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

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”。

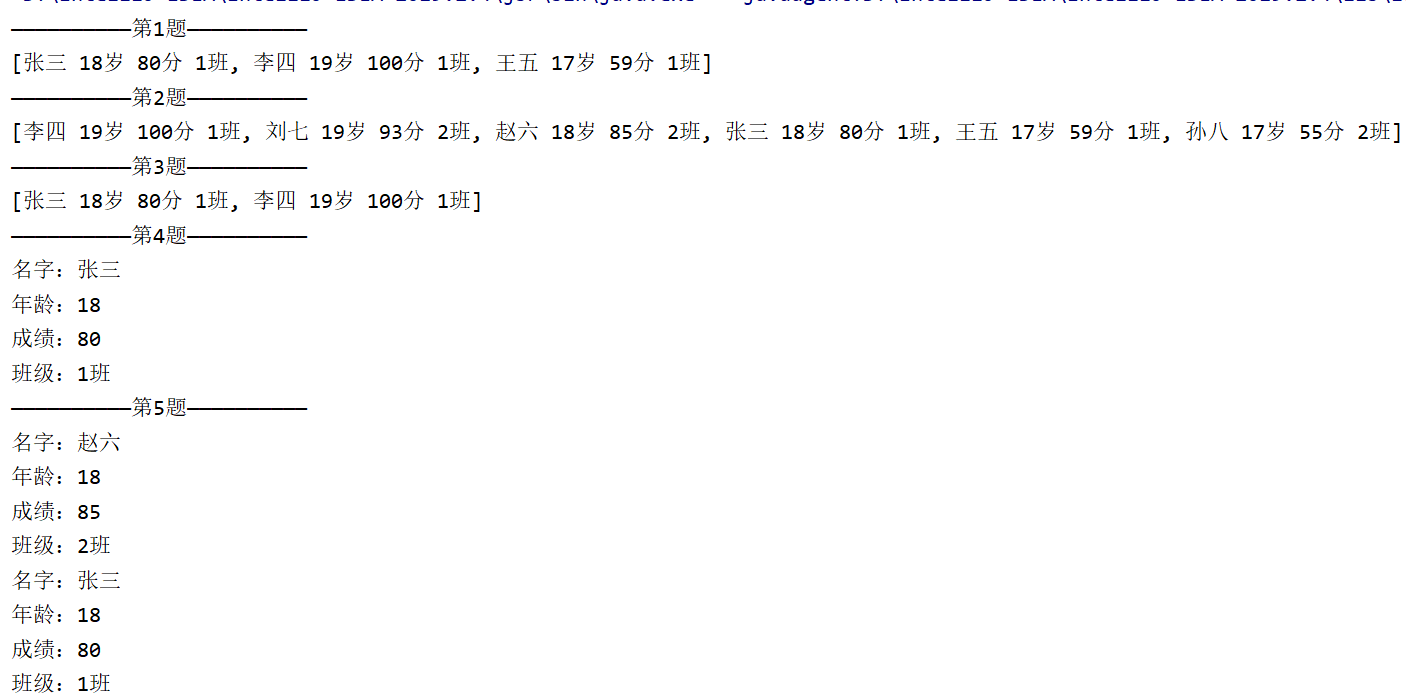
建立一个list2，包含 “赵六，18岁，85分，2班”，“刘七，19岁，93分，2班”，“孙八，17岁，55分，2班”。

**package** test;  
  
*/\*\*  
 \* 学生信息实体类  
 \* hechusheng  
 \* 2020/03/04/18:00  
 \*/***public class** Student {  
 */\*  
 学生姓名  
 \*/* **private** String **studentName**;  
 */\*  
 学生年龄  
 \*/* **private int age**;  
 */\*  
 成绩  
 \*/* **private int score**;  
 */\*  
 班级  
 \*/* **private** String **grade**;  
  
