

運算式

```
public static void 運算式() {  
    //宣告  
    int x = 0, y = 1, z = 2;  
    //指派運算式  
    x = 123;  
    //算術運算式  
    x = y + z; //一般運算式  
    x = x + y; //累加運算式  
    //算術指派運算式  
    x += y;  
    x += 1;  
    //遞增遞減運算式  
    x++;  
    ++x;  
    x--;  
    --x;  
    //比較運算式  
    if (x > y) {  
  
    }  
    while (x > y) {  
  
    }  
    //邏輯運算式  
    if (x > y && y > z) {  
  
    }  
    while (x > y && y > z) {  
  
    }  
    //位元運算式  
    int a = 12 << 3;  
  
    //三元運算子  
    int c = 10 > 20 ? 100 : 200;  
}
```

Java 的運算子→只能用在 基本資料 型態

運算子的 優先權 與 聯結性

對於各種 operators，我們須要知道它的 precedence（優先權）、associative（聯結性）、operand types（運算元型別）以及它所執行的 operation（操作）。下表說明 Java operators 的 precedence 及 operand types：

P	A	Operator	Operand types	Description
15	L	++	numeric	unary post-increment 如：n++
		--	numeric	unary post-decrement 如：n--
14	R	++	numeric	unary pre-increment 如：++n
		--	numeric	unary pre-decrement 如：--n
		+	numeric	unary plus 如：+n
		-	numeric	unary minus 如：-n
		~	integer	unary bitwise complement 如：~i
		!	boolean	unary logic negation 如：!b
13	R	new (type)	class any	object creation 如：new MyClass() casting 如：(int) i
12	L	*	numeric	multiplication 如：n1 * n2
		/	numeric	division 如：n1 / n2
		%	numeric	modulus 如：n1 % n2
11	L	+	numeric String	addition 如：n1 + n2 String concatenation 如："s1" + "s2"
		-	numeric	subtraction 如：n1 - n2

P	A	Operator	Operand types	Description
10	L	<<	integer	bitwise left shift, filling with 0s 如：i << 2
		>>	integer	bitwise right shift, filling with the sign bit 如：i >> 2
		>>>	integer	bitwise right shift, filling with 0s 如：i >>> 2
9	L	<	numeric	less than 如：n1 < n2
		<=	numeric	less than or equal to 如：n1 <= n2
		>	numeric	greater than 如：n1 > n2
		>=	numeric	greater than or equal to 如：n1 >= n2
		instanceof	numeric	greater than or equal to 如：n1 >= n2
8	L	==	any	equal 如：a1 == a2
		!=	any	not equal 如：a1 != a2
7	L	&	integer boolean	bitwise AND 如：i1 & i2 boolean AND 如：b1 & b2
6	L	^	integer boolean	bitwise XOR 如：i1 ^ i2 boolean XOR 如：b1 ^ b2
5	L		integer boolean	bitwise OR 如：i1 i2 boolean OR 如：b1 b2
4	L	&&	boolean	conditional AND 如：b1 && b2
3	L		boolean	conditional OR 如：b1 b2
2	R	?:	boolean, any, any	ternary condition 如：b ? a1 : a2
1	R	=	any	assignment 如：a1 = a2
		+=	numeric String	assignment with addition 如：n1 += n2 assignment with String concatenation 如：s1 += s2
		-=	numeric	assignment with subtraction 如：n1 -= n2
		*=	numeric	assignment with multiplication 如：n1 *= n2
		/=	numeric	assignment with division 如：n1 /= n2
		%=	numeric	assignment with modulus 如：n1 %= n2
		&=	integer boolean	assignment with bitwise AND 如：i1 &= i2 assignment with boolean AND 如：b1 &= b2
		^=	integer boolean	assignment with bitwise XOR 如：i1 ^= i2 assignment with boolean XOR 如：b1 ^= b2
		=	integer boolean	assignment with bitwise OR 如：i1 = i2 assignment with boolean OR 如：b1 = b2
		<<=	integer	assignment with bitwise left shift, filling with 0s 如：i1 <<= 3
		>>=	integer	assignment with bitwise right shift, filling with the sign bit 如：i1 >>= 3
		>>>=	integer	assignment with bitwise right shift, filling with 0s 如：i1 >>>= 3

1. 指派運算子 \rightarrow =
2. 算術運算子 \rightarrow + , - , * , / , %

```
public static void 算術運算子() {
    // + , - , * , / , %
    int grade1 = 89, grade2 = 99, grade3 = 92;

    double average = (grade1 + grade2 + grade3) / 3;
    // 這樣小數點後一定是 0 , 因為 int/int 結果是 int
    System.out.println("三科平均 = " + average);

    average = (double) (grade1 + grade2 + grade3) / 3;
    System.out.println("三科平均 = " + average);

    average = (grade1 + grade2 + grade3) / 3.0;
    System.out.println("三科平均 = " + average);
}
```

```
三科平均 = 99.0
三科平均 = 99.33333333333333
三科平均 = 99.33333333333333
```

```
public static void 換錢1() {
    int money = 987;
    int m100 = 0, m50 = 0, m10 = 0, m5 = 0, m1 = 0;

    m100 = money / 100;
    m50 = (money % 100) / 50;
    m10 = (money % 50) / 10;
    m5 = (money % 10) / 5;
    m1 = (money % 5) / 1;

    System.out.println("100 元 : " + m100);
    System.out.println("50 元 : " + m50);
    System.out.println("10 元 : " + m10);
    System.out.println("5 元 : " + m5);
    System.out.println("1 元 : " + m1);
}
```

```
100 元 : 9
50 元 : 1
10 元 : 3
5 元 : 1
1 元 : 2
```

3. 算術指派運算子 \rightarrow += , -= , *= , /= , %= \rightarrow 用來配合迴圈敘述句

```
public static void 算術指派運算子() {
    // += , -= , /= , %=
    int x = 4;
    short y = 0;

    y = (short) (y + x); // 要轉型
    y += x; // 不用轉型
}
```

