

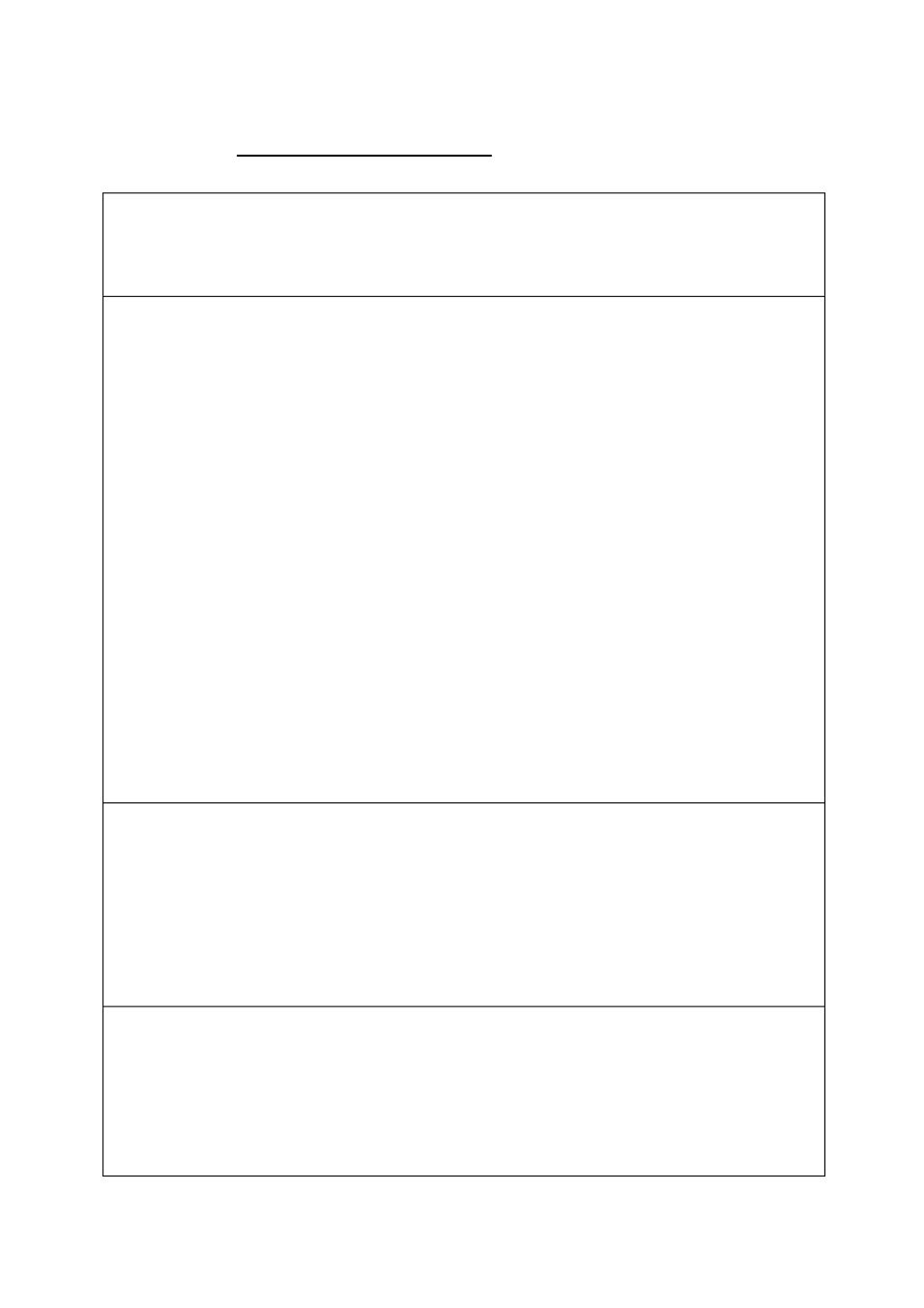
《 面向对象程序设计》

课程设计报告

|  |  |  |  |
| --- | --- | --- | --- |
| 题 | 目：学生学籍管理系统软件设计与实现 | | |
| 院 （系）： | | 信息科学与工程学院 |  |
| 专业班级： | | 计算机科学与技术 | 1201 班 |
| 学生姓名： | | 程伟 |  |
| 学 | 号： | 20121183011 |  |
| 指导教师： | | 吴奕 |  |

20 14 年 12 月 29 日至 20 15 年 1 月 9 日

华中科技大学武昌分校制



面向对象程序设计 课程设计任务书

一、设计（调查报告 / 论文）题目

学生学籍管理系统软件设计与实现

二、设计（调查报告 / 论文）主要内容

内容：

创建一个学生信息的界面，能够实现学生的学号，姓名，性别，出生日期，专业和照片的录入，修改，查询，删除和打印。这样能够详细准确的查询到学生的信息，而且能够更加直观方便的了解学生的信息。

基本功能与要求：

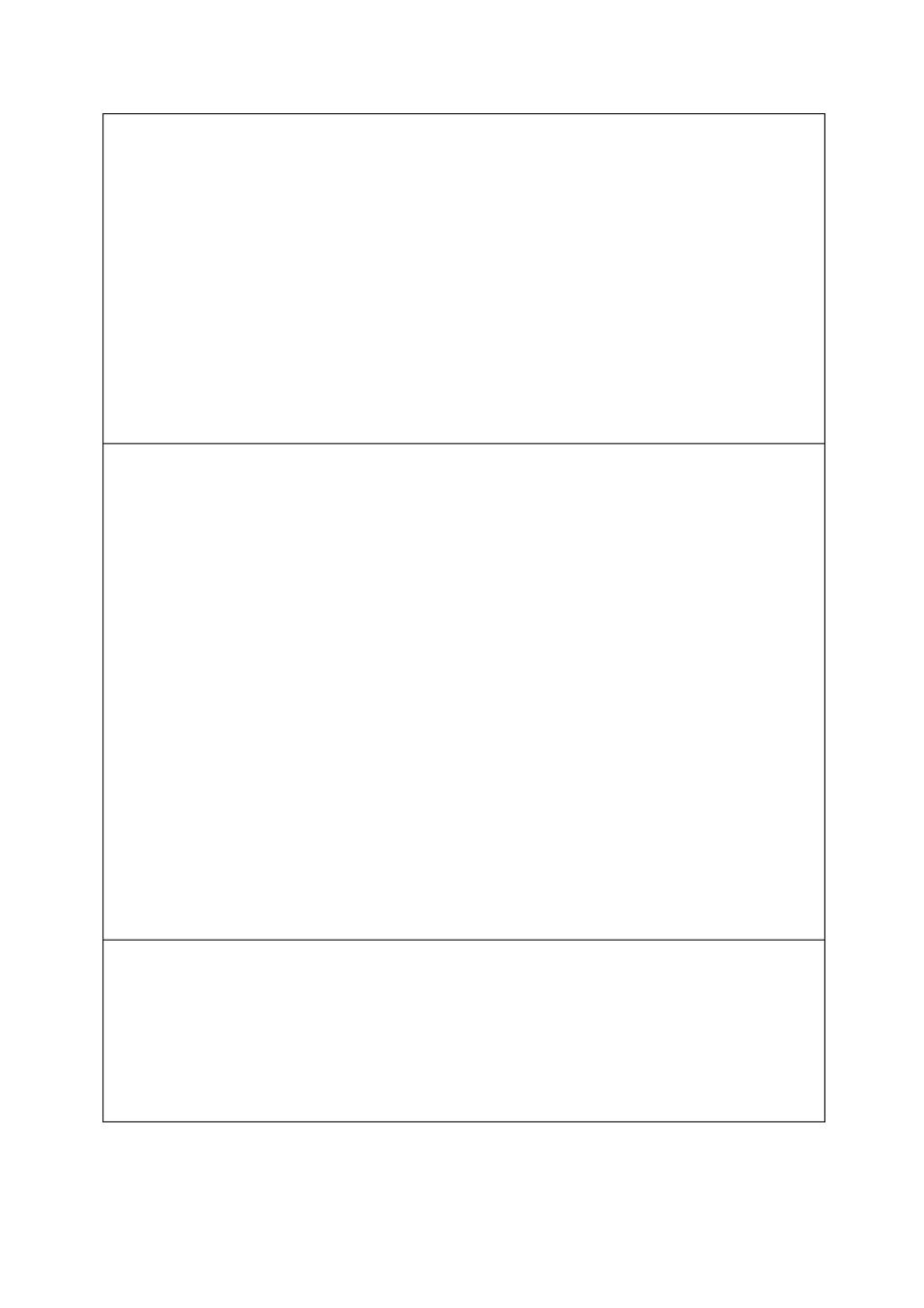
1. 录入学生信息：系统操作人员打开学生信息添加界面，输入相关的信息（姓名、性别、出生日期、入学年份、专业和学院等） 。
2. 修改学生信息：根据学生的学号查询出该学生的相关信息，修改相关条目后保存在数据库中。
3. 删除学生信息：根据学生的学号查询出该学生的相关信息，确定删除后，在数据库中删除该信息。
4. 查询、打印学生信息： 可以根据学生学号、 学生姓名、 学生性别、学生所学专业、学生所属学院来对学生信息进行查询， 所有符合查询条件的学生信息将都被显示出来。

三、原始资料

1. 《面向对象程序设计课程设计》指导书
2. 耿祥义． JAVA大学实用教程．北京：清华大学出版社， 2009．

四、要求的设计（调查 / 论文）成果

1. 课程设计报告
2. 课程设计软件



五、进程安排

布置任务，查找资料、需求分析 1 天

总体设计 1 天

详细设计 1.5 天

编制源程序实现 3.5 天

测试与修改 1 天

撰写课设报告 2 天

六、主要参考资料

1. 张广彬 . Java 课程设计案例精编（第二版） . 北京：清华大学出版社 , 2011.
2. 耿祥义 . Java 课程设计（第二版） . 北京：清华大学出版社 , 2008.
3. 耿祥义． JAVA大学实用教程．北京：清华大学出版社， 2009．
4. 邹林达．陈国君 Java2 程序设计基础．北京：清华大学出版社， 2010．

