Java基础练习3

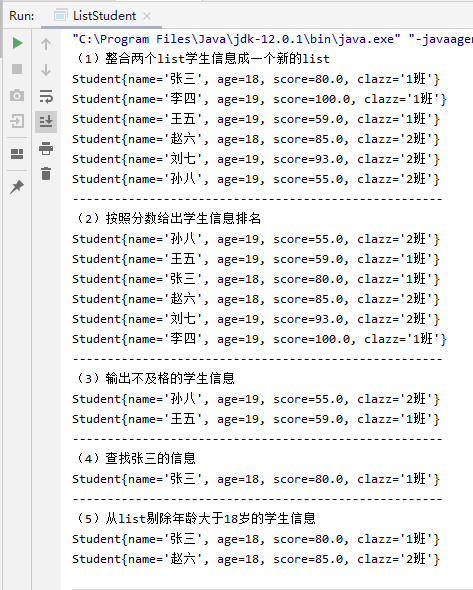
1. 建立一个实体类Student类，属性：姓名，年龄，成绩，班级

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** java3;  
  
*/\*\*  
 \* 学生实体类  
 \*/***public class** Student {  
 *//属性：姓名，年龄，成绩，班级  
 /\*\*  
 \* 学生姓名  
 \*/* **private** String **name**;  
 */\*\*  
 \* 学生年龄  
 \*/* **private int age**;  
 */\*\*  
 \* 学生成绩  
 \*/* **private float score**;  
 */\*\*  
 \* 学生班级  
 \*/* **private** String **classs**;  
  
  
  
 **public** Student(String name, **int** age, **float** score, String classs) {  
 **this**.**name** = name;  
 **this**.**age** = age;  
 **this**.**score** = score;  
 **this**.**classs** = classs;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "Student{"** +  
 **"name='"** + **name** + **'\''** +  
 **", age="** + **age** +  
 **", score="** + **score** +  
 **", clazz='"** + **classs** + **'\''** +  
 **'}'**;  
 }  
  
 **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(**float** score) {  
  
 **this**.**score** = score;  
 }  
  
 **public** String getClazz() {  
  
 **return classs**;  
 }  
  
 **public void** setClazz(String clazz) {  
 **this**.**classs** = clazz;  
 }  
}

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. **package** java3;  
     
   **import** java3.Student;  
     
   **import** java.util.ArrayList;  
   **import** java.util.Collections;  
   **import** java.util.Comparator;  
   **import** java.util.List;  
     
   */\*\*  
    \* 使用List集合操作Student对象  
    \*/***public class** ListStudent {  
    **public static void** main(String[] args) {  
    List<Student> list1 = **new** ArrayList<>();  
    List<Student> list2 = **new** ArrayList<>();  
    *//添加Student对象* 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"**, 19, 59, **"1班"**));  
    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"**, 19, 55, **"2班"**));  
     
   *//（1）整合两个list学生信息成一个新的list* List<Student> list3 = *fun1*(list1, list2);  
    *//（2）按照分数给出学生信息排名  
    fun2*(list3);  
    *//（3）输出不及格的学生信息  
    fun3*(list3);  
    *//（4）查找张三的信息  
    fun4*(list3);  
    *//（5）从list剔除年龄大于18岁的学生信息  
    fun5*(list3);  
    }  
     
     
   *//（1）整合两个list学生信息成一个新的list* **public static** List<Student> fun1(List<Student> list1, List<Student> list2){  
    System.***out***.println(**"（1）整合两个list学生信息成一个新的list"**);  
    List<Student> list3 = **new** ArrayList<>();  
    list3.addAll(list1);  
    list3.addAll(list2);  
    **for** (Student student : list3) {  
    System.***out***.println(student.toString());  
    }  
    System.***out***.println(**"-----------------------------------------------------"**);  
    **return** list3;  
    }  
     
    *//（2）按照分数给出学生信息排名* **public static void** fun2(List<Student> list3){  
    System.***out***.println(**"（2）按照分数给出学生信息排名"**);  
    Collections.*sort*(list3, **new** Comparator<Student>() {  
    @Override  
    **public int** compare(Student student1, Student student2) {  
    **float** score1 = student1.getScore();  
    **float** score2 = student2.getScore();  
    **if** (score1 > score2) {  
    **return** 1;  
    } **else if** (score1 == score2) {  
    **return** 0;  
    } **else** {  
    **return** -1;  
    }  
    }  
    });  
    **for** (Student student : list3) {  
    System.***out***.println(student.toString());  
    }  
    System.***out***.println(**"-----------------------------------------------------"**);  
    }  
     
