

Loops

Loops

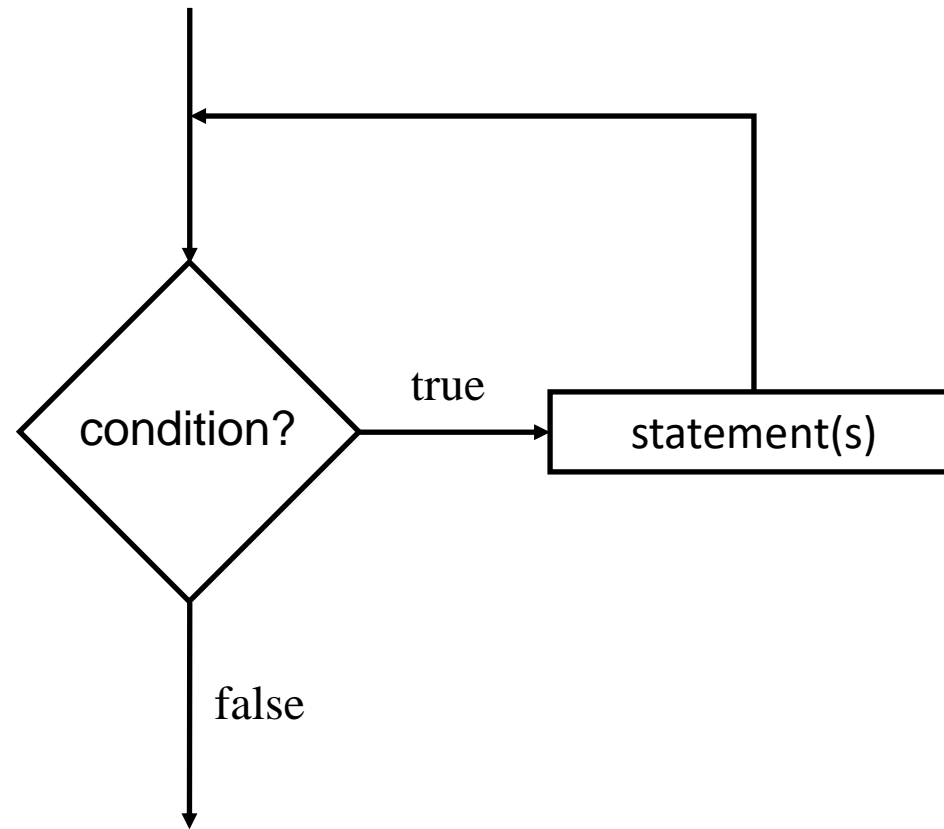
- Loops are constructs that control repeated executions of a block of statements.
- Java provides three types of loop statements:
 - **while** loops
 - **do-while** loops
 - **for** loops

The `while` Loop

- A `while` loop executes statements repeatedly while the `condition is true`.
- The syntax for the while loop is:

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

The while loop Flowchart



Example

```
int count = 0;
```

```
while (count < 100) {
```

```
    System.out.println("Welcome to Java!");
```

```
    count++;
```

```
}
```

Exercise 1

1. ให้นักศึกษาสร้างคลาสชื่อ WhileCalculation ซึ่งประกอบด้วย Methods ดังนี้
 - public void showNumber (int from, int to) ที่ใช้สำหรับแสดงจำนวนเต็มตั้งแต่ from จนถึง to โดยใช้ลูป while
 - public int sumNumber (int from, int to) ที่ใช้สำหรับหาผลรวมของจำนวนเต็มตั้งแต่ from จนถึง to โดยใช้ลูป while แล้ว return ค่าผลรวมที่ได้
 - public int gcd (int num1, int num2) ที่ใช้สำหรับหาค่าหารร่วมมาก (ห.ร.ม) ของ num1 และ num2 โดยใช้ลูป while แล้ว return ค่าหารร่วมมาก (ห.ร.ม)

Exercise 1

2. ให้นักศึกษาร่างคลาสชื่อ WhileCalculationTest เพื่อทดสอบการทำงานของคลาส WhileCalculation ดังนี้

