CHAPTER05

การเขียนโปรแกรมแบบวนซ้ำ (Iterative Programming)

บทนำ

- เราได้ทำการศึกษาการเขียนโปรแกรมแบบมีเงื่อนไขไปแล้วใน chapter04
- รูปแบบของการเขียนโปรแกรมยังมีอีกหนึ่งรูปแบบที่มีความสำคัญไม่น้อยไปกว่าการเขียนโปรแกรมแบบเรียงลำดับ และโปรแกรมแบบมีเงื่อนไข นั่นก็คือการเขียนโปรแกรมแบบวนซ้ำ
- ใน chapter05 นี้จะได้ศึกษาคำสั่งต่างๆ ที่ใช้สำหรับการเขียนโปรแกรมแบบวนซ้ำ

คำสั่งวนซ้ำ (iterative statements)

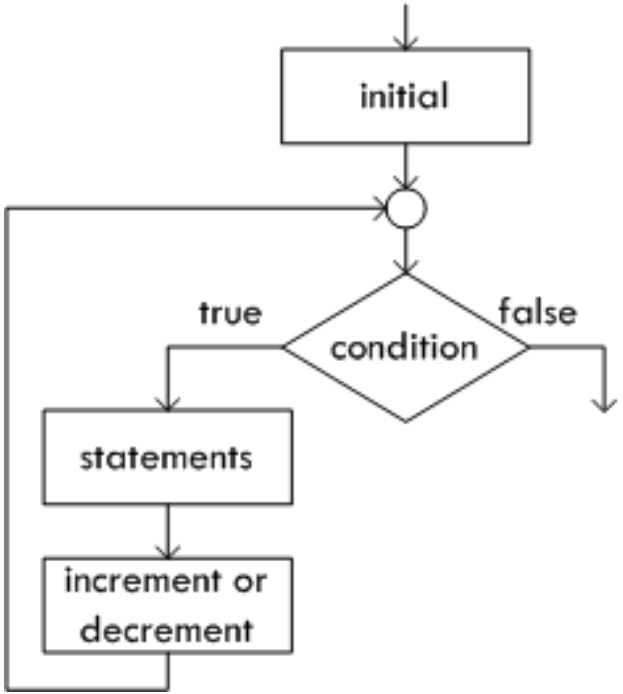
- คำสั่งวนซ้ำในภาษาจาวามีด้วยกัน 3 คำสั่ง ดังนี้
- 1. คำสั่ง for()
- 2. คำสั่ง while()
- 3. คำสั่ง do while()

1. คำสั่ง for()

- รูปแบบคำสั่ง

```
for(initial;condition;increment or decrement) {
    statements;
}
```

- ผังงาน



การทำงานของคำสั่ง for()

- 1. กำหนดค่าเริ่มต้น เช่น a=1 (initial)
- 2. ตรวจสอบเงื่อนไข (condition)
  - ถ้าเงื่อนไขเป็นจริง (true) ไปทำข้อที่ 3
  - ถ้าเงื่อนไขเป็นเท็จ (false) ไปทำข้อที่ 6
- 3. ทำงานตามคำสั่งที่ต้องการทำซ้ำ (statements)
- 4. ทำการเพิ่มค่าหรือลดค่าของค่าเริ่มต้น (increment or decrement)
- 5. กลับไปทำข้อที่ 2
- 6. จบการทำงานของคำสั่ง for()

ตัวอย่างการใช้งานคำสั่ง for()

- **ตัวอย่างที่ 1** เขียนโปรแกรมหาผลรวมของตัวเลข 1-10 โดยแสดงผลลัพธ์ทางจอภาพ

กำหนดตัวแปร

c	แทน	ตัวนับรอบการวนซ้ำ
sum	แทน	ผลรวมของตัวเลข

ตัวอย่างการใช้งานคำสั่ง for()

อัลกอริทึม

1. กำหนดตัวแปร **c, sum**
2. กำหนดค่าเริ่มต้นให้ตัวแปร **sum=0, c=1**
3. ทดสอบเงื่อนไข **c<=10**
    - 3.1 ถ้าจริง ไปทำข้อ 4
    - 3.2 ถ้าเท็จ ไปทำข้อ 7

