

常量引用、非常量引用、临时对象

 专栏收藏该内容

0 订阅 7 篇文章 订阅专栏

转载自: <https://www.cnblogs.com/littleant/archive/2012/08/01/2618846.html>
<https://www.cnblogs.com/BensonLaurip/5234555.html>
<https://blog.csdn.net/hy13684802853/article/details/87251736>

产生临时对象的情况

- 1.按值进行函数 **参数传递** 时
- 2.函数返回对象时
- 3.发生类型转化时，例如为了使函数成功调用而进行隐式 **类型转换**

C++中产生的**临时对象是不可修改的**，及**默认为const的**，**非常量引用的初始值必须是左值**，非const引用只能绑定到与该引用同类型的对象，而const引用则可以绑定到不同的但相关的对象或绑定到右值(隐式转化)，因为如果将const引用绑定到不同类型的对象（前提是被引用的类型可以转化为引用的类型），**会生成一个临时对象**，绑定的对象其实是这个临时对象，对这个临时对象操作并不会影响到被引用的对象，所以c++一般规定编译器自行构造的临时对象都是const的，这个也是c++禁止为非常量引用生成临时对象的原因。

```
1 | const int & i = 3.14;
2 | //实际过程如下
3 | const int temp(3.14);
4 | const int &i = temp;
```

C++标准的规定：**非常量的引用不能指向临时对象**

```
1 | void conv(string &str) { }
2 | int main()
3 | {
4 |     conv("dassd"); // 这里错了，编译器自动生成一个string("dassd")临时对象，不能将该临时对象传给非const引用
5 | }
```

这里将临时对象赋给了非常量引用，会报错提示非常量限定

以C++的语义来说，**如果一个程序员只想传递参数给函数，而不希望函数修改传入的参数时**，那么，或者使用**值传递**，或者采用**常量型引用(const &)**，考虑到大对象复制时产生的开销，一般使用常量型引用const &。如果函数的参数是某个类型的一个非常量的引用，那就相当于告诉编译器，程序员希望得到函数对参数的修改结果。临时变量是由编译器生成的，C++语言规范没规定编译器生成临时变量的规则，程序员无法得知由编译器生成的临时变量的名字，程序员无法访问那个临时变量。这意味着，**以引用的方式传递一个临时变量做为函数参数，如果函数内部对此临时变量做了修改，那么函数返回后，程序员无法获得函数对临时变量的修改。函数对临时变量所做出的所有更改，都将丢失**。一方面，在函数申明中，使用非常量型的引用告诉编译器你需要得到函数对某个对象的修改结果，可是你自己又不给变量起名，直接丢弃了函数的修改结果，编译器只能说：“大哥，你这是干啥呢，告诉我把结果给你，等我把结果给你了，你又直接给扔了，你这不是在玩我吗？同时，C++的标准为了防止给常量或临时变量（只有瞬间的生命周期）赋值(易产生bug)，只许使用const引用之。

因此conv函数可以改为：

```
1 | void conv(string str) { } // 值传递
2 |
3 |
4 | void conv(const string &str) { } // const引用，因为标准规定临时对象是不能更改的，所以要加上const修饰。
```

关于临时对象的总结


临时对象是有开销的，所以你应该尽可能的去除它们，然而更重要的是训练自己寻找可能建立临时对象的地方，
1.在任何时候只要见到**常量引用**（reference to const）参数，就存在建立临时对象而绑定在参数上的可能性。
2.在任何时候只要见到**返回对象**，就会有一个临时对象被建立（以后被释放）



非常量引用的初始值必须为左值的问题03-16

非常量引用的初始值必须为左值的问题 C++ 11中引入的一个**非常量**的概念就是右值引用。理解右值引用是学习“移动语义”（m...

关于“常量引用可以绑定非常量对象”的问题baidu_34263241的博客 773

引自C++PRIMER FOURTH EDITION © A nonconst reference may be only attached to an object of the same type as the referenc...

