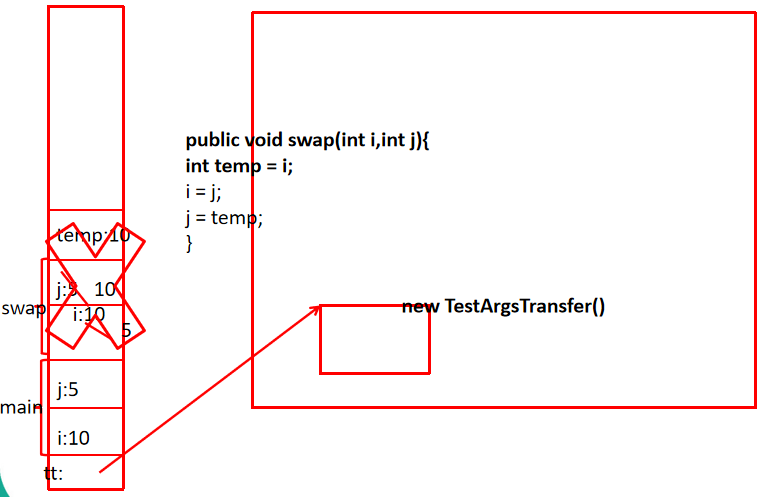
* **方法，必须有其所在类或对象调用才有意义。若方法含有参数**
* 形参：方法声明时的参数，方法小括号内的参数。
* 实参：方法调用时实际传给形参的参数值。
* **实参值如何传入方法？**

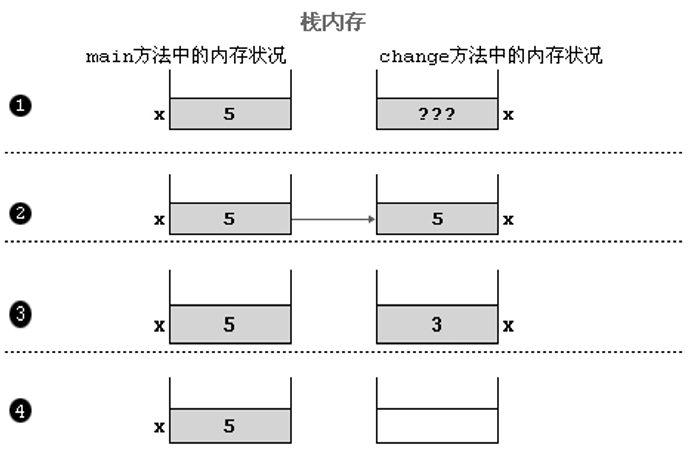
Java里方法的参数传递方式只有一种：**值传递机制**。即将实际参数值的副本（复制品）传入方法内，而参数本身不受影响。

* **方法的参数传递（形参是基本数据类型：将实参的值传递给形参的基本数据类型的变量）**

public class TestArgsTransfer {

public static void main(String[] args) {  
 int i = 10;  
 int j = 5;  
  
 TestArgsTransfer testArgsTransfer = new TestArgsTransfer();  
 testArgsTransfer.swap(i,j);  
  
 System.***out***.println("i:" + i + "\t" + "j:" + j);  
 }  
  
 /\*\*  
 \* 定义一个方法，交换两个变量的值  
 \*/  
 public void swap(int i, int j) {  
 int temp = i;  
 i = j;  
 j = temp;  
 System.***out***.println("i:" + i + "\t" + "j:" + j);  
 }  
}





* **方法的参数传递（形参是引用数据类型：将实参的引用类型变量的值(对应的堆空间的对象实体的首地址值)传递给形参的引用类型变量）**

public class TestArgsTransfer1 {

public static void main(String[] args) {  
 TestArgsTransfer1 testArgsTransfer1 = new TestArgsTransfer1();  
 DataSwap dataSwap = new DataSwap();  
  
 System.***out***.println("dataSwap.i" + dataSwap.i + " dataSwap.j" + dataSwap.j);  
  
 testArgsTransfer1.swap(dataSwap);  
 System.***out***.println("dataSwap.i" + dataSwap.i + " dataSwap.j" + dataSwap.j);  
 }  
 /\*\*  
 \* 交换元素的值  
 \*/  
 public void swap(DataSwap d){  
 int temp = d.i;  
 d.i = d.j;  
 d.j = temp;  
 }  
}  
  
class DataSwap{  
 int i = 10;  
 int j = 5;  
}

