**兰州交通大学**

**计算机科学与技术系**

**实验报告**

**课程名称： Java语言程序设计专题实验**

**实验名称： java大作业**

**专业班级： 计算机科学与技术1701**

**姓 名：唐靖（张毅峰 陈毅 陈奎斌 鄢虎强）**

**学 号： 201711014**

**指导教师： 巨涛 成 绩：**

**兰州交通大学Java大程序设计实验报告**

学号： 201711014 姓名： 唐靖

**实验名称：**

|  |
| --- |
| 实验目的 设计一个综合的java程序，将前面设计完成的java程序集成为一个综合的java程序，内容包括如下几部分：  文件：主要是通讯录程序的设计；  出版物：主要是书籍、CD和磁带信息编辑程序的设计；  工具：主要是数学计算器和日期计算器程序的设计；  游戏：主要是五子棋的设计。  其中我主是要做出版物模块。 实验内容 程序主要采用父子窗口的形式。主窗口类设计为继承自JFrame类的ApplicationFrame类，拥有菜单文件、出版物、工具和游戏，分别进行各个子功能的选择。子窗口全部设计为继承自JInternalFrame的类；  书籍子窗口：继承自JInternalFrame类的BookInternalFrame类；  CD子窗口：继承自JInternalFrame类的CDInternalFrame类；  磁带子窗口：继承自JInternalFrame类的TapeInternalFrame类；  在书籍，cd，磁带的子窗口中，分别还有增加，删除，修改的三个菜单栏，输入书籍，cd，磁带的信息在书籍，cd，磁带子窗口中，点击增加即增加，在窗口中选择即可点击删除，或者选择修改。 三、 实验环境 包括具体硬件平台，操作系统，开发环境等软硬件环境、  硬件平台：  操作系统：windows 10  开发环境：  软硬件环境： 四、实验过程（包括程序设计说明，实验步骤，经调试后正确的源程序，程序运行结果）一．程序设计说明： 源文件media.java设计了出版物父类MyMedia、书子类MyBook、CD子类MyCD和磁带子类MyTape。  源文件book.java设计了继承自JInternalFrame类的BookInternalFrame类。在这个类中有一个MyBook类对象的数组，存储了当前所有的书籍信息。在这个子窗口中可以通过选择菜单进行书籍信息的添加、修改和删除。所有的书籍信息会显示在一个列表组件中，当某本书籍的信息改变时，列表中会相应的进行改变。  为管理CD而设计了继承自JInternalFrame类的CDInternalFrame类。在这个类中有一个MyCD类对象的数组，存储了当前所有的CD信息。在这个子窗口中可以通过选择菜单进行CD信息的添加、修改和删除。所有的CD信息会显示在一个列表组件中，当某张CD 的信息改变时，列表中会相应的进行改变。  为管理磁带而设计了继承自JInternalFrame类的TapeInternalFrame类。在这个类中有一个MyTape类对象的数组，存储了当前所有的磁带信息。在这个子窗口中可以通过选择菜单进行磁带信息的添加、修改和删除。所有的磁带信息会显示在一个列表组件中，当某本磁带的信息改变时，列表中会相应的进行改变。 二．实验步骤：  1. 、创建Mybook，Mycd，MyTape,Appliction,等9个类， 2. 、其中Mybook，Mycd，Mytap三类是继承MyMdia为初始化数据和构造方法，在后面，Applictiionframe是继承JFarmed的子类，用于做图形化界面，界面中为出版物，其中含有书籍，cd，tape三类。 3. 在界面中的书籍，cd，tape三类之外，在每个BookInternalFrame，CDInternalFrame，TapeInternalFrame单独为之构造三个用户菜单项目为增加，删除，修改三个菜单。 4. 最后调用主函数即可执行本个小模块。  三．源程序： **package** 出版物;  **import** java.io.\*;  **import** java.util.Date;  **import** java.text.\*;  **class** MyBook **extends** MyMedia {  String editor; //书的编辑  Date publishDate = **null**;//出版日期  String bookISBN; //书的ISBN  //构造函数  MyBook(String bn, **float** bp, String bpr, String ba, String bi,String be,Date bpd){  **super**(bn,bp,bpr,ba);  bookISBN = bi;  editor = be;  publishDate=bpd;  }  String getEditor(){  **return** editor;  }  String getBookISBN(){  **return** bookISBN;  }  Date getPublishDate(){  **return** publishDate;  }  **void** setEditor(String editortemp){  editor=editortemp;  }  **void** setBookISBN(String bookISBNtemp){  bookISBN=bookISBNtemp;  }  **void** setPublishDate(Date publishDatetemp){  publishDate=publishDatetemp;  }  }  **package** 出版物;  **class** MyCD **extends** MyMedia{  String cdISRC; //CD的ISRC  String cdPublisher; //CD的发行商  //构造函数  MyCD(String cn, **float** cp, String cpr, String ca, String cpl,String ci){  **super**(cn,cp,cpr,ca);  cdISRC = ci;  cdPublisher = cpl;  }  String getCDISRC(){  **return** cdISRC;  }  String getCDPublisher(){  **return** cdPublisher;  }  **void** setCDISRC(String cdISRCtemp){  cdISRC=cdISRCtemp;  }  **void** setCDPublisher(String cdPublishertemp){  cdPublisher=cdPublishertemp;  }  }  **package** 出版物;  **class** MyMedia {  String mediaName; //物品名称  **float** price; //物品价格  String press; //出版社(磁带发行商)  String artist; //作者(演唱者)  MyMedia(String nametemp,**float** pricetemp,String presstemp,String artisttemp){  mediaName=nametemp;  price=pricetemp;  press=presstemp;  artist=artisttemp;  }  String getName(){  **return** mediaName;  }  **float** getPrice(){  **return** price;  }  String getPress(){  **return** press;  }  String getArtist(){  **return** artist;  }  **void** setName(String nametemp){  mediaName=nametemp;  }  **void** setPrice(**float** pricetemp){  price=pricetemp;  }  **void** setPress(String presstemp){  press=presstemp;  }  **void** setArtist(String artisttemp){  artist=artisttemp;  }  }  **package** 出版物;  **class** MyTape **extends** MyMedia {  String tapeISRC; //磁带的ISRC  //构造函数  MyTape(String tn, **float** tp,String tpr,String ta,String ti){  **super**(tn,tp,tpr,ta);  tapeISRC = ti;  }  String getTapeISRC(){  **return** tapeISRC;  }  **void** setTapeISRC(String tapeISRCtemp){  tapeISRC=tapeISRCtemp;  }  }  package 出版物;  import java.awt.\*;  import java.awt.event.\*;  import javax.swing.\*;  class ApplicationFrame extends JFrame implements ActionListener{    /\*\*  \*  \*/  private static final long serialVersionUID = 1L;  private JLayeredPane desktop;  public ApplicationFrame(){  Container ContentPane=getContentPane();  desktop=new JDesktopPane();  desktop.setOpaque(false);  ContentPane.add(desktop ,BorderLayout.CENTER);  //设置菜单  JMenuBar mbar = new JMenuBar();  this.setJMenuBar(mbar);  JMenu appMenu=new JMenu("操作");  JMenuItem exitButton=new JMenuItem("退出");  exitButton.addActionListener(this);  appMenu.add(exitButton);  mbar.add(appMenu);    JMenu MediaMenu=new JMenu("出版物");  JMenuItem bookButton=new JMenuItem("书籍");  bookButton.addActionListener(this);  MediaMenu.add(bookButton);  JMenuItem cdButton=new JMenuItem("CD");  cdButton.addActionListener(this);  MediaMenu.add(cdButton);  JMenuItem tapeButton=new JMenuItem("磁带");  tapeButton.addActionListener(this);  MediaMenu.add(tapeButton);  mbar.add(MediaMenu);      }  public void actionPerformed(ActionEvent evt){  String s = evt.getActionCommand();  if(s.equals("退出"))  {  System.exit(0);  }  if(s.equals("书籍")){  //新建BookInternalFrame对象book  BookInternalFrame book=new BookInternalFrame();  //将book添加到主窗口  desktop.add(book,JLayeredPane.DEFAULT\_LAYER);  book.show();  }  if(s.equals("CD")){  //新建CDInternalFrame对象cd  CDInternalFrame cd=new CDInternalFrame();  //将cd添加到主窗口  desktop.add(cd,JLayeredPane.DEFAULT\_LAYER);  cd.show();  }  if(s.equals("磁带")){  //新建TapeInternalFrame对象tape  TapeInternalFrame tape=new TapeInternalFrame();  //将tape添加到主窗口  desktop.add(tape,JLayeredPane.DEFAULT\_LAYER);  tape.show();  }  if(s.equals("Metal")){  setExtendedState(JFrame.MAXIMIZED\_BOTH);  }  if(s.equals("Motif")){  setExtendedState(JFrame.ICONIFIED);  }  if(s.equals("Windows")){  System.exit(0);  }  }  }  package 出版物;  import java.awt.\*;  import javax.swing.\*;  import java.awt.event.