**长沙学院计算机工程与应用数学学院**

**实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | 面向对象程序设计 | 班级 | 20软件 | 实验日期 | 3月5日 |
| 姓名 | 李佳骏 | 学号 | B20190103224 | 指导教师 | 黄志坚 |
| 成绩 |  | | | | |

一. 实验名称

JAVA集合框架

二. 实验目的及要求

1）熟练使用List、Set、Map接口的集体实现类完成对集合的” 增删改查”操作。

2）根据各个不同的场景选着合适的框架来分析问题和解决问题。

三. 实验环境

Eclipse IDE

四. 实验内容

（1）使用List存取用户信息，并做增删改查操作

（2）使用Set存取数据，并做增删改查操作

（2）使用Map存取数据，并做增删改查操作

五. 实验过程及实验结果

**（1）使用List存取用户信息，并做增删改查操作**

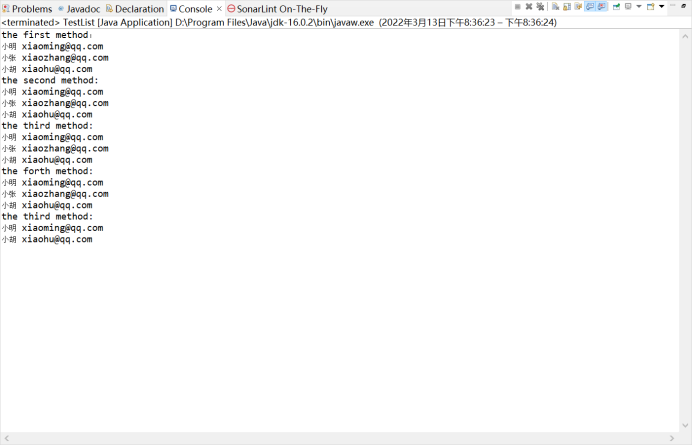
**创建一个User类**

**public** **class** User {  
 String name;  
 String email;  
  
 **public** User(String name, String email) {  
 **this**.name = name;  
 **this**.email = email;  
 }  
  
 **public** String getName() {  
 **return** **this**.name;  
 }  
  
 **public** void setName(String name) {  
 **this**.name = name;  
 }  
  
 **public** String getEmail() {  
 **return** **this**.email;  
 }  
  
 **public** void setEmail(String email) {  
 **this**.email = email;  
 }  
}

**创建一个TestList类**

**public** **class** TestList {  
 **public** static void main(String[] args) {  
 *//* **TODO** *Auto-generated method stub*  
 List<User> list = **new** ArrayList<User>();  
 list.add(**new** User("小明", "xiaoming@qq.com"));  
 User xiaozhang = **new** User("小张", "xiaozhang@qq.com");  
 User xiaohu = **new** User("小胡", "xiaohu@qq.com");  
 list.add(xiaozhang);  
 list.add(xiaohu);  
  
 System.out.println("the first method：");  
 ListIterator<User> it1 = list.listIterator();  
 **while** (it1.hasNext()) {  
 User user = it1.next();  
 System.out.println(user.getName() + " " + user.getEmail());  
 }  
  
 System.out.println("the second method:");  
 **for** (Iterator<User> it2 = list.iterator(); it2.hasNext();) {  
 User user = it2.next();  
 System.out.println(user.getName() + " " + user.getEmail());  
 }  
  
 System.out.println("the third method:");  
 **for** (User user : list) {  
 System.out.println(user.getName() + " " + user.getEmail());  
 }  
  
 System.out.println("the forth method:");  
 **for** (int i = 0; i < list.size(); ++i) {  
 System.out.println(list.get(i).getName() + " " + list.get(i).getEmail());  
 }  
  
 list.remove(1);  
  
 System.out.println("the third method:");  
 **for** (User user : list) {  
 System.out.println(user.getName() + " " + user.getEmail());  
 }  
 }  
  
}

**结果图**



**（2）使用Set存取数据，并做增删改查操作**

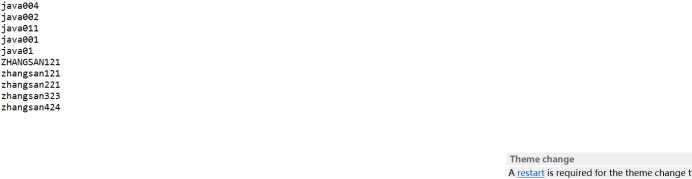
**创建一个Student 类**

**public** **class** Student {  
  
 **private** String name;  
 **private** int age;  
  
 **public** Student(String name, int age) {  
 **this**.name = name;  
 **this**.age = age;  
 }  
  
 *// 复写hashCode方法*  
 **public** int hashCode() {  
 **return** 60;  
 }  
  
 *// 复写equals方法*  
 **public** boolean equals(Object arg0) {  
 **if** (!(arg0 **instanceof** Student)) {  
 **return** **false**;  
 }  
 Student studet = (Student) arg0;  
 **return** **this**.name.equals(studet.name) && **this**.age == studet.age;  
 }  
  
 **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;  
 }  
  
}

**创建一个TestHashSet 类**

**public** **class** TestHashSet {  
  
 **public** static void main(String[] args) {  
 *//* **TODO** *Auto-generated method stub*  
 hashSet1();  
 hashSet2();  
 }  
  