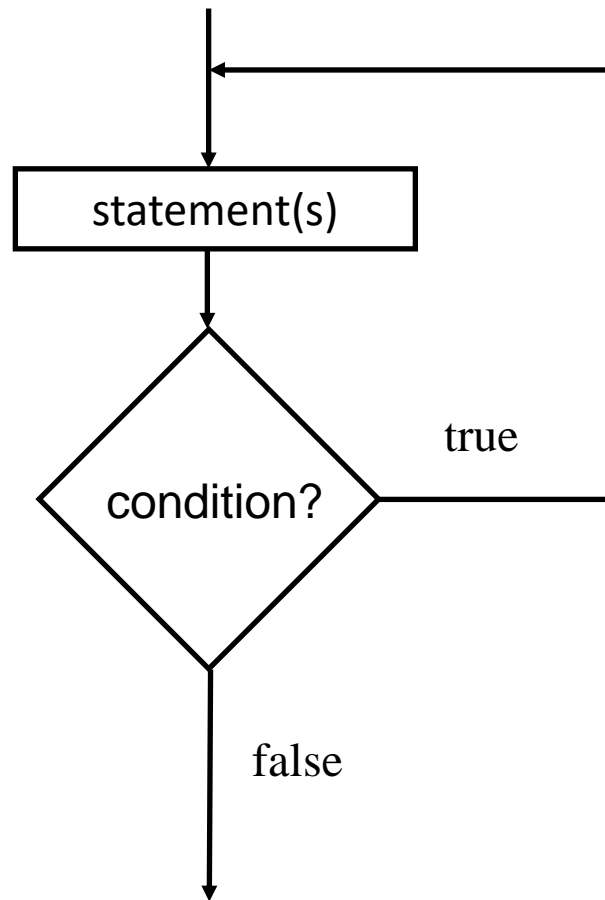
```
public class WhileCalculationTest {  
    public static void main(String[ ] args) {  
        Calculation obj = new Calculation();  
        obj.showNumber(1, 10);  
        System.out.println(obj.sumNumber(5, 100));  
        System.out.println(obj.gcd(15, 12));  
    }  
}
```

The `do-while` Loop

- A `do-while` loop is the same as a `while` loop except that it executes the loop body first and then checks the loop continuation condition.
- The syntax for the `do-while` loop is:

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


The do-while Loop Flowchart



Example

```
int count = 0
```

```
do {
```

```
    System.out.println("Welcome to Java!");
```

```
    count++;
```

```
} while (count < 100);
```

