

SQL Server+Java——数据库实验1.x 代码展示

原创

置顶 [天蝎邪月](#) 已于 2022-03-16 21:27:53 修改 5785 收藏 100

文章标签: [数据库](#) [sqlserver](#) [java](#)

于 2020-06-06 18:46:50 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_44664131/article/details/106529009

版权

从第一篇大作业博客[2020-04-27](#)到第二篇[2020-05-06](#), 第三篇[2020-05-25](#), 第四篇[2020-06-03](#)到现在, 一个多月了, 虽然其中出现了一些意外情况, 但所幸, 我的数据库大作业1.x系列可以说是顺利完成了。今天, 我就将我这一个月做的成果展示一下。

另: 文章结尾处有视频讲解演示以及程序源文件

代码展示就按照每个Java类来吧, 就不按照功能分类了。

初始登陆界面 [Login](#)



由一个文本框, 一个密码框, 两个按钮组成

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

import javax.swing.*;

public class Login extends JFrame implements ActionListener,ComponentListener
{
    private TextField tx1;
    private JPasswordField jp; //密码框
    private JButton jb1,jb2;
    private MessageJDialog jdialog;
    private ComponentEvent event;

    static String s1,s2;
    static int a; //区分操作来自于学生还是教师
    private static final long serialVersionUID = 1L;

    public Login()
    {
        super("登录界面");
        this.setBounds(600,240,320,120);
        this.setBackground(Color.lightGray);
        this.setLayout(new GridLayout(3,2));
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
```

```

//第一行
this.add(new Label("用户名",Label.CENTER));
this.tx1=new TextField(15);
this.add(this.tx1);
//第二行
this.add(new Label("密码",Label.CENTER));
this.jp=new JPasswordField(15);
this.add(this.jp);
//第三行
this.jb1 =new JButton("学生登陆") ;
this.add(this.jb1);
this.jb1.addActionListener(this );
this.jb2 =new JButton("教师登陆") ;
this.add(this.jb2);
this.jb2.addActionListener(this );

this.setVisible(true);
this.addWindowListener(new WinClose());
}

public void componentMoved(ComponentEvent e) { }

public void componentShown(ComponentEvent e) { }

public void componentHidden(ComponentEvent e) { }

public void actionPerformed(ActionEvent e)
{
s1=tx1.getText();
s2=new String(jp.getPassword());

if(!s1.equals("")&& !s2.equals(""))
{
try
{
Statement st=Connect.doConnect();
if(e.getSource()==jb1) //学生登录
{
String a0="select * from Student where Sno='\'+s1+'";
//从Student表中查找是否有该学生的学号信息
ResultSet rs=st.executeQuery(a0);
if(rs.next())
{
Search.doSearch(0, 0);
int l=Search.u; //接收来自Search的查询结果, 1为成功, 0为失败
if(l==1)
{
JOptionPane.showMessageDialog(this,"登录成功!");
a=1;
new Student();
dispose(); //关闭当前窗口
}
else
{
JOptionPane.showMessageDialog(this,"用户名或密码输入有误, 请检查输入!");
}
}
}
else

```

```

    {
        JOptionPane.showMessageDialog(this,"该用户名尚未注册，请尽快联系教师管理员进行注册！");
    }
}

else if(e.getSource()==jb2)    //教师登录
{
    Search.doSearch(0, 1);
    int l=Search.u;
    if(l==1)
    {
        JOptionPane.showMessageDialog(this,"登录成功！");
        a=2;
        new Teacher();
        dispose();
    }
    else
    {
        JOptionPane.showMessageDialog(this,"用户名或密码输入有误，请检查输入！");
    }
}
}
catch (SQLException e2)
{
    e2.printStackTrace();
}
}
else
{
    JOptionPane.showMessageDialog(this,"用户名和密码不能为空，请检查输入！");
}
}

private class MessageJDialog extends JDialog
{
    private static final long serialVersionUID = 1L;
    private JLabel jlable;
    private MessageJDialog()
    {
        super(Login.this,"提示",true);
        this.setSize(420, 110);
        this.jlable=new JLabel("",JLabel.CENTER);
        this.getContentPane().add(this.jlable);
        this.setDefaultCloseOperation(HIDE_ON_CLOSE);
        this.addComponentListener(Login.this);
    }
}

public void componentResized(ComponentEvent e)
{
    event = null;
    Component comp=event.getComponent();
    int size=(comp.getWidth()+comp.getHeight())/40;
    Font font=new Font("宋体",1,size);
    if(comp instanceof JFrame)
    {
        int n=this.getContentPane().getComponentCount();
        for(int i=0;i<n;i++)
        {

```

```

        this.getContentPane().getComponent(i).setFont(font);
    }
}
else if(comp instanceof JDialog)
{
    this.jdialog.jlable.setFont(font);
}
}

public static void main(String[] args)
{
    JOptionPane.showMessageDialog(null,"登录名为自己的学号或职工号，初始密码为00000，请登录后及时修改密码！");
    new Login();
}
}

```

之后是学生操作界面 *Student*



```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;

public class Student extends JFrame implements ActionListener
{
    private JButton jb1,jb2,jb3,jb4;
    public JComboBox<String> jcom1;
    private static String[] oper={" ","全部成绩","个人信息","本学期课表"};
    private static final long serialVersionUID = 1L;

    public Student()
    {
        super("学生登录界面");
        this.setBounds(600,240,300,100);
        this.setBackground(Color.lightGray);
        this.setLayout(new GridLayout(2,3));
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);

        //第一行
        this.add(new Label("请选择",Label.CENTER));
        this.add(this.jcom1=new JComboBox<String>(Student.oper));
        this.jcom1.addActionListener(this);
        this.jb1=new JButton("查询");
        this.add(this.jb1);
        this.jb1.addActionListener(this);
        //第二行
        this.jb2=new JButton("选课");
        this.add(this.jb2);
        this.jb2.addActionListener(this);
        this.jb3=new JButton("修改密码");
        this.add(this.jb3);
    }
}

```

```

this.setVisible(true);
this.addActionListener(this);
this.jb4=new JButton("退出");
this.add(this.jb4);
this.jb4.addActionListener(this);

this.setVisible(true);
this.addWindowListener(new WinClose());

