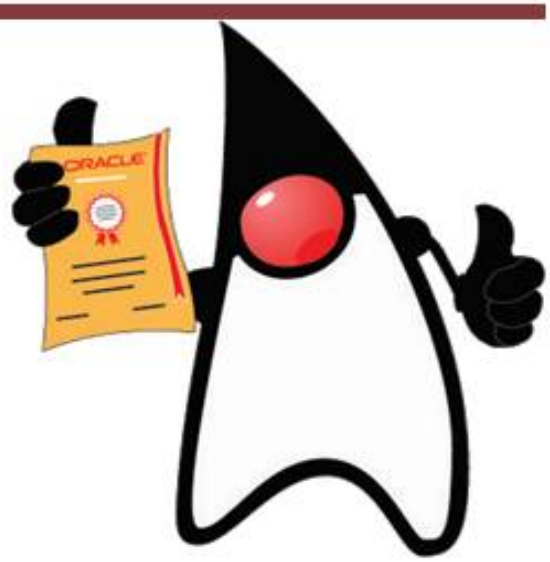
J01-07

使用 Decision Constructs (選擇結構) 和相關運算子

曾瑞君 (Jim_Tzeng)

學習目標

- 使用 relational(關係) & conditional(條件) 運算子
- 使用 if 選擇結構
- 使用 switch 選擇結構



1/3

使用 relational & conditional 運算子

Elevator 範例

```
public class MyElevator {
    public final int MAX_FLOOR = 10;
    public final int MIN_FLOOR = 1;
    public boolean open = false;
    public int currentFloor = MIN_FLOOR;
    //…電梯 開門、關門
    //…電梯 上樓、下樓
}
```

```
//…電梯 開門、關門
public void open() {
    System.out.println("Try to open door,");
    open = true;
    System.out.println("Door is open now.");
}
public void close() {
    System.out.println("Try to close door,");
    open = false;
    System.out.println("Door is closed now.");
}
```

```
//…電梯 上樓、下樓
public void up() {
    System.out.println("Elevator up...");
    currentFloor++;
    System.out.println("Now " + currentFloor + ".");
}
public void down() {
    System.out.println("Elevator down...");
    currentFloor--;
    System.out.println("Now " + currentFloor + ".");
}
```

奇怪的結果...

```
public class MyElevatorTest {
    public static void main(String args[]) {
        MyElevator myElevator = new MyElevator();
        myElevator.open();
        myElevator.close();
        myElevator.down();          // Floor is 0?!
        myElevator.up();
        myElevator.up();
        myElevator.up();
        myElevator.open();
        myElevator.close();
        myElevator.down();
        myElevator.open();
        myElevator.down();          //forget to close door first?!
        myElevator.open();          //open twice?!
    }
}
```

Relational (關係) Operators

情境	運算子	範例
是否 相等	==	int x=1; (x == 1)
是否 不相等	!=	int x=1; (x != 1)
是否 小於	<	int x=1; (x < 1)
是否 小於 等於	<=	int x=1; (x <= 1)
是否 大於	>	int x=1; (x > 1)
是否 大於 等於	>=	int x=1; (x >= 1)

Conditional (條件) Operators

情境	運算子	範例

且 (and)	&&	int x = 5;
		int y =9;
		(x < 6) && (y > 7)

或 (or)	||	int x = 5;
		int y =9;
		(x < 6) || (y > 7)

非 (not)	!	int x = 5; !(x < 6)

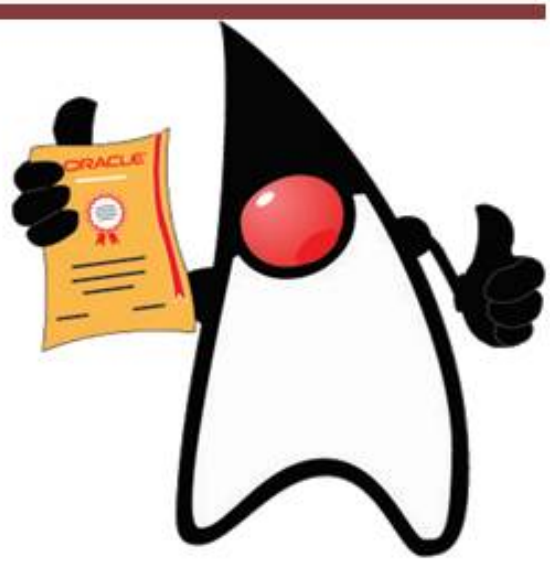
字串比較

- 使用「==」比較字串是否指向記憶體同一位址
- 使用「equals()」比較字串是否內容相同

```
public static void main(String[] args) {
        String s1 = "jim";
        String s2 = "jim";
        String s3 = new String ("jim");

        System.out.println(s1 == s2);              // true
        System.out.println(s1 == s3);              // false

        System.out.println(s1.equals(s2));         // true
        System.out.println(s1.equals(s3));         // true
}
```