[5] 邹林达． Java2 程序设计基础实验指导．北京：清华大学， 2010．

指导教师（签名）：

20 年 月 日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | 目 | 录 |  |
| 1 需求与总体设计 . .......................................................... | | | | | 1 |
| 1.1 | 需求分析 . .............................................................. | | |  | 1 |
| 1.2 | 总体设计 . .............................................................. | | |  | 1 |
| 1.3 | 设计要求 . .............................................................. | | |  | 2 |
| 1.4 | 系统主要功能图 ......................................................... | | | | 2 |
| 1.5 | 系统主要类图 | | .......................................................... | | 3 |
| 2 详细设计 ................................................................ | | | |  | 4 |
| 2.1 ManagerWindow类 ....................................................... | | | | | 4 |
| 2.1.1 | | 成员变量和方法 ...................................................... | | | 4 |
| 2.1.2 | | 详细类图 ............................................................ | |  | 4 |
| 2.2 | Student 类 . ............................................................ | | | | 5 |
| 2.2.1 | | 成员变量和方法 ...................................................... | | | 5 |
| 2.2.2 | | 详细类图 ............................................................ | |  | 5 |
| 2.3 | StudentPicture | | 类 ...................................................... | | 5 |
| 2.3.1 | | 成员变量和方法 ...................................................... | | | 5 |
| 2.3.2 | | 详细类图 ............................................................ | |  | 5 |
| 2.4 | InputStudent | | 类 ........................................................ | | 6 |
| 2.4.1 | | 成员变量和方法 ...................................................... | | | 6 |
| 2.4.2 | | 详细类图 ............................................................ | |  | 6 |
| 2.5 | Inquest 类 . ............................................................ | | | | 7 |
| 2.5.1 | | 成员变量和方法 ...................................................... | | | 7 |
| 2.5.2 | | 详细类图 ............................................................ | |  | 7 |
| 2.6 | ModifySituation | | 类 ..................................................... | | 8 |
| 2.6.1 | | 成员变量和方法 ...................................................... | | | 8 |
| 2.6.2 | | 详细类图 ............................................................ | |  | 8 |
| 2.7 | Delete 类 . ............................................................. | | | | 9 |
| 2.7.1 | | 成员变量和方法 ....................................................... | | | 9 |
| 2.7.2 | | 详细类图 . ............................................................ | | | 9 |
| 3 编码实现 ............................................................... | | | |  | 10 |
| 4 系统测试 ............................................................... | | | |  | 37 |
| 4.1ManagerWindow创建窗口 ................................................ | | | | | 37 |
| 4.2InputStudent 学生信息录入窗口 ......................................... | | | | | 37 |
| 4.3Inquest 查询，打印窗口 ................................................ | | | | | 38 |
| 4.4 | ModifySituation | | 信息修改窗口 ......................................... | | 38 |
| 4.5Delete 删除信息窗口 ................................................... | | | | | 38 |
| 总 | 结 ................................................................... | | |  | 39 |

1. 需求与总体设计

1.1 需求分析

学生信息管理系统是一个教育单位不可缺少的部分。一个功能齐全、简单易用的信

息管理系统不但能有效地减轻学校相关工作人员的工作负担， 它的内容对于学校的决策

者和管理者来说都至关重要。 所以学生信息管理系统应该能够为用户提供充足的信息和

快捷的查询手段。但一直以来人们使用传统人工的方式管理文件档案、 统计和查询数据，

随着科学技术的不断提高 , 计算机科学日渐成熟 , 其强大的功能已为人们深刻认识 , 它已

进入人类社会的各个领域并发挥着来越重要的作用。作为计算机应用的一部分 , 使用计算机对学校的各类信息进行管理 , 具有手工管理所无法比拟的优点 . 本系统完成了学生信息的增加、修改、删除、查询。

1.2 总体设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 在设计学籍管理系统时，需要编写 | | 7 个 Java 源文件： ManagerWindow.java ， | |  |
| ModifySituation.java | ，InputStudent.java | ，Inquest.java ，Delete.java | ， |  |
| Student.java ， |  |
| StudentPicture.java | 。学籍管理系统除了需要编写的 7 个 Java 源文件所给出的类外，还 | | |  |

需要 Java 系统提供的一些重要的类，如 JMenuItem，HashMap 和 File 等类。

专业.txt: 学籍管理系统通过使用名字为“专业 .txt ”的文本文件来生成专业名称列表，

以便灵活、有效地录入专业信息。允许用户自己编写“专业 .txt ”文件，只要将该文件和

应用程序存放到同一目录中即可。

ManagerWindow.java（主类）：ManagerWindow类负责创建学籍管理系统的主窗口，该类含有 main 方法，程序从该类开始执行。 ManagerWindow类的成员变量中有 4 种重要类型的对象：InputStudent 、Inquest 、ModifySituation 和 Delete 对象。ManagerWindow类的主要成员的作用将在后面详细阐述。

Student.java ：Student 类所创建的对象是 InputStudent 类的成员之一，负责封装学生的基本信息。

StudentPicture.java ：StudentPicture 类所创建的对象是 InputStudent 、Inpuest和 ModifySituation 类的成员之一，负责显示学生的照片。

InputStudent.java: InputStudent 类创建的对象是 MangerWindow类的重要成员之

一，负责提供录入学生基本信息的界面。

Inquest.java ：Inquest 类创建的对象是 MangerWindow类的重要成员之一，负责提供

查询学生基本信息及打印基本信息的界面。

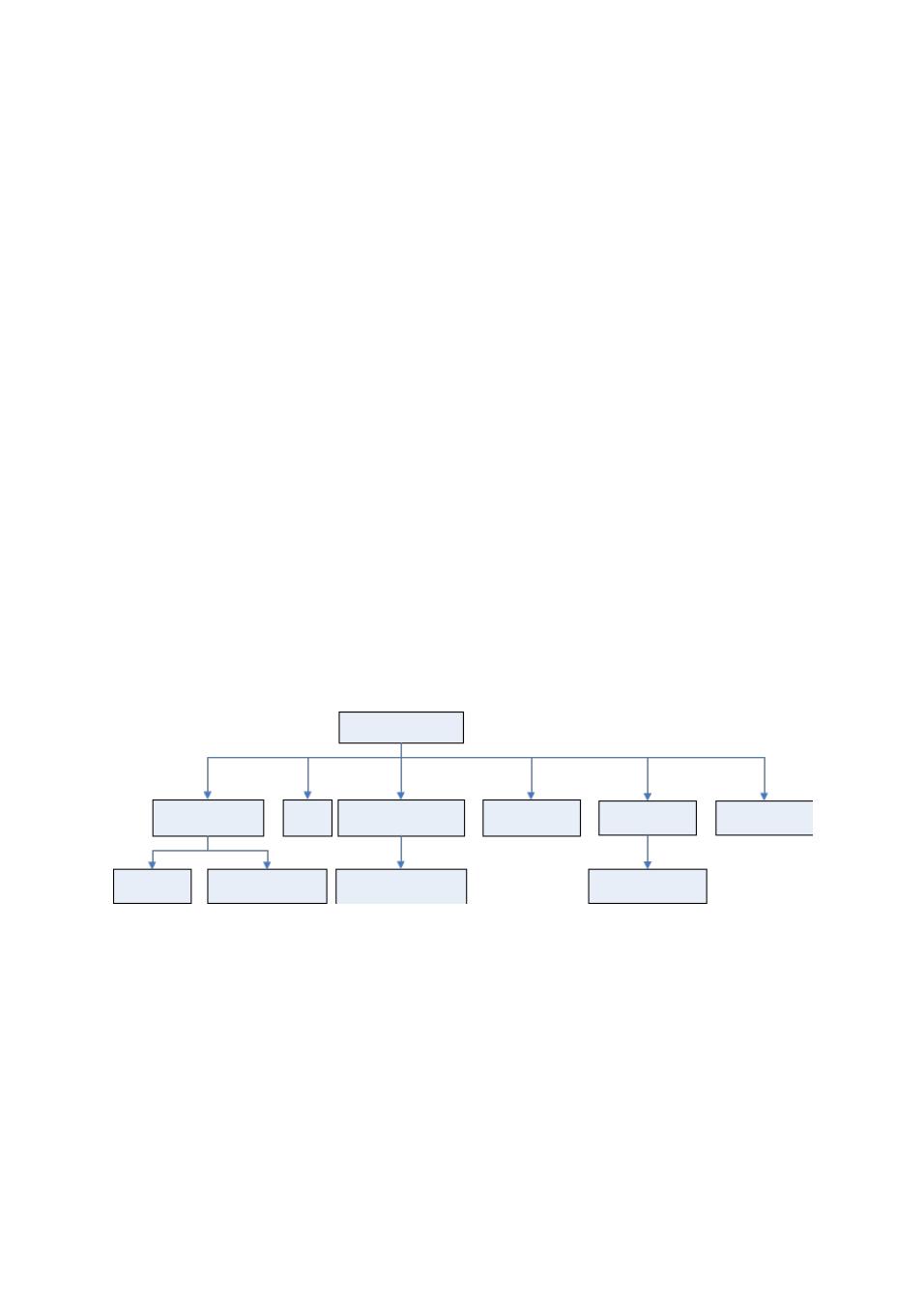
ModifySituation.java ：ModifySituation 类创建的对象是 MangerWindow类的重要成

员之一，负责提供修改学生基本信息的界面。

Delete.java ：Delete 类创建的对象是 MangerWindow类的重要成员之一，负责提供删

除学生基本信息的界面。

1



1.3 设计要求

1、录入学生信息：用户可以通过菜单选项让程序呈现“录入学生基本信息”功能的界

面，系统操作人员打开学生信息添加界面，输入相关的信息（姓名、性别、出生日期、照片、专业和学院等）。

2、修改学生信息：根据学生的学号查询出该学生的相关信息，修改相关条目后保存在

数据库中。

3、删除学生信息：根据学生的学号查询出该学生的相关信息，确定删除后，在数据库

中删除该信息。

4、查询、打印学生信息：可以根据学生学号、学生姓名、学生性别、学生所学专业、学生所属学院来对学生信息进行查询，所有符合查询条件的学生信息将都被显示出来。

1.4 系统主要功能图

系统的主要功能 :

