

参数的传递方式-1

杨振平

参数的传递方式

C++中参数有三种传递方式：

- 值传递
- 引用传递
- 指针传递

参数的传递方式（续）

1. 值传递

调用时，将实参的值传递给对应的形参，这就是值传递。

由于形参有自己独立的存储空间，又作为函数的局部变量使用，因此在函数中对形参值的任何修改都不会改变实参变量的值。

▶ 好处：

减少函数之间的数据依赖，增强了函数自身的独立性。

注意：函数的形参声明为简单类型或结构类型变量，实参与形参将采用值方式传递。

例： 验证在值传递过程中，形参的改变对实参有无影响

```
void swap(int x, int y)
```

```
{ int tmp;
```

```
    tmp = x; x = y; y = tmp;
```

```
}
```

```
int main( )
```

```
{ int a = 1, b = 2;
```

```
    cout << "Before exchange: a= " << a << ",b= " << b << endl;
```

```
    swap(a, b); // 独立语句调用
```

```
    cout << "After exchange: a= " << a << ",b= " << b << endl;
```

```
    return 0;
```

```
}
```

函数的功能-交换x和y的值

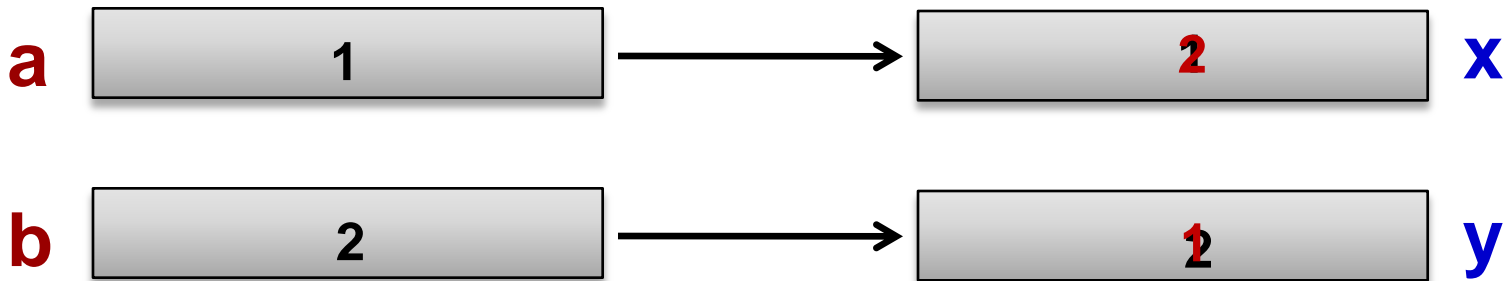
Before exchange: a= 1, b= 2

After exchange: a= 1, b= 2

结论：在值传递过程中，改变形参的值，并不会改变实参的值。

值传递过程描述如下：

- ▶ 调用前实参a, b有自己的存储空间，并有初值。
- ▶ 调用函数时，为形参x, y分配存储空间，并接收实参的值。



返回到主函数，这时实参a和b的值没有改变。

函数中，是对x与y值进行交换。
执行结束后，x与y将自动释放。