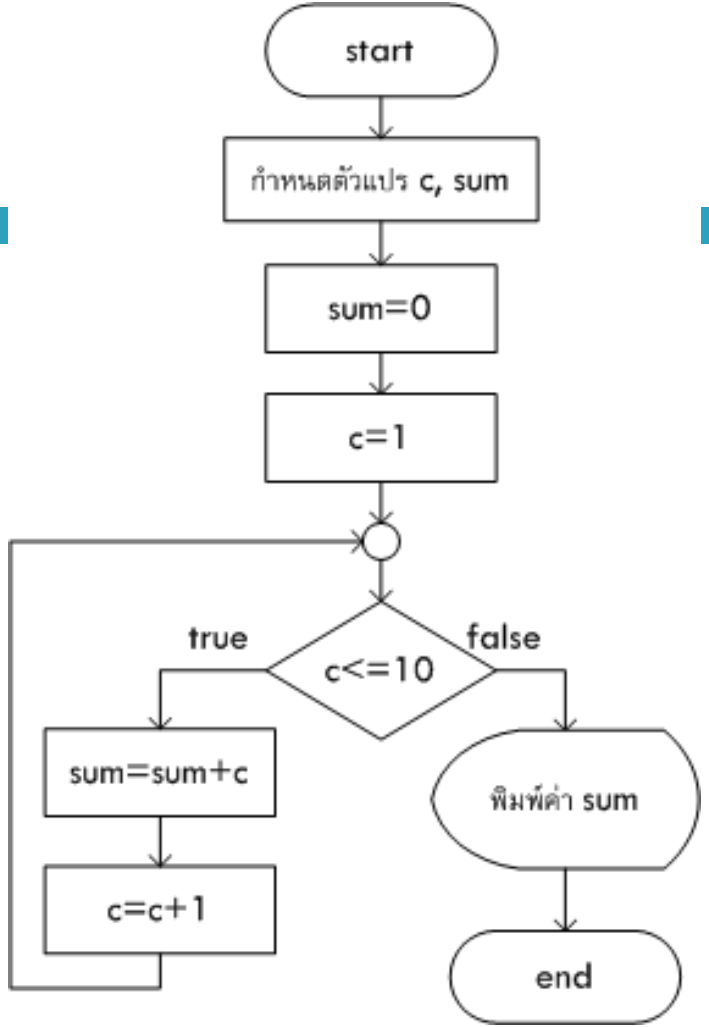
ตัวอย่างการใช้งานคำสั่ง for()

อัลกอริทึม

4. กำหนดค่า sum = sum+c
5. กำหนดค่า c = c+1
6. กลับไปทำข้อ 3
7. แสดงค่า sum
8. จบการทำงาน

ตัวอย่าง

- ผังงาน



ตัวอย่างการใช้งานคำสั่ง for()

- โค้ด

Summation.java

```
1  public class Summation {
2      public static void main(String[] args) {
3          int c,sum;
4          sum = 0;
5          for(c=1;c<=10;c=c+1){
6              sum = sum+c;
7          }
8          System.out.println("ผลรวมของเลข 1-10 คือ " + sum);
9      }
10 }
11
```

ผลลัพธ์

Javadoc | Problems | Declaration | Console

```
ผลรวมของเลข 1-10 คือ 55
```

ตัวอย่างการใช้งานคำสั่ง for()

- จากโปรแกรมที่ผ่านมาการใช้คำสั่ง **for()** ในส่วนของการเพิ่มค่า หรือลดค่า นั้นเราใช้การเพิ่มค่า **(c=c+1)** เป็นการเพิ่มค่า **c** ทีละ 1
- เราสามารถปรับปรุงโปรแกรมนี้โดยใช้การลดค่า **(c=c-1)**
- ให้นักศึกษาเปลี่ยนคำสั่งในบรรทัดที่ 8 เป็น ดังนี้

```
for(c=10;c>=1;c=c-1)
```

