实训项目一　学生管理信息系统开发实例

本项目是基于 Java 及数据库系统的“高校学生综合管理信息系统”实例，通过本系统的开发使学生全面掌握 Java 数据库应用程序开发的方法和技能。

本系统使用 JDBC-ODBC 桥接器，实现对 Microsoft Access 2000 数据库的操作和管理。

本系统主要包括系统管理、操作员管理、学生管理、课程管理和成绩管理、数据操作模板等 6 个模块。

一、系统模块设计

系统总体结构如图 2-1 所示。

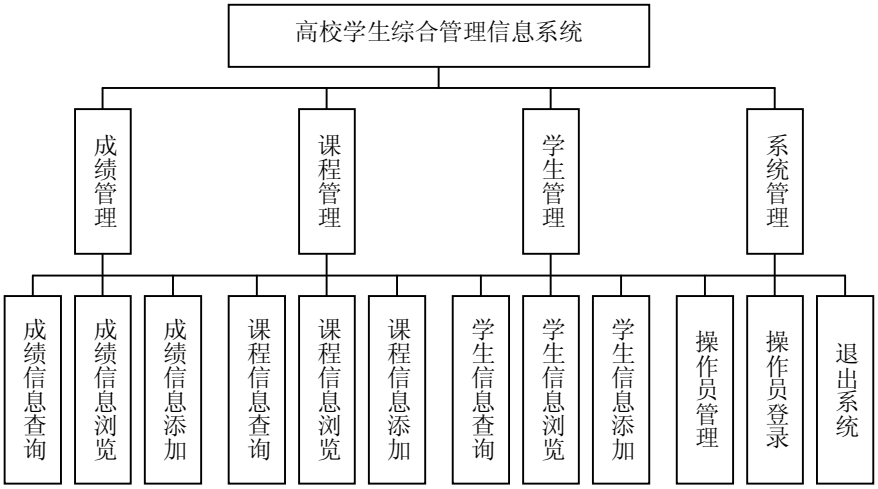

图 2-1　系统总体结构

二、数据库结构设计

该应用程序使用 Microsoft Access 数据库管理系统，创建数据库名为 student.mdb，该数据库包括“学生”表、“成绩”表、“课程”表和“操作员”表。数据源名为 student。

1. “学生”表

“学生”表结构如表 2-1 所示。

表 2-1　“学生”表结构

字段名	类型	宽度	主键	非空
学号	文本	10	是	否
姓名	文本	8	否	否
性别	文本	2	否	否
出生年月	日期/时间		否	否
简历	备注		否	否
奖学金	数字		否	是

2. “课程”表

“学生”表结构如表 2-2 所示。

表 2-2 “课程”表结构

字段名	类型	宽度	主键	非空
课程号	文本	4	是	否
课程名	文本	20	否	否
学时数	数字	3	否	否
学分数	数字	2	否	否
开课学期	文本	2	否	否
考查标志	文本	1	否	否

3. “成绩”表结构

“成绩”表结构如表 2-3 所示。

表 2-3 “成绩”表结构

字段名	类型	宽度	主键	非空
学号	文本	10	是	否
课程号	文本	4	是	否
成绩	数字	3	否	否

4. “操作员”表

“操作员”表结构如表 2-4 所示。

表 2-4 “操作员”表结构

字段名	类型	宽度	主键	非空
操作员	文本	10	是	否
密码	文本	8	否	否
权限	文本	3	否	否

三、详细设计

首先建立一个名为 CollegeMS.jpx 的工程文件，然后在工程文件中建立一个名为 mainApplication.java的应用程序，然后开始以下设计。

1. 主窗口

主窗口是应用程序的主界面，由它负责调用其他模块。根据系统模块设计的总体结构，主窗口包括系统管理、学生管理、课程管理和成绩管理等 4 个模块，如图 2-2 所示。在主窗口中通过单击相应的菜单项调用相应的功能模块。源程序为 CollegeMS\src\collegems\mainFrame.java。

建立应用程序 mainApplication.java时，并将主窗口命名为 mainFrame.java。建立应用程序后在主窗口上创建相应的菜单和菜单项，这些设计在菜单的制作中已讲述，菜单的结构可参考

图 2-1 所示的内容。

图 2-2　学生综合管理信息系统

2. 登录界面

界面设计：为了保证系统的安全，在登录界面中输入正确的操作员名称和操作员密码，才能进入主界面，否则退出应用系统。登录界面如图 2-3 所示。源程序为 CollegeMS\src\collegems\OpLogin.java。

图 2-3　登录界面

3. 系统管理模块设计

系统管理模块中包含 3 个子模块：操作员登录、操作员管理和退出系统。操作员管理子模块中又包含 3 个子模块：添加操作员、删除操作员和修改操作员。

（1）操作员登录模块。操作员登录界面如图 2-3 所示。本模块的处理过程：首先校验操作员名称是否为空，若为空，出现相应的提示信息；若不为空生成 SQL 操作语句，查询要登录的操作员名是否存在，若存在，执行下一步操作，若不存在则提示并返回；再根据操作员的名称来确定操作权限，执行主窗口中的 setEnable( )方法赋予操作员相应的操作权限。

（2）添加操作员模块。添加操作员模块的功能是添加新操作员信息，输入的数据包括操作员名称、密码和权限。添加操作员界面如图 2-4 所示。源程序为 CollegeMS\src\collegems\Opadd.java。

图 2-4　添加操作员界面

模块的处理过程：首先校验操作员名称和密码是否为空，若为空，则出现相应的提示信息；校验两次输入的密码是否一致；生成 SQL 操作语句，查询要添加的操作员是否已经存在，若存在，则提示并返回；然后执行插入操作。

（3）删除操作员模块。删除操作员模块的功能是完成操作员的删除操作，输入的数据包括操作员名称和密码。删除操作员界面如图 2-5 所示。源程序为 CollegeMS\src\collegems\ OpDele.java。

图 2-5　删除操作员界面

模块的处理过程：首先校验操作员名称和密码是否为空，若为空，则出现相应的提示信息；生成 SQL 操作语句，查询要删除的用户名是否存在，密码是否正确，若存在，执行删除，若不存在则提示并返回。

（4）修改操作员模块。修改操作员模块的功能是实现对操作员的名称和密码进行修改，输入的数据包括操作员名称、操作员原密码、操作员新密码和确认密码。修改操作员界面如图 2-6 所示。源程序为 CollegeMS\src\collegems\ OpUpdate.java

模块的处理过程：首先校验操作员名称和密码是否为空，若为空出现相应的提示信息；检验新密码和确认密码是否一致；生成 SQL 操作语句，查询要修改的操作名是否存在，密码是否正确，若存在进行修改，若不存在提示并返回。

图 2-6　修改操作员界面

4. 学生管理模块

学生管理模块中包含 3 个子模块：学生信息添加、学生信息浏览和学生信息查询。

（1）学生信息添加模块。学生信息添加模块的功能是添加新入学的学生信息，输入的数据包括学号、姓名、性别、出生年月、奖学金和简历。学生信息添加界面如图 2-7 所示。源程序为 CollegeMS\src\collegems\StudentAdd.java

图 2-7　学生信息添加界面

模块的处理过程：首先校验学号、姓名、性别、出生年月（格式为 yyyy-mm-dd）、奖学金是否为空，则若为空，则出现相应的提示信息；生成 SQL 操作语句，执行插入操作。