\*;  import java.util.Date;  import java.text.\*;  class BookInternalFrame extends JInternalFrame implements ActionListener{  private JTextField JTFname; //书籍名称文本区  private JTextField JTFprice; //书籍价格文本区  private JTextField JTFpress; //书籍出版社文本区  private JTextField JTFartist; //书籍作者文本区  private JTextField JTFeditor; //书籍编辑文本区  private JTextField JTFisbn; //书籍ISBN文本区  private JTextField JTFdate; //书籍出版日期文本区  private JList JListBook; //书籍信息列表  private static final int MAX\_ARRAY=200;  MyBook BookInfo[]; //MyBook对象数组  int BookCount; //数组中对象的个数  BookInternalFrame(){  JTFname=new JTextField(20);  JTFprice=new JTextField(20);  JTFpress=new JTextField(20);  JTFartist=new JTextField(20);  JTFeditor=new JTextField(20);  JTFisbn=new JTextField(20);  JTFdate=new JTextField(20);  JListBook=new JList();  JListBook.setVisibleRowCount(6);  JListBook.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);  JScrollPane listpane=new JScrollPane(JListBook);  JPanel bookpane=new JPanel();  bookpane.setLayout(new GridLayout(7,2));  bookpane.add(new JLabel("书籍的名称：",JLabel.RIGHT));  bookpane.add(JTFname);  bookpane.add(new JLabel("书籍的价格：",JLabel.RIGHT));  bookpane.add(JTFprice);  bookpane.add(new JLabel("书籍的出版社：",JLabel.RIGHT));  bookpane.add(JTFpress);  bookpane.add(new JLabel("书籍的作者：",JLabel.RIGHT));  bookpane.add(JTFartist);  bookpane.add(new JLabel("书籍的编辑：",JLabel.RIGHT));  bookpane.add(JTFeditor);  bookpane.add(new JLabel("书籍的ISBN：",JLabel.RIGHT));  bookpane.add(JTFisbn);  bookpane.add(new JLabel("书籍的出版日期：",JLabel.RIGHT));  bookpane.add(JTFdate);  Container contentPane = getContentPane();  contentPane.setLayout(new BorderLayout());  contentPane.add(listpane,BorderLayout.CENTER);  contentPane.add(bookpane,BorderLayout.SOUTH);  //设置菜单  JMenuBar mbar=new JMenuBar();  JMenu opeMenu=new JMenu("操作");  JMenuItem addItem=new JMenuItem("增加");  addItem.addActionListener(this);  JMenuItem altItem=new JMenuItem("修改");  altItem.addActionListener(this);  JMenuItem delItem=new JMenuItem("删除");  delItem.addActionListener(this);  opeMenu.add(addItem);  opeMenu.add(altItem);  opeMenu.add(delItem);  mbar.add(opeMenu);  this.setJMenuBar(mbar);  setTitle("信息管理专业教材管理系统");  setSize(300,400);  setResizable(true);  setClosable(true);  setMaximizable(true);  setIconifiable(true);  //初始化BookInfo数组  BookInfo= new MyBook[MAX\_ARRAY];  BookCount=0;  for(int i=0;i<MAX\_ARRAY;i++)  BookInfo[i]=null;  }  public void actionPerformed(ActionEvent evt){  String s = evt.getActionCommand();  if(s.equals("增加")){  if(BookCount==MAX\_ARRAY)  {  JOptionPane.showMessageDialog(this,  "列表已经达到最大，无法增加新内容！",  "提示",JOptionPane.INFORMATION\_MESSAGE);  return;  }  //新建一个MyBook对象  MyBook booktemp=new MyBook("",0,"","","", s, null);  //设置MyBook对象中的各项  booktemp.setName(JTFname.getText());  try{  booktemp.setPrice(Float.parseFloat(JTFprice.getText()));  }  catch(NumberFormatException e)  {  JOptionPane.showMessageDialog(this,"价格输入格式错误！",  "错误",JOptionPane.INFORMATION\_MESSAGE);  return;  }  booktemp.setPress(JTFpress.getText());  