# 实验2 集合类

一、实验目的：

1、理解List接口的作用。

2、熟悉ArrayList类的使用。

3、理解Map接口的作用。

4、熟悉HashMap常见方法的使用。

1. 实验报告要求

1、实验报告提交word文档，命名为“学号+姓名+实验2”。

2、本次实验共2题，必须给出每一题的填好**空缺**部分的**完整代码**，以及**运行成功的截图**。

3、可以附上调试程序的过程中，出现的问题、以及解决方法。【可选】

三、实验内容：

1、请在【代码一】~【代码七】填写代码，以实现购物车的常见功能。



（1）新建一个名为”ex2”的“Java”项目。在项目中，新建名为“mypackage”的包，在包中新建名为“CartItem”的Java类，CartItem类为购物项类。

package mypackage;

public class CartItem {

//商品ID

private String itemID;

//商品价格

private Double price;

//购买数量

private Integer num;

//【代码一】带三个参数的构造函数

public CartItem(String itemID, Double price, Integer num) {

this.itemID = itemID;

this.price = price;

this.num = num;

}

//【代码二】重写toString()

@Override

public String toString() {

return "CartItem{" +

"itemID='" + itemID + '\'' +

", price=" + price +

", num=" + num +

'}';

}

//【代码三】为全部的私有成员变量定义getter和setter

public String getItemID() {

return itemID;

}

public void setItemID(String itemID) {

this.itemID = itemID;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public Integer getNum() {

return num;

}

public void setNum(Integer num) {

this.num = num;

}

}

（2）在“mypackage”包中，新建名为“ShoppingCart”的Java类，ShoppingCart类为购物车类。

package mypackage;

import java.util.\*;

public class ShoppingCart {

public List<CartItem> items = new ArrayList();

//计算购物车中商品总价

public Double getTotalPrice() {

Double totalPrice = 0.0;

for (CartItem item : items) {

//【代码四】

totalPrice += item.getPrice() \* item.getNum();

}

return totalPrice;

}

//向购物车中添加商品

public void add(CartItem newItem) {

/\*\*

\* 遍历所有的商品，如果商品id相同，就将数量进行相加。 如果全部不同，则需要把商品单独添加到购物车

\*/

for (CartItem item : items) {

//比较id

if (/\*【代码五】\*/ item.getItemID().equals(newItem.getItemID())

) {

//修改item的数量

item.setNum(item.getNum() + newItem.getNum());

return;

}

}

//【代码六】把商品单独添加到购物车

items.add(newItem);

}

public void delete(String itemID) {

/\*\*

\* 使用迭代器从集合中删除商品。 使用迭代器对象.remove()方法删除

\*/

Iterator<CartItem> it = items.iterator();//拿到迭代器

while (it.hasNext()) { //hasNext():是否存在下一个对象元素

CartItem item = it.next();//拿到购物车中的商品

//判断。如果商品id和传过来的id一样，则删除

if (itemID.equals(item.getItemID())) {

it.remove();

//【代码七】跳出循环

break;

}

}

}

@Override  
public String toString() {  
 return "ShoppingCart{" +  
 "items=" + items +  
 '}';  
}

public static void main(String[] args) {

//测试

ShoppingCart cart = new ShoppingCart();

cart.add(new CartItem("C001",12.0,2));

cart.add(new CartItem("C002",15.0,1));

cart.add(new CartItem("C001",12.0,1));

System.out.println(cart);

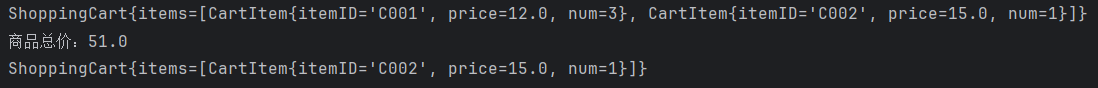
System.out.println("商品总价："+cart.getTotalPrice());

cart.delete("C001");

System.out.println(cart);