 欢迎高质量的评论，低质的评论会被折叠

  **评论**

| | |
|--|--------------------------|
| 函数参数传递 赋值调用&引用调用&非常量引用调用_wybnm... | 2-1 |
| 君希望在函数中改变其实参,则必须定义形参为 非常量引用 类型,且必须为其传递一个左值。左值就是出现在赋值操作符(operator... | |
| C++语言的常量引用问题_garulousabyss的博客_++常量引用 | 2-18 |
| 区分清楚了左值与右值,我们再来看看左值引用。左值引用根据其修饰符的不同,可以分为 非常量左值引用 (eg.1 double &r =1;)和 常... | |
| 非常量引用的初始值必须是左值 | 林多 8092 |
| 问题: void increment (double& x) {x+=1.0;} int n=5; increment(n);上述代码错误“ 非常量引用 ”的初始值必须是左值”因为 int 和do... | |
| C++ 非常量引用无效 | weixin_30781778的博客 212 |
| /* 非常量引用无效 */ #include <iostream> using namespace std; /* C++标准的规定: 非常量的引用 不能指向临时对象: 为了防止... | |
| 引用和常量引用_恩,其实你很菜...引用常量和常量引用 | 2-18 |
| const引用可以初始化为不同类型的 对象 或者初始化为右值(2.3.1节),如字面值 常量 : int i =42;// legal for const references onlycons... | |
| C++中引用、常量引用的使用,引用与指针的区别,以及什么...引用常量引用引用与指针的区别 引用引用变量 | 2-13 |
| 是一个别名,即它是某个已存在变量的另一个名字。一旦把引用初始化为某个 变量 ... | |
| 关于临时对象与常量引用、非常量引用,有轮子哥出没^_^ | znHD的博客 1004 |
| 非const引用只能绑定到与该引用同类型的 对象 。而const引用则可以绑定到不同的但相关的 对象 或绑定到右值。--C++primer 因... | |
| java中四种引用类型 | peizhiinfo 599 |
| 今天看代码，里面有一个类java.lang.ref.SoftReference把小弟弄神了，试问一下，接触java已经有3年了了哇，连lang包下面的类... | |
| C++ 常量引用用法详解_W24的博客_常量引用 | 1-26 |
| (3) 引用 的类型必须 和所引用 的类型严格匹配且不能与字面值或者某个表达式的计算结果绑定在一起,但是“ 常量引用 ”是例外(只... | |
| 为什么C++不允许将临时对象赋值给非常量引用? | whoisack的专栏 333 |
| 以C++的语义来说，如果一个程序员只想传递参数给函数，而不希望函数修改传入的参数时，那么，或者使用值传递，或者采用... | |
| java 常量对象引用_对象常量/初始化程序中的自引用_最新发布 | weixin_36382998的博客 423 |
| 有什么办法可以使以下内容在JavaScript中起作用? var foo = {a: 5, b: 6, c: this.a + this.b // Doesn't work};在当前形式中，该代码... | |
| 形参中使用常量引用，常量，普通引用的区别 | 学习记录本 4817 |
| C++中的 引用 ，给我们提供了一种区别于C指针的形参定义方式。一般我们可以将其定义为 1) 普通 引用 void foo(int &n); 2) 常... | |
| C++ 11 右值引用的理解 | werty的学习笔记 1880 |
| 虽然看得不是很懂，但是很高级的样子~~ PS: CU上面就没写出作者是谁，so不要怪我~ 右值引用的目的之一，是为了C++中一... | |
| 常量引用或指向常量的指针，其所指对象可以是非const对象 (C++) | Ivan_47的博客 769 |
| 常量引用 ：常量引用即为对const的引用，其仅对引用可参与的操作做出了限定，但对于引用的对象是否为const 常量 并未做限... | |
| 无法用"ATTRIB_GEN_STRING *"类型的值初始化"ATTRIB_GEN_NAME*"&"类型的引... | sinat_390885578的博客 2930 |
| 在ACIS中，很多下行强制（downcast）转换是安全的，如对一个ENTITY*指针强制使用(BODY*)从来没有出现过问题。... | |
| 常量指针是不是不能给非常量指针赋值? | choushengnp796550的博客 605 |
| 先上代码： #include <stdio.h> int main(int argc, const char * argv[]) { int a = 1; int *pr; const int *prr = &a; pr = pr; ... | |
| 无法用"CString"类型的值初始化"CWnd &"类型的引用(非常量限定) | sinat_40442218的博客 9051 |
| 创建MFC程序时，，显示m_strPathName,无法用"CString"类型的值初始化"CWnd &"类型的引用(非常量限定),解决方法，对DC... | |
| 常量引用和变量引用的区别理解 | 1751 |
| C++primer提到，引用就是对象的别名，换言之， | |
| 为什么C++中常量引用可以绑定非常量的对象、字面值和一般表达式 详解 | Coleum的博客 2291 |
| 在c++语言中，除两种例外情况，其他引用的类型都要和与之绑定的对象严格匹配，如int型的引用只能绑定int型的对象；并且引... | |