booktemp.setArtist(JTFartist.getText());  booktemp.setEditor(JTFeditor.getText());  booktemp.setBookISBN(JTFisbn.getText());  DateFormat df = DateFormat.getDateInstance();  try{  booktemp.setPublishDate(df.parse(JTFdate.getText()));  }catch (Exception e) {  JOptionPane.showMessageDialog(this,"日期输入格式错误！",  "错误",JOptionPane.INFORMATION\_MESSAGE);  return;  }  BookInfo[BookCount]=booktemp;  BookCount++;  String list[]=new String[BookCount];  for(int j=0;j<BookCount;j++){  list[j]="名称："+BookInfo[j].getName()  +" 价格："+BookInfo[j].getPrice()  +" 出版社："+BookInfo[j].getPress()  +" 作者："+BookInfo[j].getArtist()  +" 编辑："+BookInfo[j].getEditor()  +" ISBN:"+BookInfo[j].getBookISBN()  +" 出版日期:"+BookInfo[j].getPublishDate();  }  JListBook.setListData(list);  }  if(s.equals("修改")){  //新建一个MyBook对象  MyBook booktemp=new MyBook("",0,"","","", s, null);  //修改MyBook对象中的各项  booktemp.setName(JTFname.getText());  try{  booktemp.setPrice(Float.parseFloat(JTFprice.getText()));  }  catch(NumberFormatException e){  JOptionPane.showMessageDialog(this,"价格输入格式错误！",  "错误",JOptionPane.INFORMATION\_MESSAGE);  }  booktemp.setPress(JTFpress.getText());  booktemp.setArtist(JTFartist.getText());  booktemp.setEditor(JTFeditor.getText());  booktemp.setBookISBN(JTFisbn.getText());  DateFormat df = DateFormat.getDateInstance();  try{  booktemp.setPublishDate(df.parse(JTFdate.getText()));  }catch (Exception e) {  JOptionPane.showMessageDialog(this,"日期输入格式错误！",  "错误",JOptionPane.INFORMATION\_MESSAGE);  }  BookInfo[JListBook.getSelectedIndex()]=booktemp;  //刷新列表中的各项  String list[]=new String[BookCount];  for(int j=0;j<BookCount;j++){  list[j]="名称："+BookInfo[j].getName()  +" 价格："+BookInfo[j].getPrice()  +" 出版社："+BookInfo[j].getPress()  +" 作者："+BookInfo[j].getArtist()  +" 编辑："+BookInfo[j].getEditor()  +" ISBN:"+BookInfo[j].getBookISBN()  +" 出版日期:"+BookInfo[j].getPublishDate();  }  JListBook.setListData(list);  }  if(s.equals("删除")){  //删除选中的项，将后面的各项依次向前递推  for(int i=JListBook.getSelectedIndex();i<BookCount-1;i++)  {  BookInfo[i]=BookInfo[i+1];  }  BookCount--;  //刷新列表中的各项  String list[]=new String[BookCount];  for(int j=0;j<BookCount;j++){  list[j]="名称："+BookInfo[j].getName()  +" 价格："+BookInfo[j].getPrice()  +" 出版社："+BookInfo[j].getPress()  +" 作者："+BookInfo[j].getArtist()  +" 编辑："+BookInfo[j].getEditor()  +" ISBN:"+BookInfo[j].getBookISBN()  +" 出版日期:"+BookInfo[j].getPublishDate();  }  JListBook.setListData(list);  }  }}  package 出版物;  import java.awt.\*;  import javax.swing.\*;  import java.awt.event.\*;  import java.util.Date;  import java.text.\*;  class CDInternalFrame extends JInternalFrame implements ActionListener{  private JTextField JTFname; //CD名称文本区  private JTextField JTFprice; //CD价格文本区  private JTextField JTFpress; //CD出版社文本区  private JTextField JTFartist; //CD演唱者文本区  private JTextField JTFpublisher; //CD发行商文本区  private JTextField JTFisrc; //CDISRC文本区  private JList JListCD; //CD信息列表  private static final int MAX\_ARRAY=200;  MyCD CDInfo[]; //MyCD对象数组  int CDCount; //数组中对象的个数  CDInternalFrame(){  JTFname=new JTextField(20);  JTFprice=new JTextField(20);  JTFpress=new JTextField(20);  JTFartist=new JTextField(20);  JTFpublisher=new JTextField(20);  JTFisrc=new JTextField(20);  JListCD=new JList();  JListCD.setVisibleRowCount(6);  JListCD.