4. 遞增、遞減運算子→當變數的值須增加 1 或 減少 1→用來配合迴圈敘述句

運算方式	運算子符號	範例	1. 單純用法→以下一樣	2. 組合法
前置遞增	++	++a;	a=a+1	a=b++ 等同於-----
後置遞增	++	a++;	a+=1	a=b
前置遞減	--	--a;	a--	b=b+1
後置遞減	--	a--;	++a	

```
public static void 遞增遞減運算子1() {
    int a = 0, b = 0, c = 0, d = 0;

    a = a + 1;
    System.out.println("a=" + a);

    System.out.println("b=" + b);
    b = b + 1;

    c = c - 1;
    System.out.println("c=" + c);

    System.out.println("d=" + d);
    d = d - 1;
}
```

```
public static void 遞增遞減運算子2() {
    int a = 0, b = 0, c = 0, d = 0;

    a++;
    System.out.println("a=" + a);

    ++b;
    System.out.println("b=" + b);

    System.out.println("c=" + c++);

    System.out.println("d=" + ++d);
}
```

```
public static void 遞增遞減運算子3() {
    int a = 0, b = 0, c = 0, d = 0;

    int e;
    e = a++;
    System.out.println("後置遞增e=" + e + " a=" + a);
    e = ++b;
    System.out.println("前置遞增e=" + e + " b=" + b);
    e = c--;
    System.out.println("後置遞減e=" + e + " c=" + c);
    e = --d;
    System.out.println("前置遞減e=" + e + " d=" + d);
}
```

```
public static void 遞增遞減運算子4() {
    int i = 0, j = 0;

    System.out.println(i++ + i++ + i++);
    System.out.println(++j + ++j + ++j);
}
```

```

public static void 遞增遞減運算子5() {

    char[] str = {'a', 'b', 'c', 'd', 'e', 'f'};
    int length;
    length = str.length;
    int i = 0, j = length;

    do {
        // System.out.print(str[i++] + " ");
        System.out.println(str[i]);
        i = i + 1;

    } while (i < length);

    System.out.println("\n=====");

    do {
        // System.out.print(str[--j] + " ");
        j = j - 1;
        System.out.println(str[j]);

    } while (j > 0);

    System.out.println("\n=====");

}

```

i=0→i++		
印 str[0]→a	i=1	1<6
印 str[1]→b	i=2	2<6
印 str[2]→c	i=3	3<6
印 str[3]→d	i=4	4<6
印 str[4]→e	i=5	5<6
印 str[5]→f	i=6	6<6 (false)跳
j=6→--j		
印 str[5]→f	j=5	5>0
印 str[4]→e	j=4	4>0
印 str[3]→d	j=3	3>0
印 str[2]→c	j=2	2>0
印 str[1]→b	j=1	1>0
印 str[0]→a	j=0	0>0 (false)跳

```

a b c d e f
=====
f e d c b a
=====

```

5. 比較運算子→比較後會得到 true 與 false 布林值→用在 if 與 迴圈的比較判斷

	運算子符號		範例
數值 布林	=	數值 布林	op1 == op2
	!=		op1 != op2
數值	<	數值	op1 < op2
	<=		op1 <= op2
	>		op1 > op2
	>=		op1 >= op2

```

public static void 比較運算子1() {
    int a = 4;
    int b = 5;

    if (a > b) {

    }
    while (a > b) {

    }

}

```

```

public static void 比較運算子2() {
    int a = 123;
    int b = 456;

    System.out.println("a> b =" + (a > b));

}

```

```

public static void 比較運算子3() {
    if (true) {
        System.out.println("條件成立");
    } else {
        System.out.println("條件不成立");
    }

}

```

```

public static void 比較運算子4() {
    while (true) {
        System.out.println("條件成立");
    }

}

```

6. 邏輯運算子→常和 比較運算子一起使用→兩邊必須都是布林型態

	運算子符號		功能
布林值(比較運算式)	&&	布林值(比較運算式)	and 互相牽制
			or 互不相干
	^		xor 兩個要相反 才會得到 true
	!		not 結果相反

```
public static void 邏輯運算子1() {
    System.out.println("true && true && true || false = " + (true && true && true || false));
    System.out.println("false || false || false && true = " + (false || false || false && true));
    System.out.println("true ^ true=" + (true ^ true));
    System.out.println("true ^ false=" + (true ^ false));
    System.out.println("!true=" + (!true));
}
```

```
public static void 邏輯運算子2() {
    int grade = 95;
    if (grade >= 90 && grade <= 100) {
        System.out.println("優等");
    }
}
```

```
public static void 邏輯運算子3() {
    int a = 10, b = 10, c = 10, d = 10;
    // (== 與 != ) 的優先權 比 ( || 跟 && ) 高
    if (a == 10 && b != 10) {
    }
    // (> 與 < ) 的優先權 比 ( || 跟 && ) 高 , (&& 的優先權比 || 高)
    if (a < 10 || b < 20 && c > 30) {
    }
}
```

7. 位元運算子→用少量的記憶體(1個bit)·來記憶一個狀態是否發生。這種應用在低階語言很常被使用·用途上·做為遮罩(mask)·或稱做過濾器

	運算子符號		功能	範例
數值	&	數值	and	10 & 5
			or	10 5
	^		xor 運算	10 ^ 5
	<< · >>		左移運算·右移運算	10 << 2 10 >> 2
	~		not	

```
public static void 位元運算子() {
    int a = 12;
    int b = 38;

    System.out.println(a + " & " + b + "=" + (a & b));
    System.out.println(a + " | " + b + "=" + (a | b));
    System.out.println(a + " ^ " + b + "=" + (a ^ b));
    System.out.println(a + " << " + 2 + "=" + (a << 2));
    System.out.println(a + " >> " + 2 + "=" + (a >> 2));
    System.out.println("~" + a + "=" + (~a));
}
```

```
12 & 38 =4
12 | 38 =46
12 ^ 38 =42
12 << 2 =48
12 >> 2 =3
~ 12 =-13
```

