北京石油化工学院

BEIJING INSTITUTE OF PETROCHEMICAL TECHNOLOGY

实验报告

实验名称： 学生选课 GUI 编程实现

课程名称： Java 技术及应用

班级：大数据 181

姓名：王 叶

学号：2018310977

教师：张世博

信息工程学院计算机系

实验名称	学生选课系统简单实现	成绩评定	
评分标准	1.态度 2 分；2.报告完整性 3 分；3.代码准确性 3 分；4.调试过程 1 分；5 结果分析 1 分		
实验目的	1、充分了解分析系统需求，从学生选课角度了解系统中的实体及其关系，学会定义类中的属性以及方法； 2、基本掌握 GUI 窗体及其组件的使用（窗体组件 AWT 或 Swing 任选） 3、初步学会组件的布局 4、基本掌握事件编程的方法 。		

一、实验的业务要求：

一、说明

在**实验二**的基础上，设计图形用户界面，实现基本操作。

说明：学校有“人员”，分为“教师”和“学生”，教师教授“课程”，学生选择课程。从简化系统考虑，每名教师仅教授一门课程，每门课程的授课教师也仅有一位，每名学生选仅选一门课程。

属性示例：　　人员（编号、姓名、性别……）

教师（编号、姓名、性别、所授课程、……）

学生（编号、姓名、性别、所选课程、……）

课程（编号、课程名称、上课地点、时间、授课教师、……）

以上属性仅为示例，同学们可以自行扩展。

注：此部分在上次实验中已经初步完成，但是没有提供可供操作的图形界面，接下来，可利用上次实验所设计的基础类，再扩展界面，实现事件模型编程。

二、要求:

1、 设计 GUI 界面。

2、 编程事件模型，实现在界面上支持操作（诸如：开课、选课、退课、打印列表等操作）。

二、实验过程：

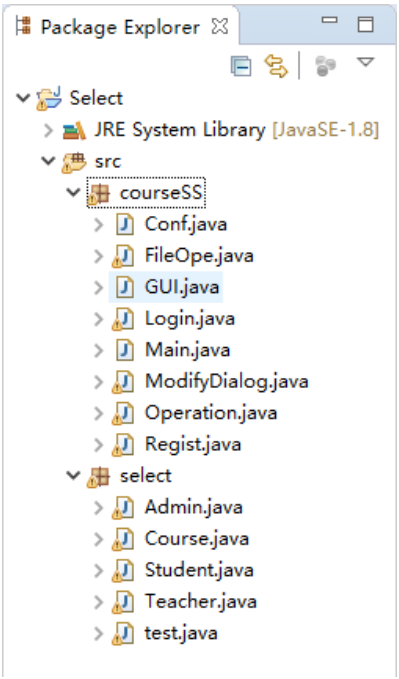
2.1 实验思路：

1. 如何组织界面

在这个项目中，需要用到一下几个界面：登录界面，注册界面，操作界面和修改界面。很明显，这些界面各自有自己的控件和事件，4 个界面应该分 4 个类，在各个类里面负责界面的界面元素和事件处理，是比较好的方法。

设计出来的类如下：

1. Login：登录界面。
2. Resgist：注册界面。
3. Operation：操作界面。
4. ModifyFrame：修改界面。



其次的类还有：

3. FileOpe：文件存储类。
4. GUI：GUI 的通用设计类。
5. Conf：静态成员保存类。

2. 如何访问文件

该项目在几个界面类中都用到了数据存储的操作，如果将 Save 操作的代码分散在多个界面类中，维护性较差。因此，有必要将数据储存操作的代码专门放在一个类中，让各个界面调用。

3. 如何保持状态

将项目划分为几个模块之后，模块之间的数据传递难度增大了。比如，在登录界面中，登录成功之后，系统就应该记住该用户的所有信息；否则到了操作界面，无法知道是谁在登录，到了修改界面，更无法显示其详细信息。

有很多方法可以保存其状态。这里采用“静态变量法”。该方法就是将各个模块之间需要共享的数据保存在其某给类的静态变量中。静态变量一旦赋值，在另一个时刻访问，仍然是这个值，因此，可以用静态变量来传送数据。

