CSDN 学院

C++语言基础

迁者 - 贺利坚

http://blog.csdn.net/sxhelijian/

http://edu.csdn.net

本节主题：

STL简介

STL是C++中泛型编程的实现

- STL：标准模板库，Standard Template Library
  - STL是一些容器——list、vector、set、map等的集合
  - STL是算法和其它一些组件的集合
  - 1996年，惠普公司免费公开了STL
  - STL是C++标准库的一个重要组成部分
- STL的主要组件
  - container（容器）
  - algorithm（算法）
  - iterator（迭代器）
  - 函数对象（function object）
- STL规模宏大，而且可以扩充，包含很多计算机基本算法和数据结构，而且将算法与数据结构完全分离，其中算法是泛型的，不与任何特定数据结构或对象类型系在一起

STL的容器

容器	所在头文件	示例
向量vector	<vector>	vector<int>
双端队列deque	<deque>	deque<double>
表list	<list>	list<Student>
队列queue	<queue>	queue<int>
堆栈stack	<stack>	
集合set	<set>	
多重集合multiset	<set>	
映射map	<map>	
多重集合multimap	<map>	

```
#include <vector>
using namespace std;
vector<int> intVector(100);
int main()
{
    intVector[20] = 50;
    ……
```

STL算法

- 算法主要包括头文件algorithm、 numeric、 functional
  - <algorithm>由一大堆模版函数组成的，每个函数有很强的独立性，涉及到比较、交换、查找、遍历操作、复制、修改、移除、反转、排序、合并等等
  - <numeric>包括加法和乘法在序列上的操作
  - <functional>定义了一些模板类，用以声明函数对象
- STL的算法非常优秀的，开发中很多相似的函数不必自己开发，只要用好函数模板
  - 高效
  - 优质

```
int iarray[SIZE];
int main()
{
   iarray[20] = 50;
   int* ip = find(iarray,iarray+SIZE,50);
```

STL迭代器

迭代器是泛化的指针，STL算法利用迭代器对存储在容器中的元素序列进行遍历，迭代器提供了访问每个元素的方法。

输入迭代器 Input iterator	向前读 Reads forward	istream
输出迭代器 Output iterator	向前写 Writes forward	ostream, inserter
前向迭代器 Forward iterator	向前读写 Read and Writes forward	
双向迭代器 Bidirectional iterator	向前向后读写 Read and Writes forward and backward	list, set, multiset, map, multimap
随机迭代器 Random access iterator	随机读写 Read and Write with random access	vector,deque,array, string

```
vector<int> intVector(100);
vector<int>::iterator intIter = find(intVector.begin(), intVector.end(), 50);
if (intIter != intVector.end()) ……
```

vector容器类示例

v.insert(pos,n,elem)：在pos位置插入n个elem数据
v.insert(pos,beg,end)：在pos位置插入在[beg,end)区间的数据
v.erase(pos)：删除pos位置的数据
v.erase(beg,end)：删除[beg,end)区间的数据

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    int a[] = {2,3,4};
    vector<int> v1;
    vector<int>::iterator iter;
    //插入元素

    //删除元素

    return 0;
}
```

```
v1.insert(v1.begin(),1,1);
v1.insert(v1.begin(),a,a+3);
v1.insert(v1.begin()+4,2,5);
cout<<"data: ";
for(iter = v1.begin() ; iter != v1.end() ; ++iter)
    cout<<*iter<<" ";
cout<<endl<<endl;
```

```
v1.erase(v1.begin(),v1.begin()+2);
v1.erase(v1.begin()+1);
cout<<"data: ";
for(iter = v1.begin() ; iter != v1.end() ; ++iter)
    cout<<*iter<<" ";
cout<<endl<<endl;
```

CSDN 学院

THANKS

本课程由 迁者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：http://ems.csdn.net/
人才服务：http://job.csdn.net/
CTO俱乐部：http://cto.csdn.net/
高校俱乐部：http://student.csdn.net/
程序员杂志：http://programmer.csdn.net/

CODE平台：https://code.csdn.net/
项目外包：http://www.csto.com/
CSDN博客：http://blog.csdn.net/
CSDN论坛：http://bbs.csdn.net/
CSDN下载：http://download.csdn.net/