（2）学生信息浏览模块。学生信息浏览模块的功能是完成对学生信息的查询、修改和删除操作，还可以通过“第一条”、“下一条”、“前一条”和“最后一条”按钮实现查询学生的所有数据。学生信息浏览界面如图 2-8 所示。源程序为 CollegeMS\src\collegems\Student-Browse.java。

模块的处理过程：首先校验班号是否为空，若为空出现相应的提示信息；根据指定查询条件进行查询操作；生成 SQL 操作语句，执行数据库的查询操作；对查询的结果集，可以通过“第一条”、“下一条”、“前一条”和“最后一条”按钮实现查询学生的所有数据，还可以对这些数据进行修改和删除操作。

图 2-8　学生信息浏览界面

（3）学生信息查询模块。学生信息查询模块的功能是完成对学生信息的查询操作，在该模块使用了 JTable 组件，一次可显示多条满足条件的记录。学生信息查询界面如图 2-9 所示。源程序为 CollegeMS\src\ collegems\StudentQuery.java。

图 2-9　学生信息查询

模块的处理过程：首先校验班号是否为空，若为空，则出现相应的提示信息；获得班号的方法是使用函数 left，取学号的前 3 个字符（在设计时把班号作为学号的前 3 位），按逐个指定的条件进行查询操作；生成 SQL 操作语句，执行数据库的查询操作。

5. *数据操作模板模块*

数据操作模板模块是数据库操作的核心和基础。由于整个应用程序要多次对数据库进行操作，因此把所需的数据库操作封装到一个类中，这样只要每次实例化这个类，然后调用这个类的相应方法即可，不用每次都执行重新创建连接对象等操作，方便了应用程序的编写，提高了开发效率。

源程序为 CollegeMS\src\collegems\ DBModeler.java。

课程管理和成绩管理两个模块与学生基本信息管理模块的基本原理、设计思想、设计过程和方法基本相同，这里不再赘述。

四、程序设计

为使学生少走弯路，我们将系统程序设计阶段的主要任务进行了分解。这里的分解就是把较大的任务拆分成一些较小的任务，期望能对学生有些帮助。主要工作如下：

1. 数据库的连接

（1）使用 Microsoft Access 2000 数据库管理系统设计数据库，数据库名为 student.mdb，数据库保存在 D:\javajc\javaData 目录下。

（2）为 student.mdb 数据库在 ODBC 管理器中配置数据源。

步骤如下：

1）打开 Windows 中的“控制面板”窗口。

2）双击“管理工具”图标，出现“管理工具”窗口。在该窗口中双击“数据源（ODBC）”图标，出现“ODBC 数据源管理器”对话框，如图 2-10 所示。

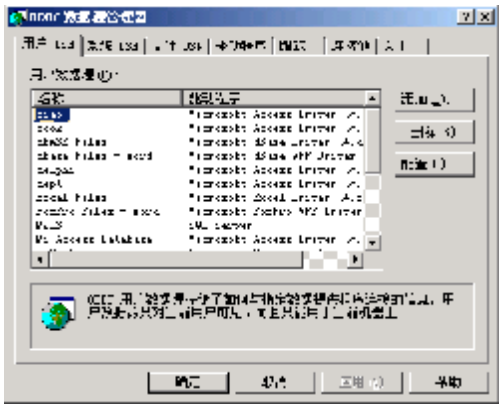

图 2-10　“ODBC 数据源管理器”对话框

3）在“ODBC 数据源管理器”对话框中，选择“用户 DSN”选项卡，单击“添加”按钮，出现“创建新数据源”对话框，如图 2-11 所示。

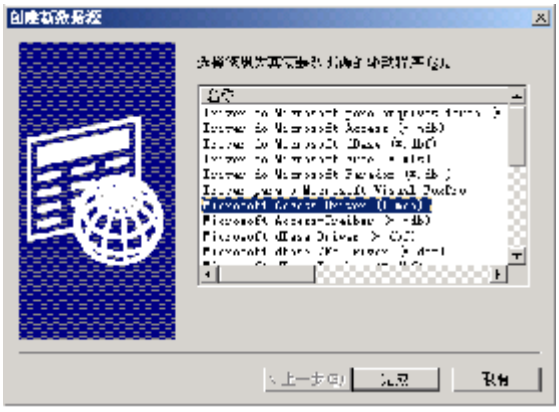

图 2-11　“创建新数据源”对话框

4）选择 Microsoft Access Driver (*.mdb)，单击“完成”按钮，出现安装 Access 数据库界

面，如图 2-12 所示。

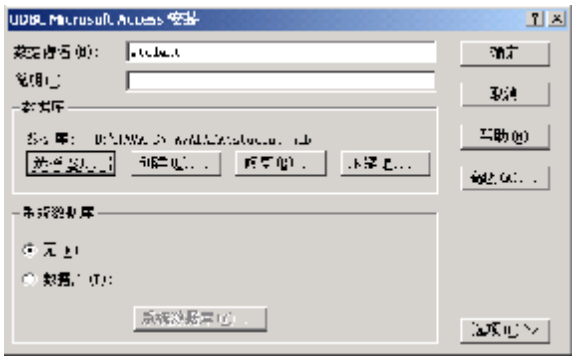

图 2-12 “ODBC Microsoft Access 安装”对话框

5）在“ODBC Microsoft Access 安装”对话框中，输入“数据源名”为 student，单击“数据库”栏内的“选择”按钮，如图 2-13 所示，出现“选择数据库”对话框。选择 Access 数据库（student.mdb）后，如图 2-12 所示，单击“确定”按钮，回到“ODBC Microsoft Access 安装”对话框。

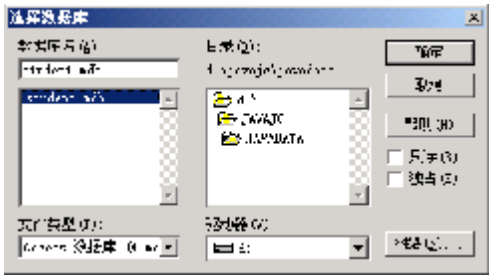

图 2-13 “选择数据库”对话框

如果要为数据源 student 设置登录用户名和密码，在“ODBC Microsoft Access 安装”对话框中单击“高级”按钮，出现“设置高级选项”对话框，如图 2-14 所示。否则单击“确定”按钮就完成了数据源的设置。

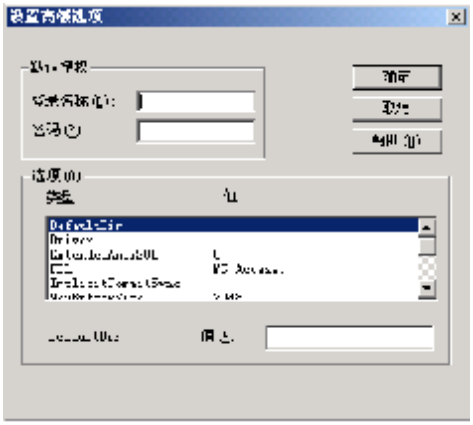

图 2-14 “设置高级选项”对话框

也可以在“登录名称”文本框里输入一个用户名，在“密码”文本框里输入口令。然后单击“确定”按钮，回到“ODBC Microsoft Access 安装”对话框，再单击“确定”按钮就完成了配置数据源的全部过程。最后，关闭“控制面板”窗口。