  
 **public** String getStudentName() {  
 **return studentName**;  
 }  
  
 **public void** setStudentName(String studentName) {  
 **this**.**studentName** = studentName;  
 }  
  
 **public int** getAge() {  
 **return age**;  
 }  
  
 **public void** setAge(**int** age) {  
 **this**.**age** = age;  
 }  
  
 **public int** getScore() {  
 **return score**;  
 }  
  
 **public void** setScore(**int** score) {  
 **this**.**score** = score;  
 }  
  
 **public** String getGrade() {  
 **return grade**;  
 }  
  
 **public void** setGrade(String grade) {  
 **this**.**grade** = grade;  
 }  
  
 **public** Student(String studentName, **int** age, **int** score, String grade){  
 **super**();  
 **this**.**studentName** = studentName;  
 **this**.**age** = age;  
 **this**.**score** = score;  
 **this**.**grade** = grade;  
 }

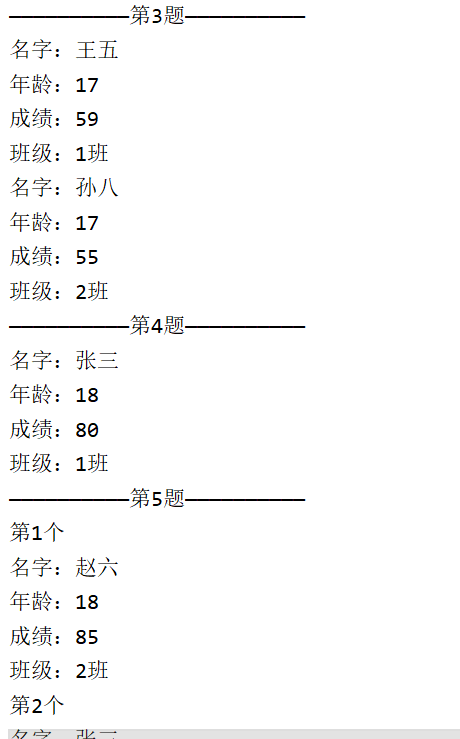
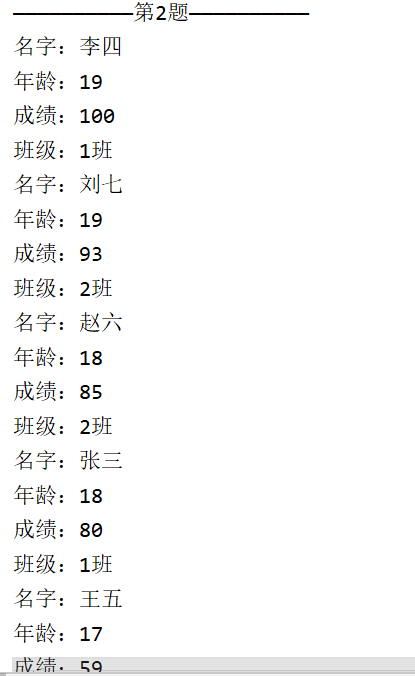
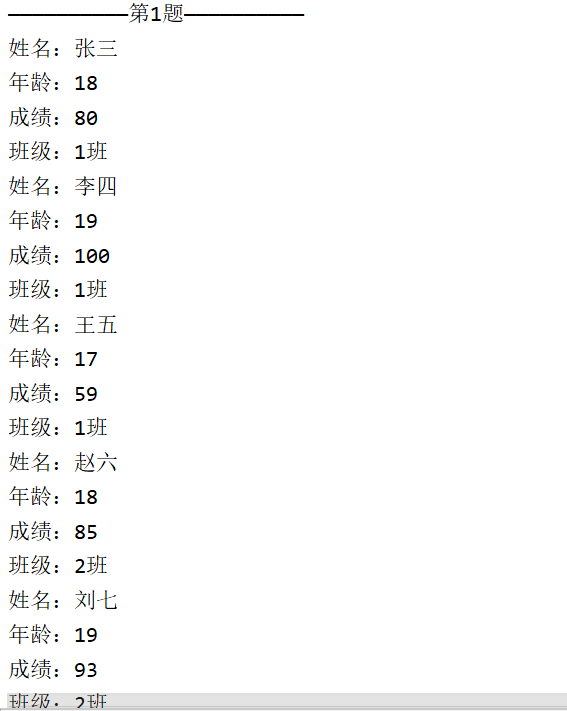
@Override  
 **public** String toString(){  
 **return** (getStudentName() +**" "**+ getAge() +**"岁"**+**" "**+ getScore() +**"分"**+**" "**+ getGrade());  
}  
}