学籍管理系统所用到的一些重要的类以及之间的组合关系如下图

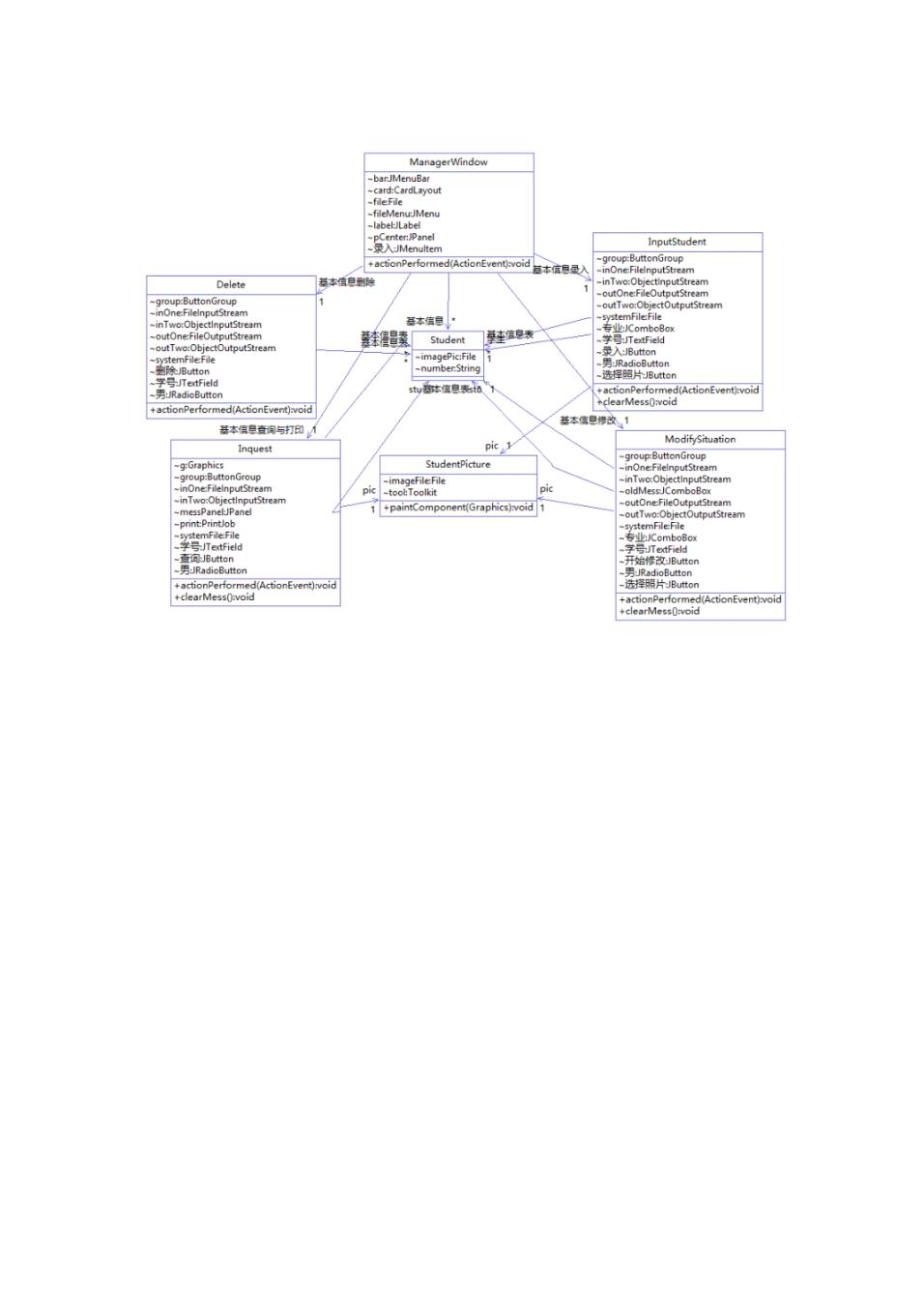
学籍管理系统除了需要编写的 7 个 Java 源文件所给出的类外，还需要 Java 系统提供的一些重要的类，如 JMenuItem，HashMap和 File 等类

**ManagerWindow**

**InputStudent** **File** **ModifySituation** **Delete** **Inquest** **HashMap**

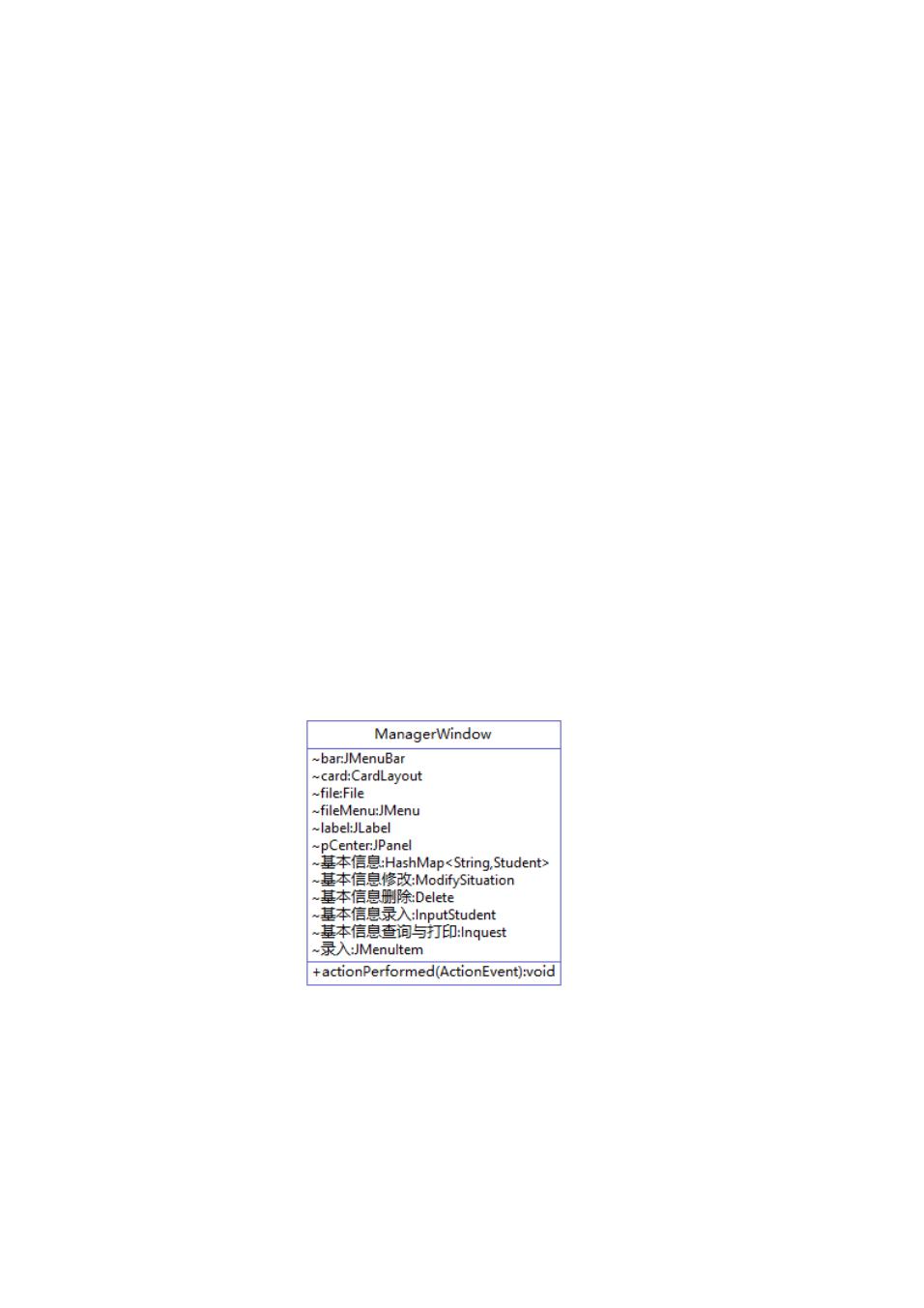
**Student** **StudentPicture** **StudentPicture** **StudentPicture**

2



1.5 系统主要类图

3



1. 详细设计

2.1 ManagerWindow 类

2.1.1 成员变量和方法

“基本信息录入”是 InputStudent 对象，负责提供录入信息的界面。

“基本信息修改”是 ModifySituation 对象，负责提供修改信息的界面。

“基本信息查询”是 Inquest 对象，负责提供查询与打印信息的界面。

“基本信息删除”是 Delete 对象，负责提供删除信息的界面。

“基本信息”是 HashMap<String,Student> 创建的散列映射，用于存储键 / 值数据对，其中的键必须是一个 String 对象，键对应的值必须是 Student 对象。

file 是 Flie 文件对象，程序将“基本信息”散列映射写入该文件中。“录入”、“修改”、“查询”、“删除”和“欢迎界面”是 JMenuItem创建的菜单项，这些菜单项都将当前窗口注册为自己的 ActionEvent 事件监视器。