    *//（3）输出不及格的学生信息* **public static void** fun3(List<Student> list3){  
    System.***out***.println(**"（3）输出不及格的学生信息"**);  
    **for** (Student student : list3) {  
    **if** (student != **null** && student.getScore() < 60) {  
    System.***out***.println(student.toString());  
    }  
    }  
    System.***out***.println(**"-----------------------------------------------------"**);  
    }  
     
    *//（4）查找张三的信息* **public static void** fun4(List<Student> list3){  
    System.***out***.println(**"（4）查找张三的信息"**);  
    **for** (Student student : list3) {  
    **if** (**"张三"**.equals(student.getName())) {  
    System.***out***.println(student.toString());  
    }  
    }  
    System.***out***.println(**"-----------------------------------------------------"**);  
    }  
     
    *//（5）从list剔除年龄大于18岁的学生信息* **public static void** fun5(List<Student> list3){  
    System.***out***.println(**"（5）从list剔除年龄大于18岁的学生信息"**);  
    List<Student> removeList = **new** ArrayList<>();  
    **for** (Student student : list3) {  
    **if** (student != **null** && student.getAge() > 18) {  
    removeList.add(student);  
    }  
    }  
    list3.removeAll(removeList);  
    **for** (Student student : list3) {  
    System.***out***.println(student.toString());  
    }  
    }  
     
   }



1. 使用Map 完成练习1的习题。
2. **package** java3;  
     
   **import** java.util.\*;  
     
   */\*\*  
    \* 使用Map集合操作Student对象  
    \*/***public class** MapStudent {  
    **public static void** main(String[] args) {  
    Map<Integer, Student> map1 = **new** HashMap<>();  
    Map<Integer, Student> map2 = **new** HashMap<>();  
    *//添加Student对象* 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"**, 19, 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"**, 19, 55, **"2班"**));  
     
     
    *//（1）整合两个list学生信息成一个新的list* Map<Integer, Student> mapAll = *function1*(map1, map2);  
    *//（2）按照分数给出学生信息排名  
    functhon2*(mapAll);  
    *//（3）输出不及格的学生信息  
    functhon3*(mapAll);  
    *//（4）查找张三的信息  
    functhon4*(mapAll);  
    *//（5）从list剔除年龄大于18岁的学生信息  
    functhon5*(mapAll);  
    }  
     
    *//（1）整合两个list学生信息成一个新的list* **public static** Map<Integer, Student> function1(Map<Integer, Student> map1, Map<Integer, Student> map2) {  
    System.***out***.println(**"（1）整合两个list学生信息成一个新的list"**);  
    Map<Integer, Student> mapAll = **new** HashMap<>();  
    mapAll.putAll(map1);  
    mapAll.putAll(map2);  
    **for** (Map.Entry<Integer, Student> entry : mapAll.entrySet()) {  
    System.***out***.println(entry.getKey() + **":"** + entry.getValue().toString());  
    }  
    System.***out***.println(**"-----------------------------------------------------"**);  
    **return** mapAll;  
    }  
     
    *//（2）按照分数给出学生信息排名* **public static void** functhon2(Map<Integer, Student> mapAll) {  
    System.***out***.println(**"（2）按照分数给出学生信息排名"**);  
    List<Map.Entry<Integer, Student>> entryList = **new** ArrayList<Map.Entry<Integer, Student>>(mapAll.entrySet());  
    Collections.*sort*(entryList, **new** Comparator<Map.Entry<Integer, Student>>() {  
    @Override  
    **public int** compare(Map.Entry<Integer, Student> integerStudentEntry1, Map.Entry<Integer, Student> integerStudentEntry2) {  
    **float** score1 = integerStudentEntry1.getValue().getScore();  
    **float** score2 = integerStudentEntry2.getValue().getScore();  
    **if** (score1 > score2) {  
    **return** 1;  
    } **else if** (score1 == score2) {  
    **return** 0;  
    } **else** {  
    **return** -1;  
    }  
    }  
    });  
     
    **for** (Map.Entry<Integer, Student> entry : entryList) {  
    System.***out***.println(entry.getValue().toString());  
    }  
    System.***out***.println(**"-----------------------------------------------------"**);  
    }  
     
    *//（3）输出不及格的学生信息* **public static void** functhon3(Map<Integer, Student> mapAll) {  
    System.***out***.println(**"（3）输出不及格的学生信息"**);  
    **for** (Map.Entry<Integer, Student> entry : mapAll.entrySet()) {  
    **if** (entry != **null** && entry.getValue().getScore() < 60) {  
    System.***out***.println(entry.getValue().toString());  
    }  
    }  
     