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

**package** test;  
  
**import** java.awt.print.Pageable;  
**import** java.util.\*;  
**import** java.util.function.Predicate;  
  
*/\*\*  
 \* 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”。  
 \*  
 \* hechusheng  
 \*/***public class** Test6 {  
  
  
 **public static void** main(String[] args) {  
 List <Student> list1 = **new** ArrayList<>();  
  
 list1.add(**new** Student(**"张三"**, 18, 80, **"1班"**));  
 list1.add(**new** Student(**"李四"**, 19, 100, **"1班"**));  
 list1.add(**new** Student(**"王五"**, 17, 59, **"1班"**));  
 *//System.out.println(list1.toString());* List <Student> list2 = **new** ArrayList<>();  
 list2.add(**new** Student(**"赵六"**, 18, 85, **"2班"**));  
 list2.add(**new** Student(**"刘七"**, 19, 93, **"2班"**));  
 list2.add(**new** Student(**"孙八"**, 17, 55, **"2班"**));  
  
  
 */\*\*  
 \* （1）整合两个list学生信息成一个新的list  
 \*/* System.***out***.println(**"——————————第1题——————————"**);  
 List<Student> newList = **new** ArrayList<>();  
 newList.addAll(list1);  
 newList.addAll(list2);  
 System.***out***.println(list1.toString());  
  
  
 */\*\*  
 \* （2）按照分数给出学生信息排名  
 \*/* System.***out***.println(**"——————————第2题——————————"**);  
 Collections.*sort*(newList,(v1,v2)->v2.getScore()-v1.getScore());  
 System.***out***.println(newList.toString());  
  
  
 */\*\*  
 \* （3）输出不及格的学生信息  
 \*/* System.***out***.println(**"——————————第3题——————————"**);  
 **for**(**int** i = 1; i <= list1.size(); i++){  
 **if**(list1.get(i).getScore()<60){  
 list1.remove(i);  
 System.***out***.println(list1.toString());  
 }  
 }  
  
  
 */\*\*  
 \* （4）查找张三的信息  
 \*/* System.***out***.println(**"——————————第4题——————————"**);  
 **for**(Student student:newList){  
 **if** (**"张三"**.equals(student.getStudentName())){  
 System.***out***.println(**"名字："**+ student.getStudentName());  
 System.***out***.println(**"年龄："**+ student.getAge());  
 System.***out***.println(**"成绩："**+ student.getScore());  
 System.***out***.println(**"班级："**+ student.getGrade());  
 **break**;  
 }  
 }  
  
  
 */\*\*  
 \* （5）从list剔除年龄大于18岁的学生信息  
 \*/* System.***out***.println(**"——————————第5题——————————"**);  
 List<Student> list3 = *filter*(newList,(str)->{**return** str.getAge()<=18;});  
 **int** count2 = 1;  
 **for**(Student student:list3){  
 System.***out***.println(**"名字："**+ student.getStudentName());  
 System.***out***.println(**"年龄："**+ student.getAge());  
 System.***out***.println(**"成绩："**+ student.getScore());  
 System.***out***.println(**"班级："**+ student.getGrade());  
 }  
 }  
 **private static** List<Student> filter(List<Student> list, Predicate<Student> pre){  
 List<Student> results = **new** ArrayList<>();  
 **for**(Student r:list){  
 **if**(pre.test(r)) *//测试是否符合要求* {  
 results.add(r);  
 }  
 }  
 **return** results;  
 }  
  