8. 三元運算子

```
public static void 三元運算子() {

    int a = 10, b = 20, c;
    /*
    if (a > b) {
        c = 100;
    } else {
        c = 200;
    }
    */
    c = a > b ? 100 : 200;
    System.out.println("c=" + c);
}
```

流程控制

```
public static void if的架構1() {

    int a = 1, b = 2, c = 3, d = 4;
    //1 種情形
    if (a > b) {

    }
    //2 種情形
    if (a > b) {

    } else {

    }
    //3 種情形以上
    if (a > b) {

    } else if (b > c) {

    } else if (c > d) {

    } else {

    }
}
```

```
public static void if的架構2() {
    //複雜的巢狀 if
    int a = 1, b = 2, c = 3, d = 4;

    if (a > b) {

        if (b > c) {

        } else if (c > d) {

        } else {

        }

    } else {

        if (d > a) {

        } else {

        }

    }
}
```

```
public static void if_中斷() {
    int x = 3;

    if (x >= 1) {
        System.out.println("x>=1");
    } else if (x >= 2) {
        System.out.println("x>=2");
    } else if (x >= 3) {
        System.out.println("x>=3");
    } else if (x >= 4) {
        System.out.println("x>=4");
    } else if (x >= 5) {
        System.out.println("x>=5");
    }
}
```

```
public static void if_連續() {
    int x = 3;

    if (x >= 1) {
        System.out.println("x>=1");
    }
    if (x >= 2) {
        System.out.println("x>=2");
    }
    if (x >= 3) {
        System.out.println("x>=3");
    }
    if (x >= 4) {
        System.out.println("x>=4");
    }
    if (x >= 5) {
        System.out.println("x>=5");
    }
}
```

```

public static void 輸入帳號密碼1() {
    String userid, password;
    final String user = "sun2000", pass = "1234";
    Scanner s = new Scanner(System.in);

    System.out.print("輸入帳號:");
    userid = s.next();
    System.out.print("輸入密碼:");
    password = s.next();

    if (userid.equals(user) && password.equals(pass)) {
        System.out.println("Pass");
    } else {
        System.out.println("No Pass");
    }
}

```

輸入帳號:sun2000

輸入密碼:1234

Pass

密碼檢核 分數等級 四則運算

使用者

密碼

```

private void btn檢核ActionPerformed(java.awt.event.ActionEvent evt) {

    String userid, password;
    final String user = "sun2000", pass = "1234";

    userid = txt使用者.getText();
    password = txt密碼.getText();

    if (userid.equals(user) && password.equals(pass)) {
        lbl通過.setText("Pass");
    } else {
        lbl通過.setText("No Pass");
    }
}

```



```

public static void if分數等級1() {
    int grade;
    Scanner s = new Scanner(System.in);
    System.out.print("請輸入分數 = ");
    grade = s.nextInt();

    if (grade >= 90 && grade <= 100) {
        System.out.println("成績：優等");
    } else if (grade >= 80 && grade <= 89) {
        System.out.println("成績：甲等");
    } else if (grade >= 70 && grade <= 79) {
        System.out.println("成績：乙等");
    } else if (grade >= 60 && grade <= 69) {
        System.out.println("成績：丙等");
    } else if (grade >= 0 && grade <= 59) {
        System.out.println("成績：丁等");
    } else {
        System.out.println("分數輸入錯誤，請檢核");
    }
}

```

```

public static void if分數等級2() {
    int grade;
    Scanner s = new Scanner(System.in);
    System.out.print("請輸入分數 = ");
    grade = s.nextInt();

    if (grade >= 90) {
        System.out.println("成績：優等");
    } else if (grade >= 80) {
        System.out.println("成績：甲等");
    } else if (grade >= 70) {
        System.out.println("成績：乙等");
    } else if (grade >= 60) {
        System.out.println("成績：丙等");
    } else {
        System.out.println("成績：丁等");
    }
}

```

```

public static void if分數等級3() {
    int grade;
    Scanner s = new Scanner(System.in);
    System.out.print("請輸入分數= ");
    grade = s.nextInt();

    if (grade >= 0 && grade <= 100) {
        if (grade >= 90) {
            System.out.println("成績：優等");
        } else if (grade >= 80) {
            System.out.println("成績：甲等");
        } else if (grade >= 70) {
            System.out.println("成績：乙等");
        } else if (grade >= 60) {
            System.out.println("成績：丙等");
        } else {
            System.out.println("成績：丁等");
        }
    } else {
        System.out.println("分數輸入錯誤，請檢核");
    }
}

```

```

public static void if分數等級4() {
    int grade;
    Scanner s = new Scanner(System.in);
    System.out.print("請輸入分數 = ");
    grade = s.nextInt();

    if(grade<0 || grade>100){
        System.out.println("分數輸入錯誤，請檢核");
    }else{
        if (grade >= 90) {
            System.out.println("成績：優等");
        } else if (grade >= 80) {
            System.out.println("成績：甲等");
        } else if (grade >= 70) {
            System.out.println("成績：乙等");
        } else if (grade >= 60) {
            System.out.println("成績：丙等");
        } else {
            System.out.println("成績：丁等");
        }
    }
}

