简单的程序诠释C++ STL算法系列之一：for_each

C++STL的非变易算法（Non-mutating algorithms）是一组不破坏操作数据的模板函数，用来对序列数据进行逐个处理、元素查找、子序列搜索、统计和匹配。

for_each用于逐个遍历容器元素，它对迭代器区间[first，last)所指的每一个元素，执行由单参数函数对象f所定义的操作。

原型：

```
template<class InputIterator, class Function>
   Function for_each(
      InputIterator _First,
      InputIterator _Last,
      Function _Func
      );
```

说明：

for_each 算法范围 [first, last) 中的每个元素调用函数 F，并返回输入的参数 f。此函数不会修改序列中的任何元素。

示例代码：

```
/*************************************************************
 * Copyright (C) Jerry Jiang
 *
 * File Name   : For_each.cpp
 * Author      : Jerry Jiang
 * Create      : 2011-9-27 19:46:44
 * Mail        : jbiaojerry@gmail.com
 * Blog        : http://blog.csdn.net/jerryjbiao
 *
 * Description : 简单的程序诠释C++ STL算法系列之一
 *   非变易算法 : 逐个遍历容器元素 for_each
 *
 *************************************************************/

#include <algorithm>
#include <list>
#include <iostream>

using namespace std;

//print为仿函数
struct print{
 int count;
 print(){count = 0;}
 void operator()(int x)
 {
  cout << x << endl;
  ++count;
 }
};

int main(void)
{
 list<int> ilist;
 //初始化
 for ( size_t i = 1; i < 10; ++i)
 {
  ilist.push_back(i);
 }
 //遍历ilist元素并打印
 print p = for_each(ilist.begin(), ilist.end(), print());
 //打印ilist元素个数
 cout << p.count << endl;

 return 0;
}
```

示例说明：

仿函数，又或叫做函数对象，是STL（标准模板库）六大组件（容器、配置器、迭代器、算法、配接器、仿函数）之一；仿函数虽然小，但却极大的拓展了算法的功能，几乎所有的算法都有仿函数版本。例如，查找算法find_if就是对find算法的扩展，标准的查找是两个元素向等就找到了，但是什么是相等在不同情况下却需要不同的定义，如地址相等，地址和邮编都相等，虽然这些相等的定义在变，但算法本身却不需要改变，这都多亏了仿函数。 仿函数之所以叫做函数对象，是因为仿函数都是定义了()函数运算操作符的类。

仿函数相关参考文章：

1、仿函数使用要领

2、何为仿函数

3、C++ 仿函数(functor)

****

C++经典书目索引及资源下载：http://blog.csdn.net/jerryjbiao/article/details/7358796

****