

简单的程序诠释C++ STL算法系列之九：equal

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。

equal算法类似于mismatch，equal算法也是逐一比较两个序列的元素是否相等，只是equal函数的返回值为bool值true/false，不是返回迭代器值。它有如下两个原型，如果迭代器区间[first1, last1)和迭代器区间[first2, first2+(last1 - first1))上的元素相等（或者满足二元谓词判断条件binary_pred），返回true，否则返回false。

函数原型：

```
template<class InputIterator1, class InputIterator2>
bool equal(
    InputIterator1 _First1,
    InputIterator1 _Last1,
    InputIterator2 _First2
);
template<class InputIterator1, class InputIterator2, class BinaryPredicate>
bool equal(
    InputIterator1 _First1,
    InputIterator1 _Last1,
    InputIterator2 _First2,
    BinaryPredicate _Comp
);
```

示例程序：

利用二元谓词判断条件absEqual，判断出两个vector向量容器的元素均绝对值相等。

```
/******
 * Copyright (C) Jerry Jiang
 * File Name   : equal.cpp
 * Author      : Jerry Jiang
 * Create Time : 2011-10-10 22:51:07
 * Mail        : jbiaojerry@gmail.com
 * Blog        : http://blog.csdn.net/jerryjbiao
 * Description : 简单的程序诠释C++ STL算法系列之九
 *              非变易算法 : 元素相等判断 equal
 *****/

#include <algorithm>
#include <vector>
#include <iostream>

using namespace std;

bool absEqual(int a, int b)
{
    return (a == abs(b) || b == abs(a)) ? true : false;
}

int main()
{
    vector<int> ivect1(5);
    vector<int> ivect2(5);

    for (vector<int>::size_type i = 0; i < ivect1.size(); ++i)
    {
        ivect1[i] = i;
        ivect2[i] = (-1) * i;
    }
    if ( equal( ivect1.begin(), ivect1.end(), ivect2.begin(), absEqual ) )
    {
        cout << "ivect1 和 ivect2 元素的绝对值完全相等" << endl;
    }
    else
    {
        cout << "ivect1 和 ivect2 元素的绝对值不完全相等" << endl;
    }
    return 0;
}
```

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

[阅读更多](#) [登录后自动展开](#)