算術運算子：＋ － ＊ ／ ％取餘數 (前)++ (後)++ (前) - - (後) - - ＋連接子

* 特別說明  
  (前)++ : 先自增1，後運算 : a++  
  (後)++ : 先運算，後自增1 : ++a  
  (前) - - : 先自減一，後運算 : a--  
  (後) - - : 先運算，後自減一：--a

連接子：＋：只能使用在string與其他數據類型變量之間使用

賦值運算子：= += -= \*= /= %=

* 特別說明  
  1. 運算的結果不會改變變量本身的數值  
   特例: short s1 = 10;  
   s1 = s1 +1; //編譯失敗:因為1為常數，所以兩者運算會變成int值  
   s1 += 1; // 編譯成功

2. 開發中，如果希望變量實現＋２的操作，有幾種方式（前提：int num = 10;）  
 方式一: num = num +2;   
 方式二: num += 2; (推薦)

開發中，如果希望變量實現＋1的操作，有幾種方式（前提：int num = 10;）

方式一: num = num +1;   
 方式二: num += 1;

方式三: num++;(推薦)

比較運算子(關係運算子): == !=不等 > < >= <= instanceof

典型代碼:  
int i =10 ;  
int j =20;

System.out.println(I == j);//false  
System.out.println(I = j);//20  
boolean b1 = true;

boolean b2 = false;

System.out.println(b2 == b1);//false

System.out.println(b2==b1);//true

* 特別說明

1. 比較運算子的結果是boolean類型
2. > < >= <= :只能使用在數值類型的數據之間
3. == 和 != :不僅可以使用在數值類型數據之間，還可以使用在其他引用類型變量之間。

Account acct1 = new Account(1000);

Account acct2 = new Account(1000);

boolean b1 = (acct1 == acct2); //比較兩個Account是否是同一個帳戶 編譯結果是:false

Boolean b2 = (acct1 != acct2);//true

邏輯運算子: & && | || ! ^

&—逻辑与| —逻辑或！—逻辑非

&& —短路与|| —短路或^ —逻辑异或

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **a** | **b** | **a&b** | **a&&b** | **a|b** | **a||b** | **!a** | **a^b** |
| **true** | **true** | true | true | true | true | false | false |
| **true** | **false** | false | false | true | true | false | true |
| **false** | **true** | false | false | true | true | true | true |
| **false** | **false** | false | false | false | false | true | false |

特別說明:  
1. 邏輯運算子操作的都是boolean類型的變量,而且結果也都是boolean

2.

&和&&:  
相同點：&和&&運算結果相同，當符號左邊是ture時，兩者都會執行符號右邊的運算

不同點：當符號左邊是false時，&繼續執行符號右邊運算，&&不再執行符號右邊的運算  
在開發中,推薦使用&&

|和||區別同理:

相同點：&和&&運算結果相同，當符號左邊是false時，兩者都會執行符號右邊的運算

不同點：當符號左邊是false時，|繼續執行符號右邊運算，||不再執行符號右邊的運算

在開發中,推薦使用||

位運算子: << >> >>> & | ^ ~

* 特別說明

1. 位運算子操作的都是整型的數據
2. ＜＜：在一定範圍內，每向左移一位，相當於＊２  
   ＞＞：在一定範圍內，每向右移一位，向當於／２

典型題目:  
1.交換兩個變量的值 2.實現60的二進制到十六進制的轉換

三元運算子

結構：（條件表達式）？表達式１：表達式２

說明：  
1.　條件表達式的結果為boolean類型

2.　根據條件表達式真或假，決定執行表達式１還式表達式２  
 如果表達式為ｔｒｕｅ，則執行表達式１

如果表達式為ｆａｌｓｅ，則執行表達式２

　　表達式１何表達式２要求是一致的

三元運算子可以嵌套使用

1. 凡是可以使用三元運算子的地方，都可以改寫為if-else，反之不成立
2. 如果程序既可以使用三元運算子，又可以使用if-else結構，那麼優先選擇三元運算符，原因簡潔

典型題目：獲取三個數的最大值  
int n1 = 12;

int n2 = 30;

int n3 = -43;

int max1 = (n1 > n2)? n1 : n2;

int max2 = (max1 > n3)? max1 : n3;

System.out.println(＂三個數中的最大值為：＂＋ max2);