第三周C/C++组技术分享

我们这周主要研究了容器相关的知识，另补充一点关于0的事情。

具体内容如下：

0. “0”

C中 宏NULL 0指针

C++中 收紧的类型检查规则 如想用NULL 用const int NULL=0；

1.容器

向量vector

1 基本操作

(1)头文件#include<vector>.

(2)创建vector对象，vector<int> vec;

(3)尾部插入数字：vec.push\_back(a);

(4)使用下标访问元素，cout<<vec[0]<<endl;记住下标是从0开始的。

(5)使用迭代器访问元素.

vector<int>::iterator it;

for(it=vec.begin();it!=vec.end();it++)

cout<<\*it<<endl;

(6)插入元素： vec.insert(vec.begin()+i,a);在第i+1个元素前面插入a;

(7)删除元素： vec.erase(vec.begin()+2);删除第3个元素

vec.erase(vec.begin()+i,vec.end()+j);删除区间[i,j-1];区间从0开始

(8)向量大小:vec.size();

(9)清空:vec.clear();

vector的元素不仅仅可以使int,double,string,还可以是结构体，但是要注意：结构体要定义为全局的，否则会出错。

Vec 提供范围检查

vector<Entry>v2=phone\_book;

int i=phone\_book[1001].num; 并不提示出错，而是赋给i随机值

Vec抛出一个out\_of\_range的异常

使异常最小化的方式 用一个try-块作为体

表-list

检索这个表，寻找给定值的元素，倾向于不采用下表访问元素 push\_front() push\_back()

映射-map（值的对偶的容器）

关联数组、字典

map<string,int>phone\_book;用第一个类型的某个值去索引，返回第二个类型的对应值。

标准容器（）

vector<T> 变长向量

list<t> 双向链表

queue<T> 队列

stack<T> 堆栈

deque<T> 双端队列

priority\_queue<T> 按值排序的队列

set<T> 集合

multiset<T> 集合，值可重复出现

map<key,value> 关联数组

multimap<key,value> 关联数组，关键字可重复出现

size() 返回容器中函数个数