}



1. 使用Map 完成练习1的习题。
2. **package** test;  
     
   **import** java.util.HashMap;  
   **import** java.util.Map;  
   **import** java.util.Set;  
     
   */\*\*  
    \* 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”。  
    \*  
    \* hechusheng  
    \* 2020/03/04 19:00  
    \*/***public class** Test7 {  
    **public static void** main(String[] args) {  
    Map<Integer, Student> map = **new** HashMap<>();  
    **int** i = 0;  
    map.put(i++, **new** Student(**"张三"**, 18, 80, **"1班"**));  
    map.put(i++, **new** Student(**"李四"**, 19, 100, **"1班"**));  
    map.put(i++, **new** Student(**"王五"**, 17, 59, **"1班"**));  
    Map<Integer, Student> map2 = **new** HashMap<>();  
    map2.put(i++, **new** Student(**"赵六"**, 18, 85, **"2班"**));  
    map2.put(i++, **new** Student(**"刘七"**, 19, 93, **"2班"**));  
    map2.put(i++, **new** Student(**"孙八"**, 17, 55, **"2班"**));  
     
    */\*\*  
    \* （1）整合两个list学生信息成一个新的list  
    \*/* System.***out***.println(**"——————————第1题——————————"**);  
    Map<Integer, Student> map3 = **new** HashMap<>();  
    map3.putAll(map);  
    map3.putAll(map2);  
    Set<Map.Entry<Integer, Student>> entrySet = map3.entrySet();  
    **int** count = 1;  
    **for** (Map.Entry<Integer, Student> entry : entrySet) {  
    Student student = entry.getValue();  
    System.***out***.println(**"姓名："** + student.getStudentName());  
    System.***out***.println(**"年龄："** + student.getAge());  
    System.***out***.println(**"成绩："** + student.getScore());  
    System.***out***.println(**"班级："** + student.getGrade());  
     
    }  
     
     
     
    */\*\*  
    \* （2）按照分数给出学生信息排名  
    \*/* System.***out***.println(**"——————————第2题——————————"**);  
    **for** ( i = 0; i < map3.size(); i++) {  
    Student student;  
    **for** (**int** j = i + 1 ; j < map3.size() ; j++){  
    **if** (map3.get(i).getScore()<map3.get(j).getScore()){  
    student = map3.get(i);  
    map3.replace(i,map3.get(j));  
    map3.replace(j,student);  
    }  
    }  
    student = map3.get(i);  
    System.***out***.println(**"名字："**+ student.getStudentName());  
    System.***out***.println(**"年龄："**+ student.getAge());  
    System.***out***.println(**"成绩："**+ student.getScore());  
    System.***out***.println(**"班级："**+ student.getGrade());  
    }  
     
     
    */\*\*  
    \* （3）输出不及格的学生信息  
    \*/* System.***out***.println(**"——————————第3题——————————"**);  
    Student findStu = **null**;  
     
    Map<Integer,Student> map4 = **new** HashMap<>();  
    **for** (Map.Entry<Integer,Student > entry:entrySet){  
    Student student = entry.getValue();  
    **if** (student.getScore()<60){  
    System.***out***.println(**"名字："**+ student.getStudentName());  
    System.***out***.println(**"年龄："**+ student.getAge());  
    System.***out***.println(**"成绩："**+ student.getScore());  
    System.***out***.println(**"班级："**+ student.getGrade());  
    }  
    **if** (**"张三"**.equals(student.getStudentName())){ *//查找张三的信息* findStu = student;  
    }  
     