}

public void actionPerformed(ActionEvent e)
{
int i=jcom1.getSelectedIndex(); //接收下拉框的选择
try
{
if (e.getSource()==jb4) //退出
{
new Login();
dispose();
}

else if(e.getSource()==jb2) //选课
{
Search.doSearch(1, 1);
new Select_C();
dispose();
}

else if(e.getSource()==jb3) //改密码
{
new Change_code();
dispose();
}

else if(e.getSource()==jb1) //查询
{
try
{
if(i!=0)
{
Search.doSearch(1, i+1);
dispose();
}
else
{
JOptionPane.showMessageDialog(this,"请选择你想进行查询的选项!");
}
}
catch (Exception e2)
{
e2.printStackTrace();
}
}
catch(SQLException w1)
{
w1.printStackTrace();
}
}
}

```

```
// public static void main(String[] args) { new Student();}  
}
```

教师登录操作界面 *Teacher*



```
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
  
public class Teacher extends JFrame implements ActionListener  
{  
    private JButton jb1,jb2,jb3,jb4,jb5;  
    public JComboBox<String> jcom1,jcom2;  
    private static String[] oper1={" ","全部学生成绩","全部学生信息","所有课程平均成绩","选课结果","课表查询"};  
    private static String[] oper2={" ","学生信息","学生成绩","个人密码","增加课程","删除课程"};  
  
    static int a; //在删除信息时使用,1为删除学生信息,2为删除课程信息  
  
    private static final long serialVersionUID = 1L;  
  
    public Teacher()  
    {  
        super("教师登录界面");  
        this.setBounds(600,240,320,180);  
        this.setBackground(Color.lightGray);  
        this.setLayout(new GridLayout(4,2));  
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);  
        this.setVisible(true);  
        this.addWindowListener(new WinClose());  
  
        //第一行  
        this.add(new Label("请选择",Label.CENTER));  
        this.jb1=new JButton("增加学生信息");  
        this.add(this.jb1);  
        this.jb1.addActionListener(this);  
        //第二行  
        this.add(this.jcom1 =new JComboBox<String>(Teacher.oper1));  
        this.jcom1.addActionListener(this);  
        this.jb2=new JButton("查询");  
        this.add(this.jb2);  
        this.jb2.addActionListener(this);  
        //第三行  
        this.add(this.jcom2 =new JComboBox<String>(Teacher.oper2));  
        this.jcom2.addActionListener(this);  
        this.jb3=new JButton("修改");  
        this.add(this.jb3);
```

```

this .jb3.addActionListener(this);
//第四行
this .jb4=new JButton("删除学生信息");
this.add(this .jb4);
this .jb4.addActionListener(this);
this .jb5=new JButton("退出");
this.add(this .jb5);
this .jb5.addActionListener(this);

}

public void actionPerformed(ActionEvent e)
{
int i=this .jcom1.getSelectedIndex(); //接收查询下拉框的选择
int j=this .jcom2.getSelectedIndex(); //接收修改下拉框的选择
if(e.getSource()==jb1) //插入学生信息
{
new Insert();
dispose();
}

else if (e.getSource()==jb2) //查询
{
try
{
if(i!=0)
{
Search.doSearch(2, i);
dispose();
}
else
{
JOptionPane.showMessageDialog(this,"请选择你想进行查询的选项!");
}
}
catch (Exception e2)
{
e2.printStackTrace();
}
}

else if (e.getSource()==jb3) //修改
{
try
{
if(j!=0)
{
a=2;
Search.doSearch(2, j+5);
dispose();
}
else
{
JOptionPane.showMessageDialog(this,"请选择你想进行修改的选项!");
}
}
catch (Exception e2)
{
e2.printStackTrace();
}
}
}

```

```

}
}

else if (e.getSource()==jb4)    // 删除学生信息
{
    a=1;
    new Delete();
    dispose();
}

    else if (e.getSource()==jb5)    //退出
{
    new Login();
    dispose();
}
}

// public static void main(String[] args) {new Teacher(); }
}

```

查询，修改操作统一在一起的类 Search

教师查询平均成绩时直接查询比较麻烦，所以我这里用了视图，先建立相应的视图，然后再查询，关闭显示界面时做一个判断，如果是查询视图的操作，先删除视图，再关闭。

这里说的查询是指学生和教师相应的查询操作，修改是指教师除修改密码操作之外的修改操作。

因为修改密码要进行的操作很难与其他操作统一在一起，所以我另建了一个修改密码的类。

这个类更相当于一个中转站，接受教师或学生的操作选择，再分别转到进行相应操作的不同的类中

```

import java.sql.*;
import javax.swing.*;

public class Search
{
    public int i;    //接收信息, 1: 来自学生端; 2: 来自教师端
    public int j;    //接受各自选项框的选择

    static ResultSet rs;

    static String Ss;    //向其他类传输字符串时的静态变量
    static int u;    //登录检验时的返回值

    static int p,q,m;

    static Statement st=Connect.doConnect();

    public static void doSearch(int i,int j) throws SQLException
    {
        if(st!=null)
        {
            if(i==0)    //登陆检验
            {
                switch(j)
                {
                    case 0:    //学生登陆检验
                        String s0="select * from Student where Sno='\'+Login.s1+'\' and Scode='\'+Login.s2+'\'";
                        ResultSet rs=st.executeQuery(s0);
                        if(rs.next())
                        {

```



```

    u=1;
}
else
{
    u=0;
}
break;
case 1: //教师登陆检验
String s1="select * from Teacher where Tno='"+Login.s1+"' and Tcode='"+Login.s2+"'";
ResultSet rs1=st.executeQuery(s1);
if(rs1.next())
{
    u=1;
}
else
{
    u=0;
}
break;
}
}
else if(i==1) //学生端操作
{
    switch(j)
    {
        case 1: //课程表显示
            String s1="select * from Course where Cnum>0";
            rs=st.executeQuery(s1);
            if(rs.next())
            {
                Ss=s1;
                p=4;

                new Output();
            }
            else
            {
                JOptionPane.showMessageDialog(null,"查询失败,请检查输入!","错误!",JOptionPane.ERROR_MESSAGE);
            }
            break;
        case 2: //学生成绩
            String s2="select * from SC where Grade >0 and Sno='"+Login.s1+"'";
            //st.execute(s1);
            rs=st.executeQuery(s2);
            if(rs.next())
            {
                Ss=s2;
                p=2;

                new Output();
            }
            else
            {
                JOptionPane.showMessageDialog(null,"查询失败,请检查输入!","错误!",JOptionPane.ERROR_MESSAGE);
            }
            break;
        case 3: //学生信息
            String s3="select Sno,Sname,Sage,Ssex,Sdept from Student where Sno='"+Login.s1+"'";
            rs=st.executeQuery(s3);
            if(rs.next())
            {
                Ss=s2;

```

