Java基础练习3

1. 建立一个实体类Student类，属性：姓名，年龄，成绩，班级

package test;  
  
*/\*\*  
 \* 学生类  
 \* psvip  
 \* 2020/3/4 19:00  
 \*/*public class Student implements Comparable<Student> {  
 */\*\*  
 \* 姓名  
 \*/* private String name;  
 */\*\*  
 \* 年龄  
 \*/* private int age;  
 */\*\*  
 \* 成绩  
 \*/* private int score;  
 */\*\*  
 \* 班级  
 \*/* private String classNum;  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public Integer getAge() {  
 return age;  
 }  
  
 public void setAge(Integer age) {  
 this.age = age;  
 }  
  
 public Integer getScore() {  
 return score;  
 }  
  
 public void setScore(Integer score) {  
 this.score = score;  
 }  
  
 public String getClassNum() {  
 return classNum;  
 }  
  
 public void setClassNum(String classNum) {  
 this.classNum = classNum;  
 }  
  
 @Override  
 public String toString() {  
 return "Student{" +  
 "name='" + name + '\'' +  
 ", age=" + age +  
 ", score=" + score +  
 ", grade='" + classNum + '\'' +  
 '}';  
 }  
  
 */\*\*  
 \* 重写compareTo方法，成绩按照从大到小排序  
 \*/* @Override  
 public int compareTo(Student stu) {  
 return stu.getScore() - this.score;  
 }  
}

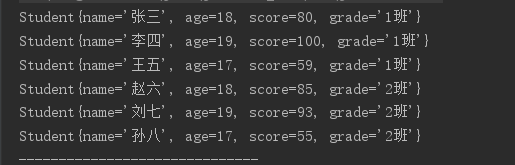
建立一个list1，包含 “张三，18岁，80分，1班”，“李四，19岁，100分，1班”，“王五，17岁，59分，1班”。

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”。

List<Student> list1 = new ArrayList<>();  
Student student1 = new Student();  
student1.setName("张三");  
student1.setAge(18);  
student1.setClassNum("1班");  
student1.setScore(80);  
list1.add(student1);  
  
Student student2 = new Student();  
student2.setName("李四");  
student2.setAge(19);  
student2.setClassNum("1班");  
student2.setScore(100);  
list1.add(student2);  
  
Student student3 = new Student();  
student3.setName("王五");  
student3.setAge(17);  
student3.setClassNum("1班");  
student3.setScore(59);  
list1.add(student3);  
  
List<Student> list2 = new ArrayList<>();  
  
Student student4 = new Student();  
student4.setName("赵六");  
student4.setAge(18);  
student4.setClassNum("2班");  
student4.setScore(85);  
  
list2.add(student4);  
  
Student student5 = new Student();  
student5.setName("刘七");  
student5.setAge(19);  
student5.setClassNum("2班");  
student5.setScore(93);  
list2.add(student5);  
  
Student student6 = new Student();  
student6.setName("孙八");  
student6.setAge(17);  
student6.setClassNum("2班");  
student6.setScore(55);  
list2.add(student6);

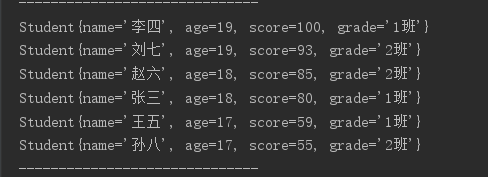
1. 整合两个list学生信息成一个新的list

List<Student> newlist=new ArrayList<>();  
newlist.addAll(list1);  
newlist.addAll(list2);  
//整合两个list  
for(Student ls:newlist){  
 System.*out*.println(ls);  
}



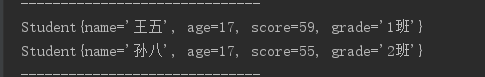
1. 按照分数给出学生信息排名

Collections.*sort*(newlist);  
for(int i=0;i<newlist.size();i++){  
 System.*out*.println(newlist.get(i));  
}



1. 输出不及格的学生信息

for(Student st :newlist){  
 if(st.getScore()<60){  
 System.*out*.println(st);  
 }  
}



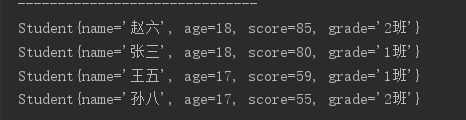
1. 查找张三的信息

for(Student st:newlist){  
 if("张三".equals(st.getName())){  
 System.*out*.println(st);  
 }  
}



1. 从list剔除年龄大于18岁的学生信息

List<Student> deleteList=new ArrayList<>();  
 for(int i=0;i<newlist.size();i++){  
 if(newlist.get(i).getAge()<=18){  
 deleteList.add(newlist.get(i));  
 }  
 }  
 //剔除年龄大于18岁的信息  
for(Student list:deleteList){  
 System.*out*.println(list);  
}



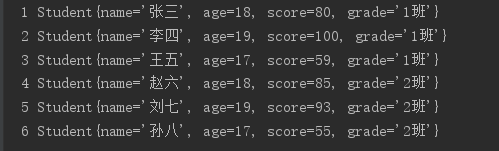
1. 使用Map 完成练习1的习题。

用map1，map2建立学生类

Map<Integer,Student> map1=new HashMap<>();  
  
Student student1 = new Student();  
student1.setName("张三");  
student1.setAge(18);  
student1.setClassNum("1班");  
student1.setScore(80);  
map1.put(1,student1);  
Student student2 = new Student();  
student2.setName("李四");  
student2.setAge(19);  
student2.setClassNum("1班");  
student2.setScore(100);  
  
map1.put(2,student2);  
Student student3 = new Student();  
student3.setName("王五");  
student3.setAge(17);  
student3.setClassNum("1班");  
student3.setScore(59);  
map1.put(3,student3);  
  
