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”。

*/\*\*  
 \* 学生信息实体类  
 \*/*public class Student {  
 */\*\*  
 \* 学生姓名  
 \*/* private String name;  
 */\*\*  
 \* 学生年龄  
 \*/* private int age;  
 */\*\*  
 \* 学生成绩  
 \*/* private int grade;  
 */\*\*  
 \* 学生班号  
 \*/* private String studentClass;  
  
 public Student(){};  
  
 */\*\*  
 \* 构造学生名字，年龄，成绩，班号  
 \** ***@param*** *name 名字  
 \** ***@param*** *age 年龄  
 \** ***@param*** *grade 成绩  
 \** ***@param*** *studentClass 班号  
 \*/* public Student(String name, int age, int grade, String studentClass) {  
 this.name = name;  
 this.age = age;  
 this.grade = grade;  
 this.studentClass = studentClass;  
 }  
  
 */\*\*  
 \* 获取学生名字  
 \** ***@return*** *学生名字  
 \*/* public String getName() {  
 return name;  
 }  
  
 */\*\*  
 \* 设置学生名字  
 \** ***@param*** *name 名字  
 \*/* public void setName(String name) {  
 this.name = name;  
 }  
 */\*\*  
 \* 获取学生年龄  
 \** ***@return*** *学生年龄  
 \*/* public int getAge() {  
 return age;  
 }  
  
 */\*\*  
 \* 设置学生年龄  
 \** ***@param*** *age 年龄  
 \*/* public void setAge(int age) {  
 this.age = age;  
 }  
 */\*\*  
 \* 获取学生成绩  
 \** ***@return*** *学生成绩  
 \*/* public int getGrade() {  
 return grade;  
 }  
  
 */\*\*  
 \* 设置学生成绩  
 \** ***@param*** *grade 成绩  
 \*/* public void setGrade(int grade) {  
 this.grade = grade;  
 }  
 */\*\*  
 \* 获取学生班号  
 \** ***@return*** *学生班号  
 \*/* public String getStudentClass() {  
 return studentClass;  
 }  
  
 */\*\*  
 \* 设置学生班号  
 \** ***@param*** *studentClass 班号  
 \*/* public void setStudentClass(String studentClass) {  
 this.studentClass = studentClass;  
 }  
  
 */\*\*  
 \* 设置学生信息  
 \** ***@param*** *name 名字  
 \** ***@param*** *age 年龄  
 \** ***@param*** *grade 成绩  
 \** ***@param*** *studentClass 班号  
 \*/* public void setData(String name, int age, int grade, String studentClass) {  
 this.name = name;  
 this.age = age;  
 this.grade = grade;  
 this.studentClass = studentClass;  
 }  
}

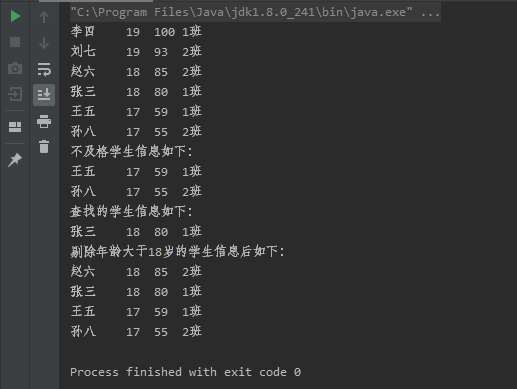
1. 整合两个list学生信息成一个新的list
2. 按照分数给出学生信息排名
3. 输出不及格的学生信息
4. 查找张三的信息
5. 从list剔除年龄大于18岁的学生信息

代码：

public static void main(String[] args) {  
 */\*\*  
 \* 建立两个list保持学生信息  
 \*/* List<Student> list1 = new ArrayList<>();  
 List<Student> list2 = 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"));  
 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"));  
 */\*\*  
 \* 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);  
  