```

        Ss=s3;
        p=1;
            new Output();
        }
        else
        {
            JOptionPane.showMessageDialog(null,"查询失败, 请检查输入!", "错误!",JOptionPane.ERROR_MESSAGE);
        }
        break;
        case 4: //本学期课表
            String s4="select * from SC where Grade =0 and Sno=\'"+Login.s1+"\'";
            rs=st.executeQuery(s4);
            if(rs.next())
            {
                Ss=s4;
                p=2;
                    new Output();
                }
            else
            {
                JOptionPane.showMessageDialog(null,"查询失败, 请检查输入!", "错误!",JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}

else if(i==2) //教师端操作
{
    switch(j)
    {
        case 1: //全部学生成绩
            String s1="select * from SC where Grade>0";
            //st.execute(s1);
            rs=st.executeQuery(s1);
            if(rs.next())
            {
                Ss=s1;
                p=2;
                    new Output();
                }
            else
            {
                JOptionPane.showMessageDialog(null,"查询失败, 请检查输入!", "错误!",JOptionPane.ERROR_MESSAGE);
            }
            break;
        case 2: //全部学生信息
            String s2="select Sno,Sname,Sage,Ssex,Sdept from Student ";
            rs=st.executeQuery(s2);
            if(rs.next())
            {
                Ss=s2;
                p=1;
                    new Output();
                }
            else
            {
                JOptionPane.showMessageDialog(null,"查询失败, 请检查输入!", "错误!",JOptionPane.ERROR_MESSAGE);
            }
            break;
        case 3: //全部课程的平均成绩
            String s0="create view C_G(Cno,A_Grade) "+

```

```

" as "+
" (SELECT Cno,avg(Grade) FROM SC group by Cno)"; //建立视图
st.executeUpdate(s00);
String s5="SELECT * FROM C_G where A_Grade >0";
rs=st.executeQuery(s5);
if(rs.next())
{
Ss=s5;
p=3;

new Output();
}
else
{
JOptionPane.showMessageDialog(null,"查询失败, 请检查输入!", "错误!",JOptionPane.ERROR_MESSAGE);
}

break;
case 4: //选课结果
String s7="select * from SC ";
//String s7="select * from SC where Grade is not null";
rs=st.executeQuery(s7);
if(rs.next())
{
Ss=s7;
p=2;

new Output();
}
else
{
JOptionPane.showMessageDialog(null,"查询失败, 请检查输入!", "错误!",JOptionPane.ERROR_MESSAGE);
}

break;
case 5: //课表信息查询
String s8="select * from Course";
rs=st.executeQuery(s8);
if(rs.next())
{
Ss=s8;
p=4;

new Output();
}
else
{
JOptionPane.showMessageDialog(null,"查询失败, 请检查输入!", "错误!",JOptionPane.ERROR_MESSAGE);
}

break;
case 6: //更改学生信息
Insert.x=1;
new Insert();
break;
case 7: //更改学生成绩
new Change_Grade();
break;
case 8: //更改密码
new Change_code();
break;
case 9: //插入课程
new Insert_C();
break;
case 10: //删除课程
new Delete();

```

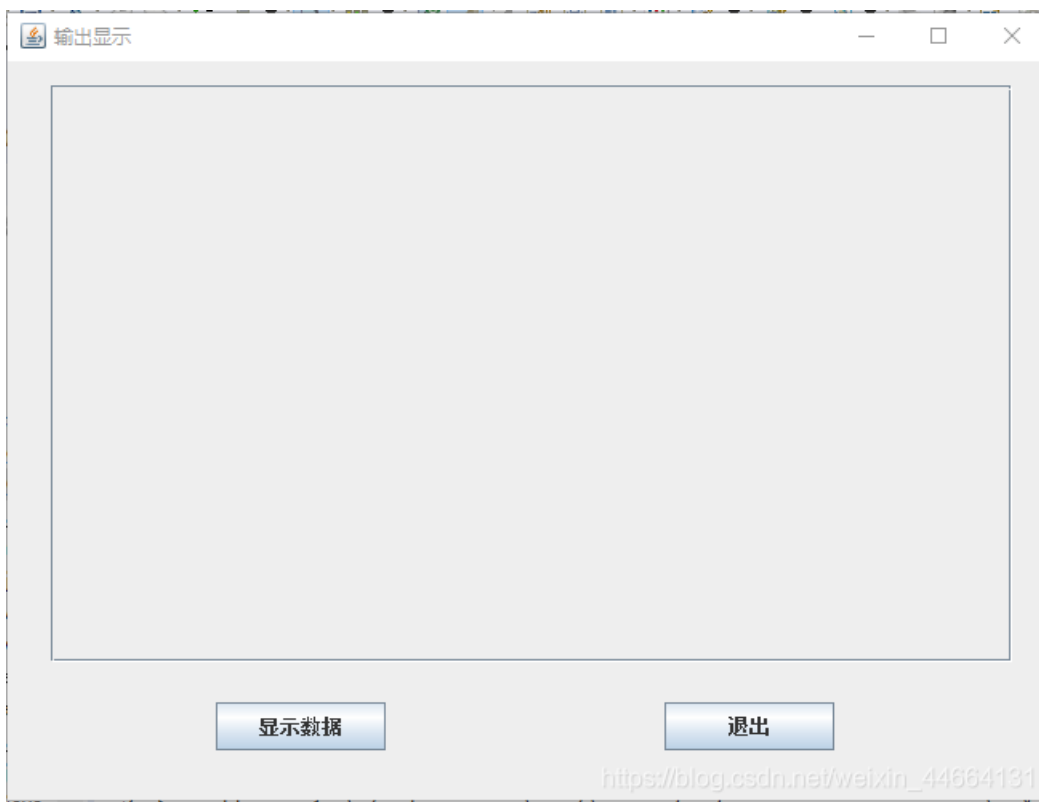
```

        new Delete();
        break;
    }
}
else
{
    JOptionPane.showMessageDialog(null, "数据库连接失败!");
    System.exit(0);
}
}

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

```

查询显示输出界面 *Output*



这个部分我借鉴了[怎么样将数据库的表在Java中界面中显示出来](#)这篇百度文库中的思路以及部分代码在Search类中进行查询或修改操作后需要进行的输出会在这里执行