2/3

使用 if 選擇結構

使用 if 和 if/else 架構

- 語法：

```
If ( boolean_expression) {
        code_block;
}
```

```
If ( boolean_expression) {
        code_block;
} else {
        code_block;
}
```

```
If ( boolean_expression) {
        code_block;
} else If ( boolean_expression) {
        code_block;
} else {
        code_block;
}
```

使用 if 解決 MyElevator 不合理的情況

```
//...電梯 開門、關門
public void open() {
        if (open) {
                System.out.println("Invalid operation!!");
        } else {
                System.out.println("Try to open door,");
                open = true;
                System.out.println("Door is open now.");
        }
}
public void close() {
        if (!open) {
                System.out.println("Invalid operation!!");
        } else {
                System.out.println("Try to close door,");
                open = false;
                System.out.println("Door is closed now.");
        }
}
```

使用 if 解決 MyElevator 不合理的情況

```
public void up() {
    if (currentFloor >= MAX_FLOOR) {
        System.out.println("Invalid operation!!");
    } else {
        if (open) {
            System.out.println("Invalid operation!! Close door first!!");
        } else {
            System.out.println("Elevator up...");
            currentFloor++;
            System.out.println("Now " + currentFloor);
        }
    }
}
public void down() {
    if (currentFloor <= MIN_FLOOR) {
        System.out.println("Invalid operation!!");
    } else {
        if (open) {
            System.out.println("Invalid operation!! Close door first!!");
        } else {
            System.out.println("Elevator down...");
            currentFloor--;
            System.out.println("Now " + currentFloor);
        }
    }
}
```

使用 if / else if / else 架構

```
public void SumMonthDaysByIf(int month) {
    if (month == 1 || month == 3 || month == 5 || month == 7
        || month == 8 || month == 10 || month == 12) {
        System.out.println("31 days");
    } else if (month == 2) {
        System.out.println("28 days");
    } else if (month == 4 || month == 6 || month == 9 || month == 11) {
        System.out.println("30 days");
    } else {
        System.out.println("Invalid month");
    }
}
```

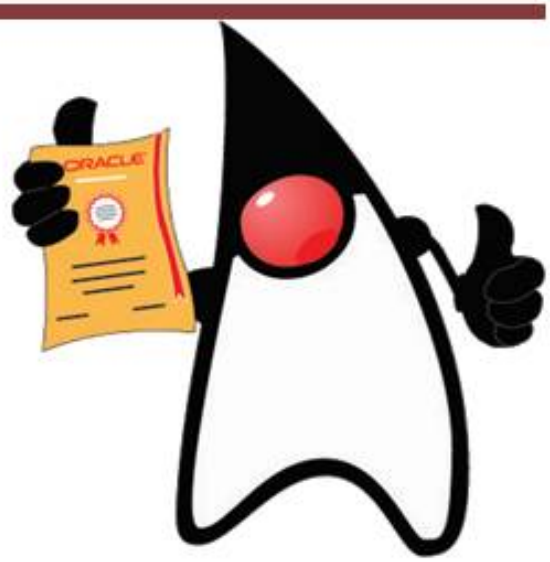
使用三元(ternary)運算子

(expression) ? value if true : value if false

```
public static void main(String[] args) {
        int a, b;
        a = 10;

        b = (a == 1) ? 20 : 30;
        System.out.println("Value of b is : " + b);

        if (a == 1) {
                b = 20;
        } else {
                b = 30;
        }
        System.out.println("Value of b is : " + b);
}
```



3/3

使用 switch 選擇結構

使用 switch架構

- 語法：

```
switch (variable) {
    case literal_value:
        <code_block>
        [break;]
    case another_literal_value:
        <code_block>
        [break;]
    [default:]
        <code_block>
}
```

- variable:：準備要測試的變數。變數型態可以是 byte, short, char, int, String
- literal_value：變數可能的值
- default：都不滿足時進入
- break：可不加。將離開 switch

使用 switch 架構

```
public void SumMonthDaysBySwitch(int month) {
    switch (month) {
    case 1: case 3: case 5: case 7:
    case 8: case 10: case 12:
        System.out.println("30 days");
        break;
    case 2:
        System.out.println("28 days");
        break;
    case 4: case 6: case 9: case 11:
        System.out.println("30 days");
        break;
    default:
        System.out.println("Invalid month");
        break;
    }
}
```

End ~~

Thank you!!