Map<Integer,Student> map2=new HashMap<>();  
Student student4 = new Student();  
student4.setName("赵六");  
student4.setAge(18);  
student4.setClassNum("2班");  
student4.setScore(85);  
map2.put(4,student4);  
  
Student student5 = new Student();  
student5.setName("刘七");  
student5.setAge(19);  
student5.setClassNum("2班");  
student5.setScore(93);  
map2.put(5,student5);  
  
Student student6 = new Student();  
student6.setName("孙八");  
student6.setAge(17);  
student6.setClassNum("2班");  
student6.setScore(55);  
map1.put(6,student6);

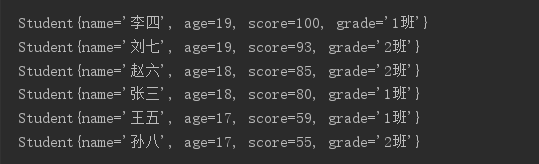
1. 整合两个map

Map<Integer,Student> newmap=new HashMap<>();  
 newmap.putAll(map1);  
 newmap.putAll(map2);  
 //合并两个map生成新的map  
 for (Map.Entry<Integer, Student> entry : newmap.entrySet()) {  
 System.*out*.println(entry.getKey() + " " + entry.getValue());  
 }



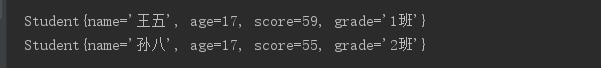
1. 按照分数给出学生信息排名

List<Map.Entry<Integer, Student>> list = new ArrayList<>(newmap.entrySet());  
Collections.*sort*(list, (o1, o2) -> o2.getValue().getScore().compareTo(o1.getValue().getScore()));  
for (Map.Entry<Integer, Student> entry : list) {  
 System.*out*.println(entry.getValue());  
}



1. 输出不及格学生信息

for (Map.Entry<Integer, Student> entry : newmap.entrySet()) {  
 if (entry.getValue().getScore() < 60) {  
 System.*out*.println(entry.getValue());  
 }  
}



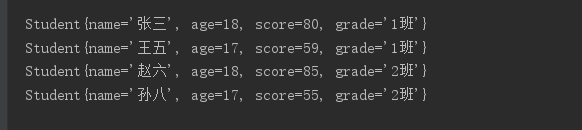
1. 查找张三信息

for (Map.Entry<Integer, Student> entry : newmap.entrySet()) {  
 if (entry.getValue().getName().equals("张三")) {  
 System.*out*.println(entry.getValue());  
 }  
}



（5）从map中剔除年龄大于18岁的学生信息

for (Iterator<Map.Entry<Integer, Student>> it = newmap.entrySet().iterator(); it.hasNext(); ) {  
 Map.Entry<Integer, Student> entry = it.next();  
 if (entry.getValue().getAge()>18) {  
 it.remove();  
 }  
}  
for (Map.Entry<Integer, Student> entry : newmap.entrySet()) {  
 System.*out*.println(entry.getValue());  
}



1. 仿照手机淘宝，设计订单和商品的实体类。

订单类

package test;  
  
*/\*\*  
 \* 订单  
 \* psvip  
 \* 2020/3/4 22：00  
 \*/*public class Order {  
 */\*\*  
 \* 订单号  
 \*/* private String orderNumber;  
 */\*\*  
 \* 订单状态  
 \*/* private String orderStatus;  
  
 */\*\*  
 \* 买家姓名  
 \*/* private String buyerName;  
 */\*\*  
 \* 买家电话  
 \*/* private String buyerPhoneNumber;  
 */\*\*  
 \* 买家地址  
 \*/* private String buyerAddress;  
  
 */\*\*  
 \* 商家名称  
 \*/* private String merchants;  
 */\*\*  
 \* 商品信息  
 \*/* private String goodsInformation;  
 */\*\*  
 \* 商品总价  
 \*/* private float goodsprize;  
 */\*\*  
 \* 运费  
 \*/* private float freight;  
 */\*\*  
 \* 实付价格  
 \*/* private float totalPrize;  
  
 */\*\*  
 \* 支付宝交易号  
 \*/* private String alipayaTransactionNumber;  
 */\*\*  
 \* 创建时间  
 \*/* private String createTime;  
 */\*\*  
 \* 付款时间  
 \*/* private String payTime;  
 */\*\*  
 \* 发货时间  
 \*/* private String deliveryTime;  
 */\*\*  
 \* 成交时间  
 \*/* private String dealTime;  
}

商品类

package test;  
  
*/\*\*  
 \* 商品  
 \* psvip  
 \* 2020/3/4 22:15  
 \*/*public class Goods {  
 */\*\*  
 \*商品名  
 \*/* private String goodsName;  
 */\*\*  
 \* 商品id  
 \*/* private String goodsId;  
 */\*\*  
 \* 商品价格  
 \*/* private String goodsPrize;  
 */\*\*  
 \* 商品介绍  
 \*/* private String goodsInformation;  
 */\*\*  
 \* 商品销量  
 \*/* private int goodsSale;  
 */\*\*  
 \* 商品库存  
 \*/* private int goodInventory;  
  
 */\*\*  
 \* 商品产地  
 \*/* private String goodsAddress;  
 */\*\*  
 \* 商品生产时间  
 \*/* private String goodsCreateTime;  
 */\*\*  
 \* 商品保质期  
 \*/* private int goodsLife;  
 */\*\*  
 \* 商品上架时间  
 \*/* private String goodsShelfTime;  
 */\*\*  
 \*商品上架人  
 \*/* private String goodsShelfPeople;  
}