```

import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

public class Output extends JFrame
{
    private static final long serialVersionUID = 1L;

    private JScrollPane scpDemo; // 滚动面板
    private JTable tabDemo; // 表格组件
    private JButton jbt, jbt1;
    int b=0; // 做一个判断区别，主要在查询平均成绩时用
}

```

```

ResultSet r;
Statement st;

public Output()
{
    super("输出显示");
    this.setSize(666, 500);
    this.setLayout(null);
    this.setLocation(340, 70);

    this .scpDemo=new JScrollPane();
    this .scpDemo.setBounds(27, 15, 600, 360);
    this .jbt=new JButton("显示数据");
    this .jbt.setBounds(130, 400, 106, 30);
    this .jbt.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ae)
        {
            jbt_ActionPerformed(ae);
        }
    });
    this .jbt1=new JButton("退出");
    this .jbt1.setBounds(410, 400, 106, 30);
    this .jbt1.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent a)
        {
            try
            {
                jbt1_ActionPerformed(a);
            }
            catch (SQLException e)
            {
                e.printStackTrace();
            }
        }
    });
    add(this .scpDemo);
    add(this .jbt);
    add(this .jbt1);

    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    this.setVisible(true);
    this.addWindowListener(new WinClose());
}

public void jbt_ActionPerformed(ActionEvent ae)
{
    st=Connect.doConnect();
    int a=Search.p; //p值不同, 输出面板表格不同
    try
    {
        if(st==null)
        {
            JOptionPane.showMessageDialog(null, "数据库连接失败!");
            System.exit(0);
        }
    }
}

```

```

r=st.executeQuery(Search.Ss);
int count=0;    //记录上一条语句查询得到的元组数量
while(r.next())
{
    count++;
}
r=st.executeQuery(Search.Ss);
/*这里划重点，必须要再执行一次，经过上面的 while 循环，r 已经到最后了，
 * 不重新执行的话，输出结果是空值，就是最后的表格数目信息是对的，但没有任何值*/
if(a==1) //学生信息的输出
{
    Object[][] array=new Object[count][5];
    count=0;
    while(r.next())
    {
        array[count][0]=r.getString("Sno");
        array[count][1]=r.getString("Sname");
        array[count][2]=Integer.valueOf(r.getInt("Sage"));
        array[count][3]=r.getString("Ssex");
        array[count][4]=r.getString("Sdept");
        count++;
    }
    String[] title={"学号 ", "姓名 ", "年龄 ", "性别 ", "系别"};
    this.tabDemo=new JTable(array,title);
}

else if(a==2) //学生成绩
{
    Object[][] array=new Object[count][3];
    count=0;
    while(r.next())
    {
        array[count][0]=r.getString("Sno");
        array[count][1]=r.getString("Cno");
        array[count][2]=Integer.valueOf(r.getInt("Grade"));
        count++;
    }
    String[] title={"学号 ", "课程号 ", "成绩"};
    this.tabDemo=new JTable(array,title);
}

else if(a==3) //平均成绩
{
    b=1;
    Object[][] array=new Object[count][2];
    count=0;
    while(r.next())
    {
        array[count][0]=r.getString("Cno");
        array[count][1]=Double.valueOf(r.getDouble("A_Grade"));
        count++;
    }
    String[] title={"课程号 ", "平均成绩"};
    this.tabDemo=new JTable(array,title);
}

else if(a==4) //课程表
{
    b=2;
    Object[][] array=new Object[count][5];

```

```

Object[][] array=new Object[count][5];
count=0;
while(r.next())
{
    array[count][0]=r.getString("Cno");
    array[count][1]=r.getString("Cname");
    array[count][2]=r.getString("Cpno");
    array[count][3]=Integer.valueOf(r.getInt("Ccredit"));
    array[count][4]=Integer.valueOf(r.getInt("Cnum"));
    count++;
}
String[] title={"课程号 ", "课程名 ", "先修课 ", " 学分", "课余量"};
this.tabDemo=new JTable(array,title);
}

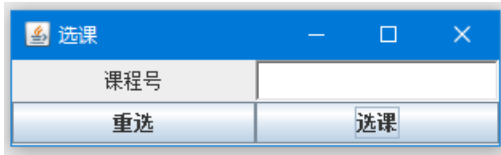
this .tabDemo.getTableHeader();
this .scpDemo.getViewPort().add(tabDemo);
}
catch (SQLException e1)
{
    JOptionPane.showMessageDialog(null, "数据源错误", "错误",JOptionPane.ERROR_MESSAGE);
}
}

public void jbt1_ActionPerformed(ActionEvent a) throws SQLException
{
    if(Login.a==1)
    {
        if(b==1)
        {
            // 删除视图
            String s00="drop view C_G";
            st.executeUpdate(s00);
        }
        new Student();
        dispose();
    }
    else if(Login.a==2)
    {
        if(b==1)
        {
            String s00="drop view C_G";
            st.executeUpdate(s00);
        }
        new Teacher();
        dispose();
    }
}

//public static void main(String[] args){ new Output();}
}

```

选课界面 [Select_C](#)



登录时系统会记住登录名，也就是登录学生的学号，再加上选择课程时输入的课程号就可以基本确定信息

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

public class Select_C extends JFrame implements ActionListener
{
    private TextField tx1;
    private JButton jbt1,jbt2;

    private static final long serialVersionUID = 1L;

    public Select_C()
    {
        super("选课");
        this.setBounds(650,540,320,90);
        this.setBackground(Color.lightGray);
        this.setLayout(new GridLayout(2,2));
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setVisible(true);
        this.addWindowListener(new WinClose());

        //第一行
        this.add(new Label("课程号",Label.CENTER));
        this.tx1=new TextField(15);
        this.add(this.tx1);
        //第二行
        this.jbt1=new JButton("选课");
        this.jbt2=new JButton("重选");
        this.add(this.jbt2);
        this.add(this.jbt1);
        this.jbt1.addActionListener(this);
        this.jbt2.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e)
    {
        String s=tx1.getText();
        if(s.equals(""))
        {
            JOptionPane.showMessageDialog(this,"输入不能为空，请检查输入!");
        }
        else
        {
            try
            {
                Statement st=Connect.doConnect();
                ResultSet rt;

                if(e.getSource()==jbt1) //选课
            {
```



```

String s0="select * from Course where Cno='"+s+"'";
//从Course表中查询是否有与输入中所对应的课序号
rt=st.executeQuery(s0);
if(rt.next())
{
String s00="select * from SC where Sno='"+Login.s1+"' and Cno='"+s+"'";
//检查该学生是否已选过该课程
rt=st.executeQuery(s00);
if(rt.next())
{
JOptionPane.showMessageDialog(this,"你已选过该课程,请勿重复选择!");
}
else
{
String s1="insert into SC values('"+Login.s1+"','"+s+"',\''0\') "; //在SC表中插入选课信息
int a=st.executeUpdate(s1);
String s2="Update Course set Cnum=Cnum-1 where Cno='"+s+"'"; //在Course表中将对应课序号的课程的课余量减

int b=st.executeUpdate(s2);
//executeUpdate的返回值是一个整数,指示受影响的行数(即更新计数)。
if(a!=0&&b!=0)
{
JOptionPane.showMessageDialog(this,"选课成功!");
dispose();
}
else
{
JOptionPane.showMessageDialog(this,"选课失败,请检查输入!");
}
}
else
{
JOptionPane.showMessageDialog(this,"该课程不存在,请重新选择!");
}
}
}
catch (SQLException e1)
{
e1.printStackTrace();
}
if(e.getSource()==jbt2) //重选
{
tx1.setText(null);
}
}
}

//public static void main(String[] args) { new Select_C(); }
}

