**重庆交通大学信息科学与工程学院**

**综合性设计性实验报告**

**班 级： 计科1703**

**姓名 学号： 黄富灵 631707060308**

**实验项目名称： 实验六、面向对象综合**

**实验项目性质： 综 合 性**

**实验所属课程： 《Java程序设计》**

**实验室(中心)：**

**指 导 教 师 ： 王桂平**

**实验完成时间： 2019年 3月 31日**

教师评阅意见：

签名： 年 月 日

实验成绩：

**一、实验题目**

设计并编写一个Java面向对象综合程序，要求尽可能包含面向对象中的以下思想或方法：

1) 类：构造方法、构造方法的重载、实例变量、类变量、局部变量等；

2) 封装：类的属性尽量隐藏，以公有的方法对外提供存取等；

3) 继承：使用多种访问控制符（private、default、protected、public），覆盖父类方法，调用父类构造方法等；

4) 多态：在类的层次结构中，父类引用变量指向不同的子类，从而可以调用不同的方法、表现出不同的行为；

5) 抽象：抽象方法和抽象类；

6) 接口：接口声明和实现；

等等。

**二、实验要求**

1) 分组要求：分组完成，自由组队，2人一组。

2) 程序要求：不少于150行代码，且具有较好的可读性（注释、正确缩进等）。

3) 文档要求：详细阐述该程序的设计思想、采用的面向对象思想或方法。

**三、设计思想、采用的面向对象思想或方法**

基于面对对象的思想，创建学生类和学生管理类，使用永真循环打印主菜单进行操作。

**四、主要程序代码**

*/\**

*\* @Coding: utf-8 \n*

*\* @Despcription: \n*

*\* @Author: Suwen SP \n*

*\* @LastEditors: Suwen SP\n*

*\* @Date: 2019-03-31 13:11:47*

*\* @LastEditTime: 2019-03-31 13:12:38*

*\*/*

*// 实现Comparable接口，重写CompareTo方法，用于功能菜单中的“1、查询所有学生信息”，根据学号进行排序*

public class StudentH implements Comparable<StudentH> {

*// 属性和构造方法可以根据题目需求进行更改*

private int id; *// 学号*

private String name; *// 姓名*

private String classNum; *// 班级*

public StudentH() {

}

public StudentH(int *id*, String *name*, String *classNum*) {

this.id = id;

this.name = name;

this.classNum = classNum;

}

public int getId() {

return id;

}

public void setId(int *id*) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String *name*) {

this.name = name;

}

public String getClassNum() {

return classNum;

}

public void setClassNum(String *classNum*) {

this.classNum = classNum;

}

@Override

public String toString() { *// 重写一下toString方法进行打印输出方法*

return "学生" + id + " [学号=" + id + ", 姓名=" + name + ", 班级=" + classNum + "]";

}

@Override

public int compareTo(StudentH *o*) { *// 重写一下compareTo方法根据学号进行排序*

return this.id - o.getId();

}

}

*/\**

*\* @Coding: utf-8 \n*

*\* @Despcription: \n*

*\* @Author: Suwen SP \n*

*\* @LastEditors: Suwen SP\n*

*\* @Date: 2019-03-31 13:13:08*

*\* @LastEditTime: 2019-03-31 13:13:39*

*\*/*

import java.util.ArrayList;

import java.util.Collections;

import java.util.InputMismatchException;

import java.util.LinkedList;

import java.util.List;

import java.util.Scanner;

public class StudentManageSystem {

private static Scanner sc;

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

*// 简单的欢迎界面*

System.out.println("---------------------------------");

System.out.println("\t" + "欢迎进入学生管理系统 ");

System.out.println("---------------------------------");

List<StudentH> list = new LinkedList<StudentH>();

sc = new Scanner(System.in);

int id = 0;

try {

while (true) {

*// 功能菜单界面，每次使用一个功能时，都会把功能菜单打印一次，便于使用者查询功能对应的数字*

System.out.println("功能菜单：");

System.out.println("----------------------------------");

System.out.println("1、查询所有学生信息");

System.out.println("2、根据学号查询信息");

System.out.println("3、增加学生信息");

System.out.println("4、删除学生信息");

System.out.println("5、修改学生信息");

System.out.println("6、退出系统");

System.out.println("(提示：为了您的使用方便，请在使用系统前，最大化控制台。)");

System.out.println("----------------------------------");

System.out.print("请输入对应的数字进行操作:");

int input = sc.nextInt(); *// 输入数字选择功能*

if (input > 0 && input < 7) { *// 输入的数字在1-6时，能执行功能，否则系统退出*

System.out.println("---------------------------------");

*// 3、增加学生信息*

if (input == 3) {

System.out.print("请输入你要添加几个学生信息：");

int num = sc.nextInt();

System.out.println("----------------------------------");

for (int w = 0; w < num; w++) {

System.out.println("请添加学生信息");

System.out.println("----------------------------------");

id++;

System.out.print("请输入学生姓名：");

String name = sc.next();

System.out.print("请输入学生班级：");

String classNum = sc.next();

list.add(new StudentH(id, name, classNum));

System.out.println("添加成功。");

System.out.println("---------------------------------");

}

}

*// 1、查询所有学生信息*

if (input == 1) {

if (list.isEmpty() == true) { *// 判断集合内是否存在学生对象的信息*

System.out.println("您还没有添加学生信息。");

} else {

System.out.println("学生信息表");

System.out.println("---------------------------------");

for (int j = 0; j < list.size(); j++) {

*// 自然排序，实现了Comparable接口并重写了CompareTo方法才能调用此方法*

Collections.sort(list);