```

你的分數 = 130
成績：優等

你的分數 = -30
成績：丁等
你的成績不好喔！

請輸入分數：= 120
你的分數 = 120
分數輸入錯誤，請檢核

請輸入分數：= 95
你的分數 = 95
成績：優等

return → 用於函數的敘述句，讓程式執行焦點跳回它所在的函數，返回之前呼叫此函數的地方，繼續執行下一個敘述句

```
public static void return測試() {
    //函數
    int x = 4, y = 5;
    int sum;
    sum = return1(x, y);
    System.out.println("sum=" + sum);
    //方法
    return2(x, y);
    System.out.println("Bye Bye");
}
```

```
public static int return1(int a, int b) {
    if (a == 0 && b == 0) {
        return 0; //中斷 + 回傳值
    }
    return a + b; //中斷 + 回傳值
}
```

```
public static void return2(int x, int y) {
    if (x == 0 && y == 0) {
        return; //中斷
    }
    System.out.println("x+y=" + (x + y));
}
```

```
public static void if分數等級() {
    int grade;
    Scanner s = new Scanner(System.in);
    System.out.print("請輸入分數 = ");
    grade = s.nextInt();
    //檢核
    if (grade > 100 || grade < 0) {
        System.out.println("你的分數輸入錯誤");
        return;
    }
    //計算
    if (grade >= 90) {
        System.out.println("成績：優等");
    } else if (grade >= 80) {
        System.out.println("成績：甲等");
    } else if (grade >= 70) {
        System.out.println("成績：乙等");
    } else if (grade >= 60) {
        System.out.println("成績：丙等");
    } else {
        System.out.println("成績：丁等");
    }
}
```

switch 可用來取代規則性巢狀 if

switch 的型態→整數 (byte · short · int · char) · String · 列舉 三種

語法：

變數或運算式的型態為==>整數，字串，列舉

```
switch( 變數 或 運算式 )
{
    case 整數常數值 或 列舉常數值 或 字串常數值 :
        //處理
    case 整數常數值 或 列舉常數值 或 字串常數值 :
        //處理
    case 整數常數值 或 列舉常數值 或 字串常數值 :
        //處理
    default :
        //處理
}
```

```
public static void switch的架構_整數() {
    int a = 66;
    final int B = 66;

    switch (a) { //整數
        case 65: //數值常數
            System.out.println("我是65");
        case B: //常數的變數
            System.out.println("我是66");
        case 'C': //字元
            System.out.println("我是67");
        case 68:
        case 69:
            System.out.println("我是68 或 69");
        default:
            System.out.println("我什麼都不是");
    }
}
```

我是66
我是67
我是68 或 69
我什麼都不是

```
public static void switch_中斷() {
    int x = 3;
    switch (x) {
        default:
            System.out.println("x>=6");
            break;
        case 5:
            System.out.println("x>=5");
            break;
        case 4:
            System.out.println("x>=4");
            break;
        case 3:
            System.out.println("x>=3");
            break;
        case 2:
            System.out.println("x>=2");
            break;
        case 1:
            System.out.println("x>=1");
            break;
    }
}
```

```
public static void switch_連續() {
    int x = 3;
    switch (x) {
        default:
            System.out.println("x>=6");
        case 5:
            System.out.println("x>=5");
        case 4:
            System.out.println("x>=4");
        case 3:
            System.out.println("x>=3");
        case 2:
            System.out.println("x>=2");
        case 1:
            System.out.println("x>=1");
    }
}
```

```
public static void switch的架構_字串() {
    String a = "abc";

    switch (a) { // 字串
        case "XYZ":
            System.out.println("我是XYZ");
            break;
        case "ABC":
            System.out.println("我是ABC");
            break;
        case "abc":
            System.out.println("我是abc");
            break;
        default:
            System.out.println("我什麼都不是");
    }
}
```

```
public static void switch的架構_列舉() {
    ColorX a=ColorX.YELLOW; //列舉變數

    switch (a) { // 列舉
        case RED:
            System.out.println("我是RED");
            break;
        case YELLOW:
            System.out.println("我是YELLOW");
            break;
        case BLUE:
            System.out.println("我是BLUE");
            break;
        default:
            System.out.println("我什麼都不是");
    }
}
```

```
public static void switch分數等級() {
    int grade;
    Scanner s = new Scanner(System.in);
    System.out.print("請輸入分數 = ");
    grade = s.nextInt();

    if (grade > 100 || grade < 0) {
        System.out.println("你的分數輸入錯誤");
        return;
    }

    switch (grade / 10) {
        case 10:
        case 9:
            System.out.println("成績：優等");
            break;
        case 8:
            System.out.println("成績：甲等");
            break;
        case 7:
            System.out.println("成績：乙等");
            break;
        case 6:
            System.out.println("成績：丙等");
            break;
        default:
            System.out.println("成績：丁等");
    }
}
```

```
private void txt分數ActionPerformed(java.awt.event.ActionEvent evt) {
    int grade;
    grade = Integer.parseInt(txt分數.getText());

    if (grade > 100 || grade < 0) {
        lbl分數等級.setText("你的分數輸入錯誤");
        return;
    }

    switch (grade / 10) {
        case 10:
        case 9:
            lbl分數等級.setText("成績：優等");
            break;
        case 8:
            lbl分數等級.setText("成績：甲等");
            break;
        case 7:
            lbl分數等級.setText("成績：乙等");
            break;
        case 6:
            lbl分數等級.setText("成績：丙等");
            break;
        default:
            lbl分數等級.setText("成績：丁等 ");
    }
}
```

```
public static void switch站別() {  
    String station;  
    StringBuilder passStat = new StringBuilder("");  
    Scanner s = new Scanner(System.in);  
  
    System.out.print("A=台北\nB=板橋\nC=中壢\nD=新竹\n輸入起站 :");  
    station = s.next();  
  
    station = station.toUpperCase();  
  
    switch (station) {  
        case "A":  
            passStat.append("台北->");  
            System.out.println("台北開車");  
        case "B":  
            passStat.append("板橋->");  
            System.out.println("板橋到了");  
        case "C":  
            passStat.append("中壢->");  
            System.out.println("中壢到了");  
        case "D":  
            passStat.append("新竹");  
            System.out.println("終站新竹");  
            break;  
        default:  
            System.out.println("選項錯誤");  
    }  
    System.out.println("你的旅程是 :\n" + passStat);  
}
```