```

数据库连接类 **Connect**

整个实验过程中几乎每个类都需要连接数据库,所以我就单独写了一个类。这个连接代码大部分内容是我百度的□之后根据我的需要改的。

```

import java.sql.*;

public class Connect
{
    public static Statement doConnect()
    {
        String JDriver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";// SQL 数据库引擎
        String connectDB = "jdbc:sqlserver://127.0.0.1:1433;DatabaseName=你想进行操作的数据库名称";// 数据源
        try
        {
            Class.forName(JDriver);// 加载数据库引擎, 返回给定字符串名的类
        }
        catch (ClassNotFoundException e)
        {
            // e.printStackTrace();
            System.out.println("加载数据库引擎失败");
            //JOptionPane.showMessageDialog(this,"学号栏不能为空!");
            System.exit(0);
        }
        System.out.println("数据库驱动成功");
        try
        {
            String user = "你自己的登录名";
            String password = "你自己的登录密码";
            Connection con = DriverManager.getConnection(connectDB, user, password);// 连接数据库对象
            System.out.println("连接数据库成功");
            Statement stmt = con.createStatement();
            return stmt;
        }
        catch (SQLException e)
        {
            e.printStackTrace();
            System.out.println("数据库连接错误");
            System.exit(0);
        }
        return null;
    }
    //public static void main(String[] args) { }
}

```

插入学生信息+修改学生信息 *Insert*



| 学号 | 姓名 | 年龄 | 性别 | 专业 |
|----|----|----|----|----|
| 0 | | 0 | | |

重新输入/再次添加 确定 退出

插入学生信息界面和修改学生信息界面的布局基本相同，要进行的操作也相近，所以我就写在了一个类里面，就是加了一个判断，在生成这个界面时作一个判断，看要进行的操作是插入还是修改。

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;

```

```

public class Insert extends JFrame implements ActionListener
{
    private TextField jtx1,jtx2,jtx3,jtx4,jtx5,jtx6,jtx7;
    private JButton jb1,jb2,jb3;

    String s6,s7;
    static int x=0;    // 做判断, 看是输入学生信息, 还是更改学生信息

    private static final long serialVersionUID = 1L;
    public Insert()
    {
        super("请输入信息");
        this.setBounds(380,240,750,120);
        this.setBackground(Color.lightGray);
        this.setLayout(new GridLayout(3,5));
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setVisible(true);
        this.addWindowListener(new WinClose());

        // 第一行
        this.add(new Label("学号",Label.CENTER));
        this.add(new Label("姓名",Label.CENTER));
        this.add(new Label("年龄",Label.CENTER));
        this.add(new Label("性别",Label.CENTER));
        this.add(new Label("专业",Label.CENTER));
        // 第二行
        this.jtx1=new TextField(15);
        this.add(this.jtx1);
        jtx1.setText("0");
        this.jtx2=new TextField(15);
        this.add(this.jtx2);
        this.jtx3=new TextField(15);
        this.add(this.jtx3);
        jtx3.setText("0");
        this.jtx4=new TextField(15);
        this.add(this.jtx4);
        this.jtx5=new TextField(15);
        this.add(this.jtx5);
        // 第三行
        this.jb2=new JButton("重新输入/再次添加");
        this.add(this.jb2);
        this.jb2.addActionListener(this );
        this.jtx6=new TextField(15);
        this.add(this.jtx6);
        jtx6.setEnabled(false);
        this.jb1=new JButton("确定");
        this.add(this.jb1);
        this.jb1.addActionListener(this );
        this.jtx7=new TextField(15);
        this.add(this.jtx7);
        jtx7.setEnabled(false);
        this.jb3=new JButton("退出");
        this.add(this.jb3);
        this.jb3.addActionListener(this );
    }

    public void actionPerformed(ActionEvent e)
    {

```

```

String s1=jtx1.getText();
String s2=jtx2.getText();
int s3=Integer.parseInt(jtx3.getText());
String s4=jtx4.getText();
String s5=jtx5.getText();

if(jtx1.getText().equals(""))
{
    JOptionPane.showMessageDialog(this,"学号栏不能为空!");
}

else
{
    if(e.getSource()==jb1)
    {
        try
        {
            Statement st=Connect.doConnect();
            if(st!=null)
            {
                if(x==1) // 修改学生信息
                {
                    String a="select * from Student where Sno='"+s1+"'";
                    ResultSet rs=st.executeQuery(a);
                    if(rs.next())
                    {
                        String Sname=rs.getString("Sname");
                        int Sage=Integer.valueOf(rs.getInt("Sage"));
                        String Ssex=rs.getString("Ssex");
                        String Sdept=rs.getString("Sdept");
                        int i=Check(s3, s4);
                        if(i==0)
                        {
                            JOptionPane.showMessageDialog(this,"学生年龄应在15至25岁，学生性别只能为‘男’或‘女’，请输入适当的学生信息!");
                        }
                    }
                }
                else
                {
                    int a1=s2.compareTo(Sname);
                    if(a1!=0) // 新输入的Sname不同，更新
                    {
                        String b1="update Student set Sname='"+s2+"' where Sno='"+s1+"'";
                        st.executeUpdate(b1);
                    }
                    if(Sage!=s3) // 新输入的Sage不同，更新
                    {
                        String b2="update Student set Sage='"+s3+"' where Sno='"+s1+"'";
                        st.executeUpdate(b2);
                    }
                    int a3=s4.compareTo(Ssex);
                    if(a3!=0) // 新输入的Ssex不同，更新
                    {
                        String b3="update Student set Ssex='"+s4+"' where Sno='"+s1+"'";
                        st.executeUpdate(b3);
                    }
                    int a4=s5.compareTo(Sdept);
                    if(a4!=0) // 新输入的Sdept不同，更新
                    {
                        String b4="update Student set Sdept='"+s5+"' where Sno='"+s1+"'";
                        st.executeUpdate(b4);
                    }
                }
            }
        }
    }
}

```