设计的类如下：

Conf：内含 5 个静态成员。

1.public static String account; //保存登录用户的账号。

2.public static String password; //保存登录用户的密码。

3.public static String number; //保存登录用户的编号。

4.public static String name; //保存登录用户的姓名。

5.public static String dept; //保存登录用户的身份。

6.public static ArrayList cours //保存登录用户的所选课程

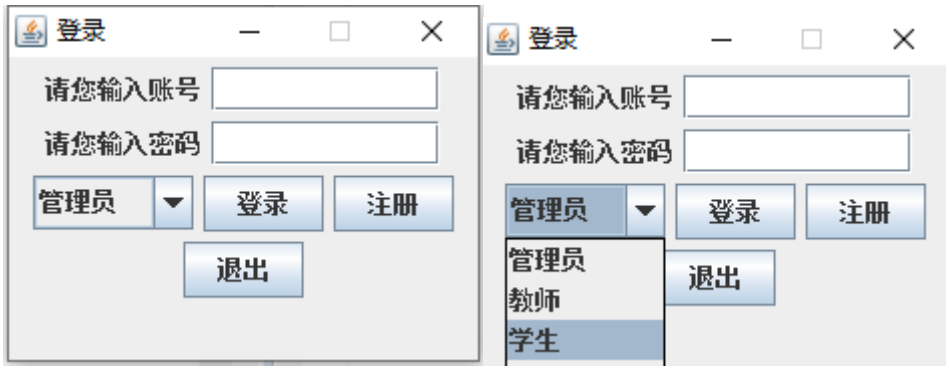
4.其他公共功能

在本项目中，界面都要显示在屏幕中间，因此，可以编写一段公用代码完成这个功能。该公用代码放在 GUI 类中。

2.2.关键技术：

1，登录界面

系统运行，出现登录界面，如图所示。



该界面出现在屏幕中间，在这个界面中：

(1)单击“登录”按钮，能够根据输入的账号密码进行登录；如果登录失败，能够提示；如果登录成功，提示登录成功之后，能到达操作界面。

(2)单击“注册”按钮，登录界面消失，出现注册界面。

(3)单击“退出”按钮，程序退出。

(4)单击“管理员”下拉菜单有四种身份选择。

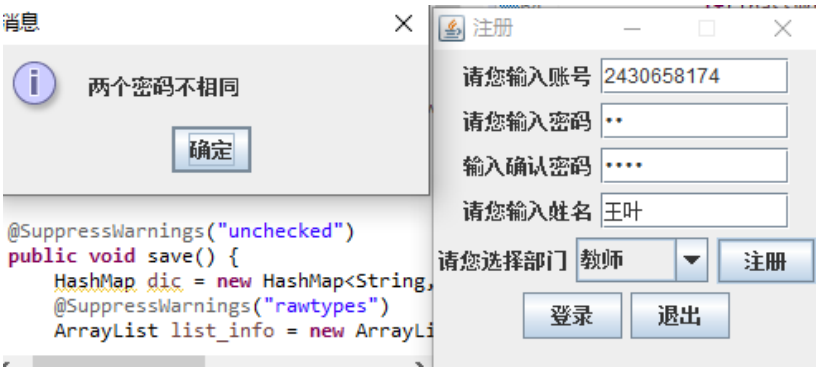
```
public class Login extends JFrame implements ActionListener {
  /*****************************定义各控件*************************/
  private JLabel lbAccount=new JLabel("请您输入账号");
  private JTextField tfAccount=new JTextField(10);
  private JLabel lbPassword=new JLabel("请您输入密码");
  private JPasswordField pfPassword=new JPasswordField(10);
  private JLabel lbDept=new JLabel("请您选择身份");
  private JComboBox cbDept=new JComboBox();
  private JButton btLogin=new JButton("登录");
  private JButton btRegister=new JButton("注册");
 private JButton btExit=new JButton("退出");
public Login() {
      /**********************界面初始化*****************************/
```

```
        super("登录");
        this.setLayout(new FlowLayout());
        this.add(lbAccount);
        this.add(tfAccount);
        this.add(lbPassword);
        this.add(pfPassword);
        this.add(cbDept);
        cbDept.addItem("管理员");
        cbDept.addItem("教师");
        cbDept.addItem("学生");
        cbDept.addItem("录课系统");
        this.add(btLogin);
        this.add(btRegister);
        this.add(btExit);
        this.setSize(240,180);
        GUI.toCenter(this);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setResizable(false);
        this.setVisible(true);
```

设置监听器：

```
        /*****************************增加监听************************/
        btLogin.addActionListener(this);
        btRegister.addActionListener(this);
            btExit.addActionListener(this);
```

实现登陆成功与失败的机制的代码实现：




```
public void actionPerformed(ActionEvent e) {
        if(e.getSource()==btRegister) {
            String password1=new String(pfPassword.getPassword());
            String password2=new String(pfPassword2.getPassword());
            if(!password1.equals(password2)) {
                JOptionPane.showMessageDialog(this,"两个密码不相同");
                return;
            }
```