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);  JScrollPane listpane=new JScrollPane(JListCD);  JPanel cdpane=new JPanel();  cdpane.setLayout(new GridLayout(6,2));  cdpane.add(new JLabel("CD的名称：",JLabel.RIGHT));  cdpane.add(JTFname);  cdpane.add(new JLabel("CD的价格：",JLabel.RIGHT));  cdpane.add(JTFprice);  cdpane.add(new JLabel("CD的出版社：",JLabel.RIGHT));  cdpane.add(JTFpress);  cdpane.add(new JLabel("CD的演唱者：",JLabel.RIGHT));  cdpane.add(JTFartist);  cdpane.add(new JLabel("CD的发行商：",JLabel.RIGHT));  cdpane.add(JTFpublisher);  cdpane.add(new JLabel("CD的ISRC：",JLabel.RIGHT));  cdpane.add(JTFisrc);  Container contentPane = getContentPane();  contentPane.setLayout(new BorderLayout());  contentPane.add(listpane,BorderLayout.CENTER);  contentPane.add(cdpane,BorderLayout.SOUTH);  //设置菜单  JMenuBar mbar=new JMenuBar();  JMenu opeMenu=new JMenu("操作");  JMenuItem addItem=new JMenuItem("增加");  addItem.addActionListener(this);  JMenuItem altItem=new JMenuItem("修改");  altItem.addActionListener(this);  JMenuItem delItem=new JMenuItem("删除");  delItem.addActionListener(this);  opeMenu.add(addItem);  opeMenu.add(altItem);  opeMenu.add(delItem);  mbar.add(opeMenu);  this.setJMenuBar(mbar);  setTitle("CD");  setSize(300,400);  setResizable(true);  setClosable(true);  setMaximizable(true);  setIconifiable(true);  //初始化CDInfo数组  CDInfo= new MyCD[MAX\_ARRAY];  CDCount=0;  for(int i=0;i<MAX\_ARRAY;i++)  CDInfo[i]=null;  }  public void actionPerformed(ActionEvent evt){  String s = evt.getActionCommand();  if(s.equals("增加")){  if(CDCount==MAX\_ARRAY){  JOptionPane.showMessageDialog(this,  "列表已经达到最大，无法增加新内容！","提示",JOptionPane.INFORMATION\_MESSAGE);  return;  }  //新建一个MyCD对象  MyCD cdtemp=new MyCD("",0,"","","","");  //设置MyCD对象中的各项  cdtemp.setName(JTFname.getText());  try{  cdtemp.setPrice(Float.parseFloat(JTFprice.getText()));  }  catch(NumberFormatException e){  JOptionPane.showMessageDialog(this,"价格输入格式错误！",  "错误",JOptionPane.INFORMATION\_MESSAGE);  return;  }  cdtemp.setPress(JTFpress.getText());  cdtemp.setArtist(JTFartist.getText());  cdtemp.setCDPublisher(JTFpublisher.getText());  cdtemp.setCDISRC(JTFisrc.getText());  CDInfo[CDCount]=cdtemp;  CDCount++;  String list[]=new String[CDCount];  for(int j=0;j<CDCount;j++){  list[j]="名称："+CDInfo[j].getName()  +" 价格："+CDInfo[j].getPrice()  +" 出版社："+CDInfo[j].getPress()  +" 演唱者："+CDInfo[j].getArtist()  +" 发行商："+CDInfo[j].getCDPublisher()  +" ISRC:"+CDInfo[j].getCDISRC();  }  JListCD.setListData(list);  }  if(s.equals("修改")){  //新建一个MyCD对象  MyCD cdtemp=new MyCD("",0,"","","","");  //修改MyCD对象中的各项  cdtemp.setName(JTFname.getText());  try{  cdtemp.setPrice(Float.parseFloat(JTFprice.getText()));  }  catch(NumberFormatException e){  