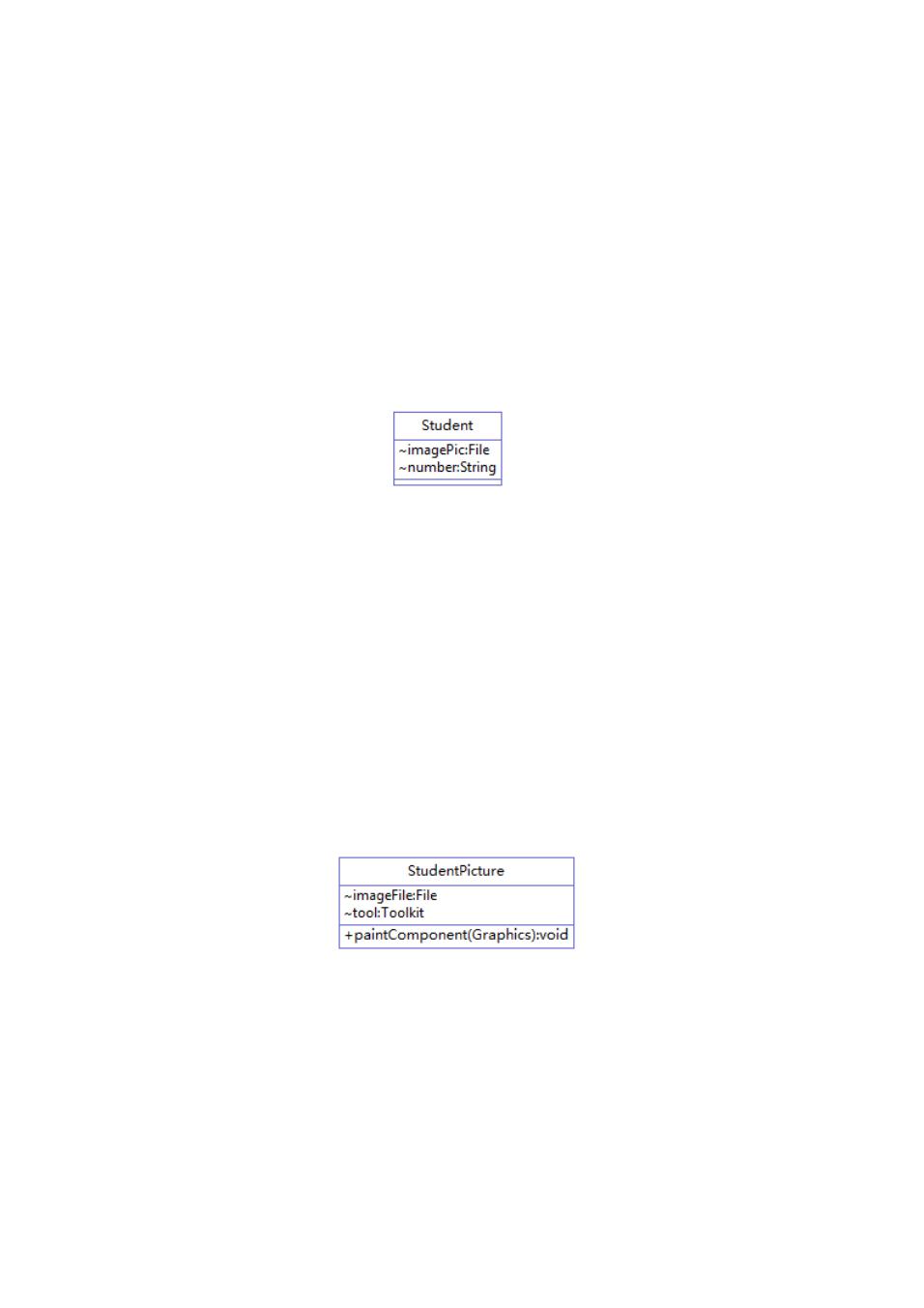
ManagerWindow（）是构造方法，负责完成窗口的初始化。

actionPerformed （ActionEvent ）方法是 ManagerWindow类实现的 ActionListener 接口中的方法。 ManagerWindow创建的窗口是“录入”、“删除”、“查询与打印”、“删除”和“欢迎界面”菜单项的 ActionEvent 事件监视器，当用户选择其中某个菜单项时，窗口将执行 actionPerformed(ActionEvent) 方法进行相应的操作。

main 方法是程序运行的入口方法。

2.1.2 详细类图

4



2.2 Student 类

2.2.1 成员变量和方法

number、name、disciping 、grade、 borth 、和 sex 是 String 对象，它们的值分别表

示学生的“学号”、“姓名”、“专业”、“年级”、“出生日期”和“性别”。

imagePic 是 File 对象，用来存放学生的照片图像和引用。

setNumber（String ）和 getNumber（）方法分别用来设置 number和获取 number的值。

setImagePic （File ）和 getImagePic （）方法分别用来设置 imagePic 的引用。

2.2.2 详细类图

2.3 StudentPicture 类

2.3.1 成员变量和方法

imageFile 是 File 对象，用来存放图像文件的引用。

tool 是 Toolkit 对象，负责创建 Image对象。

StudentPicture （）是构造方法，负责创建 pic 对象。

setImageFile （File ）方法负责设置 imageFile 对象。

paintComponent（ Graphics ）方法负责绘制根据 imageFile 文件创建的 Image 对象。

2.3.2 详细类图

5



2.4 InputStudent 类

2.4.1 成员变量和方法

“学生”是 Student 创建的对象，该对象含有和学生有关的数据。

pic 是 StudentPicture 创建的对象，负责绘制学生的照片。

“基本信息表”是 HashMap<String,Student> 创建的散列映射，用于存储键 / 值数据对，其中的键必须是一个 String 对象，键对应的值必须是 Student 对象。

“录入”、“重置”和“选择照片”是 JButton 创建的对象，这些按钮都将当前容器注册为自己的 ActionEvent 事件监视器。

systemFile 和 imagePic 是 File 声明的对象，其中 systemFile 存放的文件引用和

ManagerWindow类中的 file 的相同，程序将“基本信息表”散列映射写入 systemFile 文件中；而 imagePic 存放的文件引用有用户选择指定。

InputStudent （File ）是构造方法，用来创建“基本信息录入”对象，当创建该对象

时，通过该构造方法的参数将 ManagerWindow类中的 file 的引用传递给 systemFile 。 actionPerformed （ActionEvent ）方法是 InputStudent 类实现的 ActionListener 接

口中的方法。 InputStudent 创建的容器是“录入”、“重置”和“选择照片”按钮的 ActionEvent 事件监视器，当用户单击其中某个按钮时， 容器将执行 actionPerformed

* ActionEvent ）方法进行相应的操作。

2.4.2 详细类图

6



2.5 Inquest 类

2.5.1 成员变量和方法

pic 是 StudentPicture 创建的对象，负责绘制学生的照片。

print 是 PrintJob 对象，负责打印。

“基本信息表”是 HashMap<String,Student> 创建的散列映射，用于存储键 / 值数据对，其中的键必须是一个 String 对象，键对应的值必须是 Student 对象。程序将从 systemFile 文件中读出曾写入到该文件中的 HashMap<String，Student> 对象，并将这

个对象的引用赋给“基本信息表” ，然后查询“基本信息表”中的数据。“查询”、“打印文本照片”、“打印文本”和“打印照片”是 JButton 创建的对象，这

些按钮都将当前容器注册为自己的 ActionEvent 事件监视器。

systemFil 是 File 声明的对象，其中 systemFile 存放的文件引用和 ManagerWindow 类中的 file 的相同。

Inquest （File ）是构造方法，用来创建“基本信息查询”对象，当创建该对象时，通过该构造方法的参数将 ManagerWindow类中的 file 的引用传递给 systemFile 。 actionPerformed （ActionEvent ）方法是 InputStudent 类实现的 ActionListener 接

口中的方法。 Inquest 创建的对话框是“查询”、“打印文本照片”、“打印文本”和“打印照片”按钮的 ActionEvent 事件监视器，当用户单击其中某个按钮时，对话框将执

* actionPerformed （ActionEvent ）方法进行相应的操作。

2.5.2 详细类图

7



2.6 ModifySituation 类

2.6.1 成员变量和方法

pic 是 StudentPicture 创建的对象，负责绘制学生的照片。

“基本信息表”是 HashMap<String,Student> 创建的散列映射，用于存储键 / 值数据对，其中的键必须是一个 String 对象，键对应的值必须是 Student 对象。程序将从 systemFile 文件中读出曾写入到该文件中的 HashMap<String，Student> 对象，并将这个对象的引用赋给“基本信息表”。程序可以修改“基本信息表”中的数据，然后再将

“基本信息表”写入到 systemFile 文件中。“开始修改”、“录入修改”、“重置”和“选择照片”是 JButton 创建的对象，这些按

钮都将当前容器注册为自己的 ActionEvent 事件监视器。

systemFile 和 imagePic 是 File 声明的对象，其中 systemFile 存放的文件引用和

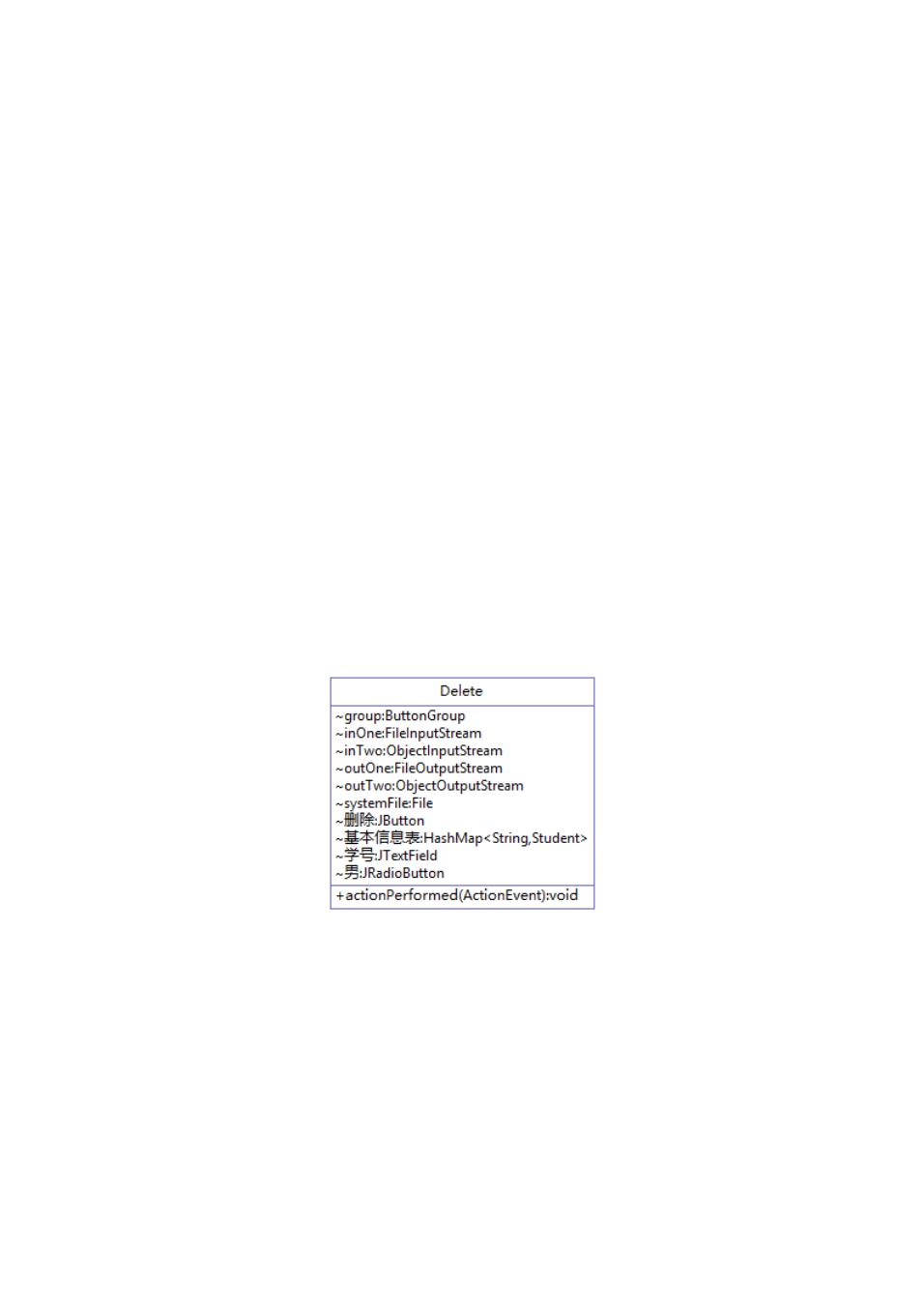
ManagerWindow类中的 file 的相同， imagePic 存放的文件引用有用户选择指定。 ModifySituation （File ）是构造方法，用来创建“基本信息修改”对象，当创建该对象时，通过该构造方法的参数将 ManagerWindow类中的 file 的引用传递给 systemFile 。 actionPerformed （ActionEvent ）方法是 InputStudent 类实现的 ActionListener 接