2. 注册界面

新用户信息录入：

```
FileOpe.updateCustomer(account,password1,name,dept);
FileOpe.getInfoByAccount(account,dept);
```

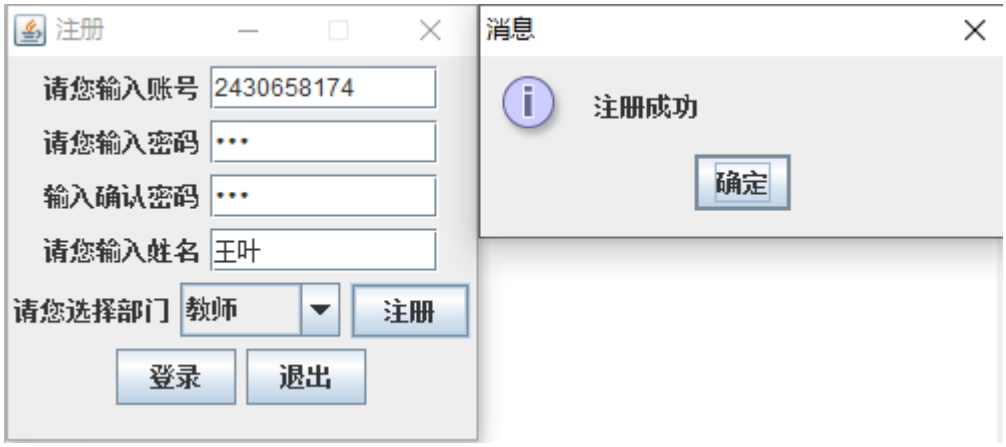
通过上面两种方法的调用实现，详细方法内容如下：

```
public static void getInfoByAccount(String account,String dept) {
    if(dept=="教师") {
        ArrayList a = new ArrayList<String>();
        a=(ArrayList<String>) Admin.teacher_map.get(account);
        Conf.account=account;
        Conf.password=(String) a.get(1);
        Conf.number= (String) a.get(2);
        Conf.name=(String) a.get(3);
          Conf.dept=dept;
    }
public static void updateCustomer(String account,String password,
                                  String name,String dept) {
    if(dept=="教师") {
        Teacher wf = new Teacher();
        wf.set_tc_name(name);
        wf.ac_pw(account,password);
        System.out.println("name"+wf.tc_name);
        System.out.println("number"+wf.tc_number);
        wf.save();
        wf.scanMyslef();
    }
```

仍是通过 Account 作为 keys 值组建字典 Map 以及调用 value 值。

```
{111=[111, 111, 2, 111, []], 2430658174=[2430658174, 111, 4, 王叶, []]}
查询个人信息:
[2430658174, 111, 4, 王叶, []]
```

此时 Console 命令台自动返回相关数据，从中可发现 key 值为 Account，并因为 static 静态量，可以多次存储。但是如果想真正实现存储，仍因存储到文件或数据库中，由于时间和知识储备不足，暂时只能存到静态的字典里。



3. 登陆后界面：

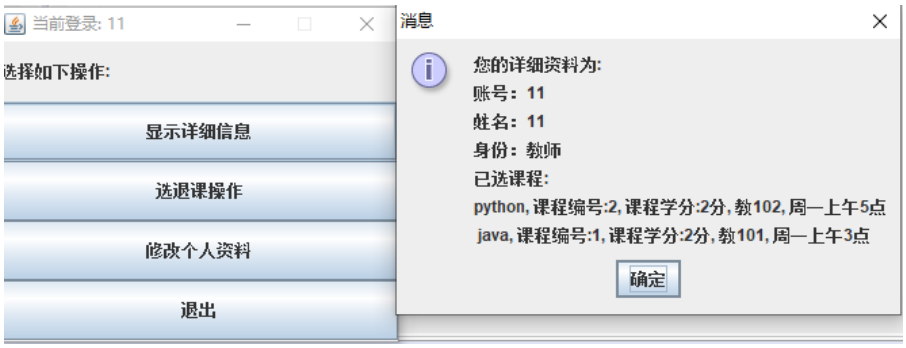
图中看到 fieldtxt 信息为当前账户信息和一个 frame 里面的 4 个按钮。

```
public Operation() {
    /*****************************界面初始化************************/
    super("当前登录: "+Conf.account);
    this.setLayout(new GridLayout(5,1));
    this.add(lblWelcome);
    this.add(btQuery);
    this.add(btselect);
    this.add(btModify);
    this.add(btExit);
    this.setSize(300,250);
    GUI.toCenter(this);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setResizable(false);
    this.setVisible(true);
    /*********************增加监听*******************************/
    btQuery.addActionListener(this);
    btselect.addActionListener(this);
    btModify.addActionListener(this);
    btExit.addActionListener(this);
```

```
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==btQuery) {
            String message="您的详细资料为:\n";
            message+="账号:  "+Conf.account+"\n";
            message+="姓名:  "+Conf.name+"\n";
            message+="身份:  "+Conf.dept+"\n";
            message+="已选课程:\n"+Conf.cours+"\n";
            JOptionPane.showMessageDialog(this,message);
        }
        else if(e.getSource()==btModify) {
            new ModifyDialog(this);
        }
        else if(e.getSource()==btselect) {

        }
        else {
            JOptionPane.showMessageDialog(this,"谢谢光临");
            System.exit(0);
        }
    }
      }
```