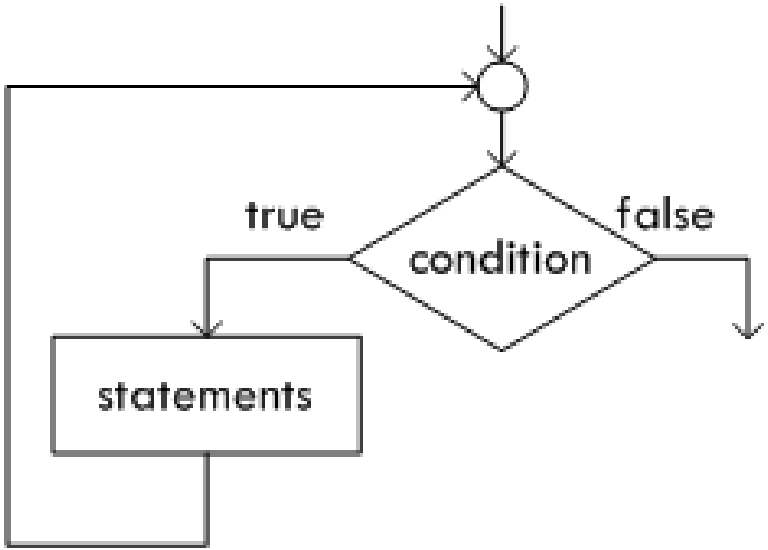
- สังเกตผลรัน เหมือนหรือต่างกันอย่างไร?

2. คำสั่ง while()

- รูปแบบคำสั่ง

```
while(condition) {
    statements;
}
```

- ผังงาน



การทำงานคำสั่ง while()

- 1. ตรวจสอบเงื่อนไข (condition)
  - ถ้าจริง ไปทำข้อ 2
  - ถ้าเท็จ ไปทำข้อ 4
- 2. ทำคำสั่งที่ต้องการทำซ้ำ
- 3. กลับไปทำข้อ 1 ใหม่
- 4. ออกจากคำสั่ง while()

ตัวอย่างการใช้งานคำสั่ง while()

- **ตัวอย่างที่ 2** ให้ปรับปรุงโปรแกรมในตัวอย่างที่ 1 โดยเปลี่ยนคำสั่งวนซ้ำจาก for() มาใช้เป็นคำสั่ง while()

ข้อสังเกต จากผังงานของคำสั่ง for() และคำสั่ง while() มีลักษณะการทำงานที่คล้ายกันก็คือ จะมีการตรวจสอบเงื่อนไขก่อนเสมอจึงจะทำงานในลูป (loop) โดยเงื่อนไขจะต้องเป็นจริงเท่านั้นจึงจะทำงานในลูป แต่ถ้าเงื่อนไขเป็นเท็จ ก็จะไม่ทำงานในลูป

ดังนั้นอัลกอริทึม และผังงานจึงมีการทำงานเหมือนกัน

ตัวอย่างการใช้งานคำสั่ง while()

- โค้ด

Summation.java

```
1  public class Summation {
2      public static void main(String[] args) {
3          int c,sum;
4          sum = 0;
5          c = 1;
6          while(c<=10){
7              sum = sum + c;
8              c = c + 1; //ใช้คำสั่ง c++ หรือ ++c แทนก็ได้
9          }
10         System.out.println("ผลรวมของเลข 1-10 คือ " + sum);
11     }
12 }
13
```

ผลลัพธ์

Javadoc | Problems | Declaration | Console

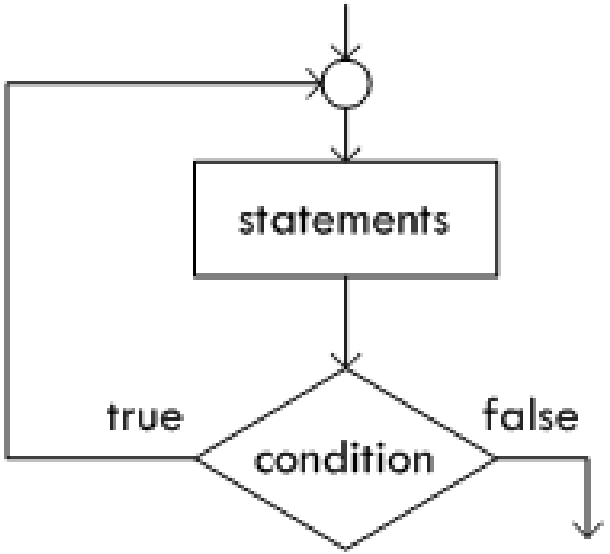
```
ผลรวมของเลข 1-10 คือ 55
```

3. คำสั่ง do while()