（3）利用 JDBC-ODBC 驱动程序访问数据源 student 所对应的 Access 数据库 student.mdb，在“成绩”表中插入数据，如表 2-5 所示。

表 2-5　“成绩”表内容

学号	课程号	成绩
951001	0001	100
951003	0001	89
954006	0002	78
953008	0002	90

源程序如下：

```
package crsj;
import java.sql.*;
//在成绩表中插入数据 (数据源名为 student)
public class InsertData {
  public static void main(String[] args) {

    try {//加载 JDBC 数据库驱动程序
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
          System.out.println ("成功加载 JDBC-ODBC 驱动程序！");
           }
    catch (ClassNotFoundException ee) {
            System.out.println ("加载 JDBC-ODBC 驱动程序失败！");
            System.out.println (ee.getMessage ());
          return;
          }
      try
        { //建立与数据源 student 的连接，用户名和口令为空
          Connection conn=DriverManager.getConnection("jdbc:odbc:
              student","","");
          // 创建 Statement 对象
          Statement  Stmt=conn.createStatement();
           //向数据库发送 SQL 语句（利用 Statement 对象）
          Stmt.executeUpdate("insert into 成绩 values('951001','0001',100)");
          Stmt.executeUpdate("insert into 成绩 values('951003','0001',89)");
          Stmt.executeUpdate("insert into 成绩 values('954006','0002',78)");
          Stmt.executeUpdate("insert into 成绩 values('953008','0002',90)");
          Stmt.executeUpdate("insert into 成绩 values('951001','0002',89)");
```

```
            System.out.println ("在成绩表插入数据成功!");
            conn.close();
           }

        catch(SQLException e1)
         {
           System.out.println ("在成绩表插入数据失败!");
           System.err.println("SQLException: " + e1.getMessage());
           return;
         }
    }
}
```

2. 数据查询（一）

（1）新建项目。

新建一个项目，项目名为 StudentMIS，并将这个项目保存在自己的文件夹中，不要采用默认路径保存。

（2）创建一个 Java GUI 窗体。

1）单击“文件”→“新建文件”命令，出现“新建文件”对话框，如图 2-15 所示。在“新建文件”对话框中，选择“类别”列表框中的“Java GUI 窗体”，再在“文件类型”列表框中选择“JFrame 窗体”，单击“下一步”按钮。

图 2-15 “新建文件”对话框

2）单击“下一步”按钮后，出现“新建 JFrame 窗体”对话框，如图 2-16 所示。用户在“类名”文本框中输入文件名，并选择包，单击“完成”按钮。

（3）在新建好的 JQueryFrame 窗体中设计界面，如图 2-17 所示。

图 2-16　“新建 JFrame 窗体”对话框

图 2-17　“学生成绩查询”界面

（4）编写代码。

1）添加“确定”按钮的代码。

```
private void jBokActionPerformed(java.awt.event.ActionEvent evt) {
//取得要查询的学号
String sno1=jTFsno.getText();
   String kno1=jTFkno.getText();
      try
         {
            //装载驱动
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         }
         catch(ClassNotFoundException ce)
         {
           System.out.println("SQLException:"+ce.getMessage());
         }
```

```
            try {
                //建立连接
                    Connection con = DriverManager.getConnection(
                       "jdbc:odbc:student","","");
                    Statement stmt = con.createStatement();
                //执行 SQL 语句
                    ResultSet rs = stmt.executeQuery("select * from 成绩 where 学
号='" +sno1 + "' and 课程号='"+kno1+"'");
                //处理结果
                    if (rs.next()) {
                      jTFchengji.setText(rs.getString("成绩"));

                    }
                    else
                JOptionPane.showMessageDialog(null,"对不起，没有该学号的成绩！");
                //断开连接
                    rs.close();
                    stmt.close();
                    con.close();
                  }
                  catch (SQLException ex) {

                    System.out.println("SQLException:" + ex.getMessage());
                  }
      }
```

2）添加“取消”按钮的代码。

```
private void jBcancelActionPerformed(java.awt.event.ActionEvent evt) {
   this.dispose() ;
}
```

3）添加“清空”按钮的代码。

```
private void jBclearActionPerformed(java.awt.event.ActionEvent evt) {
   jTFsno.setText("");
   jTFkno.setText("");
   jTFchengji.setText("");
   }
```

3. 数据查询（二）

（1）打开项目。

在“数据查询”（一）中，已经新建了一个项目，项目名为 StudentMIS，将这个项目打开。

（2）创建一个 Java GUI 窗体。

1）单击“文件”→“新建文件”命令，出现“新建文件”对话框，如图 2-15 所示。在“新建文件”对话框中，选择“类别”列表框中的“Java GUI 窗体”，再选择“文件类型”列表框中的“JFrame 窗体”，单击“下一步”按钮。

2）单击“下一步”按钮后，出现“新建 JFrame 窗体”对话框，如图 2-16 所示。用户在“类名”文本框中输入文件名 JComQueryFrame，把包名选择为 StudentMIS，单击“完成”按钮。

（3）新建好的 JComQueryFrame 窗体设计界面如图 2-18 所示。

图 2-18　“学生成绩浏览”界面

（4）编写代码。

1）在类里面定义一个 rs 数据结果集对象。

```
ResultSet rs;
```

2）添加“查询”按钮的代码。

```
private void jBtnQueryActionPerformed(java.awt.event.ActionEvent evt) {
    //取得要查询的学号
    String sno1=jTFsno.getText();
    try {
        //装载驱动
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    } catch(ClassNotFoundException ce) {
        System.out.println("SQLException:"+ce.getMessage());
    }
       try {
       //建立连接
       Connection con = DriverManager.getConnection(
               "jdbc:odbc:student","","");
        Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_ SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
        //执行 SQL 语句
        rs = stmt.executeQuery("select * from 成绩 where 学号='" +
                 sno1 + "'");
        //处理结果
               } catch (SQLException ex) {
         System.out.println("SQLException:" + ex.getMessage());
    }
         try {
        //判断返回的结果集是否为空，若不为空显示第一条记录
        if(!rs.equals(null) )
        {  rs.next();
```

```
            //显示当前记录
        jTFsno.setText(rs.getString("学号"));
        jTFkno.setText(rs.getString("课程号")) ;
        jTFchengji.setText(rs.getString("成绩"));
           }
         }//若出现异常，弹出警告对话框
         catch (SQLException ex) {
             JOptionPane.showMessageDialog(null," 查询失败!");
        }
}
```

3）添加“第一条”按钮的代码。

```
private void jBtnFirstActionPerformed(java.awt.event.ActionEvent evt) {
try {
    rs.first();
    jTFsno.setText(rs.getString("学号"));
    jTFkno.setText(rs.getString("课程号")) ;
    jTFchengji.setText(rs.getString("成绩"));
       }
       catch (SQLException ex) {
       }
}
```

4）添加“上一条”按钮的代码。

```
private void jBtnPreActionPerformed(java.awt.event.ActionEvent evt) {
try {
   if(!rs.isFirst() )
    {rs.previous();
      jTFsno.setText(rs.getString("学号"));
      jTFkno.setText(rs.getString("课程号")) ;
      jTFchengji.setText(rs.getString("成绩"));
      }
   }
       catch (SQLException ex) {}
}
```