口中的方法。 InputStudent 创建的容器是“开始修改”、“录入修改”、“重置”和“选择照片”按钮的 ActionEvent 事件监视器，当用户单击其中某个按钮时，容器将执行

actionPerformed （ActionEvent ）方法进行相应的操作。

2.6.2 详细类图

8



2.7 Delete 类

2.7.1 成员变量和方法

“基本信息表”是 HashMap<String,Student> 创建的散列映射，用于存储键 / 值数据对，其中的键必须是一个 String 对象，键对应的值必须是 Student 对象。程序将从 systemFile 文件中读出曾写入到该文件中的 HashMap<String，Student> 对象，并将这

个对象的引用赋给“基本信息表”。程序可以删除“基本信息表”中的特定数据，然后再将“基本信息表”写入到 systemFile 文件中。

“删除”是 JButton 创建的对象，这些按钮都将当前容器注册为自己的 ActionEvent 事件监视器。

systemFile 和 imagePic 是 File 声明的对象，其中 systemFile 存放的文件引用和ManagerWindow类中的 file 的相同。

Delete （File ）是构造方法，用来创建“基本信息删除”对象，当创建该对象时，通过该构造方法的参数将 ManagerWindow类中的 file 的引用传递给 systemFile 。

actionPerformed （ActionEvent ）方法是 InputStudent 类实现的 ActionListener 接口中的方法。 InputStudent 创建的容器是“删除”按钮的 AcionEvent 事件监视器，当用户单击该按钮时，容器将执行 actionPerformed （ActionEvent ）方法进行相应的操作。

2.7.2 详细类图

9

3 编码实现

源代码

package 学生学籍管理系统 ;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.io.\*;

import java.util.HashMap;

import java.util.\*;

import javax.swing.filechooser.\*;

public class ManagerWindow extends JFrame implements ActionListener{ public static void main(String args[]){

new ManagerWindow();

}

InputStudent 基本信息录入 ;

ModifySituation 基本信息修改 ;

Inquest 基本信息查询与打印 ;

Delete 基本信息删除 ;

JMenuBar bar;

JMenu fileMenu;

JMenuItem 录入, 修改, 查询与打印 , 删除, 欢迎界面 ;

HashMap<String,Student> 基本信息 =null;

File file=null;

CardLayout card=null;

JLabel label=null;

JPanel pCenter;

public ManagerWindow(){

录入 =new JMenuItem("录入学生基本信息 ");

修改 =new JMenuItem("修改学生基本信息 ");

查询与打印 =new JMenuItem("查询与打印学生基本信息 "); 删除 =new JMenuItem("删除学生基本信息 "); 欢迎界面 =new JMenuItem("欢迎界面 ");

bar=new JMenuBar();

10

fileMenu=new JMenu(" 菜单选项 ");

fileMenu.add( 录入);

fileMenu.add( 修改);

fileMenu.add( 查询与打印 );

fileMenu.add( 删除);

fileMenu.add( 欢迎界面 );

bar.add(fileMenu);

setJMenuBar(bar);

label=new JLabel(" 学籍管理系统 ",JLabel.CENTER); label.setIcon(new ImageIcon("welcome.jpg"));

label.setFont(new Font(" 隶书",Font.BOLD,36)); label.setHorizontalTextPosition(SwingConstants.CENTER); label.setForeground(Color.red);

基本信息 =new HashMap<String,Student>();

录入 .addActionListener(this);

修改 .addActionListener(this);

查询与打印 .addActionListener(this);

删除 .addActionListener(this);

欢迎界面 .addActionListener(this);

card=new CardLayout();

pCenter=new JPanel();

pCenter.setLayout(card);

file=new File(" 基本信息 .txt");

if(!file.exists()){

try{

FileOutputStream out=new FileOutputStream(file); ObjectOutputStream objectOut=new ObjectOutputStream(out);

objectOut.writeObject( 基本信息 );

objectOut.close();

out.close();

}

catch(IOException e){}

}

基本信息录入 =new InputStudent(file);

基本信息修改 =new ModifySituation(file);

基本信息查询与打印 =new Inquest(file);

11

基本信息删除 =new Delete(file);

pCenter.add(" 欢迎界面 ",label);

pCenter.add(" 录入界面 ", 基本信息录入 );

pCenter.add(" 修改界面 ", 基本信息修改 );

pCenter.add(" 删除界面 ", 基本信息删除 );

add(pCenter,BorderLayout.CENTER);

validate();

setVisible(true);

setBounds(100,50,460,280);

setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

int n=JOptionPane.showConfirmDialog(null," 确认退出吗 ?","

确认对话框 ",

JOptionPane.YES\_NO\_OPTION );

if(n==JOptionPane.YES\_OPTION)

System.exit(0);

}});

validate();

}

public void actionPerformed(ActionEvent e){

if(e.getSource()== 录入){

基本信息录入 .clearMess();

card.show(pCenter," 录入界面 ");

}

else if(e.getSource()== 修改){

基本信息修改 .clearMess();

card.show(pCenter," 修改界面 ");

}

else if(e.getSource()== 查询与打印 ){

基本信息查询与打印 .clearMess();

基 本 信 息 查 询 与 打

印.setLocation(getBounds().x+getBounds().width,getBounds().y);

基本信息查询与打印 .setVisible(true);

}

else if(e.getSource()== 删除)

12

card.show(pCenter," 删除界面 ");

else if(e.getSource()== 欢迎界面 )

card.show(pCenter," 欢迎界面 ");

}

}

class Student implements Serializable{

String number,name,discipling,grade,borth,sex;

File imagePic;

public void setNumber(String number){

this.number=number;

}

public String getNumber(){

return number;

}

public void setName(String name){

this.name=name;

}

public String getName(){

return name;

}

public void setDiscipling(String discipling){

this.discipling=discipling;

}

public String getDisciping(){

return discipling;

}

public void setGrade(String grade){

this.grade=grade;

}

public String getGrade(){

return grade;

}

public void setBorth(String borth){

this.borth=borth;

}

13

public String getBorth(){

return borth;

}

public void setSex(String sex){

this.sex=sex;

}

public String getSex(){

return sex;

}

public void setImagePic(File image){

imagePic=image;

}

public File getImagePic(){

return imagePic;

}

}

class StudentPicture extends JPanel{

File imageFile;

Toolkit tool;

public StudentPicture(){

tool=getToolkit();

}

public void setImage(File imageFile){

this.imageFile=imageFile;

repaint();

}

public void paintComponent(Graphics g){

super.paintComponent(g);

int w=getBounds().width;

int h=getBounds().height;

if(imageFile!=null){

Image image=tool.getImage(imageFile.getAbsolutePath()); g.drawImage(image,0,0,w,h,this);

}

else

14

g.drawString(" 没有选择照片图像 ",20,30);

}

}

class InputStudent extends JPanel implements ActionListener{ Student 学生=null;

StudentPicture pic;

HashMap<String,Student> 基本信息表 =null;

JTextField 学号, 姓名 , 年级 , 出生 ;

JButton 选择照片 ;

JComboBox 专业;

JRadioButton 男, 女;

ButtonGroup group=null;

JButton 录入 , 重置 ;

FileInputStream inOne=null;

ObjectInputStream inTwo=null;

FileOutputStream outOne=null;

ObjectOutputStream outTwo=null;

File systemFile,imagePic;

public InputStudent(File file){

systemFile=file;

pic=new StudentPicture();

学号=new JTextField(5);

姓名=new JTextField(5);

专业=new JComboBox();

try{

FileReader inOne=new FileReader(" 专业.txt"); BufferedReader inTwo= new BufferedReader(inOne); String s=null;

int i=0;

while((s=inTwo.readLine())!=null)

专业.addItem(s);

inOne.close();

inTwo.close();

}

catch(IOException exp){

15

专业 .addItem(" 数学 ");

专业 .addItem(" 计算机科学与技术 ");

}

年级=new JTextField(5);

出生=new JTextField(5);

选择照片 =new JButton(" 选择");

group=new ButtonGroup();

男=new JRadioButton(" 男",true);

女=new JRadioButton(" 女",false);

group.add( 男 );

group.add( 女 );

录入=new JButton(" 录入 ");

重置=new JButton(" 重置 ");

录入.addActionListener(this);

选择照片 .addActionListener(this);

重置.addActionListener(this);

Box box1=Box.createHorizontalBox();

box1.add(new JLabel(" 学号 :",JLabel.CENTER));

box1.add( 学号);

