

原简单的程序诠释C++ STL算法系列之六：count

C++STL的非变异算法（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。

count算法用于计算容器中的某个给定值的出现次数。它有两个使用原型，均计算迭代器区间[first, last)上等于value值的元素个数n，区别在于计数n是直接返回还是引用返回。

函数原型：

```
template<class InputIterator, class Type>
    typename iterator_traits<InputIterator>::difference_type count(
        InputIterator _First,
        InputIterator _Last,
        const Type& _Val
    );
```

```
template<class InputIterator, class T> inline
    size_t count(
        InputIterator First,
        InputIterator Last,
        const T& Value
    )
```

示例代码：

```

/*****
 * Copyright (C) Jerry Jiang
 * File Name   : count.cpp
 * Author      : Jerry Jiang
 * Create Time : 2011-10-6 20:25:37
 * Mail        : jbiaojerry@gmail.com
 * Blog        : http://blog.csdn.net/jerryjbiao
 * Description : 简单的程序诠释C++ STL算法系列之六
 * 非变易算法 : 统计等于某值的容器元素个数 count
 *****/
#pragma warning(disable:4786)

#include <iostream>
#include <algorithm>
#include <list>
#include <string>
#include <vector>

using namespace std;

int main()
{
    list<int> ilist;
    for (list<int>::size_type index = 0; index < 100; ++index)
    {
        ilist.push_back( index % 20 );
    }

    list<int>::difference_type num = 0;
    int value = 9;
    num = count(ilist.begin(), ilist.end(), value);

    cout << "链表中元素等于value的元素的个数: "
         << num << endl;

    vector<string> vecString;
    vecString.push_back("this");
    vecString.push_back("is");
    vecString.push_back("a");
    vecString.push_back("test");
    vecString.push_back("program");
    vecString.push_back("is");

    string valString("is");

    ptrdiff_t result = count(vecString.begin(), vecString.end(), valString);

    cout << "容器中元素为is的元素个数: "
         << result << endl;

    return 0;
}

```

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