- รูปแบบคำสั่ง

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

- ผังงาน



การทำงานคำสั่ง do while()

- 1. ทำคำสั่งที่ต้องการวนซ้ำ (statements)
- 2. ตรวจสอบเงื่อนไข (condition)
  - 2.1 ถ้าจริง ไปทำข้อ 1
  - 2.2 ถ้าเท็จ ไปทำข้อ 3
- 3. จบการทำงาน

ตัวอย่างการใช้งานคำสั่ง do while()

- **ตัวอย่างที่ 3** เขียนโปรแกรมรับชื่อ และคะแนนสอบของนักเรียนทางแป้นพิมพ์ มีเงื่อนไขคือ หากคะแนนสอบมีค่าน้อยกว่า 60 ถือว่าไม่ผ่านเกณฑ์ ทำงานจนกว่าผู้ใช้โปรแกรมไม่ต้องการทำงานต่อ ซึ่งผู้ใช้จะต้องป้อนตัวอักษร **“N”** หรือ **“n”** จึงออกจากโปรแกรม

ตัวอย่างการใช้งานคำสั่ง do while()

กำหนดตัวแปร

stdName	แทน	ชื่อนักเรียน
score	แทน	คะแนนสอบ
result	แทน	ผลสอบ
cont	แทน	ความต้องการทำงานต่อ
	“Y” หรือ “y”	ต้องการทำงานต่อ
	“N” หรือ “n”	ไม่ต้องการทำงานต่อ

ตัวอย่างการใช้งานคำสั่ง do while()

อัลกอริทึม

1. กำหนดตัวแปร **stdName, score, result, cont**
2. กำหนดค่าเริ่มต้นให้ **stdName** = “” และ **cont** = “Y”
3. รับค่าชื่อนักเรียน และคะแนนสอบ
4. ทดสอบคะแนนสอบ < 60 เป็นจริง หรือเท็จ
    - 4.1 ถ้าจริง ให้ **result** = “ไม่ผ่านเกณฑ์” แล้วไปทำข้อ 5
    - 4.2 ถ้าเท็จ ให้ **result** = “ผ่านเกณฑ์” แล้วไปทำข้อ 5