显示详细信息的操作展示及代码：



修改个人资料操作展示及代码：

```
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==btModify) {
            String password1=new String(pfPassword.getPassword());
            String password2=new String(pfPassword2.getPassword());
            if(!password1.equals(password2)) {
                JOptionPane.showMessageDialog(this,"两个密码不相同");
                return;
            }
            String name=tfName.getText();
            String dept=(String)cbDept.getSelectedItem();
            //将新的值存入静态变量
            Conf.password=password1;
            Conf.name=name;
            Conf.dept=dept;
            FileOpe.updateCustomer(Conf.account,password1,name,dept);
            JOptionPane.showMessageDialog(this,"修改成功");
        }
        else {
            this.dispose();
        }
          }
```

通过重载覆盖原有信息。

4. GUI 设置类：

```
   public class GUI {
    public static void toCenter(Component comp) {
        GraphicsEnvironment ge=GraphicsEnvironment.getLocalGraphicsEnvironment();
        Rectangle
rec=ge.getDefaultScreenDevice().getDefaultConfiguration().getBounds();
        comp.setLocation(((int)(rec.getWidth()-comp.getWidth())/2),
                ((int)(rec.getHeight()-comp.getHeight()))/2);


    }
}
```

5. 重点实验技术：

1 GUI 控件的类 extend 与监听器的事件 implements：

```
public class Login extends JFrame implements ActionListener
```

2 定义各控件：

```
private JLabel lbAccount=new JLabel("请您输入账号");
private JTextField tfAccount=new JTextField(10);
```

3 界面初始化设置即 add 方法添加组件：

```
this.add(btLogin);
this.add(btRegister);
```

```
        this.add(btExit);
        this.setSize(240,180);
        GUI.toCenter(this);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setResizable(false);
       this.setVisible(true);
```

4 添加监听器：

```
btLogin.addActionListener(this);
btRegister.addActionListener(this);
```

5 添加事件：

```
    public void actionPerformed(ActionEvent e) {
     if(e.getSource()==btRegister) {
         String password1=new String(pfPassword.getPassword());
         String password2=new String(pfPassword2.getPassword());
         if(!password1.equals(password2)) {
             JOptionPane.showMessageDialog(this,"两个密码不相同");
             return;
         }
```

6 显示、关闭窗口：

显示： this.setVisible(true);

