์เรย์ a และอาร์เรย์ b
- (5) แสดงผลลัพธ์จากการ Complement ของอาร์เรย์ a และอาร์เรย์ b

```
} //End of main
} //End of class
```