Box box2=Box.createHorizontalBox();

box2.add(new JLabel(" 姓名 :",JLabel.CENTER));

box2.add( 姓名);

Box box3=Box.createHorizontalBox();

box3.add(new JLabel(" 性别 :",JLabel.CENTER));

box3.add( 男);

box3.add( 女);

Box box4=Box.createHorizontalBox();

box4.add(new JLabel(" 专业 :",JLabel.CENTER));

box4.add( 专业);

Box box5=Box.createHorizontalBox();

box5.add(new JLabel(" 年级 :",JLabel.CENTER));

box5.add( 年级);

Box box6=Box.createHorizontalBox();

box6.add(new JLabel(" 出生 :",JLabel.CENTER));

box6.add( 出生);

Box boxH=Box.createVerticalBox();

16

boxH.add(box1);

boxH.add(box2);

boxH.add(box3);

boxH.add(box4);

boxH.add(box5);

boxH.add(box6);

boxH.add(Box.createVerticalGlue());

JPanel picPanel=new JPanel();

picPanel.setBackground(Color.green);

picPanel.add(new JLabel(" 选择照片 :",JLabel.CENTER));

picPanel.add( 选择照片 );

JPanel putButton=new JPanel();

putButton.add( 录入);

putButton.add( 重置);

JPanel messPanel=new JPanel();

messPanel.add(boxH);

messPanel.setBackground(Color.cyan);

putButton.setBackground(Color.red);

setLayout(new BorderLayout());

JSplitPane splitV=

new JSplitPane(JSplitPane.VERTICAL\_SPLIT,picPanel,pic);

JSplitPane splitH=new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT,messPanel,splitV);

add(splitH,BorderLayout.CENTER);

add(putButton,BorderLayout.SOUTH);

validate();

}

public void actionPerformed(ActionEvent e){

if(e.getSource()== 录入){

String number="";

number= 学号.getText();

if(number.length()>0){

try {

inOne=new FileInputStream(systemFile);

inTwo=new ObjectInputStream(inOne);

基本信息表 =(HashMap<String,Student>)inTwo.readObject();

17

inOne.close();

inTwo.close();

}

catch(Exception ee){}

if( 基本信息表 .containsKey(number)){

String warning=" 该生基本信息已存在 , 请到修改页面修改 !";

JOptionPane.showMessageDialog(this,warning," 警告",

JOptionPane.WARNING\_MESSAGE);

}

else{

String m=" 基本信息将被录入 !";

int ok=JOptionPane.showConfirmDialog(this,m," 确认",

JOptionPane.YES\_NO\_OPTION,JOptionPane.INFORMATION\_MESSAGE); if(ok==JOptionPane.YES\_OPTION){

String name= 姓名.getText();

String discipling=(String) 专业 .getSelectedItem();

String grade= 年级 .getText();

String borth= 出生 .getText();

String sex=null;

if( 男.isSelected())

sex= 男.getText();

else

sex= 女.getText();

学生 =new Student();

学生 .setNumber(number);

学生 .setName(name);

学生 .setDiscipling(discipling);

学生 .setGrade(grade);

学生 .setBorth(borth);

学生 .setSex(sex);

学生 .setImagePic(imagePic);

try { outOne=new FileOutputStream(systemFile);

outTwo=new ObjectOutputStream(outOne);

基本信息表 .put(number, 学生);

outTwo.writeObject( 基本信息表 );

18

outTwo.close();

outOne.close();

clearMess();

}

catch(Exception ee){}

}

}

}

else{

String warning=" 必须要输入学号 !";

JOptionPane.showMessageDialog(this,warning," 警 告

",JOptionPane.WARNING\_MESSAGE);

}

}

else if(e.getSource()== 选择照片 ){ JFileChooser chooser=new JFileChooser(); FileNameExtensionFilter filter = new FileNameExtensionFilter(

"JPG & GIF Images", "jpg", "gif"); chooser.setFileFilter(filter);

int state=chooser.showOpenDialog(null); File choiceFile=chooser.getSelectedFile(); if(choiceFile!=null&&state==JFileChooser.APPROVE\_OPTION){

选择照片 .setText(choiceFile.getName());

imagePic=choiceFile;

pic.setImage(imagePic);

pic.repaint();

}

}

else if(e.getSource()== 重置){

clearMess();

}

}

public void clearMess(){

学号.setText(null);

姓名.setText(null);

年级.setText(null);

19

出生.setText(null);

选择照片 .setText(" 选择");

imagePic=null;

pic.setImage(imagePic);

pic.repaint();

}

}

class Inquest extends JDialog implements ActionListener{ StudentPicture pic;

PrintJob print;

Graphics g=null;

HashMap<String,Student> 基本信息表 ;

JTextField 学号, 姓名 , 专业 , 年级 , 出生;

JRadioButton 男, 女;

JButton 查询 , 打印文本及照片 , 打印文本 , 打印照片 ; ButtonGroup group=null;

FileInputStream inOne=null;

ObjectInputStream inTwo=null;

File systemFile=null;

JPanel messPanel;

Student stu;

public Inquest(File file){

setTitle(" 查询对话框 ");

setModal(false);

systemFile=file;

pic=new StudentPicture();

学号=new JTextField(6);

查询=new JButton(" 查询 ");

学号.addActionListener(this);

查询.addActionListener(this);

打印文本及照片 =new JButton(" 打印文本及照片 "); 打印文本 =new JButton(" 打印文本 ");

打印照片 =new JButton(" 打印照片 ");

打印文本及照片 .addActionListener(this);

打印文本 .addActionListener(this);

20

打印照片 .addActionListener(this);

打印文本及照片 .setEnabled(false);

打印文本 .setEnabled(false);

打印照片 .setEnabled(false);

姓名=new JTextField(5);

姓名.setEditable(false);

专业=new JTextField(5);

专业.setEditable(false);

年级=new JTextField(5);

年级.setEditable(false);

出生=new JTextField(5);

出生.setEditable(false);

男=new JRadioButton(" 男",false);

女=new JRadioButton(" 女",false);

group=new ButtonGroup();

group.add( 男 );

group.add( 女 );

Box box1=Box.createHorizontalBox();

box1.add(new JLabel(" 学号 :",JLabel.CENTER));

box1.add( 学号);

box1.add( 查询);

Box box2=Box.createHorizontalBox();

box2.add(new JLabel(" 姓名 :",JLabel.CENTER));

box2.add( 姓名);

Box box3=Box.createHorizontalBox();

box3.add(new JLabel(" 性别 :",JLabel.CENTER));

box3.add( 男);

box3.add( 女);

Box box4=Box.createHorizontalBox();

box4.add(new JLabel(" 专业 :",JLabel.CENTER));

box4.add( 专业);

Box box5=Box.createHorizontalBox();

box5.add(new JLabel(" 年级 :",JLabel.CENTER));

box5.add( 年级);

Box box6=Box.createHorizontalBox();

box6.add(new JLabel(" 出生 :",JLabel.CENTER));

21

box6.add( 出生);

Box boxH=Box.createVerticalBox();

boxH.add(box1);

boxH.add(box2);

boxH.add(box3);

boxH.add(box4);

boxH.add(box5);

boxH.add(box6);

boxH.add(Box.createVerticalGlue());

messPanel=new JPanel();

messPanel.add(boxH);

JSplitPane splitH=new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT,messPanel,pic);

add(splitH,BorderLayout.CENTER);

JPanel pSouth=new JPanel();

pSouth.add( 打印文本及照片 );

pSouth.add( 打印文本 );

pSouth.add( 打印照片 );

add(pSouth,BorderLayout.SOUTH);

validate();

setVisible(false);

setSize(360,230);

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

setVisible(false);

}});

}

public void actionPerformed(ActionEvent e){

if(e.getSource()== 查询 ||e.getSource()== 学号){ String number="";

number= 学号 .getText();

if(number.length()>0){

try {

inOne=new FileInputStream(systemFile);

inTwo=new ObjectInputStream(inOne);

基本信息表 =(HashMap<String,Student>)inTwo.readObject();

22

inOne.close();

inTwo.close();

}

catch(Exception ee){}

if( 基本信息表 .containsKey(number)){

stu= 基本信息表 .get(number);

姓名.setText(stu.getName());

专业.setText(stu.getDisciping());

年级.setText(stu.getGrade());

出生.setText(stu.getBorth());

if(stu.getSex().equals(" 男"))

男.setSelected(true);

else

女.setSelected(true);

pic.setImage(stu.getImagePic());

pic.repaint();

打印文本及照片 .setEnabled(true);

打印文本 .setEnabled(true);

打印照片 .setEnabled(true);

}

else{

打印文本及照片 .setEnabled(false);

打印文本 .setEnabled(false);

打印照片 .setEnabled(false);

String warning=" 该学号不存在 !";

JOptionPane.showMessageDialog(this,warning," 警告 ",

JOptionPane.WARNING\_MESSAGE);

clearMess();

}

}

else{

打印文本及照片 .setEnabled(false);

打印文本 .setEnabled(false);

打印照片 .setEnabled(false);

String warning=" 必须要输入学号 !";

JOptionPane.showMessageDialog(this,warning," 警告",

23

JOptionPane.WARNING\_MESSAGE);

}

}

else if(e.getSource()== 打印文本及照片 ){

try{

print=getToolkit().getPrintJob(new JFrame()," 打 印 ",new

Properties());

g=print.getGraphics();

g.translate(120,200);// 在打印页的坐标 (120,200) 开始打印文本内容

int w=messPanel.getBounds().width;

messPanel.printAll(g);

g.translate(w,0); // 在打印页的坐标 (120+w,200) 处打印照片

pic.printAll(g);

姓名.setText(stu.getName());

g.dispose();

print.end();

}

catch(Exception exp){}

}

else if(e.getSource()== 打印文本 ){

try{

print=getToolkit().getPrintJob(new JFrame()," 打 印 ",new

Properties());

g=print.getGraphics();

g.translate(120,200);// 在打印页的坐标 (120,200) 开始打印文本内容

messPanel.printAll(g);

g.dispose();

print.end();

}

catch(Exception exp){}

}

else if(e.getSource()== 打印照片 ){

try{

print=getToolkit().getPrintJob(new JFrame()," 打 印 ",new

Properties());

g=print.getGraphics();