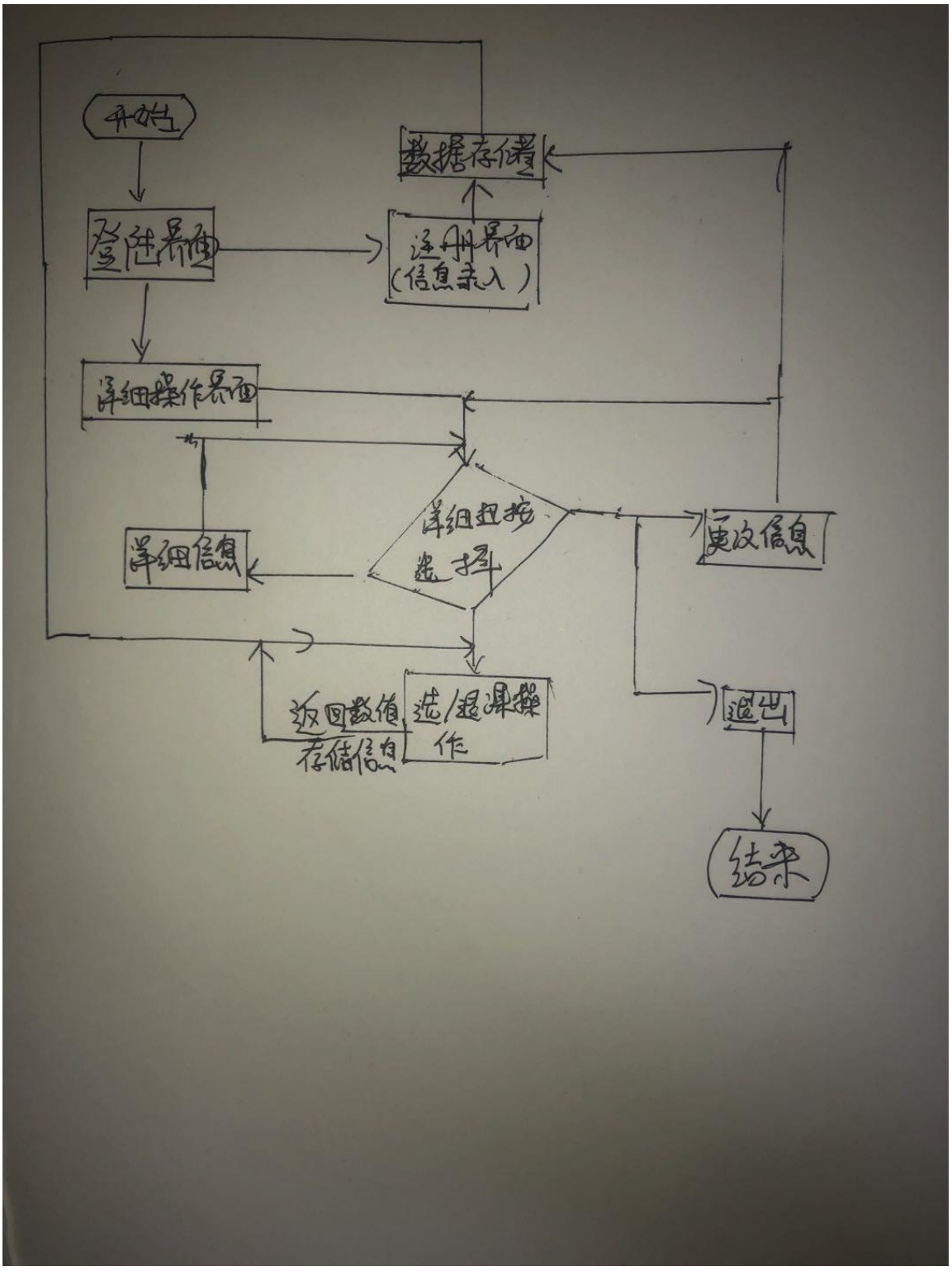
关闭：

主界面主要是有两个 JButton，一个是通过按钮事件调起另一个 JFame，另一个是关闭当前窗体。

1、关闭当前窗体不能使用 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)方法，可以使用 setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

2、通过 JButton 事件不能使用 exit（），这样会使得整个程序的窗体全部关闭，可以使用 dispose（）；这样就只关闭了当前窗体。

三、流程图


开始
数据存储
登陆界面
注册界面
(信息录入)
详细操作界面
详细信息
详细查找
选择
更改信息
返回数值
存储信息
选/退课操作
退出
结束

四、实验感想

此次实验熟悉了 GUI 的大体操作方法，并且通过 Button 等组件，体验到了面向对象编程的好处，在编写各种事件时，只需调用所需对象及其相应方法和变量即可

但是此次 GUI 实验仍有许多不满的地方和以后需要进一步了解的知识

1，数据存储：在此次实验中所有的数据只存储到了静态 static 量中，虽然再一次操作中可以存储多次信息，但是关掉程序后 数据信息就会消失，但是如果存储到数据库中，甚至文件中也可以永久保存数据信息。因此通过此次实验感到自己对数据存储方法的匮乏。

2，代码臃冗：在存储和调出中因有多个类别的对象，虽然操作一样但是还是写了三遍极度相似的代码。下次写代码要注意。

其次在本次实验中也有很多满意的地方：

在设计 CUI 类中想到了代码复用的问题，因为窗体界面都要显示在屏幕中间，因此，可以编写一段公用代码完成这个功能。该公用代码放在 GUI 类中。 也想到了将项目划分为几个模块之后，模块之间的数据传递难度增大了。比如，在登录界面中，登录成功之后，系统就应该记住该用户的所有信息；否则到了操作界面，无法知道是谁在登录，到了修改界面，更无法显示其详细信息。

有很多方法可以保存其状态。这里采用“静态变量法”。该方法就是将各个模块之间需要共享的数据保存在其某给类的静态变量中。静态变量一旦赋值，在另一个时刻访问，仍然是这个值，因此，可以用静态变量来传送数据。

不光是熟悉了 Java GUI 的编程操作，更理解了 Java 的编程思想。