 **public** static void hashSet1() {  
 HashSet<String> hashSet = **new** HashSet<String>();  
 hashSet.add("java001");  
 hashSet.add("java01");  
 hashSet.add("java011");  
 hashSet.add("java002");  
 hashSet.add("java004");  
 *// 使用常用迭代器获取输出内容*  
 Iterator<String> iterator = hashSet.iterator();  
 **while** (iterator.hasNext()) {  
 String next = iterator.next();  
 System.out.println(next);  
 }  
 }  
  
 **public** static void hashSet2() {  
 HashSet<Student> hashSet = **new** HashSet<Student>();  
 hashSet.add(**new** Student("ZHANGSAN1", 21));  
 hashSet.add(**new** Student("zhangsan1", 21));  
 hashSet.add(**new** Student("zhangsan2", 21));  
 hashSet.add(**new** Student("zhangsan3", 23));  
 hashSet.add(**new** Student("zhangsan4", 24));  
 *// 使用常用迭代器获取值*  
 Iterator<Student> iterator = hashSet.iterator();  
 **while** (iterator.hasNext()) {  
 Student next = (Student) iterator.next();  
 System.out.println(next.getName() + "" + next.getAge());  
 }  
 }  
  
}

**结果图**



**（3）使用Map存取数据，并做增删改查操作**

**创建一个Student 类**

**public** **class** Student {  
   
 **public** String id;  
 **public** String name;  
   
 **public** Student(String id,String name) {  
 **this**.id = id;  
 **this**.name = name;  
 }  
   
}

**创建一个TestMap 类**

**public** **class** TestMap {  
  
 **public** Map<String, Student> student;  
  
 **public** TestMap() {  
 **this**.student = **new** HashMap<String, Student>();  
 }  
  
 **public** void testPut() {  
 Scanner sc = **new** Scanner(System.in);  
 int i = 0;  
 **while** (i < 3) {  
 System.out.println("请输入学生ID:");  
 String stuID = sc.next();  
 Student stu = student.get(stuID);  
 **if** (stu == **null**) {  
 System.out.println("请输入学生姓名：");  
 String stuName = sc.next();  
 student.put(stuID, **new** Student(stuID, stuName));  
 } **else** {  
 System.out.println("该学生ID已被占用");  
 **continue**;  
  
 }  
 }  
 sc.close();  
 }  
  
 **public** void testKeySet() {  
 Set<String> keySet = student.keySet();  
 System.out.println("总共有" + student.size() + "学生");  
 **for** (String stuID : keySet) {  
 Student stu = student.get(stuID);  
 **if** (stu != **null**) {  
 System.out.println("学生" + stu.name);  
 }  
 }  
 }  
  
 **public** void testEntrySet() {  
 Set<Entry<String, Student>> entrySet = student.entrySet();  
 **for** (Entry<String, Student> entry : entrySet) {  
 System.out.println("取得键：" + entry.getKey());  
 System.out.println("对应值为：" + entry.getValue().name);  
 }  
 }  
  
 **public** void testRemove() {  
 Scanner sc = **new** Scanner(System.in);  
 **while** (**true**) {  
 System.out.println("请输入要删除的学生的ID:");  
 String stuID = sc.next();  
 Student stu = student.get(stuID);  
 **if** (stu == **null**) {  
 System.out.println("输入学生的ID不存在！");  
 **continue**;  
 }  
 student.remove(stuID);  
 System.out.println("成功删除学生" + stu.name);  
 **break**;  
 }  
 sc.close();  
 testEntrySet();  
 }  
  
 **public** void testModify() {  
 Scanner sc = **new** Scanner(System.in);  
 System.out.println("请输入要修改的学生的ID:");  
 **while** (**true**) {  
 String id = sc.next();  
 Student stu = student.get(id);  
 **if** (stu == **null**) {  
 System.out.println("输入学生的ID不存在！");  
 **continue**;  
 }  
 System.out.println("当前学生是：" + stu.name);  
 System.out.println("请输入新的学生");  
 String name = sc.next();  
 student.put(id, **new** Student(id, name));  
 System.out.println("修改成功!");  
 **break**;  
 }  
 sc.close();  
   
 }  
 **public** void testContainsKey() {  
 Scanner sc = **new** Scanner(System.in);  
 System.out.println("请输入学生的ID:");  
 String stuID = sc.next();  
 System.out.println("输入的ID为" + stuID + ",在学生列表中是否存在“" + student.containsKey(stuID));  
 **if**(student.containsKey(stuID)) {  
 System.out.println("学生的姓名为" + student.get(stuID).name);  
 }  
  
 System.out.println("请输入学生的姓名:");  
 String name = sc.next();  
 **if**(student.containsValue(**new** Student(**null**, name))) {  
 System.out.println("存在学生" + name);  
 } **else** {  
 System.out.println("学生不存在！");  
 }  
 }  
 **public** static void main(String[] args) {  
 *//* **TODO** *Auto-generated method stub*  
 TestMap mt = **new** TestMap();  
 mt.testPut();  
 mt.testKeySet();  
 }  
  
}

六. 实验小结

（1）java集合框架主要包括Collection和Map两种类型。其中Collection又包括List、Set、Queue。Map中存储的是键值对的映射。

（2）List是有序集合，可以用索引取值，且允许有重复元素。

（3）Set是无序集合，不能通过索引取值，允许有重复元素，有且只有一个null元素。

（4）Map是映射集，元素以键值对的形式存储，键不可重复，值可重复，通过键取值。

（5）效率上，规则集set比线性表更高效。

七. 教师评阅意见