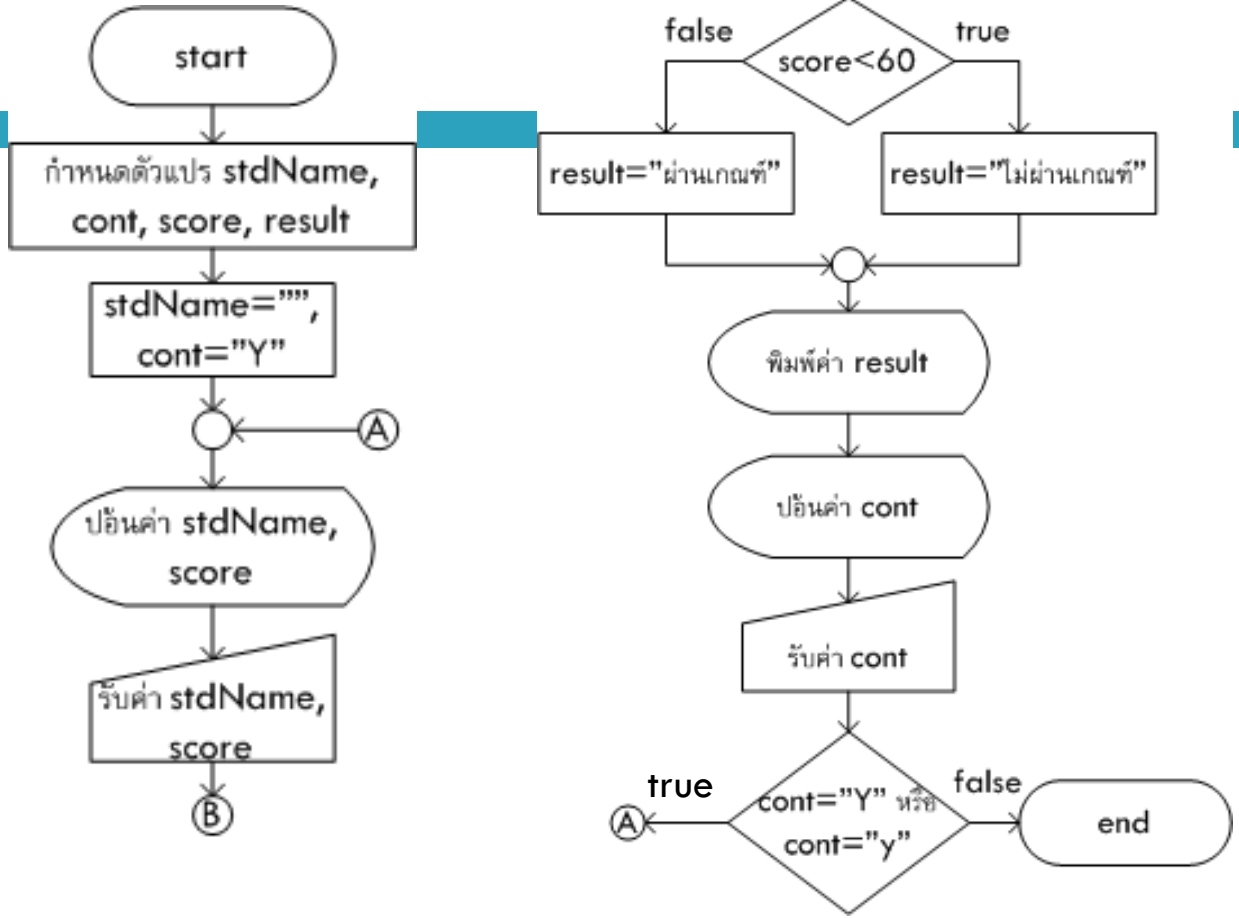
ตัวอย่างการใช้งานคำสั่ง do while()

อัลกอริทึม

5. พิมพ์ค่าผลสอบ
6. รับค่า **cont**
7. ทดสอบเงื่อนไข **cont** = “Y” หรือ “y”
    - 7.1 ถ้าจริง ให้ไปทำข้อ 3
    - 7.2 ถ้าเท็จ ให้ไปทำข้อ 8
8. จบการทำงาน