5）添加“下一条”按钮的代码。

```
private void jBtnNextActionPerformed(java.awt.event.ActionEvent evt) {
try {
    if(!rs.isLast())
    {rs.next();
      jTFsno.setText(rs.getString("学号"));
      jTFkno.setText(rs.getString("课程号")) ;
      jTFchengji.setText(rs.getString("成绩"));
      }
    }
    catch (SQLException ex) {}
}
```

6）添加“最末条”按钮的代码。

```
private void jBtnLastActionPerformed(java.awt.event.ActionEvent evt) {
try {
if(!rs.isLast())
       {  rs.next();
          jTFsno.setText(rs.getString("学号"));
          jTFkno.setText(rs.getString("课程号")) ;
          jTFchengji.setText(rs.getString("成绩"));
       }
   }
   catch (SQLException ex) {}
}
```

4. 数据添加（一）

（1）打开项目。

在数据查询（一）中，已经新建了一个项目，项目名为 StudentMIS，将这个项目打开。

（2）创建一个 Java GUI 窗体。

1）单击“文件”→“新建文件”命令，出现“新建文件”对话框。在“新建文件”对话框中，选择“类别”列表框中的“Java GUI 窗体”，再选择“文件类型”列表框中的“JFrame 窗体”，单击“下一步”按钮。

2）单击“下一步”按钮后，出现“新建 JFrame 窗体”对话框。用户在“类名”文本框中输入文件名 JAddFrame，把包名选择为 StudentMIS，单击“完成”按钮。

（3）新建好的 JAddFrame 窗体设计界面如图 2-19 所示。

图 2-19　“学生成绩添加”界面

（4）编写代码。

1）添加“确定”按钮的代码。

```
private void jBokActionPerformed(java.awt.event.ActionEvent evt) {
   String sno1=jTFsno.getText();
   String kno1=jTFkno.getText();
   String chengji1=jTFchengji.getText();
        try
        {
                 //装载驱动
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```
            }
            catch(ClassNotFoundException ce)
            {
              System.out.println("SQLException:"+ce.getMessage());
            }
            try {
                //建立连接
                Connection con = DriverManager.getConnection(
                   "jdbc:odbc:student","","");
                Statement stmt = con.createStatement();
                //执行 SQL 语句
                ResultSet rs = stmt.executeQuery("select * from 成绩 where 学号='"
+sno1 + "' and 课程号='"+kno1+"'");
                if (rs.next()) {
                JOptionPane.showMessageDialog(null," 对不起，该成绩信息已存在！ ");
                }
                else  //否则插入记录
                  {
                    stmt.executeUpdate("insert  into  成 绩   values('"+sno1+"',
'"+kno1+"',"+chengji1+")");
                  JOptionPane.showMessageDialog(null," 记录已经成功添加！ ");
                  }
                //断开连接
                rs.close();
                stmt.close();
                con.close();
              }
                catch (SQLException ex) {
                System.out.println("SQLException:" + ex.getMessage());
              }
    }
```

2）添加“取消”按钮的代码。

```
private void jBcancelActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose() ;
}
```

3）添加“清空”按钮的代码。

```
private void jBclearActionPerformed(java.awt.event.ActionEvent evt) {
   jTFsno.setText("");
   jTFkno.setText("");
   jTFchengji.setText("");
   }
```

5. 数据添加（二）

（1）打开项目。

在“数据查询（一）”中，已经新建了一个项目，项目名为 StudentMIS，将这个项目打开。

（2）创建一个 Java GUI 窗体。

1）单击“文件”→“新建文件”命令，出现“新建文件”对话框。在“新建文件”对话框中，选择“类别”列表框中的“Java GUI 窗体”，再选择“文件类型”列表框中的“JFrame 窗体”，单击“下一步”按钮。

2）出现“新建 JFrame 窗体”对话框。用户在“类名”文本框中输入文件名 JComAddFrame，把包名选择为 StudentMIS，单击“完成”按钮。

（3）在新建好的 JComAddFrame窗体中设计界面，如图 2-19 所示。

（4）编写代码。

1）添加“确定”按钮的代码。

```
private void jBokActionPerformed(java.awt.event.ActionEvent evt) {
     String sno1=jTFsno.getText();
     String kno1=jTFkno.getText();
     String chengji1=jTFchengji.getText();

     if(sno1.trim().equals(""))
     JOptionPane.showMessageDialog(null,"学号不能为空！");
     else {
     if(kno1.trim().equals(""))
     JOptionPane.showMessageDialog(null,"课程号不能为空！");
    else {
     if(chengji1.trim().equals(""))
     JOptionPane.showMessageDialog(null,"成绩不能为空！");
    else {
     int chengji=Integer.parseInt(chengji1);
     if(chengji>100||chengji<0)
           JOptionPane.showMessageDialog(null,"成绩要在 0-100 之间");
       else {
         try
          {
                 //装载驱动
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
          }
            catch(ClassNotFoundException ce)
          {
            System.out.println("SQLException:"+ce.getMessage());
          }
          try {
             //建立连接
             Connection con = DriverManager.getConnection(
                "jdbc:odbc:student","","");
             Statement stmt = con.createStatement();
             //执行 SQL 语句
             ResultSet rs = stmt.executeQuery("select * from 成绩 where 学号='"
+sno1 + "' and 课程号='"+kno1+"'");
              if (rs.next()) {
           JOptionPane.showMessageDialog(null,"对不起，该成绩信息已存在！");
```

```
                }
                else  //否则插入记录
                  {
                    stmt.executeUpdate("insert  into  成 绩  values('"+sno1+"',
'"+kno1+"',"+chengji1+")");
                  JOptionPane.showMessageDialog(null," 记录已经成功添加！");
                  }
                //断开连接
                rs.close();
                stmt.close();
                con.close();
              }
              catch (SQLException ex) {
                System.out.println("SQLException:" + ex.getMessage());
              }
          }
          }
       }
      }
    }
```

2）添加“取消”按钮的代码。

```
 private void jBcancelActionPerformed(java.awt.event.ActionEvent evt) {
 this.dispose() ;
 }
```

3）添加“清空”按钮的代码。

```
private void jBclearActionPerformed(java.awt.event.ActionEvent evt) {
    jTFsno.setText("");
    jTFkno.setText("");
    jTFchengji.setText("");
   }
```

6. 数据修改

（1）打开项目。

在“数据查询（一）”中，已经新建了一个项目，项目名为 StudentMIS，将这个项目打开。

（2）创建一个 Java GUI 窗体。

1）单击“文件”→“新建文件”命令，出现“新建文件”对话框。在“新建文件”对话框中，选择“类别”列表框中的“Java GUI 窗体”，再选择“文件类型”列表框中的“JFrame 窗体”，单击“下一步”按钮。

2）出现“新建 JFrame 窗体”对话框。用户在“类名”文本框中输入文件名 JModifyFrame，把包名选择为 StudentMIS，单击“完成”按钮。

（3）新建好的 JModifyFrame 窗体设计界面，如图 2-20 所示。

图 2-20　“学生成绩修改”界面

（4）编写代码。

1）添加“查询”按钮的代码。

```
private void jBQueryActionPerformed(java.awt.event.ActionEvent evt) {
    String sno1=jTFsno.getText();
    String kno1=jTFkno.getText();
        try
          {
//装载驱动
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
          }
          catch(ClassNotFoundException ce)
          {
            System.out.println("SQLException:"+ce.getMessage());
          }
          try {
                //建立连接
                Connection con = DriverManager.getConnection(
                   "jdbc:odbc:student","","");
                Statement stmt = con.createStatement();
                //执行 SQL 语句
                ResultSet rs = stmt.executeQuery("select * from 成绩 where
学号='" + sno1 + "' and 课程号='"+kno1+"'");
                //处理结果
                if (rs.next()) {
                  jTFchengji.setText(rs.getString("成绩"));
                }
                else
              JOptionPane.showMessageDialog(null,"对不起，没有该学号的成绩！");
                //断开连接
                rs.close();
                stmt.close();
                con.close();
```

```
                    }
                    catch (SQLException ex) {
                      System.out.println("SQLException:" + ex.getMessage());
                    }
        // TODO 将在此处添加您的处理代码:
        }
```