    **if** (student.getAge()<=18){ *//将年龄小于等于18的学生信息加入map4* map4.put(count++,student);  
    }  
    }  
     
     
    */\*\*  
    \* （4）查找张三的信息  
    \*/* System.***out***.println(**"——————————第4题——————————"**);  
    **if** (findStu!=**null**){  
    System.***out***.println(**"名字："**+ findStu.getStudentName());  
    System.***out***.println(**"年龄："**+ findStu.getAge());  
    System.***out***.println(**"成绩："**+ findStu.getScore());  
    System.***out***.println(**"班级："**+ findStu.getGrade());  
    }  
     
     
    */\*\*  
    \* （5）从list剔除年龄大于18岁的学生信息  
    \*/* System.***out***.println(**"——————————第5题——————————"**);  
    Set<Map.Entry<Integer,Student>> entrySet2 = map4.entrySet();  
    **int** count3 = 1;  
    **for** (Map.Entry<Integer,Student> entry:entrySet2){  
    Student student = entry.getValue();  
    System.***out***.println(**"第"**+ count3++ +**"个"**);  
    System.***out***.println(**"名字："**+ student.getStudentName());  
    System.***out***.println(**"年龄："**+ student.getAge());  
    System.***out***.println(**"成绩："**+ student.getScore());  
    System.***out***.println(**"班级："**+ student.getGrade());  
    }  
     
    }  
   }



1. 仿照手机淘宝，设计订单和商品的实体类。

*/\*\*  
 \* 订单实体类  
 \* hechusehng  
 \* 2020/03/04/20:00  
 \*/***public class** Order {  
 */\*\*  
 \* 店铺名称  
 \*/* **private** String **storeName**;  
 */\*\*  
 \* 商品类型  
 \*/* **private** String **productType**;  
 */\*\*  
 \*购买商品数量  
 \*/* **private int purchaseQuantity**;  
 */\*\*  
 \* 配送方式  
 \*/* **private** String **deliveryMethod**;  
 */\*\*  
 \* 订单备注  
 \*/* **private** String **Note**;  
 */\*\*  
 \* 购买人  
 \*/* **private** String **Buyer**;  
 */\*\*  
 \* 购买时间  
 \*/* **private** String **buyTime**;  
 */\*\*  
 \* 商品单价  
 \*/* **private int Price**;  
 */\*\*  
 \* 订单总价  
 \*/* **private int totalPrice**;  
 */\*\*  
 \* 商品款式  
 \*/* **private** String **Style**;

*/\*\*  
 \* 商品实体类  
 \* hechusheng  
 \* 2020/03/04/20:21  
 \*/***public class** Product {  
 */\*\*  
 \* 商品编号  
 \*/* **private int ne\_products\_id**;  
 */\*\*  
 \* 商品类型编号  
 \*/* **private int ne\_classify\_id**;  
 */\*\*  
 \* 商品品牌编号  
 \*/* **private int ne\_brand\_id**;  
 */\*\*  
 \* 商品型号  
 \*/* **private** String **ne\_model**;  
 */\*\*  
 \* 商品介绍  
 \*/* **private** String **ne\_introduce**;  
 */\*\*  
 \* 商品描述  
 \*/* **private** String **ne\_bewrite**;  
 */\*\*  
 \* 商品小图  
 \*/* **private** String **ne\_print**;  
 */\*\*  
 \* 商品中图  
 \*/* **private** String **ne\_picture**;  
 */\*\*  
 \* 商品价格  
 \*/* **private double ne\_price**;  
 */\*\*  
 \* 商品库存  
 \*/* **private int ne\_number**;  
 */\*\*  
 \* 商品上传时间  
 \*/* **private** String **ne\_time**;  
 */\*\*  
 \* 商品状态编号  
 \*/* **private int ne\_status\_id**;  
 */\*\*  
 \* //上架和下架，0代表上架，1代表下架  
 \*/* **private** String **ne\_grounding**;