ตัวอย่าง

- ผังงาน



ตัวอย่าง

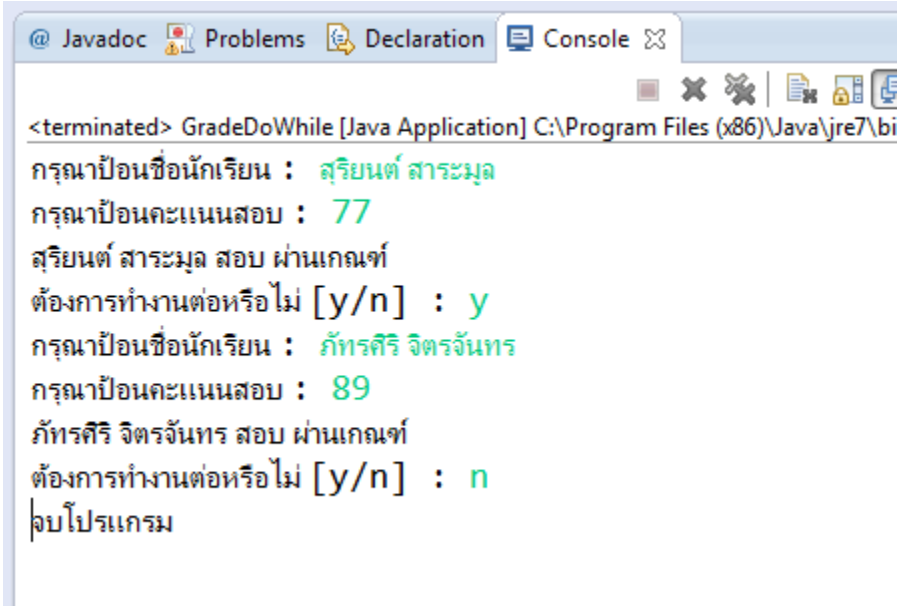
- โค้ด

GradeDoWhile.java

```java
import java.util.Scanner;
public class GradeDoWhile {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String stdName = "",cont = "Y",result;
        int score;
        do{
            System.out.print("กรุณาป้อนชื่อนักเรียน : ");
            stdName = sc.nextLine();
            System.out.print("กรุณาป้อนคะแนนสอบ : ");
            score = sc.nextInt();
            if(score<60)
                result = "ไม่ผ่านเกณฑ์";
            else
                result = "ผ่านเกณฑ์";
            System.out.println(stdName + " สอบ " + result);
            System.out.print("ต้องการทำงานต่อหรือไม่ [y/n] : ");
            cont = sc.next();
            sc.nextLine();
        }while((cont.equals("Y")||cont.equals("y")));
        System.out.println("จบโปรแกรม");
    }
}
```

ตัวอย่างการใช้งานคำสั่ง do while()

- ผลลัพธ์



Nested Loop

- **nested loop** คือโปรแกรมที่มีคำสั่งวนซ้ำแบบซ้อนกันอยู่ภายใน
- ตัวอย่างเช่น

```
1  public class ExampleNestedLoop {
2      public static void main(String[] args) {
3          int x,y,result;
4          result=0;
5          x=1;
6          while(x<=10){
7              y=1;
8              while(y<=5){
9                  result=result+y;
10                 y++;
11             }
12             x++;
13         }
14         System.out.println("Result : " + result);
15     }
16 }
```

Nested Loop

ตัวอย่างการใช้งาน nested loop

- **ตัวอย่างที่ 4** โปรแกรมสั่งพิมพ์เครื่องหมายดอกจัน (*) ตามจำนวนคอลัมน์ และจำนวนแถวที่รับเข้ามาทางแป้นพิมพ์

กำหนดตัวแปร

col	แทน	จำนวนคอลัมน์
row	แทน	จำนวนแถว
x	แทน	จำนวนวนรอบแถว
y	แทน	จำนวนวนรอบคอลัมน์

ตัวอย่างการใช้งาน nested loop

อัลกอริทึม

1. กำหนดตัวแปร col, row, x, y
2. รับค่าจำนวนคอลัมน์ และจำนวนแถว
3. กำหนดค่าเริ่มต้น x=0
4. ทดสอบค่า x<row
    - 4.1 ถ้าเป็นจริง ให้ไปทำข้อ 5 และ 6
    - 4.2 ถ้าเป็นเท็จ ให้ไปทำข้อ 7