2）添加“修改”按钮的代码。

```
    private void jBModifyActionPerformed(java.awt.event.ActionEvent evt) {
        String sno1=jTFsno.getText();
        String kno1=jTFkno.getText();
        String chengji1=jTFchengji.getText();
             try
               {
                            //装载驱动
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
               }
               catch(ClassNotFoundException ce)
               {
                 System.out.println("SQLException:"+ce.getMessage());
               }
               try {
                      //建立连接
                      Connection con = DriverManager.getConnection(
                         "jdbc:odbc:student","","");
                      Statement stmt = con.createStatement();
                      //执行 SQL 语句
                         stmt.executeUpdate("update  成绩  set  成绩 ="+chengji1+"
where 学号='"+sno1+"' and 课程号='"+kno1+"'");
                       JOptionPane.showMessageDialog(null," 记录已经成功修改! ");

                      //断开连接
                      stmt.close();
                      con.close();
                    }
                    catch (SQLException ex) {
                      System.out.println("SQLException:" + ex.getMessage());
                    }
        }
```

3）添加“取消”按钮的代码。

```
    private void jBcancelActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose() ;
    }
```

4）添加“清空”按钮的代码。

```
    private void jBclearActionPerformed(java.awt.event.ActionEvent evt) {
        jTFsno.setText("");
        jTFkno.setText("");
```

```
        jTFchengji.setText("");
      }
```

7. 数据删除

（1）打开项目。

在“数据查询（一）”中，已经新建了一个项目，项目名为 StudentMIS，将这个项目打开。

（2）创建一个 Java GUI 窗体。

1）单击“文件”→“新建文件”命令，出现“新建文件”对话框。在“新建文件”对话框中，选择“类别”列表框中的“Java GUI 窗体”，再选择“文件类型”列表框中的“JFrame 窗体”，单击“下一步”按钮。

2）出现“新建 JFrame 窗体”对话框。用户在“类名”文本框中输入文件名 JDeleteFrame，把包名选择为 StudentMIS，单击“完成”按钮。

（3）新建好的 JDeleteFrame 窗体设计界面如图 2-21 所示。

图 2-21　“学生删除模块”界面

（4）编写代码。

1）添加“查询”按钮的代码。

```
private void jBQueryActionPerformed(java.awt.event.ActionEvent evt) {
        //取得要查询的学号
        String sno1=jTFsno.getText();
        String kno1=jTFkno.getText();
        try {
            //装载驱动
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        } catch(ClassNotFoundException ce) {
            System.out.println("SQLException:"+ce.getMessage());
        }
        try {
            //建立连接
            Connection con = DriverManager.getConnection(
                    "jdbc:odbc:student","","");
            Statement stmt = con.createStatement();
            //执行 SQL 语句
            ResultSet rs = stmt.executeQuery("select * from 成绩 where 学号='" +
```

```
                sno1 + "' and 课程号='"+kno1+"'");
            //处理结果
            if (rs.next()) {
               jTFchengji.setText(rs.getString("成绩"));

            } else
            JOptionPane.showMessageDialog(null,"对不起，没有该学号的成绩！");
            //断开连接
            rs.close();
            stmt.close();
            con.close();
         } catch (SQLException ex) {

            System.out.println("SQLException:" + ex.getMessage());
         }
      }
```

2）添加“删除”按钮的代码。

```
   private void jBDeleteActionPerformed(java.awt.event.ActionEvent evt) {
      String sno1=jTFsno.getText(); //取得要查询的学号
      String kno1=jTFkno.getText();
               try
                {
                              //装载驱动
                 Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                }
                catch(ClassNotFoundException ce)
                {
                  System.out.println("SQLException:"+ce.getMessage());
                }
                try {
                       //建立连接
               Connection con = DriverManager.getConnection(
                          "jdbc:odbc:student","","");
               Statement stmt = con.createStatement();
                       //执行 SQL 语句
      if(JOptionPane.showConfirmDialog(null,"确定要删除本记录吗？","确认",
JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE)==0)
        {stmt.executeUpdate("delete from 成绩 where 学号='" + sno1 + "' and 课程
号='"+kno1+"'");
      JOptionPane.showMessageDialog(null, "记录已经成功删除！");
      jTFchengji.setText("");
                }
                stmt.close();
                con.close();
             }
             catch (SQLException ex) {
```

```
                System.out.println("SQLException:" + ex.getMessage());
            }
        }
```

3）添加“取消”按钮的代码。

```
private void jBcancelActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose() ;
}
```

4）添加“清空”按钮的代码。

```
private void jBclearActionPerformed(java.awt.event.ActionEvent evt) {
    jTFsno.setText("");
    jTFkno.setText("");
    jTFchengji.setText("");
   }
```

8. 系统登录

（1）打开项目。

在“数据查询（一）”中，已经新建了一个项目，项目名为 StudentMIS，将这个项目打开。

（2）创建应用程序的主窗体。

1）在项目中新建一个名为 JMainFrame 的 JFrame 窗体。窗体的界面自行设计，可以添加一些项目名称、制作者信息等内容，如图 2-22 所示。

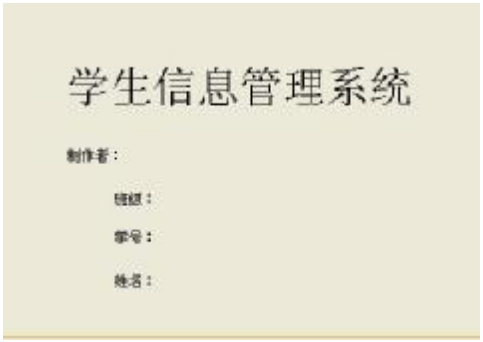

图 2-22　主窗体界面

2）将 JMainFrame 设置为项目的主类。当运行主项目时，需要选择主类，可将 JMainFrame 选择为主类。

（3）制作菜单。

主要的步骤如下：

1）在主窗体上添加 JMenuBar。

2）JMenuBar 上添加 4 个 Jmenu。

3）为各个 JMenu 添加相应的 JmenuItem，如图 2-23 所示。

（4）为各菜单项添加事件，即通过菜单打开对应的窗口。

举例：现在要为“成绩信息添加”菜单项编写事件代码，则步骤如下：

1）找到该菜单项。

2）添加 Action 事件。

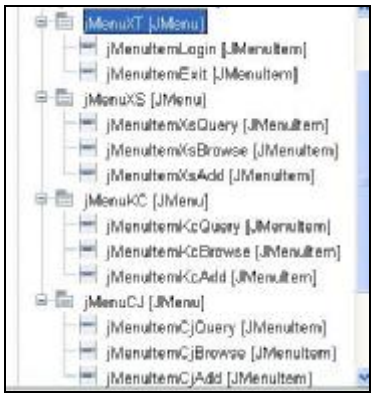

图 2-23　菜单

3）代码参考：

```
JAddFrame addf=new JAddFrame();  // JAddFrame 为"成绩信息添加"对应的窗口
    addf.pack();
    addf.setVisible(true);
```