A=台北

B=板橋

C=中壢

D=新竹

輸入起站 :a

台北開車

板橋到了

中壢到了

終站新竹

你的旅程是：

台北->板橋->中壢->新竹

```
public static void switch三元運算子() {  
  
    int cartype, money = 0, money2 = 0, money3 = 0, money4 = 0, sum = 0;  
    String cdplayer, sunroof, autotrans;  
  
    Scanner s = new Scanner(System.in);  
  
    System.out.print("請選擇:(1)跑車 (2)轎車(豪華型) (3)轎車(經濟型) : ");  
    cartype = s.nextInt();  
    System.out.println();  
  
    System.out.print("是否加裝CD音響?(y/n) : ");  
    cdplayer = s.next();  
  
    System.out.print("是否加裝天窗?(y/n) : ");  
    sunroof = s.next();  
  
    System.out.print("是否該裝成自動排檔?(y/n) : ");  
    autotrans = s.next();  
  
    switch (cartype) {  
        case 1:  
            money = 800000;  
            break;  
        case 2:  
            money = 650000;  
            break;  
        case 3:  
            money = 450000;  
            break;  
        default:  
            System.out.println("謝謝惠顧!\n");  
    }  
  
    money2 = (cdplayer.equals("y")) ? 4000 : 0;  
    money3 = (sunroof.equals("y")) ? 5000 : 0;  
    money4 = (autotrans.equals("y")) ? 3000 : 0;  
  
    sum = money + money2 + money3 + money4;  
    System.out.printf("總價格為:%d\n", sum);  
    System.out.println("謝謝惠顧!\n");  
}
```

請選擇:(1)跑車 (2)轎車(豪華型) (3)轎車(經濟型) : 1

是否加裝CD音響?(y/n) : y

是否加裝天窗?(y/n) : y

是否該裝成自動排檔?(y/n) : y

總價格為:812000

謝謝惠顧!

```
public enum Count {

    加, 減, 乘, 除;

}
```

```
public static void 算術運算1() {
    int x, y;
    int sum;
    Scanner s = new Scanner(System.in);

    System.out.print("輸入數字1=");
    x = s.nextInt();
    System.out.print("輸入數字2=");
    y = s.nextInt();

    sum = 四則運算1(x, y, '+');
    System.out.println("x+y=" + sum);
    sum = 四則運算1(x, y, '-');
    System.out.println("x-y=" + sum);
    sum = 四則運算1(x, y, '*');
    System.out.println("x*y=" + sum);
    sum = 四則運算1(x, y, '/');
    System.out.println("x/y=" + sum);
    sum = 四則運算1(x, y, '#');
    System.out.println("x#y=" + sum);
}

public static int 四則運算1(int x, int y, char op) {

    switch (op) {
        case '+':
            return x + y;
        case '-':
            return x - y;
        case '*':
            return x * y;
        case 'X':
        case 'x':
            return x * y;
        case '/':
            return x / y;
        default:
            System.out.println("運算子錯誤");
            return 0;
    }
}
```

```
public static void 算術運算2() {
    int x, y;
    int sum;
    Scanner s = new Scanner(System.in);

    System.out.print("輸入數字1=");
    x = s.nextInt();
    System.out.print("輸入數字2=");
    y = s.nextInt();

    sum = 四則運算2(x, y, Count.加);
    System.out.println("x+y=" + sum);
    sum = 四則運算2(x, y, Count.減);
    System.out.println("x-y=" + sum);
    sum = 四則運算2(x, y, Count.乘);
    System.out.println("x*y=" + sum);
    sum = 四則運算2(x, y, Count.除);
    System.out.println("x/y=" + sum);
}

public static int 四則運算2(int x, int y, Count op) {

    switch (op) {
        case 加:
            return x + y;
        case 減:
            return x - y;
        case 乘:
            return x * y;
        case 除:
            return x / y;
        default:
            return 0;
    }
}
```

```
輸入數字1=20
輸入數字2=10
x+y=30
x-y=10
x*y=200
x/y=2
```



```
private void btn等於ActionPerformed(java.awt.event.ActionEvent evt) {

    int x = Integer.parseInt(txt數字1.getText());
    int y = Integer.parseInt(txt數字2.getText());
    int sum = 0;

    if (rdo加.isSelected()) {
        sum = 四則運算(x, y, Count.加);
    }
    if (rdo減.isSelected()) {
        sum = 四則運算(x, y, Count.減);
    }
    if (rdo乘.isSelected()) {
        sum = 四則運算(x, y, Count.乘);
    }
    if (rdo除.isSelected()) {
        sum = 四則運算(x, y, Count.除);
    }
    lbl答案.setText(String.valueOf(sum));
}
```

```
private void 四則運算共用事件(java.awt.event.ActionEvent evt) {
    //1.
    // btn等於ActionPerformed(evt);
    //2.
    JRadioButton tmp = (JRadioButton) evt.getSource();
    int x = Integer.parseInt(txt數字1.getText());
    int y = Integer.parseInt(txt數字2.getText());
    int sum = 0;

    if (tmp == rdo加) {
        sum = 四則運算(x, y, Count.加);
    }
    if (tmp == rdo減) {
        sum = 四則運算(x, y, Count.減);
    }
    if (tmp == rdo乘) {
        sum = 四則運算(x, y, Count.乘);
    }
    if (tmp == rdo除) {
        sum = 四則運算(x, y, Count.除);
    }
    lbl答案.setText(String.valueOf(sum));
}
```