The **for** Loop

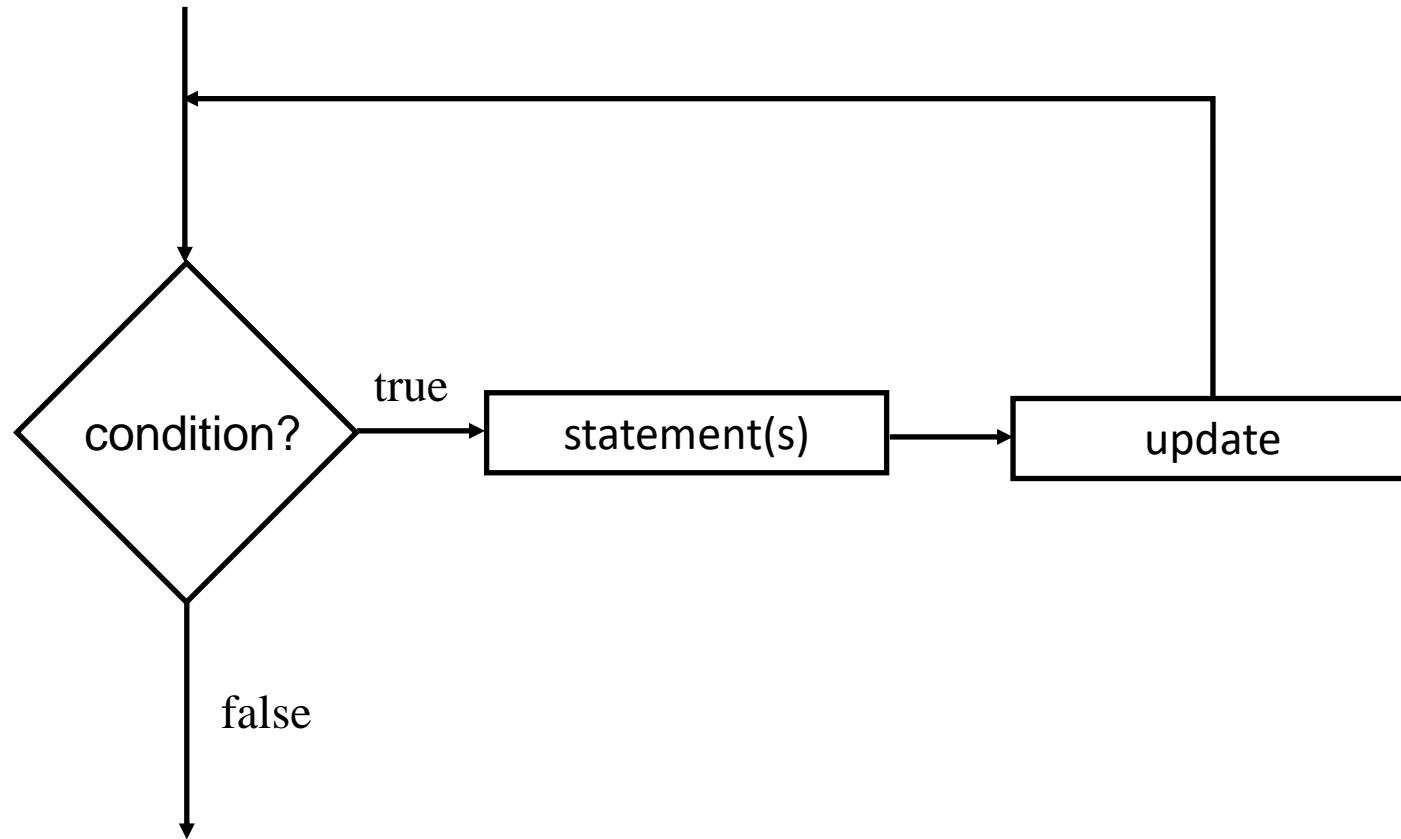
- A for **loop** has a concise syntax for writing loops.
- The syntax for the for loop is:

```
for (initialization; condition; update) {  
    statement(s);  
}
```

The **for** Loop

- The **for** loop control has three parts:
 - The first part (initialization) is executed (one time) before the execution of the code block.
 - The second part (condition) defines the condition for executing the code block.
 - The third part (update) is executed (every time) after the code block has been executed.

The for Loop Flowchart



Example

```
for (int i = 0; i < 100; i++) {  
    System.out.println ("Welcome to Java!");  
}
```

Exercise 2

1. ให้นักศึกษาสร้างคลาสชื่อ ForCalculation ซึ่งประกอบด้วย Methods ดังนี้
 - public void showNumber (int from, int to) ที่ใช้สำหรับแสดงจำนวนเต็มตั้งแต่ from จนถึง to โดยใช้ลูป for
 - public int sumNumber (int from, int to) ที่ใช้สำหรับหาผลรวมของจำนวนเต็มตั้งแต่ from จนถึง to โดยใช้ลูป for แล้ว return ค่าผลรวมที่ได้
 - public int gcd (int num1, int num2) ที่ใช้สำหรับหาค่าหารร่วมมาก (ห.ร.ม) ของ num1 และ num2 โดยใช้ลูป for แล้ว return ค่าหารร่วมมาก (ห.ร.ม) ที่ได้

Exercise 2

2. ให้นักศึกษาร่างคลาสชื่อ ForCalculationTest เพื่อทดสอบการทำงานของคลาส ForCalculation ดังนี้

```
public class WhileCalculationTest {  
    public static void main(String[ ] args) {  
        Calculation obj = new Calculation();  
        obj.showNumber(5, 10);  
        System.out.println(obj.sumNumber(5, 100));  
        System.out.println(obj.gcd(15, 12));  
    }  
}
```


Keywords `break` and `continue`

- The `break` and `continue` keywords provide additional controls in a loop.
- The `break` statement breaks out of the loop.
- The `continue` statement breaks out of the current iteration in the loop.

