Map容器

Map容器是关联型容器，键值对应着实值，所有元素根据键值自动排序。Map中的所有元素都是pair对组，pair的第一个元素为键值，第二个元素为实值，Map不允许有相同的键值，但允许有相同的实值。

与set容器一样，不允许通过迭代器修改map的键值，因为键值与map的排列有关，如果修改键值，那么map的结构会被破坏。Map的实值是可以修改的。

二．Multimap

Multimap与map的操作特性类似，位移不同就是multimap允许键值重复。

Map和multimap都是用红黑树作为底层实现的。

三，map的常用操作

1. map构造函数

map<T1, T2> mapTT;//map默认构造函数:

map(const map &mp);//拷贝构造函数

1. map赋值操作

map& operator=(const map &mp);//重载等号操作符

swap(mp);//交换两个集合容器

1. map大小操作

size();//返回容器中元素的数目

empty();//判断容器是否为空

1. map插入元素操作

map.insert(...); //往容器插入元素，返回pair<iterator,bool>

map<int, string> mapStu;

// 第一种 通过pair的方式插入对象

mapStu.insert(pair<int, string>(3, "小张"));

// 第二种 通过pair的方式插入对象

mapStu.inset(make\_pair(-1, "校长"));

// 第三种 通过value\_type的方式插入对象

mapStu.insert(map<int, string>::value\_type(1, "小李"));

// 第四种 通过数组的方式插入值

mapStu[3] = "小刘";

mapStu[5] = "小王";

1. map删除操作

clear();//删除所有元素

erase(pos);//删除pos迭代器所指的元素，返回下一个元素的迭代器。

erase(beg,end);//删除区间[beg,end)的所有元素 ，返回下一个元素的迭代器。

erase(keyElem);//删除容器中key为keyElem的对组。

1. map查找操作

find(key);//查找键key是否存在,若存在，返回该键的元素的迭代器；/若不存在，返回map.end();

count(keyElem);//返回容器中key为keyElem的对组个数。对map来说，要么是0，要么是1。对multimap来说，值可能大于1。

lower\_bound(keyElem);//返回第一个key>=keyElem元素的迭代器。

upper\_bound(keyElem);//返回第一个key>keyElem元素的迭代器。

equal\_range(keyElem);//返回容器中key与keyElem相等的上下限的两个迭代器。