迴圈→資料累加・陣列

```
public static void 單層迴圈1() {
    int i;
    for (i = 1; i <= 10; i++) {
        System.out.println("for 我愛你" + i + "次");
    }
    //-----
    i = 1;
    while (i <= 10) {
        System.out.println("while 我愛你" + i + "次");
        i++;
    }
    //-----
    i = 1;
    do {
        System.out.println("do while 我愛你" + i + "次");
        i++;
    } while (i <= 10);
}
```

```
public static void 單層迴圈2() {
    int i;
    for (i = 10; i >= 1; i--) {
        System.out.println("for 我愛你" + i + "次");
    } //-----
    i = 10;
    while (i >= 1) {
        System.out.println("while 我愛你" + i + "次");
        i--;
    }
    //-----
    i = 10;
    do {
        System.out.println("do while 我愛你" + i + "次");
        i--;
    } while (i >= 1);
}
```

for 我愛你1次
for 我愛你2次
for 我愛你3次
for 我愛你4次
for 我愛你5次
for 我愛你6次
for 我愛你7次
for 我愛你8次
for 我愛你9次
for 我愛你10次

for 我愛你10次
for 我愛你9次
for 我愛你8次
for 我愛你7次
for 我愛你6次
for 我愛你5次
for 我愛你4次
for 我愛你3次
for 我愛你2次
for 我愛你1次

```
public static void 單層迴圈3() {
    int i;
    int sum = 0; //必須 init

    for (i = 1; i <= 10; i++) {
        sum = sum + i;
        System.out.println("for i=" + i + "    sum = " + sum);
    }
    //-----
    sum = 0;
    i = 1;
    while (i <= 10) {
        sum = sum + i;
        System.out.println("while i=" + i + "    sum = " + sum);
        i++;
    }
    //-----
    sum = 0;
    i = 1;
    do {
        sum = sum + i;
        System.out.println("do while i=" + i + "    sum = " + sum);
        i++;
    } while (i <= 10);
}
```

for i=1	sum = 1
for i=2	sum = 3
for i=3	sum = 6
for i=4	sum = 10
for i=5	sum = 15
for i=6	sum = 21
for i=7	sum = 28
for i=8	sum = 36
for i=9	sum = 45
for i=10	sum = 55

```

public static void 雙層迴圈1() {
    for (int i = 1; i <= 10; i++) {

        for (int j = 1; j <= 10; j++) {
            System.out.print(" i=" + i + " j=" + j);
        }

        System.out.println();
    }
}

```

```

i=1 j=1 i=1 j=2 i=1 j=3 i=1 j=4 i=1 j=5 i=1 j=6 i=1 j=7 i=1 j=8 i=1 j=9 i=1 j=10
i=2 j=1 i=2 j=2 i=2 j=3 i=2 j=4 i=2 j=5 i=2 j=6 i=2 j=7 i=2 j=8 i=2 j=9 i=2 j=10
i=3 j=1 i=3 j=2 i=3 j=3 i=3 j=4 i=3 j=5 i=3 j=6 i=3 j=7 i=3 j=8 i=3 j=9 i=3 j=10
i=4 j=1 i=4 j=2 i=4 j=3 i=4 j=4 i=4 j=5 i=4 j=6 i=4 j=7 i=4 j=8 i=4 j=9 i=4 j=10
i=5 j=1 i=5 j=2 i=5 j=3 i=5 j=4 i=5 j=5 i=5 j=6 i=5 j=7 i=5 j=8 i=5 j=9 i=5 j=10
i=6 j=1 i=6 j=2 i=6 j=3 i=6 j=4 i=6 j=5 i=6 j=6 i=6 j=7 i=6 j=8 i=6 j=9 i=6 j=10
i=7 j=1 i=7 j=2 i=7 j=3 i=7 j=4 i=7 j=5 i=7 j=6 i=7 j=7 i=7 j=8 i=7 j=9 i=7 j=10
i=8 j=1 i=8 j=2 i=8 j=3 i=8 j=4 i=8 j=5 i=8 j=6 i=8 j=7 i=8 j=8 i=8 j=9 i=8 j=10
i=9 j=1 i=9 j=2 i=9 j=3 i=9 j=4 i=9 j=5 i=9 j=6 i=9 j=7 i=9 j=8 i=9 j=9 i=9 j=10
i=10 j=1 i=10 j=2 i=10 j=3 i=10 j=4 i=10 j=5 i=10 j=6 i=10 j=7 i=10 j=8 i=10 j=9 i=10 j=10

```

雙層迴圈 2

```

我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你
我愛你我恨你我恨你我恨你

```

```

public static void 雙層迴圈3() {
    for (int i = 1; i <= 10; i++) {
        System.out.print(" i=" + i + " j=");

        for (int j = 1; j <= 10; j++) {

            if (j < 10) {
                System.out.print(j + " , ");
            } else {
                System.out.print(j);
            }
            // System.out.print(j < 10 ? j + " , " : j);
        }
        System.out.println();
    }
}

```

```

i=1 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=2 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=3 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=4 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=5 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=6 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=7 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=8 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=9 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10
i=10 j=1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10

```

```

public static void 九九乘法1() {

    for (int i = 1; i <= 9; i++) {
        for (int j = 1; j <= 9; j++) {
            System.out.print(i + "*" + j + "=" + i * j + " ");
            if (i * j < 10) {
                System.out.print(" ");
            }
        }
        System.out.println();
    }
}

```

```

1*1=1 1*2=2 1*3=3 1*4=4 1*5=5 1*6=6 1*7=7 1*8=8 1*9=9
2*1=2 2*2=4 2*3=6 2*4=8 2*5=10 2*6=12 2*7=14 2*8=16 2*9=18
3*1=3 3*2=6 3*3=9 3*4=12 3*5=15 3*6=18 3*7=21 3*8=24 3*9=27
4*1=4 4*2=8 4*3=12 4*4=16 4*5=20 4*6=24 4*7=28 4*8=32 4*9=36
5*1=5 5*2=10 5*3=15 5*4=20 5*5=25 5*6=30 5*7=35 5*8=40 5*9=45
6*1=6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36 6*7=42 6*8=48 6*9=54
7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49 7*8=56 7*9=63
8*1=8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64 8*9=72
9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81

```

```

public static void 九九乘法2() {

    for (int i = 1; i <= 9; i++) {
        for (int j = 1; j <= 9; j++) {
            System.out.printf("%d*%d=%2d ", i, j, i * j);
        }
        System.out.println();
    }
}