```

        st.executeUpdate(b4);
    }
    String s7="恭喜你, 更改成功! ";
    jtx7.setText(s7);
}
    }
    else
    {
        JOptionPane.showMessageDialog(this,"学号输入错误, 无法更改, 请检查输入! ");
    }
}
    }
    else //增加学生信息
    {
        String a="select * from Student where Sno=\'"+s1+"\'";
        ResultSet rs=st.executeQuery(a);
        if(rs.next())
        {
            JOptionPane.showMessageDialog(this,"该学号已存在, 请勿重复添加! ");
        }
        else
        {
            int i=Check(s3, s4);
            if(i==0)
            {
                JOptionPane.showMessageDialog(this,"学生年龄应在15至25岁, 学生性别只能为‘男’或‘女’, 请输入适当的学生信息!! ");
            }
            else
            {
                String a1 = "insert into Student (Sno,Sname,Sage,Ssex,Sdept,Score)
                    + " values(\'" + s1 + "\',\'" + s2 + "\',\'" + s3 + "\',\'" + s4 + "\',\'" + s5 + "\',\'00000\');"
                ;

                //插入学生信息, 初始学生密码为00000
                st.executeUpdate(a1);
                //System.out.println("插入数据成功");
                String s7="恭喜你, 插入成功! ";
                jtx7.setText(s7);
            }
        }
    }
}
    }
    else
    {
        JOptionPane.showMessageDialog(this,"数据库连接失败! ");
        System.exit(0);
    }
}
}

catch (SQLException e1)
{
    e1.printStackTrace();
    //System.out.println("数据库连接错误");
    String s6="很遗憾, 插入失败! ";
    jtx6.setText(s6);
    System.exit(0);
}
}
}

if(e.getSource()==jb2)
{

```

```

jtx1.setText("0");
jtx2.setText(null);
jtx3.setText("0");
jtx4.setText(null);
jtx5.setText(null);
jtx6.setText(null);
jtx7.setText(null);
}

else if(e.getSource()==jb3)
{
    new Teacher();
    dispose();
}
}

public int Check(int Sage,String Ssex) //检查学生年龄和性别输入是否合法
{
    int a=Sage;
    String aS=Ssex;
    if((a>=15&&a<=25)&&(aS.compareTo("男")==0||aS.compareTo("女")==0))
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

// public static void main(String[] args) { new Insert(); }
}

```

插入课程信息 *Insert_C*



| 课序号 | 课程名 | 先修课 | 学分 |
|-----------|-----|-----|----|
| 0 | | 2 | 0 |
| 重新输入/再次添加 | | 确定 | 退出 |

也是插入相关的操作，本来我想和上面的 *Insert* 类写成一个类来的，但是学生和课程需要的文本框数不一样，写在一起就真的是个大工程了，所以就又写了一个。

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;

public class Insert_C extends JFrame implements ActionListener
{
    private TextField jtx1,jtx2,jtx3,jtx4,jtx5;
    private JButton jb1,jb2,jb3;

    String s6,s7;
}

```

```

private static final long serialVersionUID = 1L;

public Insert_C()
{
    super("请输入信息");
    this.setBounds(400,240,650,120);
    this.setBackground(Color.lightGray);
    this.setLayout(new GridLayout(3,4));
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    this.setVisible(true);
    this.addWindowListener(new WinClose());

    this.add(new Label("课序号",Label.CENTER));
    this.add(new Label("课程名",Label.CENTER));
    this.add(new Label("先修课",Label.CENTER));
    this.add(new Label("学分",Label.CENTER));
    //第二行
    this.jtx1=new TextField(15);
    this.add(this.jtx1);
    jtx1.setText("0");
    this.jtx2=new TextField(15);
    this.add(this.jtx2);
    this.jtx3=new TextField(15);
    this.add(this.jtx3);
    jtx3.setText("2");
    this.jtx4=new TextField(15);
    this.add(this.jtx4);
    jtx4.setText("0");
    //第三行
    this.jtx5=new TextField(15);
    this.add(this.jtx5);
    jtx5.setEnabled(false);
    this.jb2=new JButton("重新输入/再次添加");
    this.add(this.jb2);
    this.jb2.addActionListener(this );

    this.jb1=new JButton("确定");
    this.add(this.jb1);
    this.jb1.addActionListener(this );
    this.jb3=new JButton("退出");
    this.add(this.jb3);
    this.jb3.addActionListener(this );
}

public void actionPerformed(ActionEvent e)
{
    String s1=jtx1.getText();
    String s2=jtx2.getText();
    int s4=Integer.parseInt(jtx4.getText());
    String s3=jtx3.getText();

    if(jtx1.getText().equals(""))
    {
        JOptionPane.showMessageDialog(this,"课序号不能为空!");
    }

    else
    {
        if(e.getSource()==jb1) // 确定按钮

```

```

{
    try
    {
        Statement st=Connect.doConnect();
        if(st!=null)
        {
            String a0="select * from Course where Cno='"+s1+"'";
            ResultSet rs=st.executeQuery(a0);
            if(rs.next())
            {
                JOptionPane.showMessageDialog(this,"该课序号已存在, 请勿重复添加! ");
            }
            else
            {
                String a = "insert into Course (Cno,Cname,Cpno,Ccredit,Cnum)"
                    + " values('\'' + s1 + '\',\'' + s2 + '\',\'' + s3 + '\',\'' + s4 + '\',\''20\''";
                //插入课程信息, 初始课余量默认20
                st.executeUpdate(a);
                //System.out.println("插入数据成功");
                String s7="恭喜你, 插入成功! ";
                jtx5.setText(s7);
            }
        }
        else
        {
            JOptionPane.showMessageDialog(this,"数据库连接失败! ");
            System.exit(0);
        }
    }
    catch (SQLException e1)
    {
        e1.printStackTrace();
        //System.out.println("数据库连接错误");
        String s6="很遗憾, 插入失败! ";
        jtx5.setText(s6);
        System.exit(0);
    }
}

if(e.getSource()==jb2) //重新输入
{
    jtx1.setText("0");
    jtx2.setText(null);
    jtx3.setText(null);
    jtx4.setText("0");
    jtx5.setText(null);
//    jtx6.setText(null);
}

else if(e.getSource()==jb3) //退出
{
    new Teacher();
    dispose();
}
}

//public static void main(String[] args) { new Insert_C(); }
}