*// 比较排序，必须定义另外一个自定义的比较类，这里比较类的类名为MyComparator*

*// Collections.sort(list, new MyComparator());*

System.out.println(list.get(j).toString());

}

}

System.out.println("---------------------------------");

}

*// 2、根据学号查询信息*

if (input == 2) {

if (list.isEmpty() == true) {

System.out.println("您还没有添加学生信息。");

} else {

System.out.print("请输入你要查找的学生学号：");

int index = sc.nextInt();

List<Integer> listID = new ArrayList<Integer>();

for (int c = 0; c < list.size(); c++) {

listID.add(list.get(c).getId());

}

for (int q = 0; q < list.size(); q++) {

if (listID.contains(index)) {

if (index == list.get(q).getId()) {

System.out.println(list.get(q).toString());

break;

} else {

continue;

}

} else {

System.out.println("没有该学生信息。");

break;

}

}

}

System.out.println("---------------------------------");

}

*// 4、删除学生信息*

if (input == 4) {

if (list.isEmpty() == true) {

System.out.println("您还没有添加学生信息。");

} else {

System.out.print("请输入你要删除的学生学号：");

int index = sc.nextInt();

List<Integer> listID = new ArrayList<Integer>();

for (int c = 0; c < list.size(); c++) {

listID.add(list.get(c).getId());

}

for (int q = 0; q < list.size(); q++) {

if (listID.contains(index)) {

if (index == list.get(q).getId()) {

list.remove(q);

System.out.println("删除成功。");

break;

} else {

continue;

}

} else {

System.out.println("没有该学生信息。");

break;

}

}

}

System.out.println("---------------------------------");

}

*// 5、修改学生信息*

if (input == 5) {

if (list.isEmpty() == true) {

System.out.println("您还没有添加学生信息。");

} else {

System.out.print("请输入你要修改的学生学号：");

int index = sc.nextInt();

List<Integer> listID = new ArrayList<Integer>();

for (int c = 0; c < list.size(); c++) {

listID.add(list.get(c).getId());

}

for (int q = 0; q < list.size(); q++) {

if (listID.contains(index)) {

if (index == list.get(q).getId()) {

System.out.print("此学生原信息:");

*// 打印要修改的学生信息的原信息*

System.out.println(list.get(q).toString());

System.out.println("请修改此学生信息：");

System.out.println("----------------------------------");

System.out.print("请重新输入学生姓名：");

String name = sc.next();

System.out.print("请重新输入学生班级：");

String classNum = sc.next();

list.remove(q); *// 先删除该学生所有的信息*

id = index;

*// 重新添加该学生信息*

list.add(new StudentH(id, name, classNum));

System.out.println("修改成功。");

break;

} else {

continue;

}

} else {

System.out.println("没有该学生信息。");

break;

}

}

}

System.out.println("----------------------------------");

}

*// 6、退出系统*

if (input == 6) {

System.out.println("系统已退出，欢迎下次使用。。。");

System.exit(0);

}

} else {

System.out.println("----------------------------------");

System.out.println("输入有误，系统退出。");

System.exit(0);

}

}

*// 输入时，如果输入的不是数字，会捕捉InputMismatchException异常并提示“输入有误，系统退出”*

} catch (InputMismatchException *e*) {

System.out.println("----------------------------------");

System.out.println("输入有误，系统退出。");

} catch (Exception *e*) {

System.out.println("----------------------------------");

System.out.println("其他错误。");

e.printStackTrace();

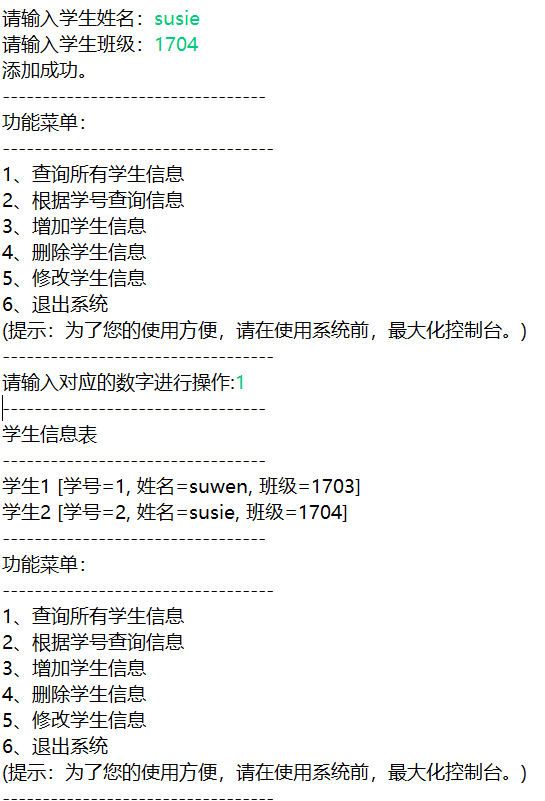
} finally {

System.exit(0); *// 退出系统*

}

}

}

**五、测试结果及说明**

**六、实验体会**

使用面向对象的思维方法，其实是一个把业务逻辑从具体的编程技术当中抽象出来的过程，而这个抽象的过程是自上而下的，非常符合人类的思维习惯，也就是先不考虑问题解决的细节，把问题的最主要的方面抽象成为一个简单的框架，集中精力思考如何解决主要矛盾，然后在解决问题的过程中，再把问题的细节分割成一个一个小问题，再专门去解决细节问题。