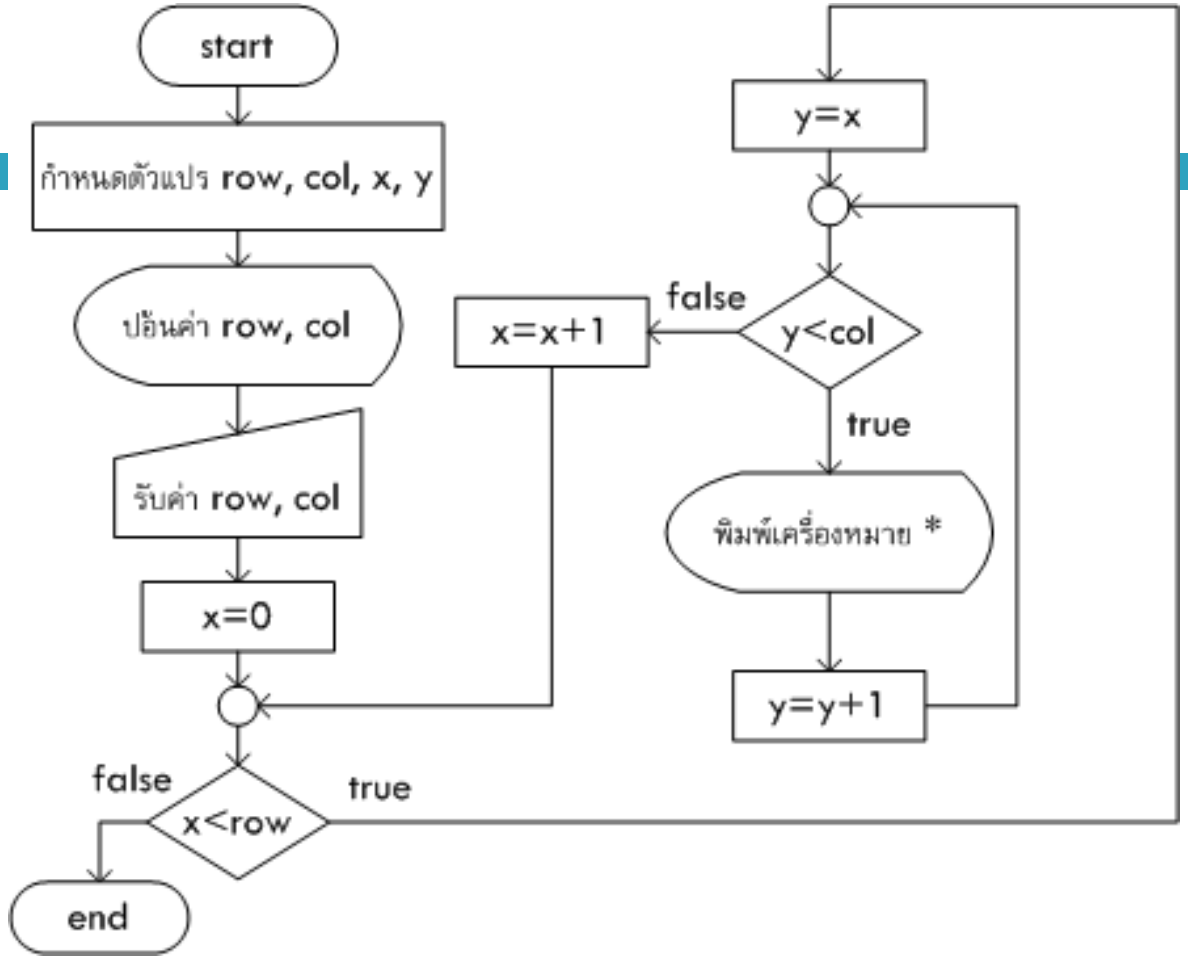
ตัวอย่างการใช้งาน nested loop

อัลกอริทึม

5. กำหนดค่าเริ่มต้น y=x
6. ทดสอบค่า y<col
    - 6.1 ถ้าเป็นจริง พิมพ์ดอกจัน แล้วบวกค่า y ทีละ 1 แล้วไปทำข้อ 6
    - 6.2 ถ้าเป็นเท็จ บวกค่า x ทีละ 1 แล้วไปทำข้อ 4
7. จบการทำงาน