```

```

1*1= 1 1*2= 2 1*3= 3 1*4= 4 1*5= 5 1*6= 6 1*7= 7 1*8= 8 1*9= 9
2*1= 2 2*2= 4 2*3= 6 2*4= 8 2*5=10 2*6=12 2*7=14 2*8=16 2*9=18
3*1= 3 3*2= 6 3*3= 9 3*4=12 3*5=15 3*6=18 3*7=21 3*8=24 3*9=27
4*1= 4 4*2= 8 4*3=12 4*4=16 4*5=20 4*6=24 4*7=28 4*8=32 4*9=36
5*1= 5 5*2=10 5*3=15 5*4=20 5*5=25 5*6=30 5*7=35 5*8=40 5*9=45
6*1= 6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36 6*7=42 6*8=48 6*9=54
7*1= 7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49 7*8=56 7*9=63
8*1= 8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64 8*9=72
9*1= 9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81

```

```

public static void 寫檔案() {
    try {
        FileWriter fw = new FileWriter("file1.txt", true);
        PrintWriter pw = new PrintWriter(fw);
        pw.println("AAA");
        pw.println("BBB");

        pw.flush();
        pw.close();
    } catch (IOException e) {
        System.out.println(e);
    }
}

```

```

public static void 讀檔案() {
    try {
        FileReader fr = new FileReader("file1.txt");
        BufferedReader br = new BufferedReader(fr);

        String data;
        while ((data = br.readLine()) != null) {
            System.out.println(data);
        }
    } catch (IOException e) {
        System.out.println(e);
    }
}

```

```

public static void 寫檔案_九九乘法() {
    try {
        FileWriter fw = new FileWriter("九九乘法表.txt", false);
        PrintWriter pw = new PrintWriter(fw);
        for (int i = 1; i <= 9; i++) {
            for (int j = 1; j <= 9; j++) {
                pw.printf("%d*%d=%2d ", i, j, i * j);
            }
            pw.println();
        }
        pw.flush();
        pw.close();
    } catch (IOException e) {
        System.out.println(e);
    }
}

```

跳躍敘述句

1. break ; →用在 switch · for · while · do...while 強迫跳離此 break 所在的敘述句 · 迴圈 · 無窮迴圈。

```

public static void break1() {

    for (int i = 1; i <= 10; i++) {
        if (i > 6) {
            break;
        }
        System.out.print(i + " ");
    }
}

```

1 2 3 4 5 6

```

public static void break2() {

    AA:
    for (int i = 1; i <= 10; i++) {
        for (int j = 1; j <= 10; j++) {
            if (j > 5) {
                break AA; //加上標記一次可跳兩個迴圈
            }
            System.out.print(j + " ");
        }
    }
}

```

2. continue ; →在某條件運算下 · 強迫中止某次迴圈的執行 · 利用 continue 回到迴圈的起點。

```

public static void continue1() {
    for (int i = 1; i <= 20; i++) {
        if (i % 7 == 0) {
            continue;
        }
        System.out.print(i + " ");
    }
}

```

1 2 3 4 5 6 8 9 10 11 12 13 15 16 17 18 19 20

```

public static void 輸入檢核(){

    Scanner s = new Scanner(System.in);
    int x;

    for(int i=1;i<=3;i++){
        System.out.print("x" + i + "=");
        x=s.nextInt();
        if(x<1 || x>10){
            System.out.println("error (1-10 之間)");
            i--;
            continue;
        }
    }
}

```

x1=13
error (1-10 之間)
x1=5
x2=78
error (1-10 之間)
x2=5
x3=47
error (1-10 之間)
x3=2

```

public static void 分數等級_迴圈() {
    int grade;
    Scanner s = new Scanner(System.in);
    String ans = "Y";
    do {
        System.out.print("請輸入分數 = ");
        grade = s.nextInt();

        if (grade < 0 || grade > 100) {
            System.out.println("你的分數 輸入錯誤");
            continue;
        }
        switch (grade / 10) {
            case 10:
            case 9:
                System.out.println("成績：優等");
                break;
            case 8:
                System.out.println("成績：甲等");
                break;
            case 7:
                System.out.println("成績：乙等");
                break;
            case 6:
                System.out.println("成績：丙等");
                break;
            default:
                System.out.println("成績：丁等");
        }
        System.out.print("繼續嗎：<Y/N>");
        ans = s.next();
    } while (ans.equals("Y") || ans.equals("y"));
}

```

```

public static void 輸入帳號密碼2() {
    String str1 = null;

    boolean ok = false;
    final String password = "ymca";

    Scanner s1 = new Scanner(System.in);

    for (int i = 1; i <= 3; i++) {
        System.out.print("請輸入密碼：");
        str1 = s1.next();

        if (str1.equals(password)) {
            ok = true;
            break;
        }

        System.out.println("密碼錯誤，無法進入");
        System.out.println();
    }
    if (ok) {
        System.out.println("Login成功，歡迎進入");
    } else {
        System.out.println("非法進入");
    }
}

```

請輸入分數 = 74
 成績：乙等
 繼續嗎：<Y/N>y
 請輸入分數 = 130
 你的分數 輸入錯誤
 請輸入分數 = 65
 成績：丙等
 繼續嗎：<Y/N>y
 請輸入分數 = 32
 成績：丁等
 繼續嗎：<Y/N>

