[1. C++ 中 \* 和 \*&的区别 2](#_Toc428879235)

[2. 在C++/CLI中，\*&的写法是^% 3](#_Toc428879236)

[3. C# 中的ref和out参数 4](#_Toc428879237)

[3-1. 什么时候用ref和out 4](#_Toc428879238)

[3-2. Out必须要求的调用函数内部至少对传入的对象初始化一次 4](#_Toc428879239)

[3-3. 什么时候用ref，什么时候用out 4](#_Toc428879240)

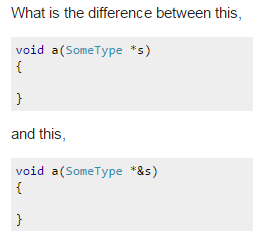
[3-4. C# ref与out关键字的使用与区别 5](#_Toc428879241)

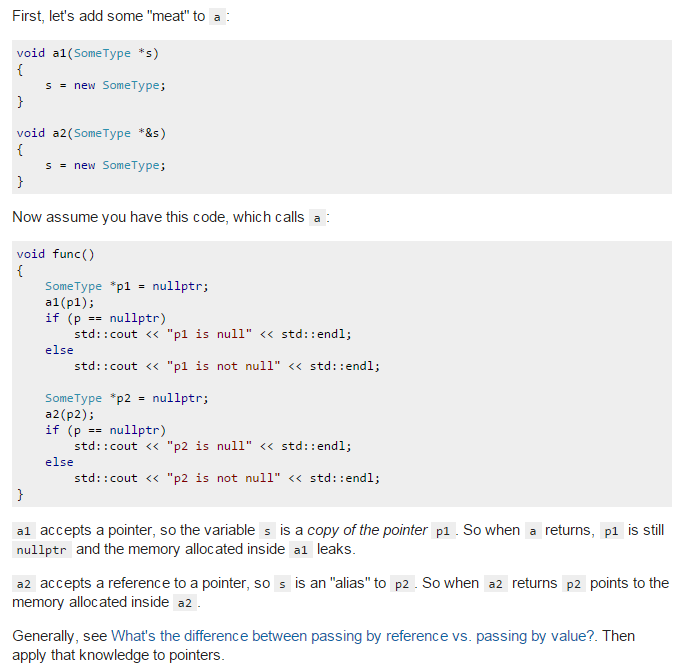
# C++ 中 \* 和 \*&的区别

其实主要的使用场景在函数参数的定义中。

即需要在函数中对传入的空指针进行初始化的情况

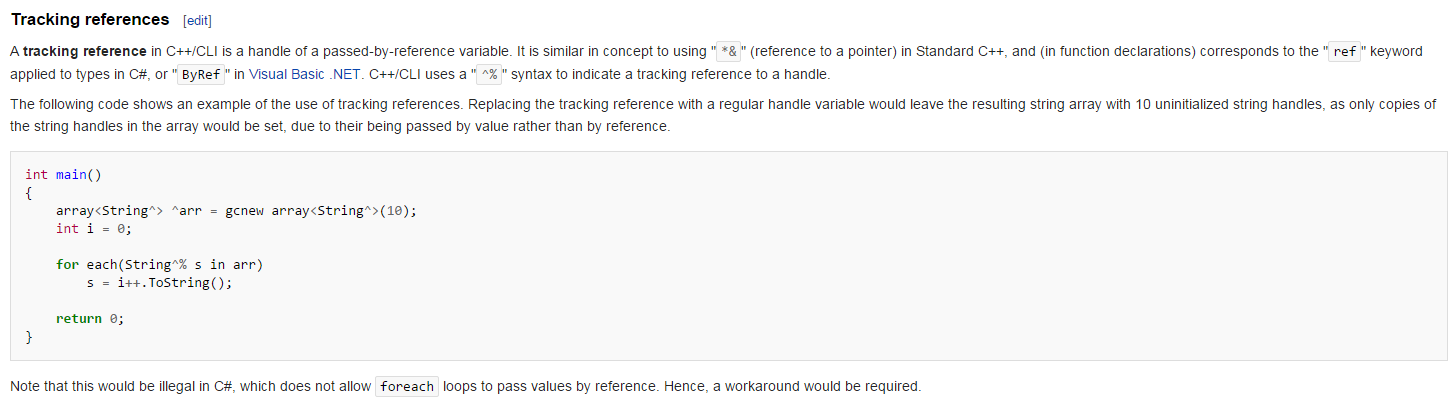
http://stackoverflow.com/questions/24718874/what-is-the-difference-between-and-in-c





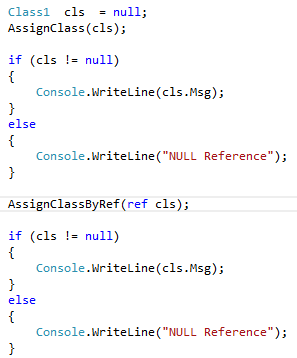
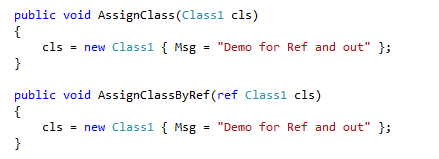
# 在C++/CLI中，\*&的写法是^%

<https://en.wikipedia.org/wiki/C%2B%2B/CLI>



# C# 中的ref和out参数

C#中也存在同样的问题，如果需要在调用函数中为一个对象赋值，则必须加ref或者out，否则函数调用结束后，对象并不会被赋值成功



## 什么时候用ref和out

Ref和out起到的作用是一样的，只是用的时候必须一致：即定义的时候用ref，调用的时候也必须使用ref，即使对于数组对象，用ref和out也都可以。

## Out必须要求的调用函数内部至少对传入的对象初始化一次

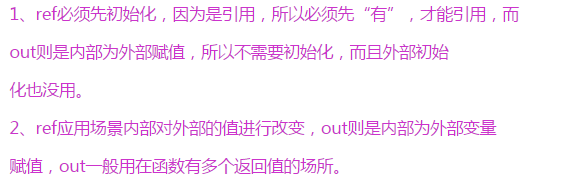
Out参数比ref参数特殊的一点是，在调用函数内部，编译器要求必须对传入的参数初始化一次，否则提示编译错误。而且必须保证在任何分之下都必须初始化一次。如果初始化是在if 判断里的，就不行。不管out参数在调用之前是否已经初始化过。

Ref则没有这个限制。

可能编译器认为out的参数必须是内部给他初始化的，所以必须初始化。

## 什么时候用ref，什么时候用out

基本的使用场景是：



对于简单数据类型，如int， 如果需要在swap函数中交换其值，则定义时用ref

对于引用数据类型，如果需要在调用函数中初始化，用out好了，虽然用ref也可以。

所以可以简单划分为： 值对象，用ref； 饮用对象，用out

## C# ref与out关键字的使用与区别

<C# ref与out关键字的使用与区别.pdf>