24

int w=messPanel.getBounds().width;

g.translate(120+w,200);// 在打印页的坐标 (120+w,200)处打印照片

pic.printAll(g);

g.dispose();

print.end();

}

catch(Exception exp){}

}

}

public void clearMess(){

学号.setText(null);

姓名.setText(null);

年级.setText(null);

出生.setText(null);

专业.setText(null);

pic.setImage(null);

pic.repaint();

}

}

class ModifySituation extends JPanel implements ActionListener{ StudentPicture pic;

HashMap<String,Student> 基本信息表 =null;

JTextField 学号, 姓名, 年级, 出生 ;

JComboBox 专业 ;

JButton 选择照片 ;

JRadioButton 男, 女;

ButtonGroup group=null;

JButton 开始修改 , 录入修改 , 重置;

FileInputStream inOne=null;

ObjectInputStream inTwo=null;

FileOutputStream outOne=null;

ObjectOutputStream outTwo=null;

File systemFile,imagePic;

JComboBox oldMess;

Student stu=null;

public ModifySituation(File file){

25

systemFile=file;

pic=new StudentPicture();

学号=new JTextField(6);

姓名=new JTextField(6);

专业=new JComboBox();

try{

FileReader inOne=new FileReader(" 专业.txt"); BufferedReader inTwo= new BufferedReader(inOne); String s=null;

int i=0;

while((s=inTwo.readLine())!=null)

专业.addItem(s);

inOne.close();

inTwo.close();

}

catch(IOException exp){

专业.addItem(" 数学");

专业.addItem(" 计算机科学与技术 ");

}

年级=new JTextField(6);

出生=new JTextField(6);

选择照片 =new JButton(" 选择 ");

group=new ButtonGroup();

男=new JRadioButton(" 男",true);

女=new JRadioButton(" 女",false);

group.add( 男);

group.add( 女);

oldMess=new JComboBox();

开始修改 =new JButton(" 开始修改 ");

录入修改 =new JButton(" 录入修改 ");

录入修改 .setEnabled(false);

选择照片 .setEnabled(false);

重置=new JButton(" 重置");

学号.addActionListener(this);

开始修改 .addActionListener(this);

录入修改 .addActionListener(this);

26

重置.addActionListener(this);

选择照片 .addActionListener(this);

Box box1=Box.createHorizontalBox();

box1.add(new JLabel(" 输入要修改信息的学号 :",JLabel.CENTER));

box1.add( 学号 );

box1.add( 开始修改 );

Box box2=Box.createHorizontalBox();

box2.add(new JLabel("( 新) 姓名 :",JLabel.CENTER));

box2.add( 姓名 );

Box box3=Box.createHorizontalBox();

box3.add(new JLabel("( 新) 性别 :",JLabel.CENTER));

box3.add( 男);

box3.add( 女);

Box box4=Box.createHorizontalBox();

box4.add(new JLabel("( 新) 专业 :",JLabel.CENTER));

box4.add( 专业 );

Box box5=Box.createHorizontalBox();

box5.add(new JLabel("( 新) 年级 :",JLabel.CENTER));

box5.add( 年级 );

Box box6=Box.createHorizontalBox();

box6.add(new JLabel("( 新) 出生 :",JLabel.CENTER));

box6.add( 出生 );

Box box7=Box.createHorizontalBox();

box7.add(new JLabel(" 学生已有的数据 :",JLabel.CENTER));

box7.add(oldMess);

Box boxH=Box.createVerticalBox();

boxH.add(box1);

boxH.add(box2);

boxH.add(box3);

boxH.add(box4);

boxH.add(box5);

boxH.add(box6);

boxH.add(box7);

boxH.add(Box.createVerticalGlue());

JPanel picPanel=new JPanel();

picPanel.setBackground(Color.green);

27

picPanel.add(new JLabel(" 选择照片 :",JLabel.CENTER));

picPanel.add( 选择照片 );

JPanel putButton=new JPanel();

putButton.setBackground(Color.yellow);

putButton.add( 录入修改 );

putButton.add( 重置 );

JPanel messPanel=new JPanel();

messPanel.setBackground(Color.pink);

messPanel.add(boxH);

setLayout(new BorderLayout());

JSplitPane splitV=

new JSplitPane(JSplitPane.VERTICAL\_SPLIT,picPanel,pic);

JSplitPane splitH=new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT,messPanel,splitV);

add(splitH,BorderLayout.CENTER);

add(putButton,BorderLayout.SOUTH);

validate();

}

public void actionPerformed(ActionEvent e){

if(e.getSource()== 开始修改 ||e.getSource()== 学号){ oldMess.removeAllItems();

String number="";

imagePic=null;

stu=null;

number= 学号.getText();

if(number.length()>0){

try {

inOne=new FileInputStream(systemFile);

inTwo=new ObjectInputStream(inOne);

基本信息表 =(HashMap<String,Student>)inTwo.readObject();

inOne.close();

inTwo.close();

}

catch(Exception ee){}

if( 基本信息表 .containsKey(number)){

录入修改 .setEnabled(true);

28

选择照片 .setEnabled(true);

stu= 基本信息表 .get(number);

oldMess.addItem(" 姓名 :"+stu.getName());

oldMess.addItem(" 专业 :"+stu.getDisciping());

oldMess.addItem(" 年级 :"+stu.getGrade());

oldMess.addItem(" 出生日期 :"+stu.getBorth());

if(stu.getSex().equals(" 男"))

* .setSelected(true);

else

* .setSelected(true);

imagePic=stu.getImagePic();

pic.setImage(imagePic);

pic.repaint();

姓名 .setText(stu.getName());

年级 .setText(stu.getGrade());

出生 .setText(stu.getBorth());

}

else{

录入修改 .setEnabled(false);

选择照片 .setEnabled(false);

String warning=" 该学号不存在 !";

JOptionPane.showMessageDialog(this,warning," 警告 ",

JOptionPane.WARNING\_MESSAGE);

clearMess();

}

}

else{

录入修改 .setEnabled(false);

选择照片 .setEnabled(false);

String warning=" 必须要输入学号 !";

JOptionPane.showMessageDialog(this,warning," 警 告

",JOptionPane.WARNING\_MESSAGE);

clearMess();

}

}

else if(e.getSource()== 录入修改 ){

29

String number="";

number= 学号.getText();

if(number.length()>0){

try {

inOne=new FileInputStream(systemFile);

inTwo=new ObjectInputStream(inOne);

基本信息表 =(HashMap<String,Student>)inTwo.readObject();

inOne.close();

inTwo.close();

}

catch(Exception ee){}

if( 基本信息表 .containsKey(number)){

String question=" 该生基本信息已存在 , 您想修改他 ( 她 ) 的基本信息

吗?";

JOptionPane.showMessageDialog(this,question," 警告",

JOptionPane.QUESTION\_MESSAGE);

String m=" 基本信息将被修改 !";

int ok=JOptionPane.showConfirmDialog(this,m," 确认",

JOptionPane.YES\_NO\_OPTION,JOptionPane.INFORMATION\_MESSAGE); if(ok==JOptionPane.YES\_OPTION){

String name= 姓名 .getText();

if(name.length()==0)

name=stu.getName();

String discipling=(String) 专业 .getSelectedItem();

if(discipling==null)

discipling=stu.getDisciping();

String grade= 年级.getText();

if(grade.length()==0)

grade=stu.getGrade();

String borth= 出生.getText();

if(borth.length()==0)

borth=stu.getBorth();

String sex=null;

if( 男.isSelected())

sex= 男.getText();

30

else

sex= 女.getText();

if(imagePic==null)

imagePic=stu.getImagePic();

Student 学生 =new Student();

学生.setNumber(number);

学生.setName(name);

学生.setDiscipling(discipling);

学生.setGrade(grade);

学生.setBorth(borth);

学生.setSex(sex);

学生.setImagePic(imagePic);

try{

outOne=new FileOutputStream(systemFile);

outTwo=new ObjectOutputStream(outOne);

基本信息表 .put(number, 学生 );

outTwo.writeObject( 基本信息表 );

outTwo.close();

outOne.close();

clearMess();

}

catch(Exception ee){}

录入修改 .setEnabled(false);

选择照片 .setEnabled(false);

}

else if(ok==JOptionPane.NO\_OPTION){

录入修改 .setEnabled(true);

选择照片 .setEnabled(true);

}

}

else{

String warning=" 该学号没有基本信息 , 不能修改 !";

JOptionPane.showMessageDialog(this,warning," 警告",

JOptionPane.WARNING\_MESSAGE);

录入修改 .setEnabled(false);

选择照片 .setEnabled(false);

31

clearMess();

}

}

else{

String warning=" 必须要输入学号 !";

JOptionPane.showMessageDialog(this,warning," 警 告

",JOptionPane.WARNING\_MESSAGE);

录入修改 .setEnabled(false);

clearMess();

录入修改 .setEnabled(false);

选择照片 .setEnabled(false);

}

}

else if(e.getSource()== 选择照片 ){ JFileChooser chooser=new JFileChooser(); FileNameExtensionFilter filter = new FileNameExtensionFilter(

"JPG & GIF Images", "jpg", "gif"); chooser.setFileFilter(filter);

int state=chooser.showOpenDialog(null); File choiceFile=chooser.getSelectedFile(); if(choiceFile!=null&&state==JFileChooser.APPROVE\_OPTION){

选择照片 .setText(choiceFile.getName());

imagePic=choiceFile;

pic.setImage(imagePic);

pic.repaint();

}

}

else if(e.getSource()== 重置){

clearMess();

录入修改 .setEnabled(false);

选择照片 .setEnabled(false);

}

}

public void clearMess(){

学号.setText(null);

姓名.setText(null);

32

年级.setText(null);

出生.setText(null);

oldMess.removeAllItems();

选择照片 .setText(" 选择");

imagePic=null;

pic.setImage(imagePic);

pic.repaint();

}

}

class Delete extends JPanel implements ActionListener{ HashMap<String,Student> 基本信息表 =null; JTextField 学号, 姓名, 专业, 年级 , 出生 ;

JRadioButton 男, 女;

JButton 删除;

ButtonGroup group=null;

FileInputStream inOne=null;

ObjectInputStream inTwo=null;