請輸入密碼：abfd
 密碼錯誤，無法進入

 請輸入密碼：afsz
 密碼錯誤，無法進入

 請輸入密碼：ymca
 Login成功，歡迎進入

```

public static void 換錢2() {

    Scanner s = new Scanner(System.in);
    int money;
    int m100 = 0, m50 = 0, m10 = 0, m5 = 0, m1 = 0;

    //      System.out.print("Input money ( < 1000 ) : ");
    //      money = s.nextInt();
    //      while (money >= 1000) {
    //          System.out.println("money error");
    //          System.out.print("Input money ( < 1000 ) : ");
    //          money = s.nextInt();
    //      }

    while (true) {
        System.out.printf("Input money ( < 1000 ) : ");
        money = s.nextInt();

        if (money < 1000) {
            break;
        }
        System.out.println("Error , mony >=1000");
    }

    for (; money >= 100; money -= 100, m100++);
    for (; money >= 50; money -= 50, m50++);
    for (; money >= 10; money -= 10, m10++);
    for (; money >= 5; money -= 5, m5++);
    for (; money >= 1; money -= 1, m1++);

    System.out.println("100 元 : " + m100);
    System.out.println("50 元 : " + m50);
    System.out.println("10 元 : " + m10);
    System.out.println("5 元 : " + m5);
    System.out.println("1 元 : " + m1);
}

```

```

public static void 亂數() {
    //亂數種子的值 會分佈在 >=0 and <1 之間

    System.out.println(Math.random());
    System.out.println(Math.random());
    System.out.println(Math.random());
    System.out.println(Math.random());
    System.out.println(Math.random());
    System.out.println(Math.random());
    System.out.println((int) (Math.random() * 100) + 1);
    System.out.println((int) (Math.random() * 100) + 1);
    System.out.println((int) (Math.random() * 100) + 1);
    System.out.println((int) (Math.random() * 100) + 1);
    System.out.println((int) (Math.random() * 100) + 1);
    System.out.println((int) (Math.random() * 100) + 1);
    System.out.println((int) (Math.random() * 100) + 1);
}

```

```

Input money ( < 1000 ) : 1200
Error , mony >=1000
Input money ( < 1000 ) : 987
100 元 : 9
50 元 : 1
10 元 : 3
5 元 : 1
1 元 : 2

```

```

0.5582312466685769
0.338135535691618
0.21777207144953703
0.721959849912308
0.3891328497524945
0.6781019382580535
72
88
54
75
37
3

```

作業：

- 三數判斷→輸入 $x \cdot y \cdot z$
比較出 →三數相等，兩數相等，三數皆不相等
- 閏年 與 平年→輸入 year
條件 →可以被 4 整除 但不能被 100 整除 或 可以被 400 整除
- 旅費判斷 1→輸入 money →[3-1] if 連續效果 [3-2] if 中斷效果
<10000→金額錯誤 >=10000 在台灣玩，>=20000 在日本玩，>=40000 在美國玩，>=60000 以上 在歐洲玩
- 旅費判斷 2→輸入 money →[4-1] switch 連續效果 [4-2] switch 中斷效果
<10000→金額錯誤 >=10000 在台灣玩，>=20000 在日本玩，>=40000 在美國玩，>=60000 以上 在歐洲玩
- 汽車價錢計算

- 總分與平均_單層迴圈→grade 分數 · total 總分 · average 平均
分數 1：
分數 2：
分數 3：
總分：
平均：
- 總分與平均_雙層迴圈→grade 分數 · total 總分 · average 平均→有 3 個學生，每個學生有 3 科成績

學生 1:	學生 2:	學生 3:
分數 1:	分數 1:	分數 1:
分數 2:	分數 2:	分數 2:
分數 3:	分數 3:	分數 3:
總分:	總分:	總分:
平均:	平均:	平均:

- [8-1] 輸入 n 求 n 的質數
[8-2] 輸入 n 求 最接近 n 的最大與最小質數 → 例 最接近 100 的最大質數是 101 最小質數 97

9. 猜數字遊戲

```

public static void 猜數字遊戲() {
    Scanner s1 = new Scanner(System.in);
    int ran, input = 0, n = 0, x = 1, y = 100;

    ran = (int) (Math.random() * 100) + 1;
    System.out.println("取到的亂數是 " + ran);

    do {

    } while (ran != input);

    if (ran == input) {

    }

}

```

```

取到的亂數是 51
1-100=>2
太小
2-100=>130
不在2-100範圍內，請重新輸入
2-100=>40
太小
40-100=>65
太大
40-65=>50
太小
50-65=>55
太大
-- 歹勢，已經五次，挑戰失敗--

```

```

取到的亂數是 62
1-100=>65
太大
1-65=>23
太小
23-65=>50
太小
50-65=>62
你猜對了
你猜了4次

```

10. 存錢計算

小明・媽媽 每天給他 5 元，每次從中拿 2.5 元 存入罐中，目標是存上 100 元，但每隔 5 天，就會從罐中取出 6 元，如果一直按照這樣的規律，小明多少天才可以存到 100 元 ~

```

public static void 存到100元() {

}

```

```

第5天花去 6 元，還剩 6.5元！
第10天花去 6 元，還剩 13.0元！
第15天花去 6 元，還剩 19.5元！
第20天花去 6 元，還剩 26.0元！
第25天花去 6 元，還剩 32.5元！
第30天花去 6 元，還剩 39.0元！
第35天花去 6 元，還剩 45.5元！
第40天花去 6 元，還剩 52.0元！
第45天花去 6 元，還剩 58.5元！
第50天花去 6 元，還剩 65.0元！
第55天花去 6 元，還剩 71.5元！
第60天花去 6 元，還剩 78.0元！
第65天花去 6 元，還剩 84.5元！
第70天花去 6 元，還剩 91.0元！
要經過連續儲存 74天，才能存上 100 元！

```


11. 印出某一年的日曆

```

public static boolean isLeapYear(int year) {
    return ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0));
}

public static int firstWeekDayOfYear(int year) {
    long day = year * 365;
    for (int i = 1; i < year; i++) {
        if (isLeapYear(i)) {
            day += 1;
        }
    }
    return (int) day % 7;
}

```

請輸入一個年份2017

2017年

1月

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
01	02	03	04	05	06	07
08	09	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

2月

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			01	02	03	04
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

3月

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			01	02	03	04
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	