Nested Loops

- A loop can be nested inside another loop.
- Nested loops consist of an outer loop and one or more inner loops. Each time the outer loop is repeated, the inner loops are reentered, and started anew.

ตัวอย่าง

```
public class Drawing {  
    public void drawRightTriangle() {  
        for (int i = 0; i < 5; i++) {  
            for (int j = 0; j <= i; j++) {  
                System.out.print("* ");  
            }  
            System.out.println();  
        }  
    }  
}
```

Exercise 3

- ให้นักศึกษาสร้างคลาสชื่อ `RightTriangle` ซึ่งประกอบด้วย `Methods` ที่ใช้สำหรับวาดแพทเทิร์น ตามจำนวนแถวที่กำหนด (n) ดังนี้
 - `public void printStar (int n)` สำหรับใช้วาด Right Triangle Pattern ด้วย `*`
 - `public void printNumber(int n)` สำหรับใช้วาด Right Triangle Pattern ด้วยตัวเลข
 - `public void printAlphabet(int n)` สำหรับใช้วาด Right Triangle Pattern ด้วยตัวอักษร

```
*  
* *  
* * *  
* * * *  
* * * * *
```

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5
```

```
A  
A B  
A B C  
A B C D  
A B C D E
```

Exercise 3

2. ให้นักศึกษาร่างคลาสชื่อ RightTriangleTest เพื่อทดสอบการทำงานของคลาส RightTriangle ดังนี้

```
public class RightTriangleTest {  
    public static void main(String[] args) {  
        RightTriangle obj = new RightTriangle();  
        obj.printStar(5);  
        System.out.println();  
        obj.printNumber(5);  
        System.out.println();  
        obj.printAlphabet(5);  
    }  
}
```

Exercise 4

- ให้นักศึกษาสร้างคลาสชื่อ Pyramid ซึ่งประกอบด้วย Methods ที่ใช้สำหรับวาดแพทเทิร์นตามจำนวนแถวที่กำหนด (ค่า n) ดังนี้
 - public void printStar (int n) สำหรับใช้วาด Pyramid Pattern ด้วย *
 - public void printNumber(int n) สำหรับใช้วาด Pyramid Pattern ด้วยตัวเลข
 - public void printAlphabet(int n) สำหรับใช้วาด Pyramid Pattern ด้วยตัวอักษร

```
*  
* *  
* * *  
* * * *  
* * * * *
```

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5
```

```
A  
A B  
A B C  
A B C D  
A B C D E
```

Exercise 4

2. ให้นักศึกษาร่างคลาสชื่อ PyramidTest เพื่อทดสอบการทำงานของคลาส Pyramid ดังนี้

```
public class PyramidTest {  
    public static void main(String[] args) {  
        Pyramid obj = new Pyramid();  
        obj.printStar(5);  
        System.out.println();  
        obj.printNumber(5);  
        System.out.println();  
        obj.printAlphabet(5);  
    }  
}
```

Exercise 5

- ให้นักศึกษาสร้างคลาสชื่อ LeftTriangle ซึ่งประกอบด้วย Methods ที่ใช้สำหรับวาดแพทเทิร์นตามจำนวนแถวที่กำหนด (n) ดังนี้
 - public void printStar (int n) สำหรับใช้วาด Left Triangle Pattern ด้วย *
 - public void printNumber(int n) สำหรับใช้วาด Left Triangle Pattern ด้วยตัวเลข
 - public void printAlphabet(int n) สำหรับใช้วาด Left Triangle Pattern ด้วยตัวอักษร

```
  *
 * *
* * *
* * * *
* * * * *
```

```
  1
 1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
  A
  A B
 A B C
A B C D
A B C D E
```


Exercise 5

2. ให้นักศึกษาร่างคลาสชื่อ LeftTriangleTest เพื่อทดสอบการทำงานของคลาส LeftTriangle ดังนี้

```
public class LeftTriangleTest {  
    public static void main(String[] args) {  
        LeftTriangle obj = new LeftTriangle();  
        obj.printStar(5);  
        System.out.println();  
        obj.printNumber(5);  
        System.out.println();  
        obj.printAlphabet(5);  
    }  
}
```

THANK YOU