FileOutputStream outOne=null;

ObjectOutputStream outTwo=null;

File systemFile=null;

public Delete(File file){

systemFile=file;

学号=new JTextField(10);

删除=new JButton(" 删除");

学号.addActionListener(this);

删除.addActionListener(this);

姓名=new JTextField(10);

姓名.setEditable(false);

专业=new JTextField(10);

专业.setEditable(false);

年级=new JTextField(10);

年级.setEditable(false);

出生=new JTextField(10);

出生.setEditable(false);

男=new JRadioButton(" 男",false);

33

女=new JRadioButton(" 女",false);

group=new ButtonGroup();

group.add( 男);

group.add( 女);

Box box1=Box.createHorizontalBox();

box1.add(new JLabel(" 输入要删除的学号 :",JLabel.CENTER));

box1.add( 学号 );

box1.add( 删除 );

Box box2=Box.createHorizontalBox();

box2.add(new JLabel(" 姓名:",JLabel.CENTER));

box2.add( 姓名 );

Box box3=Box.createHorizontalBox();

box3.add(new JLabel(" 性别:",JLabel.CENTER));

box3.add( 男);

box3.add( 女);

Box box4=Box.createHorizontalBox();

box4.add(new JLabel(" 专业:",JLabel.CENTER));

box4.add( 专业 );

Box box5=Box.createHorizontalBox();

box5.add(new JLabel(" 年级:",JLabel.CENTER));

box5.add( 年级 );

Box box6=Box.createHorizontalBox();

box6.add(new JLabel(" 出生:",JLabel.CENTER));

box6.add( 出生 );

Box boxH=Box.createVerticalBox();

boxH.add(box1);

boxH.add(box2);

boxH.add(box3);

boxH.add(box4);

boxH.add(box5);

boxH.add(box6);

boxH.add(Box.createVerticalGlue());

JPanel pCenter=new JPanel();

pCenter.add(boxH);

setLayout(new BorderLayout());

add(pCenter,BorderLayout.CENTER);

34

validate();

}

public void actionPerformed(ActionEvent e){

if(e.getSource()== 删除||e.getSource()== 学号 ){ String number="";

number= 学号 .getText();

if(number.length()>0){

try {

inOne=new FileInputStream(systemFile);

inTwo=new ObjectInputStream(inOne);

基本信息表 =(HashMap)inTwo.readObject();

inOne.close();

inTwo.close();

}

catch(Exception ee){}

if( 基本信息表 .containsKey(number)){

Student stu=(Student) 基本信息表 .get(number);

姓名.setText(stu.getName());

专业.setText(stu.getDisciping());

年级.setText(stu.getGrade());

出生.setText(stu.getBorth());

if(stu.getSex().equals(" 男"))

男.setSelected(true);

else

女.setSelected(true);

String m=" 确定要删除该学号及全部信息吗 ?";

int ok=JOptionPane.showConfirmDialog(this,m," 确认 ",

JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE); if(ok==JOptionPane.YES\_OPTION){

基本信息表 .remove(number);

try{

outOne=new FileOutputStream(systemFile);

outTwo=new ObjectOutputStream(outOne);

outTwo.writeObject( 基本信息表 );

outTwo.close();

outOne.close();

35

学号 .setText(null);

姓名 .setText(null);

专业 .setText(null);

年级 .setText(null);

出生 .setText(null);

}

catch(Exception ee){}

}

else if(ok==JOptionPane.NO\_OPTION){

学号.setText(null);

姓名.setText(null);

专业.setText(null);

年级.setText(null);

出生.setText(null);

}

}

else{

String warning=" 该学号不存在 !";

JOptionPane.showMessageDialog(this,warning," 警告 ",

JOptionPane.WARNING\_MESSAGE);

学号 .setText(null);

}

}

else{

String warning=" 必须要输入学号 !";

JOptionPane.showMessageDialog(this,warning," 警告 ",

JOptionPane.WARNING\_MESSAGE);

}

}

}

}

36



4 系统测试

4.1ManagerWindow 创建窗口

菜单选项： JMenultem

4.2InputStudent 学生信息录入窗口

37



4.3Inquest 查询，打印窗口

4.4 ModifySituation 信息修改窗口

4.5Delete 删除信息窗口

38



总 结

本次面向对象程序课程设计历时两周时间，说简单又不是很简单，说复杂也都是我们了解过的东西。总而言之，这两周时间的付出我们得到了应有的收获，学到了不少知识，也更加深刻的了解和认识到了 Java。

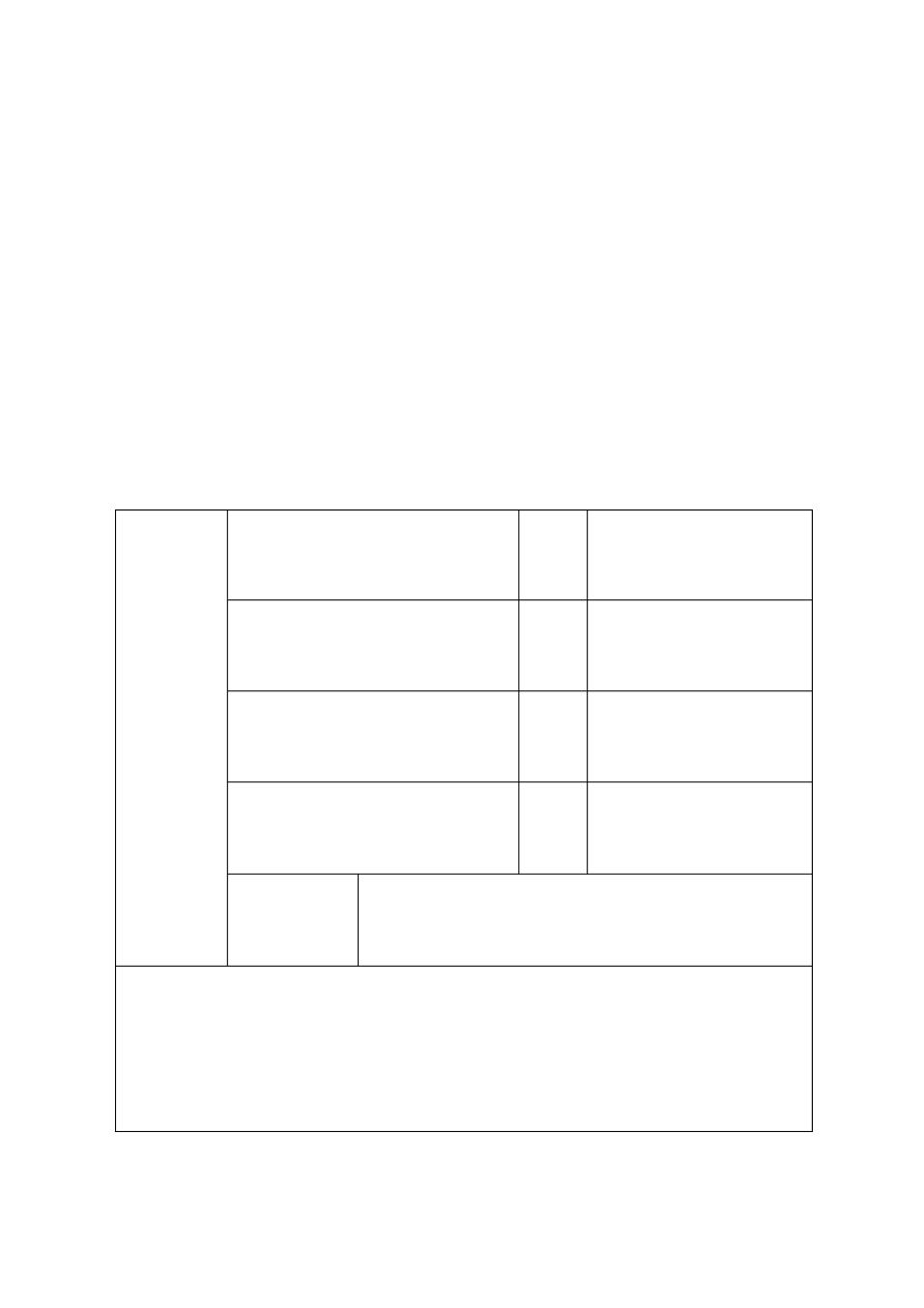
首先，我们都会确定自己的课题，熟悉该课题主要涉及的目的，熟悉课程涉及的要求。编写程序之前首先要进行总体设计，确定该设计需要有那几个部分组成，并确定各部分的功能，确定主类。分析完成之后画出程序流程图，然后开始编写程序，程序编写这一个环节特别重要，一定不能出问题，所以一定要仔细，认真。这个地方也是出错最多的一个环节，然后就一直在翻书，然后找一些资料，或者跟同学讨论一下问题所在，慢慢的程序也就完整的写了出来。其实学生学籍管理系统的代码还不算很复杂，它的几个类之间还是有很多相似之处的，没各类就算单独出现错误还是比较容易查出来的。总体来说，这次课程设计还是比较顺利的，没有什么特别难的问题，写的东西还是有点多的，不过最后还是把它完成了，也算是有所收获。

这次的课程设计不仅仅巩固了 java 的知识，也让我对编程的步骤有了新的理解， 以前经常就是先看程序，然后清理各部分之间的关系，最后才会画出流程图。但是现在我们做课程设计都是一开始就画出整个程序的流程图，跟以前那种错误的方法完全相反。

两周的时间过的很快，我们学会了很多，不仅是知识上的，还有实践的东西，也感谢学校能开设课程设计让我们学习。当然，还得感谢老师和同学的帮助，当我在遇到一些问题准备放弃的时候，同学会给我一些鼓励，让我能继续坚持下去。也许就是这样，我在烦

躁的时候，同学就能轻松的帮我找出我的问题， 让我慢慢的完成了我的课程设计， 总言之，我相信我们以后能学到更多的东西，不断的进步，这些对我们以后的生活和学习都有不小的帮助。

39



课程设计成绩评定表

项 目 比例 得 分

平时成绩（百分制记分） 30%

成

绩

业务考核成绩（百分制记分） 70%

评

定

总评成绩（百分制记分） 100%

评定等级 优 良 中 及格 不及格

指导教师（签名）：

20 年 月 日

40