}

}



2、请在【代码一】~【代码七】填写代码，实现通过学号，对学生基本信息的管理。

（1）在“mypackage”包中，新建名为“Student”的Java类。Student为学生类。

package mypackage;

public class Student {

private String id;

private String name;

private int grade;

@Override

public String toString() {

return "Student{" + "id=" + id + ", name=" + name + ", grade=" + grade + '}';

}

public Student(String id, String name, int grade) {

this.id = id;

this.name = name;

this.grade = grade;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getGrade() {

return grade;

}

public void setGrade(int grade) {

this.grade = grade;

}

}

（2）在“mypackage”包中，新建名为“StudentManager”的Java类。StudentManager类用于实现对学生信息的增删改查的操作。

package mypackage;  
import java.util.\*;  
public class StudentManager {  
 //【代码一】声明HashMap集合对象  
 HashMap<String, Student> map = new HashMap<>();  
  
 public StudentManager() {  
 //创建集合对象  
 map = new HashMap();  
 //创建学生对象  
 Student s = new Student("181305", "xws", 100);  
 Student s1 = new Student("181306", "xxs", 90);  
 Student s2 = new Student("181307", "wws", 99);  
 /\*【代码二】根据自己的学号、姓名以及100分的成绩，创建Student对象;  
 并将该Student对象添加到map集合中\*/  
 Student s3 = new Student("226709029", "刘俊", 100);  
 map.put("226709029", s3);  
 //学生对象添加到集合  
 map.put("181305", s);  
 map.put("181306", s1);  
 map.put("181307", s2);  
 }  
  
 //显示所有学生信息  
 public void showStu() {  
 if (map.isEmpty()) {  
 System.*out*.println("没有学生");  
 } else {  
 System.*out*.println("一共有" + map.size() + "个学生");  
 for (String key : map.keySet()) {  
 System.*out*.println(key + ":" + map.get(key));  
 }  
 }  
 }  
  
 //添加一个学生  
 public void addStu(Student student) {  
 //【代码三】  
 String key = student.getId();  
 map.put(key, student);  
 System.*out*.println("添加学生成功");  
 showStu();  
 }  
  
 //根据学号，查找一个学生  
 public void selectStu(String id) {  
 if (map.containsKey(id)) {  
 Student stu = /\*【代码四】\*/ map.get(id);  
 System.*out*.println(stu);  
 } else {  
 System.*out*.println("没有这个学号的学生");  
 }  
 }  
  
 //根据学号更新学生基本信息  
 public void updateStu(String id, String name, int grade) {  
 if (/\*【代码五】\*/ map.containsKey(id)) {  
 Student stu = map.get(id);  
 //【代码六】设置学生的姓名及成绩  
 stu.setName(name);  
 stu.setGrade(grade);  
 map.put(id, stu);  
 System.*out*.println("学生信息更新成功");  
 showStu();  
 } else {  
 System.*out*.println("没有这个学号的学生");  
 }  
 }  
  
 //根据学号删除学生  
 public void deleteStu(String id) {  
 if (map.containsKey(id)) {  
 //【代码七】  
 map.remove(id);  
 System.*out*.println("学生删除成功");  
 showStu();  
 } else {  
 System.*out*.println("没有这个学号的学生");  
 }  
 }  
  
 public static void main(String[] args) {  
 StudentManager studentManager = new StudentManager();  
 studentManager.showStu();  
 System.*out*.println("=============================================");  
 studentManager.selectStu("990001");  
 studentManager.selectStu("181305");  
 System.*out*.println("=============================================");  
 studentManager.addStu(new Student("990002", "lucy", 90));  
 System.*out*.println("=============================================");  
 studentManager.deleteStu("990001");  
 studentManager.deleteStu("990002");  
 System.*out*.println("=============================================");  
 studentManager.updateStu("990001", "mary", 90);  
 studentManager.updateStu("181305", "mary", 90);  
  
 }  
}



文本

描述已自动生成