ตัวอย่าง

- ผังงาน



ตัวอย่าง

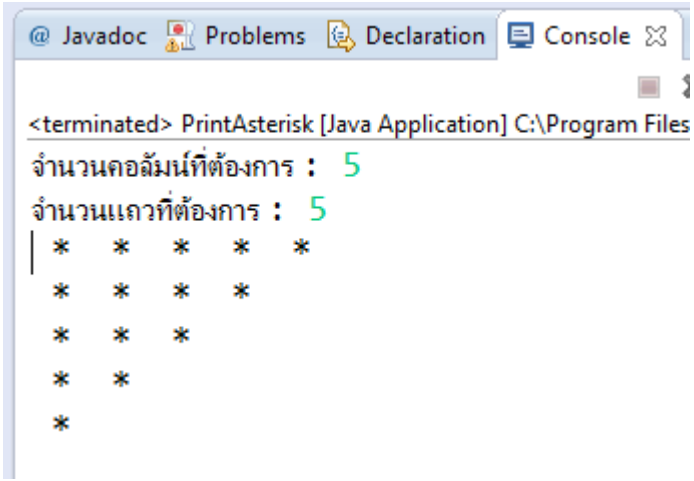
- โค้ด

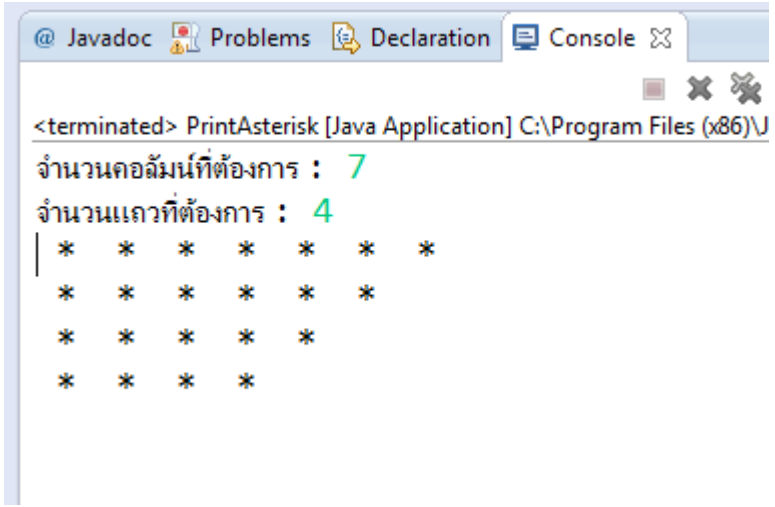
PrintAsterisk.java

```java
import java.util.Scanner;
public class PrintAsterisk {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int col,row,x,y;
        System.out.print("จำนวนคอลัมน์ที่ต้องการ : ");
        col = sc.nextInt();
        System.out.print("จำนวนแถวที่ต้องการ : ");
        row = sc.nextInt();
        for(x=0;x<row;x++){
            for(y=x;y<col;y++){
                System.out.print(" * ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

ตัวอย่างการใช้งาน nested loop

- ผลลัพธ์





คำสั่ง break และ continue

- คำสั่ง **break** เป็นการสั่งให้ ออกจากคำสั่งเงื่อนไข หรือออกจากคำสั่งวนซ้ำ หรือวนลูป โดยหยุดการตรวจสอบเงื่อนไขถัดไป
- คำสั่ง **continue** เป็นการสั่งให้ วนซ้ำ หรือวนลูปต่อไป โดยไม่ทำคำสั่งที่เหลือในลูปนั้น

ตัวอย่างการใช้คำสั่ง break

BreakStatement.java

```
1  public class BreakStatement {
2      public static void main(String[] args) {
3          int i,sum = 0;
4          for(i=1;i<=5;i++){
5              if(i==3){
6                  break;
7              }
8              sum=sum+i;
9          }
10         System.out.println("The sum is " + sum);
11     }
12 }
13
```

Javadoc | Problems | Declaration | Console

```
The sum is 3
```

ตัวอย่างการใช้คำสั่ง continue

ContinueStatment.java

```
1  public class ContinueStatment {
2      public static void main(String[] args) {
3          int i,sum=0;
4          for(i=1;i<=5;i++){
5              if(i==3){
6                  continue;
7              }
8              sum=sum+i;
9          }
10         System.out.println("The sum is " + sum);
11     }
12 }
13
```

Javadoc | Problems | Declaration | Console

```
The sum is 12
```

เอกสารอ้างอิง

พนิดา พาณิชกุล. การเขียนโปรแกรมคอมพิวเตอร์เบื้องต้นด้วยภาษา Java. กรุงเทพฯ : เค ที พี คอมพ์ แอนด์ คอนซัลท์, 2548.

บัญชา ปะสีละเตสัง. สร้างเว็บไซต์ด้วย HTML5 ร่วมกับ CSS3 และ jQuery. กรุงเทพฯ : ซีเอ็ด ยูเคชั่น, 2556.

End.