题目：

1. 建立一个实体类Student类，属性：姓名，年龄，成绩，班级

建立一个list1，包含 “张三，18岁，80分，1班”，“李四，19岁，100分，1班”，“王五，17岁，59分，1班”。

建立一个list2，包含 “赵六，18岁，85分，2班”，“刘七，19岁，93分，2班”，“孙八，17岁，55分，2班”。

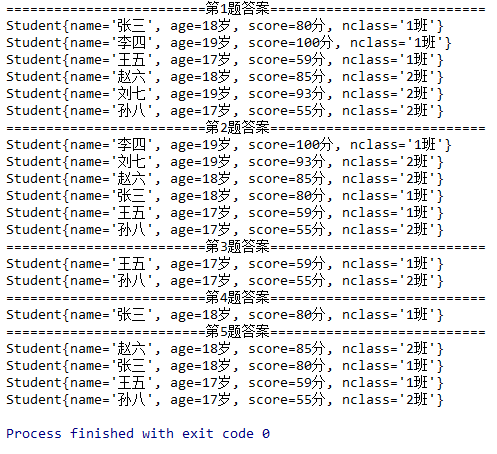
1. 整合两个list学生信息成一个新的list
2. 按照分数给出学生信息排名
3. 输出不及格的学生信息
4. 查找张三的信息
5. 从list剔除年龄大于18岁的学生信息
6. 使用Map 完成练习1的习题。

3.仿照手机淘宝，设计订单和商品的实体类。

*/\*\*  
 \* Student实体类  
 \*  
 \** ***@author*** *haoao  
 \*/***public class** Student {  
 */\*\*  
 \* 姓名  
 \*/* **private** String **name**;  
 */\*\*  
 \* 年龄  
 \*/* **private int age**;  
 */\*\*  
 \* 成绩  
 \*/* **private int score**;  
 */\*\*  
 \* 班级  
 \*/* **private** String **nclass**;  
  
 **public** Student() {  
 }  
  
 **public** Student(String name, **int** age, **int** score, String nclass) {  
 **this**.**name** = name;  
 **this**.**age** = age;  
 **this**.**score** = score;  
 **this**.**nclass** = nclass;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
 **public int** getAge() {  
 **return age**;  
 }  
  
 **public void** setAge(**int** age) {  
 **this**.**age** = age;  
 }  
  
 **public float** getScore() {  
 **return score**;  
 }  
  
 **public void** setScore(**int** score) {  
 **this**.**score** = score;  
 }  
  
 **public** String getNclass() {  
 **return nclass**;  
 }  
  
 **public void** setNclass(String nclass) {  
 **this**.**nclass** = nclass;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "Student{"** +  
 **"name='"** + **name** + **'\''** +  
 **", age="** + **age** + **"岁"** +  
 **", score="** + **score** + **"分"** +  
 **", nclass='"** + **nclass** + **'\''** +  
 **'}'**;  
 }  
}

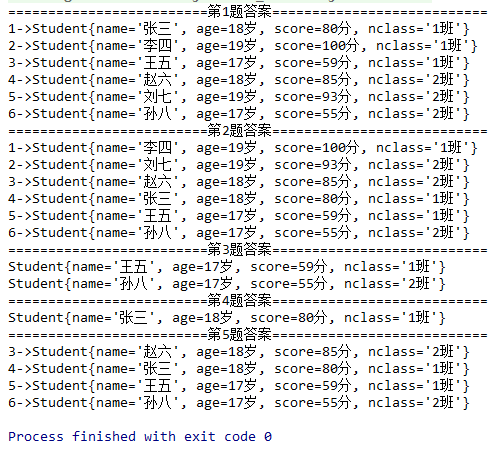
**public class** Demo1 {  
 **public static void** main(String[] args) {  
 List<Student> list1 = **new** ArrayList<>();  
 Student s1 = **new** Student(**"张三"**, 18, 80, **"1班"**);  
 Student s2 = **new** Student(**"李四"**, 19, 100, **"1班"**);  
 Student s3 = **new** Student(**"王五"**, 17, 59, **"1班"**);  
  
 list1.add(s1);  
 list1.add(s2);  
 list1.add(s3);  
  