 */\*\*  
 \* 按照分数给出学生排名信息  
 \*/* List<Student> temp = new ArrayList<>();  
 */\*\*  
 \* 冒泡排序法  
 \* if判断出成绩较高的学生  
 \*/* for (int i = 0; i < list3.size(); i++) {  
 for (int j = 0; j < list3.size() - i - 1; j++) {  
 if (list3.get(j).getGrade() < list3.get(j + 1).getGrade()) {  
 */\*\*  
 \* 调换顺序  
 \*/* temp.add(list3.get(j));  
 list3.set(j, list3.get(j + 1));  
 list3.set(j + 1, temp.get(0));  
 temp.remove(0);  
 }  
 }  
 }  
  
 for (Student element : list3) {  
 System.*out*.println(element.getName() + "\t" + element.getAge() + "\t" + element.getGrade() + "\t" + element.getStudentClass());  
 }  
  
 */\*\*  
 \* 输出不及格学生  
 \*/* System.*out*.println("不及格学生信息如下：");  
 for (Student element : list3) {  
 if (element.getGrade() < 60) {  
 System.*out*.println(element.getName() + "\t" + element.getAge() + "\t" + element.getGrade() + "\t" + element.getStudentClass());  
 }  
 }  
  
 */\*\*  
 \* 查找张三信息  
 \*/* System.*out*.println("查找的学生信息如下：");  
 for (Student element : list3) {  
 if (element.getName() == "张三") {  
 System.*out*.println(element.getName() + "\t" + element.getAge() + "\t" + element.getGrade() + "\t" + element.getStudentClass());  
 }  
 }  
  
 */\*\*  
 \* 从list剔除年龄大于18岁的学生信息  
 \*/* System.*out*.println("剔除年龄大于18岁的学生信息后如下：");  
 */\*\*  
 \* 利用迭代器剔除list中的学生信息  
 \*/*

Iterator<Student> it = list3.iterator();  
 while(it.hasNext()) {  
 Student item = it.next();  
 if (item.getAge() > 18) {  
 it.remove();  
 }  
 else {  
 System.*out*.println(item.getName()+"\t"+item.getAge()+"\t"+item.getGrade()+"\t"+item.getStudentClass());  
 }  
 }  
}

运行结果：

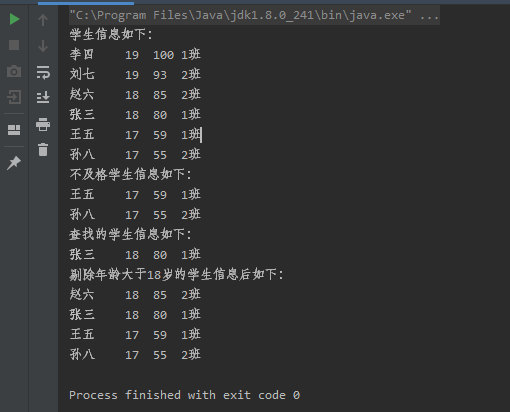


1. 使用Map 完成练习1的习题。

代码：

public static void main(String[] args) {  
 */\*\*  
 \* 建立两个map保存学生信息  
 \*/* Map<Integer,Student> map1 = new HashMap<>();  
 Map<Integer,Student> map2 = new HashMap<>();  
 map1.put(0,new Student("张三", 18, 80, "1班"));  
 map1.put(1,new Student("李四", 19, 100, "1班"));  
 map1.put(2,new Student("王五", 17, 59, "1班"));  
 map2.put(3,new Student("赵六", 18, 85, "2班"));  
 map2.put(4,new Student("刘七", 19, 93, "2班"));  
 map2.put(5,new Student("孙八", 17, 55, "2班"));  
  
 */\*\*  
 \* 整合两个map学生信息成一个新的map  
 \*/* Map<Integer,Student> map3 = new HashMap<>();  
 map3.putAll(map1);  
 map3.putAll(map2);  
  
 */\*\*  
 \* 按照分数给出学生排名信息  
 \* 冒泡排序  
 \* if判断出成绩较高的学生  
 \*/* Map<Integer,Student> temp = new HashMap<>();  
 for (int i = 0; i < map3.size(); i++) {  
 for (int j = 0; j < map3.size() - i - 1; j++) {  
 if (map3.get(j).getGrade() < map3.get(j + 1).getGrade()) {  
 */\*\*  
 \* 调换顺序  
 \*/* temp.put(0,map3.get(j));  
 map3.put(j,map3.get(j+1));  
 map3.put(j+1,temp.get(0));  
 temp.remove(0);  
 }  
 }  
 }  
 System.*out*.println("学生信息如下：");  
 for(Map.Entry<Integer,Student> entry : map3.entrySet()){  
 System.*out*.println(entry.getValue().getName()+"\t"+entry.getValue().getAge()+"\t"+entry.getValue().getGrade()+"\t"+entry.getValue().getStudentClass());  
 }  
  