    System.***out***.println(**"-----------------------------------------------------"**);  
    }  
     
    *//（4）查找张三的信息* **public static void** functhon4(Map<Integer, Student> mapAll) {  
    System.***out***.println(**"（4）查找张三的信息"**);  
    **for** (Map.Entry<Integer, Student> entry : mapAll.entrySet()) {  
    **if** (entry != **null** && **"张三"**.equals(entry.getValue().getName())) {  
    System.***out***.println(entry.getValue().toString());  
    }  
    }  
    System.***out***.println(**"-----------------------------------------------------"**);  
    }  
     
    *//（5）从list剔除年龄大于18岁的学生信息* **public static void** functhon5(Map<Integer, Student> mapAll) {  
    System.***out***.println(**"（5）从list剔除年龄大于18岁的学生信息"**);  
    *//把map中的Student对象转换成Iterator迭代器进行迭代* Iterator<Student> iterator = mapAll.values().iterator();  
    *//当迭代器中还有元素时，继续循环遍历* **while** (iterator.hasNext()) {  
    *//迭代器中的元素都是Student对象* Student student = iterator.next();  
    **if** (student != **null** && student.getAge() > 18) {  
    iterator.remove();  
    }  
    }  
    **for** (Map.Entry<Integer, Student> entry : mapAll.entrySet()) {  
    System.***out***.println(entry.getValue().toString());  
    }  
    }  
   }



1. 仿照手机淘宝，设计订单和商品的实体类。

订单类

1. **package** java3;  
     
   **public class** Order {  
    */\*\*  
    \* 商品编号  
    \*/* **private** String **code**;  
     
    */\*\*  
    \* 商品名称  
    \*/* **private** String **name**;  
     
    */\*\*  
    \* 商品价格  
    \*/* **private** String **price**;  
     
    */\*\*  
    \* 商品运费  
    \*/* **private** String **freight**;  
     
    */\*\*  
    \* 运险费  
    \*/* **private** String **RiskPremium**;  
     
    */\*\*  
    \* 收货人  
    \*/* **private** String **Consignee** ;  
     
    */\*\*  
    \* 收货人电话  
    \*/* **private** String **ConsigneeTelephone** ;  
     
    */\*\*  
    \*收货人地址  
    \*/* **private** String **ConsigneeAddress** ;  
     
    */\*\*  
    \* 评价  
    \*/* **private** String **evaluate** ;  
     
    */\*\*  
    \* 交易时间  
    \*/* **private** String **TransactionTime** ;  
     
    */\*\*\*  
    \* 创建时间  
    \*/* **private** String **creatTime**;  
     
    */\*\*\*  
    \* 创建人  
    \*/* **private** String **creatUser**;  
     
    */\*\*  
    \* 修改时间  
    \*/* **private** String **updateTime**;  
     
    */\*\*  
    \* 修改人  
    \*/* **private** String **updateUser**;  
      
   }

商品类

**package** java3;  
  
**import** java.util.Date;  
**import** java.util.List;  
  
*/\*\*  
 \* 淘宝商品实体类  
 \*/***public class** Goods {  
 */\*\*  
 \* 商品编号  
 \*/* **private** String **goodsId**;  
 */\*\*  
 \* 商品名称  
 \*/* **private** String **goodsName**;  
 */\*\*  
 \* 促销价格  
 \*/* **private float salePrice**;  
 */\*\*  
 \* 原价  
 \*/* **private float originalPrice**;  
 */\*\*  
 \* 评价  
 \* 一件商品可以有很多评价  
 \*/* **private** List<String> **goodsComments**;  
 */\*\*  
 \* 图片保存路径  
 \* 一件商品可以有多张图片  
 \*/* **private** List<String> **goodsImgUrl**;  
 */\*\*  
 \* 商品数量  
 \*/* **private int goodsSum**;  
 */\*\*  
 \* 已销售数量  
 \*/* **private int saleSum**;  
 */\*\*  
 \* 上架时间  
 \*/* **private** Date **releaseTime**;  
 */\*\*  
 \* 下架时间  
 \*/* **private** Date **withdrawalTime**;  
  
 */\*\*\*  
 \* 创建时间  
 \*/* **private** String **creatTime**;  
  
 */\*\*\*  
 \* 创建人  
 \*/* **private** String **creatUser**;  
  
 */\*\*  
 \* 修改时间  
 \*/* **private** String **updateTime**;  
  
 */\*\*  
 \* 修改人  
 \*/* **private** String **updateUser**;