```


删除信息 Delete



用来删除学生信息和课程信息，由于在数据库中我将Student表中的Sno，Course表中的Cno设成了主码，所以删除时只要确定主码就能确定要删除的元组，一个接受输入信息的文本框就足够了。

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;

public class Delete extends JFrame implements ActionListener
{
    private TextField tex1,tex2;
    private JButton jb1,jb2,jb3;

    String s6,s7;

    private static final long serialVersionUID = 1L;

    public Delete()
    {
        super("删除界面");
        this.setBounds(500,250,440,100);
        this.setBackground(Color.lightGray);
        this.setLayout(new GridLayout(2,3));
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setVisible(true);
        this.addWindowListener(new WinClose());

        //第一行
        this.add(new Label("请输入要删除元组的编号",Label.LEFT));
        this.tex1=new TextField(15);
        this.add(this.tex1);
        this.tex2=new TextField(15);
        this.add(this.tex2);
        tex2.setEnabled(false);
        //第二行
        this.jb1=new JButton("重新输入/再次删除");
        this.add(this.jb1);
        this.jb1.addActionListener(this );
        this.jb2=new JButton("删除");
        this.add(this.jb2);
        this.jb2.addActionListener(this );
        this.jb3=new JButton("退出");
        this.add(this.jb3);
        this.jb3.addActionListener(this );
    }

    public void actionPerformed(ActionEvent e)
    {
        String s1=tex1.getText();
```

```

if(tex1.getText().equals(""))
{
    JOptionPane.showMessageDialog(this, "输入栏不能为空!");
}
else
{
    if(e.getSource()==jb2)
    {
        int p=Teacher.a;
        if(p==1) //删除学生信息
        {
            try
            {
                Statement st=Connect.doConnect();
                if(st!=null)
                {
                    String a0="select * from Student where Sno=\'"+s1+"\'";
                    //在数据库中查询是否有与输入相对应的元组
                    ResultSet rs=st.executeQuery(a0);
                    if(rs.next())
                    {
                        String a = "delete from Student where Sno=\'"+s1+"\'";
                        st.executeUpdate(a);
                        //System.out.println("插入数据成功");
                        String s7="恭喜你，删除成功! ";
                        tex2.setText(s7);
                    }
                    else
                    {
                        JOptionPane.showMessageDialog(this, "该学号不存在，删除失败! ");
                    }
                }
            }
            else
            {
                JOptionPane.showMessageDialog(this, "数据库连接失败! ");
                System.exit(0);
            }
        }
        st.close();//关闭命令对象连接
    }
    catch (SQLException e1)
    {
        e1.printStackTrace();
        //System.out.println("数据库连接错误");
        String s6="很遗憾，删除失败! ";
        tex2.setText(s6);
        System.exit(0);
    }
}

else if(p==2) //删除课程信息
{
    try
    {
        Statement st=Connect.doConnect();
        if(st!=null)
        {
            String a0="select * from Course where Cno=\'"+s1+"\'";
            ResultSet rs=st.executeQuery(a0);

```

```

        ResultSet rs=st.executeQuery(a0);
        if(rs.next())
        {
            String a = "delete from Course where Cno='"+s1+"'";
            st.executeUpdate(a);
            //System.out.println("插入数据成功");
            String s7="恭喜你，删除成功！";
            tex2.setText(s7);
        }
        else
        {
            JOptionPane.showMessageDialog(this,"该课序号不存在，删除失败！");
        }

    }
    else
    {
        JOptionPane.showMessageDialog(this,"数据库连接失败！");
        System.exit(0);
    }

    st.close();// 关闭命令对象连接
}

catch (SQLException e1)
{
    e1.printStackTrace();
    //System.out.println("数据库连接错误");
    String s6="很遗憾，删除失败！";
    tex2.setText(s6);
    System.exit(0);
}
}
}

if(e.getSource()==jb1)
{
    //JOptionPane.showMessageDialog(this,"eeeeee！");
    tex1.setText(null);
    tex2.setText(null);
}
if(e.getSource()==jb3)
{
    new Teacher();
    dispose();
}
}

// public static void main(String[] args) { new Delete(); }
}

```

修改学生成绩 [Change_Grade](#)



在SC表中进行操作，(Sno, Cno)是主码，根据这两项就能唯一确定元组。

先判断：1.是否有该学号的学生；2.是否有该课程号；3.该学生是否选修了该课程

然后再修改。

这里的修改我好像弄麻烦了，我是先删除相应的元组信息，然后重新插入，额，直接更新不就行了吗？我忘了我当时咋想的了。

。。

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

public class Change_Grade extends JFrame implements ActionListener
{
    private TextField tx1,tx2,tx3,tx4;
    private JButton jb1,jb2;

    String s6;

    private static final long serialVersionUID = 1L;

    public Change_Grade()
    {
        super("输入界面");
        this.setBounds(600,240,500,120);
        this.setBackground(Color.lightGray);
        this.setLayout(new GridLayout(3,3));
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setVisible(true);
        this.addWindowListener(new WinClose());

        //第一行
        this.add(new Label("学号",Label.CENTER));
        this.add(new Label("课程号",Label.CENTER));
        this.add(new Label("成绩",Label.CENTER));
        //第二行
        this.tx1=new TextField(15);
        this.add(this.tx1);
        this.tx2=new TextField(15);
        this.add(this.tx2);
        this.tx3=new TextField(15);
        this.add(this.tx3);
        //第三行
        this.tx4=new TextField(15);
        this.add(this.tx4);
        tx4.setEnabled(false);
        this.jb1=new JButton("重新输入");
        this.add(this.jb1);
        this.jb1.addActionListener(this );
        this.jb2=new JButton("确定");
        this.add(this.jb2);
    }
}
```