4）注意：修改 JAddFrame 窗口的 defaultCloseOperation属性，将值修改为 DISPOSE。原因是关闭该窗口只是隐藏，并不是退出整个系统。然后把其余菜单事件添加完整。

（5）制作登录窗口。

出于安全性的考虑，系统应包含登录窗口，当用户输入正确的用户名和密码时，可以进入系统操作，否则不能进入。

1）界面的设计。

- 在项目中新建一个名为 JLoginFrame的 JFrame 窗体。
- 界面可以参考图 2-24。

图 2-24　登录界面

注意：密码后面的组件是密码框 JPasswordField，不是文本框。

2）代码编写。

- “登录”按钮参考代码。

```
        String user=jTextFielduser.getText();// 取得用户名处文本框的值，即用户输入的值，赋值给字符串变量 user
        String pw=new String(jPasswordFieldpw.getPassword());
```

```
        //注意：PasswordField 用的是 getPassword()，并且返回的是字符数组，所以必须由字符数组生成字符串
        if(user.equalsIgnoreCase("admin") && pw.equalsIgnoreCase("123456"))
            //如果用户名为 admin 并且密码为 123456。注意字符串的比较用 equalsIgnoreCase 而不是==
        { JOptionPane.showMessageDialog(this,"恭喜您！成功登录！");
          this.dispose();
          JMainFrame mf=new  JMainFrame();
          mf.pack();
          mf.setVisible(true);
        }

            //输入正确时，弹出对话框显示
        else
            JOptionPane.showMessageDialog(this,"对不起，输入有误！");
          //输入错误时，弹出对话框显示
```

- “取消”按钮参考代码。

```
        jTextFielduser.setText("");//将用户名的文本框清空
        jPasswordFieldpw.setText("");//将密码框清空
```

3）登录窗口的使用。

由于项目运行时，应先出现登录窗口，当验证成功时再显示主窗口，所以我们修改 JMainFrame 的 main 方法的代码。

将“new JMainFrame().setVisible(true);”修改为“new JLoginFrame().setVisible(true);”。

9. 菜单制作举例

按照以下步骤操作，就可以创建一个菜单结构。

（1）创建一个名为 Jcd 的项目，其中包含一个主类 JmenuTest，然后在项目中新建一个名为 DemoJMenu 的 Jframe 窗体。

（2）向 DemoJMenu 中添加一个 JMenuBar，在对象检查器窗口中选中它，右击，如图 2-25 所示，将其变量名称修改为 jMenuBarGlobal。

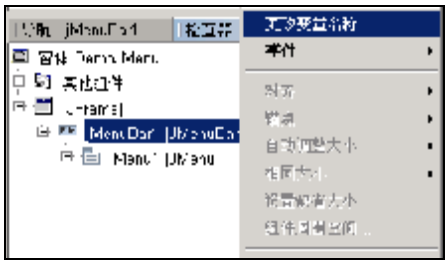

图 2-25　对象检查器

（3）在对象检查器窗口中选中 jMenuBarGlobal 节点下的 jMenu1 节点，该 J 节点是添加 JMenuBar 时自动添加的，将其变量名称修改为 jMenuFile。在“属性”对话框中将其 text 属性的信息改为 File。

（4）在对象检查器窗口中右击 jMenuBarGlobal 节点，在弹出的菜单中单击“添加 JMenu”命令，如图 2-26 所示，向 jMenuBarGlobal 中添加一个 JMenu。将该 JMenu 的名称修

改为 jMenuEdit，并将其 text 属性的值修改为 Edit。

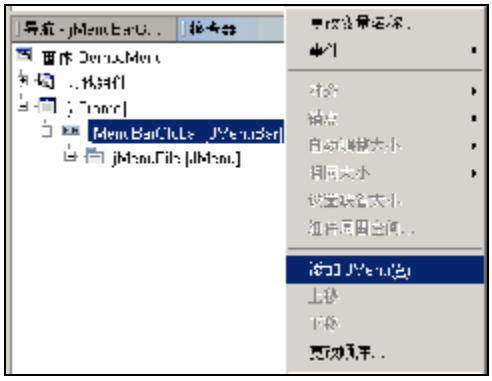

图 2-26　添加 JMenu

（5）按步骤（4）所述再向 jMenuBarGlobal 中添加一个名称为 jMenuAbout，text 属性值为 About 的 JMenu。此时，GUI 设计器中内容如图 2-27 所示。

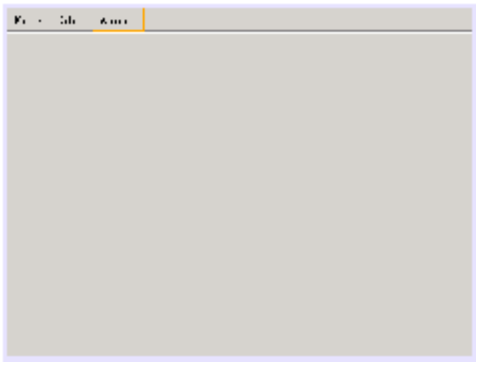

图 2-27　菜单栏

（6）在对象检查器中右击 jMenuFile 节点，依次单击右键菜单中的“添加”→JMenuItem 选项，如图 2-28 所示，向 jMenuFile 中添加一个 JMenuItem，将其名称修改为 jMenuItemOpenFile，将其 text 属性的值改为 OpenFile。

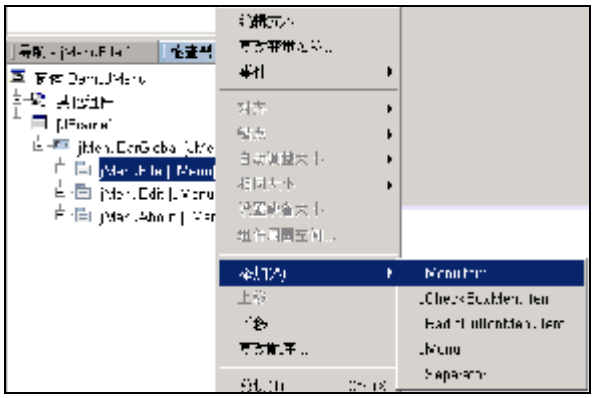

图 2-28　添加菜单项

（7）为 jMenuItemOpenFile 设置加速器。在 jMenuItemOpenFile 的属性面板中，单击 accelerator 属性右侧的…按钮，弹出如图 2-29 所示的窗口。