JOptionPane.showMessageDialog(this,"价格输入格式错误！",  "错误",JOptionPane.INFORMATION\_MESSAGE);  }  cdtemp.setPress(JTFpress.getText());  cdtemp.setArtist(JTFartist.getText());  cdtemp.setCDPublisher(JTFpublisher.getText());  cdtemp.setCDISRC(JTFisrc.getText());  CDInfo[JListCD.getSelectedIndex()]=cdtemp;  //刷新列表中的各项  String list[]=new String[CDCount];  for(int j=0;j<CDCount;j++){  list[j]="名称："+CDInfo[j].getName()  +" 价格："+CDInfo[j].getPrice()  +" 出版社："+CDInfo[j].getPress()  +" 演唱者："+CDInfo[j].getArtist()  +" 发行商："+CDInfo[j].getCDPublisher()  +" ISRC:"+CDInfo[j].getCDISRC();  }  JListCD.setListData(list);  }  if(s.equals("删除")){  //删除选中的项，将后面的各项依次向前递推  for(int i=JListCD.getSelectedIndex();i<CDCount-1;i++){  CDInfo[i]=CDInfo[i+1];  }  CDCount--;  //刷新列表中的各项  String list[]=new String[CDCount];  for(int j=0;j<CDCount;j++){  list[j]="名称："+CDInfo[j].getName()  +" 价格："+CDInfo[j].getPrice()  +" 出版社："+CDInfo[j].getPress()  +" 演唱者："+CDInfo[j].getArtist()  +" 发行商："+CDInfo[j].getCDPublisher()  +" ISRC:"+CDInfo[j].getCDISRC();  }  JListCD.setListData(list);  }  }  }  package 出版物;  import java.awt.\*;  import javax.swing.\*;  import java.awt.event.\*;  import java.util.Date;  import java.text.\*;  class TapeInternalFrame extends JInternalFrame implements ActionListener{  private JTextField JTFname; //Tape名称文本区  private JTextField JTFprice; //Tape价格文本区  private JTextField JTFpress; //Tape出版社文本区  private JTextField JTFartist; //Tape演唱者文本区  private JTextField JTFisrc; //TapeISRC文本区  private JList JListTape; //Tape信息列表  private static final int MAX\_ARRAY=200;  MyTape TapeInfo[]; //MyTape对象数组  int TapeCount; //数组中对象的个数  TapeInternalFrame(){  JTFname=new JTextField(20);  JTFprice=new JTextField(20);  JTFpress=new JTextField(20);  JTFartist=new JTextField(20);  JTFisrc=new JTextField(20);  JListTape=new JList();  JListTape.setVisibleRowCount(6);  JListTape.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);  JScrollPane listpane=new JScrollPane(JListTape);  JPanel tapepane=new JPanel();  tapepane.setLayout(new GridLayout(5,2));  tapepane.add(new JLabel("Tape的名称：",JLabel.RIGHT));  tapepane.add(JTFname);  tapepane.add(new JLabel("Tape的价格：",JLabel.RIGHT));  tapepane.add(JTFprice);  tapepane.add(new JLabel("Tape的出版社：",JLabel.RIGHT));  tapepane.add(JTFpress);  tapepane.add(new JLabel("Tape的演唱者：",JLabel.RIGHT));  tapepane.add(JTFartist);  tapepane.add(new JLabel("Tape的ISRC：",JLabel.RIGHT));  tapepane.add(JTFisrc);  Container contentPane = getContentPane();  contentPane.setLayout(new BorderLayout());  contentPane.add(listpane,BorderLayout.CENTER);  contentPane.add(tapepane,BorderLayout.SOUTH);  //设置菜单  JMenuBar mbar=new JMenuBar();  JMenu opeMenu=new JMenu("操作");  JMenuItem addItem=new JMenuItem("增加");  addItem.addActionListener(this);  JMenuItem altItem=new JMenuItem("修改");  altItem.addActionListener(this);  JMenuItem delItem=new JMenuItem("删除");  delItem.addActionListener(this);  opeMenu.add(addItem);  opeMenu.add(altItem);  opeMenu.add(delItem);  