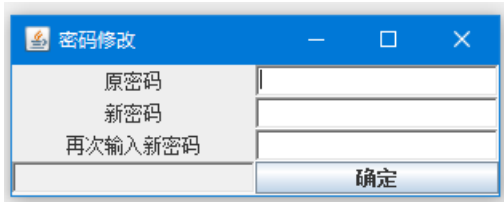
```

        JOptionPane.showMessageDialog(this, "学生成绩应在0至100分,请输入正确的学生信息!");
    }
    else
    {
        String a00="步骤一错误, 修改失败";
        tx4.setText(a00);
    }
}
else
{
    JOptionPane.showMessageDialog(this, "该学生没有选修该课程, 修改失败!");
}
}
else
{
    JOptionPane.showMessageDialog(this, "该课程不存在, 修改失败!");
}
}
else
{
    JOptionPane.showMessageDialog(this, "该学生不存在, 修改失败!");
}
}
else
{
    JOptionPane.showMessageDialog(this, "数据库连接失败!");
}
}
}
catch (SQLException e2)
{
    e2.printStackTrace();
}
}
}
    if(e.getSource()==jb1)
    {
        tx1.setText(null);
        tx2.setText(null);
        tx3.setText(null);
    }
}

public int Chack(String Grade) //检测成绩输入是否在合法范围内
{
    int i=Integer.valueOf(Grade);
    if(i>=0&&i<=100)
        return 1;
    else
        return 0;
}

// public static void main(String[] args) { new Change_Grade(); }
}

```



这里有多重判断，具体描述详见我之前的博客，这里就不多说了。

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

public class Change_code extends JFrame implements ActionListener
{
    Statement st;

    private TextField tx1,tx2,tx3,tx4;
    private JButton jbt1;

    private static final long serialVersionUID = 1L;

    public Change_code()
    {
        super("密码修改");
        this.setBounds(610,250,320,120);
        this.setBackground(Color.lightGray);
        this.setLayout(new GridLayout(4,2));
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setVisible(true);
        this.addWindowListener(new WinClose());

        //第一行
        this.add(new Label("原密码",Label.CENTER));
        this.tx1=new TextField(15);
        this.add(this.tx1);
        //第二行
        this.add(new Label("新密码",Label.CENTER));
        this.tx2=new TextField(15);
        this.add(this.tx2);
        //第三行
        this.add(new Label("再次输入新密码",Label.CENTER));
        this.tx3=new TextField(15);
        this.add(this.tx3);
        //第四行
        this.tx4=new TextField(15);
        this.add(this.tx4);
        tx4.setEnabled(false);
        this.jbt1 =new JButton("确定") ;
        this.add(this.jbt1);
        this.jbt1.addActionListener(this );
    }

    public void actionPerformed(ActionEvent e)
    {
        String s1=tx1.getText();
        String s2=tx2.getText();
        String s3=tx3.getText();
    }
}
```

```

String s3=tx3.getText();

if(s1.length()<=0||s2.length()<=0||s3.length()<=0)
{
    JOptionPane.showMessageDialog(this,"输入不能有空值, 请检查输入!");
}
else
{
    if(e.getSource()==jbt1)
    {
        try
        {
            st=Connect.doConnect();
            int p=Login.a;
            String s = null;
            int x=s1.compareTo(Login.s2);    // 检验原密码与登录密码是否相同
            if(x==0)
            {
                int y=s1.compareTo(s2);    // 检验新密码与原密码是否相同
                if(y==0)
                {
                    JOptionPane.showMessageDialog(this,"原密码与新密码不能相同, 请重新输入!");
                    tx2.setText(null);
                }
            }
            else
            {
                int z=s2.compareTo(s3);    // 检验两次输入的新密码是否相同
                if(z!=0)
                {
                    JOptionPane.showMessageDialog(this,"两次输入的密码不一致, 请重新输入!");
                    tx3.setText(null);
                }
            }
            else
            {
                if(p==1)    // 修改学生密码
                {
                    s="update Student set Scode='"+s2+"' where Sno='"+Login.s1+"'";
                }
                else if(p==2) // 修改教师密码
                {
                    s="update Teacher set Tcode='"+s2+"' where Tno='"+Login.s1+"'";
                }
                int q=st.executeUpdate(s);
                if(q!=0)
                {
                    JOptionPane.showMessageDialog(this,"更改成功, 请重新登录!");
                    new Login();
                    dispose();
                }
            }
            else
            {
                JOptionPane.showMessageDialog(this,"更改失败, 请重试!");
            }
        }
        }
    }
}
else
{
    JOptionPane.showMessageDialog(this,"原密码输入错误, 请检查输入!");
}
}

```



```
}
catch (SQLException e2)
{
    e2.printStackTrace();
}
}
}
}

//public static void main(String[] args) {new Change_code(); }
}
```

还有最后一个，

WinClose

这个是使用界面关闭相应操作时系统需要生成的

```
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class WinClose implements WindowListener {

    @Override
    public void windowOpened(WindowEvent e) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowClosing(WindowEvent e) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowClosed(WindowEvent e) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowIconified(WindowEvent e) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowDeiconified(WindowEvent e) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowActivated(WindowEvent e) {
        // TODO Auto-generated method stub
    }

    @Override
    public void windowDeactivated(WindowEvent e) {
        // TODO Auto-generated method stub
    }

}
```

OK

以上就是我此次数据库大作业1.x所有的代码了。

终于完成大作业了，开心。



敲开心的

代码有很多不太完善的地方，我也是知道的，比如有的类里面多重的 if-else 语句。还有一些重复代码。也可能有很多我没想到的bug。所以希望各位读者能不吝赐教，有什么问题或者新的想法一起交流。

关于这次实验的实验报告我已经正在写了，再录一段程序讲解视频，这次大作业就算是圆满完成了。

等实验报告和讲解视频完成我也会将链接放在这里。

所以，诸位，再会喽~~

程序的源文件 链接：[百度网盘](#) 提取码：a2z3

程序视频讲解 链接：[数据库作业：学生信息管理系统](#)

实验报告 链接：[百度网盘](#) 提取码：avlb

数据库文件 链接：[百度网盘](#) 提取码：1w9a

注意：更改学生成绩的**Change_Grade**类有更新，上面的代码我更新了，源文件没有，如果有使用的话请注意一下。

哦，对了，差点忘说，我的作业代码和其中的操作是根据我的数据库设置而来的，所以如果你的设置和我不同的话，呃——好吧，你们都懂的。

然后就是我的数据库设置我在视频讲解的时候会讲，还有就是看完视频的时候，感觉还好的话，可以给我点个赞。



眼神暗示