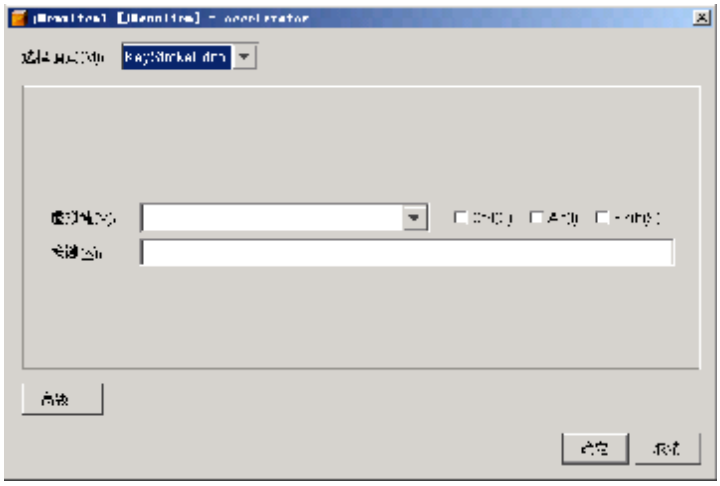

图 2-29　设置加速器快捷键窗口

（8）通过在“虚拟键”下拉列表框中选择要使用的按键或者单击键盘上的相应按键修改 Key Stroke 文本框的值来为 jMenuItemOpenFile 设置加速器的按键。还可以通过选择 Ctrl、Alt 或者 Shift 复选框设置加速器的组合键。这里将使用 Ctrl+O 作为加速器的组合键，如图 2-30 所示。单击“确定”按钮完成设置。

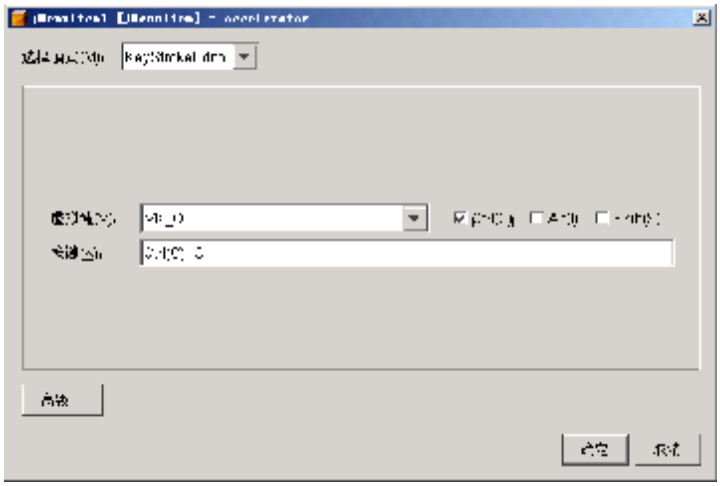

图 2-30　完成加速器快捷键设置

（9）按照步骤（6）向 jMenuFile 中添加名称为 jMenuItemSaveFile的 JMenuItem，将其 text 属性值修改为 Save File。按照步骤（8）所述为 jMenuItemSaveFile设置加速器，并将加速器组合键设置为 Ctrl+S。

（10）在对象检查器窗口中右击 jMenuFile 节点，依次单击右键菜单中的“添加”→Jseparator 命令，如图 2-31 所示，向 jMenuFile添加一个 JSeparator（菜单分隔条）。

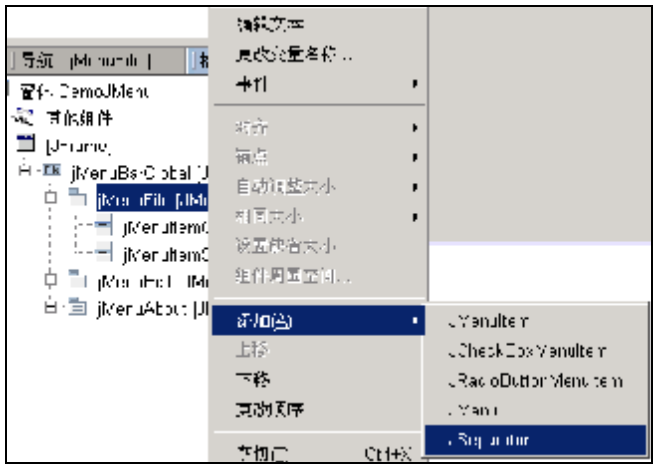

图 2-31 添加 JSeparator

（11）向 jMenuFile 中添加名称为 jMenuOthers 的 JMenu，如图 2-32 所示，将其 text 属性值修改为 Others。

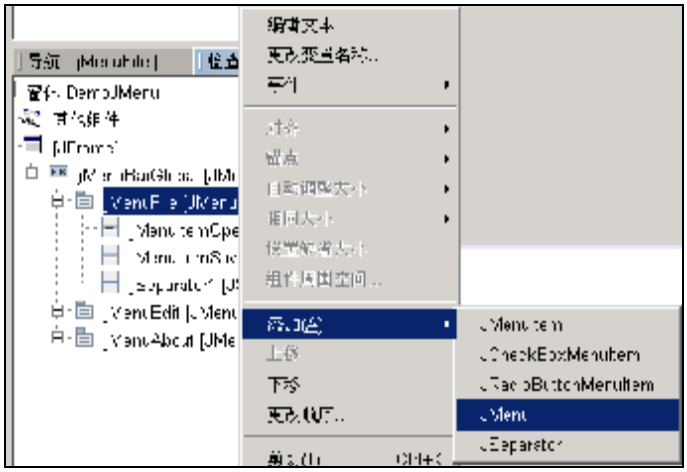

图 2-32 添加子菜单

（12）按照步骤（6）向 jMenuOthers 中分别添加名称为 jMenuItemSetting 和 jMenuItem-SaveAs 的 JmenuItem，并将其 text 属性值分别修改为 Setting 和 SaveAs。

（13）向 jMenuFile 中添加名称为 jMenuItemExit 的 JMenuItem，将 text 属性值修改为 Exit，并将加速器组合键设置为 Ctrl+E。

（14）分别为 jMenuItemOpenFile、jMenuItemSaveFile和 jMenuItemExit添加 ActionEvent 事件的处理方法，添加方法：选中相应的菜单项，右击，从弹出的快捷菜单中单击“事件”→Action→ActionPerformed命令。并向相应方法添加如下代码。

- 首先在源代码的最顶端添加如下代码：

```
import javax.swing.*;//包含 swing 包
```

- 向 jMenuItemOpenFile的 ActionPerformend方法中添加如下代码：

```
JfileChooser. fileChooser=new JFileChooser();
```

```
fileChooser.showopenDialog(this);// 打开文件选择器对话框
```

- 向 jMenuItemSaveFile的 ActionPerformed方法中添加如下代码：

```
JFileChooser fileChooser=new JFileChooser();
fileChooser.s showSaveDialog(this);// 打开文件保存对话框
```

- 向 jMenuItemExit的 ActionPerformed方法中添加如下代码：

```
System.exit(0);
```

（15）将下列代码添加到 JMenuTest 类的 main 方法中。

```
new DemoJMenu ().setVisible(true); // 创建 DemoJFrame 窗口类对象并且显示它
```

