STL 笔记:

What is STL?

STL(标准模板库)是一套强大的C++模板类，提供了通用的模板类和函数，这些模板类和函数可以实现多种流行和常用的算法和数据结构。

组成部件:

容器: list,vector,map

算法:对容器类容的初始化，排序，搜索和转换

迭代器:用于便利操作

在c++标准中，STL被组织为下面的13个头文件：<algorithm>、<deque>、<functional>、<iterator>、<vector>、<list>、<map>、

<memory>、<numeric>、<queue>、<set>、<stack>和<utility>。

个人觉得，STL的思想为：

容器内放数据，只是数据结构，算法是一些高阶算法，两者是独立的，迭代器是其纽带，让开发者可以利用容器的迭代器来调用其相关算法。

1. Vector

Vector基于模板实现，需要包含头文件<vector>

1. push\_back   在数组的最后添加一个数据  
   2.pop\_back    去掉数组的最后一个数据   
   3.at                得到编号位置的数据  
   4.begin           得到数组头的指针  
   5.end             得到数组的最后一个单元+1的指针  
   6．front        得到数组头的引用  
   7.back            得到数组的最后一个单元的引用  
   8.max\_size     得到vector最大可以是多大  
   9.capacity       当前vector分配的大小  
   10.size           当前使用数据的大小  
   11.resize         改变当前使用数据的大小，如果它比当前使用的大，者填充默认值  
   12.reserve      改变当前vecotr所分配空间的大小  
   13.erase         删除指针指向的数据项  
   14.clear          清空当前的vector  
   15.rbegin        将vector反转后的开始指针返回(其实就是原来的end-1)  
   16.rend          将vector反转构的结束指针返回(其实就是原来的begin-1)  
   17.empty        判断vector是否为空  
   18.swap         与另一个vector交换数据

测试代码:



代码测试:

//============================================================================

// Name : fiu.cpp

// Author : sun2019

// Version :

// Copyright : Your copyright notice

// Description : Hello World in C++, Ansi-style

//============================================================================

#include <iostream>

#include<vector>

using namespace std;

int main() {

//初始化

vector<int> a;

vector<int> a1(10); //10个元素

vector<int> a2(10, 1); //10个大小为1的元素

vector<int> a3(a); //用a来初始化a3

int initarray[] = { 0, 1, 2, 3, 4, 5 }; //用数组来初始化

vector<int> a4(initarray, initarray + 5);

for (int i = 0; i < 5; i++)

cout << "a4[" << i << "]" << a4[i] << endl;

vector<int>::iterator t;

cout << "迭代器" << endl;

for (t = a4.begin(); t != a4.end(); t++) {

cout << \*t << endl;

}

//在begin处+一个1000

a2.insert(a2.begin(), 1000);

for (t = a2.begin(); t != a2.end(); t++) {

cout << "a2" << \*t << endl;

}

//在begin处 到 begin+5处添加5个100

a3.insert(a3.begin(), 5, 100);

for (t = a3.begin(); t != a3.end(); t++) {

cout << "a3" << \*t << endl;

}

//将a4全部元素添加到a

a.insert(a.begin(), a4.begin(), a4.end());

for (t = a.begin(); t != a.end(); t++) {

cout << "a" << \*t << endl;

}

a.erase(a.begin()); //将第一个元素删除

a.erase(a.begin(),a.begin()+3); //删除其之间的全部元素

return 0;

}

结果截图:



1. List

Lists将元素按顺序储存在链表中. 与 向量(vectors)相比, 它允许快速的插入和删除，但是随机访问却比较慢.  
assign() 给list赋值   
back() 返回最后一个元素   
begin() 返回指向第一个元素的迭代器   
clear() 删除所有元素   
empty() 如果list是空的则返回true   
end() 返回末尾的迭代器   
erase() 删除一个元素   
front() 返回第一个元素   
get\_allocator() 返回list的配置器   
insert() 插入一个元素到list中   
max\_size() 返回list能容纳的最大元素数量   
merge() 合并两个list   
pop\_back() 删除最后一个元素   
pop\_front() 删除第一个元素   
push\_back() 在list的末尾添加一个元素   
push\_front() 在list的头部添加一个元素   
rbegin() 返回指向第一个元素的逆向迭代器   
remove() 从list删除元素   
remove\_if() 按指定条件删除元素   
rend() 指向list末尾的逆向迭代器   
resize() 改变list的大小   
reverse() 把list的元素倒转   
size() 返回list中的元素个数   
sort() 给list排序   
splice() 合并两个list   
swap() 交换两个list   
unique() 删除list中重复的元素