1. set的基本概念

简介：

- 所有元素都会在插入时自动排序

本质：

- set/multiset 属于关联式容器，底层结构是用二叉树实现。

set / multiset的区别：

- set不允许有重复的元素；
- multise允许有重复的元素。

2.set的构造和赋值

构造：

- set<Elemtype> st; //默认构造函数
- set<const set &st>; //拷贝构造函数

```
set<int> st;    //创建set容器

//插入数据，只有insert方式
st.insert(12);
st.insert(41);
st.insert(2);
st.insert(5);

//遍历容器
for(set<int>::iterator it = st.begin(); it != st.end(); it++)
    cout<<*it<<" ";
```

结果（自动排序）：

```
2 5 12 41
```

字符串的排序（字典序排序）：

```
//插入数据，只有insert方式
st.insert("abj");
st.insert("abc");
st.insert("abd");
st.insert("das");
```

```
Microsoft Visual Studio
abc abd abj das
F:\VS\草稿\Debug\草稿
要在调试停止时自动关
按任意键关闭此窗口.
```

3.常用函数接口

- st.size(); //返回容器元素个数
- st.empty(); //容器判空

- st.swap(st0); //交换st和st0容器中的元素
- st.erase(st.begin()); //删除迭代器begin位置的元素
- st.erase("das"); //直接删除元素“das”
- st.clear(); //清空容器中的元素
- st.find(key); //查找元素key是否存在，存在返回该元素的迭代器；不存在，返回st.end()
- st.count(key); //查找元素key的个数，并返回(set容器只有一个，multiset可以有多个)

4.使用set构造结构体，对结构体排序

首先：在main函数之前写一个仿函数

如下：

```
class comparePerson
{
public:
bool operator()(const Person& p1, const Person& p2) const
{
//按照年龄升序排序（若降序排序只要将 < 改成 > ）
return p1.age < p2.age;
}
};
```

同时在创建容器时使用如下命名方式：

```
set<Person,comparePerson> st; //创建set容器
```

之后，将结构体插入set容器中，就会自动排序

详细代码如下：

```cpp
class comparePerson
{
public:
    bool operator()(const Person& p1, const Person& p2) const
    {
        //按照年龄降序排序
        return p1.age < p2.age;
    }
};
int main()
{
    set<Person,comparePerson> st;    //创建set容器
    int n;
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        Person p;
        cin >> p.name >> p.age;
        st.insert(p);
    }
    for (auto it = st.begin(); it != st.end(); it++)
    {
        cout << it->name << " " << it->age << endl;
    }
```

结果如下:

```
4
张飞 23
关羽 18
刘备 15
赵云 14
赵云 14
刘备 15
关羽 18
张飞 23
```