### 12.10作业

#### 使用链表和映射存放多个图书信息，遍历并输出。其中商品属性：编号，名称，单价，出版社；使用商品编号作为映射中的key。

**package cn.zh.collection;**

**import java.util.\*;**

**public class Test {**

**public static void main(String[] args) {**

**//new几个Books对象**

**Books books1 = new Books(1001, "Java", 888.0, "Sun出版社");**

**Books books2 = new Books(1002, "Python", 257.0, "Python出版社");**

**Books books3 = new Books(1003, "C++", 155.0, "C++出版社");**

**//然后放进HashMap中,Key是图书的编号,value是图书对象**

**HashMap<Integer, Books> booksHashMap = new HashMap<>();**

**booksHashMap.put(books1.getNum(), books1);**

**booksHashMap.put(books2.getNum(), books2);**

**booksHashMap.put(books3.getNum(), books3);**

**List<HashMap<Integer, Books>> booksList = new ArrayList<>();**

**booksList.add(booksHashMap);**

**System.out.println("编号\t名称\t单价\t出版社");**

**for (HashMap<Integer, Books> h : booksList) {**

**//获取key**

**Set<Integer> key = h.keySet();**

**for (Integer i : key) {**

**//再通过key找value**

**System.out.println(i**

**+ "\t" + h.get(i).getName()**

**+ "\t" + h.get(i).getPrice()**

**+ "\t" + h.get(i).getPress());**

**}**

**}**

**}**

**}**

**class Books {**

**/\*\***

**\* num 编号**

**\* name 名称**

**\* price 价格**

**\* press 出版社**

**\*/**

**private int num;**

**private String name;**

**private double price;**

**private String press;**

**public Books() {**

**}**

**Books(int num, String name, double price, String press) {**

**super();**

**this.num = num;**

**this.name = name;**

**this.price = price;**

**this.press = press;**

**}**

**int getNum() {**

**return num;**

**}**

**String getName() {**

**return name;**

**}**

**double getPrice() {**

**return price;**

**}**

**String getPress() {**

**return press;**

**}**

**}**

* **由控制台按照固定格式输入学生信息，包括学号，姓名，年龄信息，当输入的内容为exit退出；将输入的学生信息分别封装到一个Student对象中，再将每个Student对象加入到一个集合中，要求集合中的元素按照年龄大小正序排序；最后遍历集合，将集合中学生信息写入到记事本，每个学生数据占单独一行。**

**public class Student implements Comparable<Student> {**

**private Integer stuId;**

**private String name;**

**private Integer age;**

**public Student() {**

**}**

**public Student(Integer stuId, String name, Integer age) {**

**this.stuId = stuId;**

**this.name = name;**

**this.age = age;**

**}**

**/\*\***

**\* getter和setter方法**

**\*/**

**public Integer getStuId() {**

**return stuId;**

**}**

**public void setStuId(Integer stuId) {**

**this.stuId = stuId;**

**}**

**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;**

**}**

**@Override**

**public int compareTo(Student stu) {**

**return this.age - stu.age;**

**}**

**public String toString(){**

**return "Student [age = " + age + ",name = " + name +",stuId = " + stuId + "]";**

**}**

**}**

**public class TestStudent {**

**public static void main(String[] args) {**

**Set<Student> stuSet = saveStudentInfo();**

**outputInfo(stuSet);**

**}**

**private static Set<Student> saveStudentInfo() {**

**Scanner input = new Scanner(System.in);**

**/\*\***

**\* 创建TreeSet保存学生信息**

**\*/**

**Set<Student> stuSet = new TreeSet<Student>();**

**while (true){**

**System.out.println("请输入学生信息（学号#姓名#年龄）");**

**String inputData = input.nextLine();**

**/\*\***

**\* 判断是否退出 inputData.equals("exit")**

**\*/**

**if ("exit".equals(inputData)){**

**break;**

**}**

**/\*\***

**\* 把用户输入的学生信息分割为String[]**

**\*/**

**String [] info = inputData.split("#");**

**/\*\***

**\* 将输入信息封装到Student对象中**

**\*/**

**Student stu = new Student(Integer.parseInt(info[0]),info[1],**

**Integer.parseInt(info[2]));**

**/\*\***

**\* 将学生对象加入集合**

**\*/**

**stuSet.add(stu);**

**}**

**return stuSet;**

**}**

**private static void outputInfo(Set<Student> stuSet){**

**File file = new File("e:/student.txt");**

**/\*\***

**\* 创建文件输出流对象**

**\*/**

**FileWriter fw = null;**

**try {**

**fw = new FileWriter(file);**

**Iterator<Student> it = stuSet.iterator();**

**while (it.hasNext()){**

**String info = it.next().toString();**

**fw.write(info);**

**fw.write("\r\n");**

**}**

**} catch (IOException e) {**

**e.printStackTrace();**

**}finally {**

**try {**

**fw.close();**

**} catch (IOException e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**}**