```java
package cn.sxt.collection;
import java.util.Map;
import java.util.TreeMap;
/**
  * 测试TreeMap的使用
  * @author 江
  *
  */
public class TestTreeMap {
   public static void main(String[] args) {
          Map<Integer,String> treemap1=new TreeMap<>();
          treemap1.put(20,"aa");
          treemap1.put(6, "bb");
          treemap1.put(3, "cc");

          //按照key递增的方式进行排序
          for(Integer key:treemap1.keySet()) {
                    System.out.println(key+"--"+treemap1.get(key));
          }

          Map<Emp,String> treemap2=new TreeMap<>();
          treemap2.put(new Emp(100,"张三",50000), "张三不错");
          treemap2.put(new Emp(200,"李四",5000), "李四不行");
          treemap2.put(new Emp(500,"王五",10000), "王五还行");
          treemap2.put(new Emp(50,"赵六",10000), "赵六是个开心果");

          //按照Emp的方式进行排序
          for(Emp key:treemap2.keySet()) {
                    System.out.println(key+"--"+treemap2.get(key));
          }
```

```
    }
}
class Emp implements Comparable<Emp>{
        int id;
        String name;
        double salary;

        public Emp(int id, String name, double salary) {
                super();
                this.id = id;
                this.name = name;
                this.salary = salary;
        }

        @Override
        public String toString() {
                return "id"+id+"name"+name+"salary"+salary;
        }
        @Override
        public int compareTo(Emp o) { //负数：小
于                      0：等于                正数：大于
                if(this.salary>o.salary) {
                        return 1;
                }else if(this.salary<o.salary) {
                        return -1;
                }else {
                        if(this.id>o.id) {
                                return 1;
                        }else if(this.id<o.id){
```

```
                    return -1;
                }else {
                    return 0;
                }
            }

        }


}
```