１．針對於方法內變量的賦值舉例：

System.out.println("\*\*\*\*\*\*\*\*\*\*\*基本數據類型：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**int** m = 10;

**int** n = m;

System.out.println("m = " + m + ", n = " + n);

n = 20;

System.out.println("m = " + m + ", n = " + n);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*引用數據類型：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Order o1 = **new** Order();

o1.orderId = 1001;

Order o2 = o1;//賦值以後，o1和o2的地址值相同，都指向了堆空間中同一個對象實體

System.out.println("o1.orderId = " + o1.orderId + ",o2.orderId = " +o2.orderId);

o2.orderId = 1002;

System.out.println("o1.orderId = " + o1.orderId + ",o2.orderId = " +o2.orderId);

規則：

如果變量是基本數據類型，此時賦值的是變量所保存的的數據值

如果變量是引用數據類型，此時賦值的是變量所保存的的數據的地址值

２．針對於方法的參數概念

形參：方法定義時，聲明的小括號內的參數

實參：方法調用時，實際傳遞給形參的數據

３．Java中參數傳遞機制：值傳遞

規則：

如果參數是基本數據類型，此時實參賦給形參的是實參真實存儲的數據值。

如果參數是引用數據類型，此時實參賦給形參的是實參存儲數據的地址值。

推廣：

如果變量是基本數據類型，此時賦值的是變量所保存的的數據值

如果變量是引用數據類型，此時賦值的是變量所保存的的數據的地址值

４．典型例題與內存解析：

《例題一》



《例題二》