（16）编译并运行项目，界面如图 2-33 所示。

（17）依次选择 File→Others 选项，界面如图 2-34 所示。

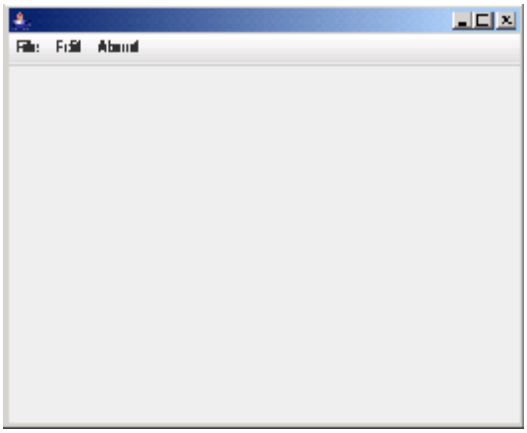

图 2-33　程序运行界面

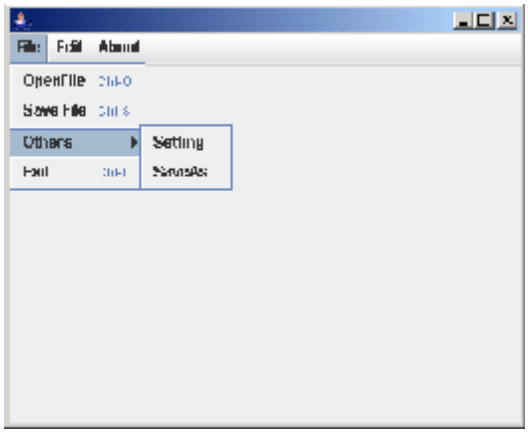

图 2-34　Others 子菜单

（18）单击 File→OpenFile 命令，打开“打开”对话框，界面如图 2-35 所示。

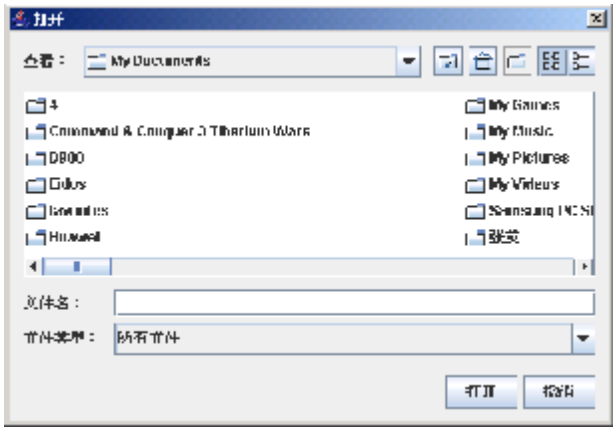

图 2-35　“打开”对话框

提示：本程序并没有真正地从选择的文件中读取内容。

（19）在如图 2-34 所示的状态下，按下 Ctrl+S 组合键后，会打开“保存”对话框，界面如图 2-36 所示。

（20）单击 File→Exit 命令，或者按下 Ctrl+E 组合键，即可退出程序。

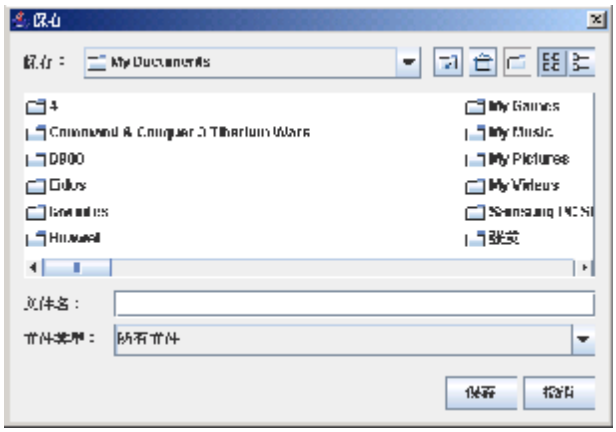

图 2-36 “保存”对话框

实训项目二 “俄罗斯方块”游戏的设计与实现

一、模块划分及功能说明

（1）ErsBlocksGame.java：游戏主类，继承自 JFrame 类，负责游戏的全局控制。内含：

1）一个 GameCanvas 画布类的实例引用。

2）一个保存当前活动块（ErsBlock）实例的引用。

3）一个保存当前控制面板（ControlPanel）实例的引用。

（2）ErsBox.java：方格类，是组成块的基本元素，用自己的颜色来表示块的外观。

（3）ControlPanel.java：控制面板类，继承自 JPanel，上边安放预显窗口、等级、得分、控制按钮，主要用来控制游戏进程。

（4）ErsBlock.java：块类，继承自线程类（Thread），由 4×4 个方格（ErsBox）构成一个块，控制块的移动、下落、变形等。

（5）GameCanvas.java：画布类，内有“行数×列数”个方格类实例，继承自 JPanel 类。ErsBlock 线程类动态改变画布类的方格颜色，画布类通过检查方格颜色来体现 ErsBlock 块的移动情况。

二、模块功能的实现

1. ErsBlockGame 类说明

ErsBlockGame 中用到的方法具体说明如下。

（1）main()：程序入口函数。

（2）ErsBlockGame()：构造函数，初始化窗体界面。

（3）setSize()：窗口大小。

（4）setLocation()：利用屏幕和主窗体的大小差值，使窗口居中。

（5）createMenu()：建立并设置窗口菜单，依次添加：

1）菜单栏：bar；

2）菜单项：mGame、mControl、mWindowStyle;

3）子菜单：

① mGame: miNewGame、miSetBlockColor、miSetBackColor、miTurnHarder、miTurnEasier、miExit;

② mControl：miPlay、miPause、miResume、miStop;

③ mWindowStyle：miAsWindows、miAsMotif、miAsMetal。

（6）依次添加各子菜单的事件监听器。

1）NewGame：新游戏。

- stopGame()：先停止原来的游戏（判断 ErsBlock 类对象 Block 是否为空，如不为空，即调用它的 pauseMove()方法，即 stopping 为 true）。
- reset()：重新还原状态（分别调用 ctrlPanel.reset()控制面板复位和 canvas.reset()画布复位）。
- setLevel()：设置级别，初始值为 5。
- miSetBlockColor：设置活动的方块的颜色（JColorChooser.showDialog选择颜色对话框）。
- miSetBackColor：设置画布的背景色（同上）。
- miTurnHarder：设置高一级的级别，getLevel()与 MAX_LEVEL 的比较，再调用 setLevel()。
- miTurnEasier：设置低一级的级别，getLevel()与 Min_LEVEL 的比较，再调用 setLevel();
- miExit：退出。
- miPlay：开始游戏，调用 playGame()→this.play()→this.reset()→新建线程并启动 start()→调用 Game（此类实现 Runnable 接口）的 run()方法→判断 block 是否为空，判断 checkFullLine()是否有满行，判断游戏是否结束；随机产生画布类的方块类型和控制面板类的方块类型。
- miPause：暂停游戏（调用 pauseGame()）。
- miResume：唤醒游戏，调用 resumeGame()→判断 ErsBlock 类对象 Block 是否为空，如不为空，即调用它的 resumeMove()方法，即 stopping 为 false。
- miStop：停止游戏，调用 stopGame()→判断 ErsBlock 类对象 Block 是否为空，如不为空，即调用它的 stopMove()方法，即 moving 为 false。
- miAsWindows：设置为 Windows 界面，调用 this.setWindowStyle()，再调用 canvas，ctrlPanel 的 fanning()。
- miAsMotif：设置为 motif 界面。
- miAsMetal：设置为 metal 界面。
- mguanyu：“关于”对话框。

2）canvas：创建画布类对象（调用 canvas 的构造函数）。

3）ctrlPanel：创建控制面板类对象（调用 ctrlPanel 的构造函数）。

4）addWindowListener：添加窗口适配器（调用 stopGame()）。

5）addComponentListener：添加组件适配器（调用 canvas.fanning()）。

6）show()：显示窗体。

7）canvas.fanning()：根据窗口的大小，自动调整方格的尺寸。

8）以下为外部类调用的方法：