 List<Student> list2 = **new** ArrayList<>();  
 Student s4 = **new** Student(**"赵六"**, 18, 85, **"2班"**);  
 Student s5 = **new** Student(**"刘七"**, 19, 93, **"2班"**);  
 Student s6 = **new** Student(**"孙八"**, 17, 55, **"2班"**);  
  
 list2.add(s4);  
 list2.add(s5);  
 list2.add(s6);  
  
 List<Student> list3 = **new** ArrayList<>();  
 list3.addAll(list1);  
 list3.addAll(list2);  
 System.***out***.println(**"=========================第1题答案==========================="**);  
 **for** (Student student : list3) {  
 System.***out***.println(student);  
 }  
  
 *//用Collections里的sort方法进行排序* Collections.*sort*(list3, **new** Comparator<Student>() {  
 *//重写比较的规则* @Override  
 *//根据成绩降序排序* **public int** compare(Student o1, Student o2) {  
 **return** (**int**) (o2.getScore() - o1.getScore());  
 }  
 });  
 System.***out***.println(**"=========================第2题答案==========================="**);  
 *//按照分数给出学生信息排名* **for** (Student student : list3) {  
 System.***out***.println(student);  
 }  
  
 System.***out***.println(**"=========================第3题答案==========================="**);  
 *//输出不及格的学生信息* **for** (Student student : list3) {  
 **if** (student.getScore() < 60) {  
 System.***out***.println(student);  
 }  
 }  
  
 System.***out***.println(**"=========================第4题答案==========================="**);  
 *//查找张三的信息* **for** (Student student : list3) {  
 **if** (student.getName() == **"张三"**) {  
 System.***out***.println(student);  
 }  
 }  
   
 *//从list剔除年龄大于18岁的学生信息* List<Student> listx = **new** ArrayList<>();  
 **for** (Student student : list3) {  
 **if** (student.getAge() > 18) {  
 listx.add(student);  
 }  
 }  
 System.***out***.println(**"=========================第5题答案==========================="**);  
 list3.removeAll(listx);  
 **for** (Student student : list3) {  
 System.***out***.println(student);  
 }  
 }  
}

运行结果：



*/\*\*  
 \* 使用 Map 完成练习1的习题。  
 \*/***public class** Demo2 {  
 **public static void** main(String[] args) {  
 Map<Integer, Student> map1 = **new** HashMap<>();  
 Map<Integer, Student> map2 = **new** HashMap<>();  
 Map<Integer, Student> map3 = **new** HashMap<>();  
  
 map1.put(1, **new** Student(**"张三"**, 18, 80, **"1班"**));  
 map1.put(2, **new** Student(**"李四"**, 19, 100, **"1班"**));  
 map1.put(3, **new** Student(**"王五"**, 17, 59, **"1班"**));  
  
 map2.put(4, **new** Student(**"赵六"**, 18, 85, **"2班"**));  
 map2.put(5, **new** Student(**"刘七"**, 19, 93, **"2班"**));  
 map2.put(6, **new** Student(**"孙八"**, 17, 55, **"2班"**));  
 System.***out***.println(**"=========================第1题答案==========================="**);  
 *//整合到一个新的map* map3.putAll(map1);  
 map3.putAll(map2);  
 **for** (Map.Entry<Integer, Student> entry : map3.entrySet()) {  
 Integer key = entry.getKey();  
 Student value = entry.getValue();  
 System.***out***.println(key + **"->"** + value);  
 }  
  
 *//将value取出来放进list中* ArrayList<Student> list = **new** ArrayList<>();  
 **for** (Map.Entry<Integer, Student> en : map3.entrySet()) {  
 list.add(en.getValue());  
 }  
 *//用Collections里的sort方法进行排序* Collections.*sort*(list, **new** Comparator<Student>() {  
 *//重写比较的规则* @Override  
 *//根据成绩降序序排序* **public int** compare(Student o1, Student o2) {  
 **return** (**int**) (o2.getScore() - o1.getScore());  
 }  
 });  
 *//排序后重新put进map3* **for** (**int** i = 0; i < list.size(); i++) {  
 map3.put(i + 1, list.get(i));  
 }  
 System.***out***.println(**"=========================第2题答案==========================="**);  
 *//遍历* **for** (Map.Entry<Integer, Student> entry : map3.entrySet()) {  
 Integer key = entry.getKey();  
 Student value = entry.getValue();  
 System.***out***.println(key + **"->"** + value);  
 }  
 System.***out***.println(**"=========================第3题答案==========================="**);  
  