 */\*\*  
 \* 输出不及格学生信息  
 \*/* System.*out*.println("不及格学生信息如下：");  
 for (Map.Entry<Integer,Student> entry : map3.entrySet()) {  
 if (entry.getValue().getGrade() < 60) {  
 System.*out*.println(entry.getValue().getName()+"\t"+entry.getValue().getAge()+"\t"+entry.getValue().getGrade()+"\t"+entry.getValue().getStudentClass());  
 }  
 }  
  
 */\*\*  
 \* 查找张三信息  
 \*/* System.*out*.println("查找的学生信息如下：");  
 for (Map.Entry<Integer,Student> entry : map3.entrySet()) {  
 if (entry.getValue().getName() == "张三") {  
 System.*out*.println(entry.getValue().getName()+"\t"+entry.getValue().getAge()+"\t"+entry.getValue().getGrade()+"\t"+entry.getValue().getStudentClass());  
 }  
 }  
  
 */\*\*  
 \* 从list剔除年龄大于18岁的学生信息  
 \*/* System.*out*.println("剔除年龄大于18岁的学生信息后如下：");  
 */\*\*  
 \* 利用迭代器剔除以及打印相关学生信息  
 \*/* for(Iterator<Map.Entry<Integer,Student>> it = map3.entrySet().iterator();it.hasNext();) {  
 Map.Entry<Integer,Student> item = it.next();  
 if (item.getValue().getAge() > 18) {  
 it.remove();  
 }  
 else {  
 System.*out*.println(item.getValue().getName()+"\t"+item.getValue().getAge()+"\t"+item.getValue().getGrade()+"\t"+item.getValue().getStudentClass());  
 }  
 }  
}

运行结果：



1. 仿照手机淘宝，设计订单和商品的实体类。

订单实体类代码：

import java.util.ArrayList;  
import java.util.List;  
  
*/\*\*  
 \* 淘宝订单实体类  
 \*/*public class Order {  
 */\*\*  
 \* 订单编号  
 \*/* private int orderId;  
 */\*\*  
 \* 卖家id  
 \*/* private int sellerId;  
 */\*\*  
 \* 买家id  
 \*/* private int buyerId;  
 */\*\*  
 \* 下单时间  
 \*/* private String timePay;  
 */\*\*  
 \* 订单完成时间  
 \*/* private String timeFinish;  
 */\*\*  
 \* 商品名称  
 \*/* private String goodsName;  
 */\*\*  
 \* 商品价格  
 \*/* private short goodsPrice;  
 */\*\*  
 \* 运费  
 \*/* private short freight;  
 */\*\*  
 \* 收货地址  
 \*/* private String orderGoal;  
 */\*\*  
 \* 物流信息  
 \*/* private List<String> logistics = new ArrayList<>();  
 */\*\*  
 \* 订单评价  
 \*/* private String evaluation;  
}

商品实体类代码：

*/\*\*  
 \* 淘宝商品实体类  
 \*/*public class Goods {  
 */\*\*  
 \* 商品名称  
 \*/* private String goodsName;  
 */\*\*  
 \* 商品价格  
 \*/* private short goodsPrice;  
 */\*\*  
 \* 商品库存  
 \*/* private int goodsInventory;  
 */\*\*  
 \* 商品销量  
 \*/* private int goodsSales;  
 */\*\*  
 \* 商品id  
 \*/* private int goodsId;  
 */\*\*  
 \* 商品类别  
 \*/* private String goodsKind;  
 */\*\*  
 \* 父级id  
 \*/* private int parentId;  
 */\*\*  
 \* 子级id  
 \*/* private int childId;  
 */\*\*  
 \* 卖家id  
 \*/* private int sellerId;  
 */\*\*  
 \* 买家id  
 \*/* private int buyerId;  
 */\*\*  
 \* 商品上架时间  
 \*/* private String timePutaway;  
 */\*\*  
 \* 商品被购买时间  
 \*/* private String timeBuy;  
 */\*\*  
 \* 商品评价数量  
 \*/* private int evaluationNum;  
}