mbar.add(opeMenu);  this.setJMenuBar(mbar);  setTitle("Tape");  setSize(300,400);  setResizable(true);  setClosable(true);  setMaximizable(true);  setIconifiable(true);  //初始化TapeInfo数组  TapeInfo= new MyTape[MAX\_ARRAY];  TapeCount=0;  for(int i=0;i<MAX\_ARRAY;i++)  TapeInfo[i]=null;  }  public void actionPerformed(ActionEvent evt){  String s = evt.getActionCommand();  if(s.equals("增加")){  if(TapeCount==MAX\_ARRAY){  JOptionPane.showMessageDialog(this,  "列表已经达到最大，无法增加新内容！","提示",JOptionPane.INFORMATION\_MESSAGE);  return;  }  //新建一个MyTape对象  MyTape tapetemp=new MyTape("",0,"","","");  //设置MyTape对象中的各项  tapetemp.setName(JTFname.getText());  try{  tapetemp.setPrice(Float.parseFloat(JTFprice.getText()));  }  catch(NumberFormatException e){  JOptionPane.showMessageDialog(this,"价格输入格式错误！",  "错误",JOptionPane.INFORMATION\_MESSAGE);  return;  }  tapetemp.setPress(JTFpress.getText());  tapetemp.setArtist(JTFartist.getText());  tapetemp.setTapeISRC(JTFisrc.getText());  TapeInfo[TapeCount]=tapetemp;  TapeCount++;  String list[]=new String[TapeCount];  for(int j=0;j<TapeCount;j++){  list[j]="名称："+TapeInfo[j].getName()  +" 价格："+TapeInfo[j].getPrice()  +" 出版社："+TapeInfo[j].getPress()  +" 演唱者："+TapeInfo[j].getArtist()  +" ISRC:"+TapeInfo[j].getTapeISRC();  }  JListTape.setListData(list);  }  if(s.equals("修改")){  //新建一个MyTape对象  MyTape tapetemp=new MyTape("",0,"","","");  //修改MyTape对象中的各项  tapetemp.setName(JTFname.getText());  try{  tapetemp.setPrice(Float.parseFloat(JTFprice.getText()));  }  catch(NumberFormatException e){  JOptionPane.showMessageDialog(this,"价格输入格式错误！",  "错误",JOptionPane.INFORMATION\_MESSAGE);  }  tapetemp.setPress(JTFpress.getText());  tapetemp.setArtist(JTFartist.getText());  tapetemp.setTapeISRC(JTFisrc.getText());  TapeInfo[JListTape.getSelectedIndex()]=tapetemp;  //刷新列表中的各项  String list[]=new String[TapeCount];  for(int j=0;j<TapeCount;j++){  list[j]="名称："+TapeInfo[j].getName()  +" 价格："+TapeInfo[j].getPrice()  +" 出版社："+TapeInfo[j].getPress()  +" 演唱者："+TapeInfo[j].getArtist()  +" ISRC:"+TapeInfo[j].getTapeISRC();  }  JListTape.setListData(list);  }  if(s.equals("删除")){  //删除选中的项，将后面的各项依次向前递推  for(int i=JListTape.getSelectedIndex();i<TapeCount-1;i++){  TapeInfo[i]=TapeInfo[i+1];  }  TapeCount--;  //刷新列表中的各项  String list[]=new String[TapeCount];  for(int j=0;j<TapeCount;j++){  list[j]="名称："+TapeInfo[j].getName()  +" 价格："+TapeInfo[j].getPrice()  +" 出版社："+TapeInfo[j].getPress()  +" 演唱者："+TapeInfo[j].getArtist()  +" ISRC:"+TapeInfo[j].getTapeISRC();  }  JListTape.setListData(list);  }  }  }  **package** 出版物;  **import** javax.swing.JFrame;  **public** **class** Application{  **public** **static** **void** main(String[] args){  JFrame jf=**new** ApplicationFrame();  jf.~~show~~();  }  } 结果: IMG_256  IMG_256    **增加：**    删除    修改：     1. **出现的问题及解决方法**   对于图形界面的使用的不熟悉，以至于有能多看不懂的错误，和对于图形界面的构造的不明白，经过观看书本和雪梨老师的视频课程了解到了一些关于图形化界面的知识，然后在网络上查找了一些资料，和原先就对本次的程序有点熟悉，因此完成本次程序。   1. **实验总结**   对于本次实验，我了解到了程序的图形化界面是很有趣的，没有原来那么无聊的，学习到了如何把一个程序形象化的变成图形化界面。 |