 *//输出不及格的学生信息* **for** (Map.Entry<Integer, Student> entry : map3.entrySet()) {  
 **if** (entry.getValue().getScore() < 60) {  
 System.***out***.println(entry.getValue());  
 }  
 }  
 System.***out***.println(**"=========================第4题答案==========================="**);  
  
 **for** (Map.Entry<Integer, Student> entry : map3.entrySet()) {  
 **if** (entry.getValue().getName() == **"张三"**) {  
 System.***out***.println(entry.getValue());  
 }  
 }  
  
 System.***out***.println(**"=========================第5题答案==========================="**);  
 *//从list剔除年龄大于18岁的学生信息* Set<Integer> set = map3.keySet();  
 ArrayList list1 = **new** ArrayList<Integer>();  
 **for** (Integer integer : set) {  
 **if** (map3.get(integer).getAge() > 18) {  
 list1.add(integer);  
 }  
 }  
 **for** (**int** i = 1; i <= list1.size(); i++) {  
 map3.remove(i);  
 }  
 **for** (Map.Entry<Integer, Student> entry : map3.entrySet()) {  
 System.***out***.println(entry.getKey() + **"->"** + entry.getValue());  
 }  
  
  
 }  
}

运行结果：



仿照手机淘宝，设计订单和商品的实体类。

*/\*\*  
 \* 订单实体类  
 \** ***@author*** *haoao  
 \*/***public class** Order {  
 */\*\*  
 \* 订单id  
 \*/* **private int orderId**;  
 */\*\*  
 \* 交易号  
 \*/* **private** String **businessNum**;  
 */\*\*  
 \* 订单状态 付款/没付款  
 \*/* **private boolean state**;  
 */\*\*  
 \* 物流动态  
 \*/* **private** String **logistics**;  
 */\*\*  
 \* 创建时间  
 \*/* **private** Date **createTime**;  
 */\*\*  
 \* 付款时间  
 \*/* **private** Date **paymentTime**;  
 */\*\*  
 \* 买家名字  
 \*/* **private** String **buyerName**;  
 */\*\*  
 \* 买家地址  
 \*/* **private** String **buyerAddress**;  
 */\*\*  
 \* 买家电话  
 \*/* **private** String **buyerPhone**;  
 */\*\*  
 \* 卖家名字  
 \*/* **private** String **sallerName**;  
 */\*\*  
 \* 卖家名字  
 \*/* **private** String **sallerPhone**;  
 */\*\*  
 \* 商品各种信息  
 \*/* **private** List<Goods> **goods**;  
}

*/\*\*  
 \* 商品实体类  
 \** ***@author*** *haoao  
 \*/***public class** Goods {  
 */\*\*  
 \* 商品id  
 \*/* **private int goodsId**;  
 */\*\*  
 \* 商品名称  
 \*/* **private** String **goodsName**;  
 */\*\*  
 \* 商品上级id  
 \*/* **private int parentId**;  
 */\*\*  
 \* 商品价格  
 \*/* **private double goodsPrice**;  
 */\*\*  
 \* 商品库存  
 \*/* **private int goodsStock**;  
 */\*\*  
 \* 商品图片  
 \*/* **private int**[] **goodPicture**;  
 */\*\*  
 \* 商品款式  
 \*/* **private** String **goodsStyle**;  
 */\*\*  
 \* 商品重量  
 \*/* **private** String **goodseight**;  
 */\*\*  
 \* 商品产地  
 \*/* **private** String **goodsPlace**;  
 */\*\*  
 \* 商品简述  
 \*/* **private** String **goodsSketch**;  
}