

**《Java实习报告》**



|  |  |
| --- | --- |
| 学 院 | **信息工程学院** |
| 专业名称 | **计算机科学与技术** |
| 班 级 | **101011901** |
| 学 号 | **191027** |
| 学生姓名 | **杨乃宸** |
| 成 绩 |  |

|  |  |  |
| --- | --- | --- |
| **项目一：家庭记账软件** | **得分** |  |
| 1. **实验内容**   **程序有一个相对美观的简约界面。**  **软件操作简易能记录家庭的收入和支出，打印收支明细表。**  **功能采用下拉式菜单编写。**   1. **问题描述**   **后台控制器显示主窗口状态。**  **操作内容均有详细提示。**  **操作时打印组件相关功能，便于维护。**  **登记内容须为浮点型正数，每次记账抹零。**  **输入不合法时弹窗会进行一定程度的异常处理。**   1. **项目功能**   **键盘默选首个控件。**  **计算账户当前余额。**  **显示连续操作次数。**  **反复记录收支类别。**  **账单记录具体金额。**  **记账内容备注。**  **累计账户查询。**  **退出。**   1. **项目分析**     **程序运行 菜单里点按钮（如图）**    **收入登记1000元劳务费 显示一些内容（如图）**  **支出登记2000元（如图）物业费**  **查明细界面**    **退出**   1. **项目设计**   **开项目继承→规划各变量→初始化窗体容器→组件调试→事件处理→布局管理→实现菜单→对话框→删添包→添加一些个性化想法→修改外观及文本内容→反复检查是否符合项目报告要求**   1. **系统实现**   **package Software;**  **import java.awt.Color;**  **import java.awt.Font;**  **import java.awt.event.ActionEvent;**  **import java.awt.event.ActionListener;**  **import java.awt.event.WindowEvent;**  **import java.awt.event.WindowListener;**  **import javax.swing.BoxLayout;**  **import javax.swing.JFrame;**  **import javax.swing.JLabel;**  **import javax.swing.JMenu;**  **import javax.swing.JMenuBar;**  **import javax.swing.JMenuItem;**  **import javax.swing.JOptionPane;**  **import javax.swing.JPanel;**  **import javax.swing.SwingConstants;**  **import javax.swing.WindowConstants;**  **@SuppressWarnings("serial")**  **public class Projc01 extends JFrame {**  **String Details [] = new String [ 10000 ] ;**  **int DLC ;**  **public static void main ( String [] args ) {**  **Projc01 P1 = new Projc01 ( ) ;**  **P1.Details [ 0 ] = " 收 支 账户金额 收支金额 说 明 \n" ;**  **P1.DLC = 0 ;**  **int Balance = 0 ;**  **int [ ] Income = new int [ 10000 ] ;**  **Income [ Balance ] = 10000 ;**  **Color Rainbow = new Color ( 100 , 200 , 250 ) ; // 定义一部分常用的量**  **// String Choice [] = { "a" , "b" , "c" } ;**  **JFrame JF1 = new JFrame (" -----家庭收支记账软件----- ") ; // 整块 框架Win**  **// ButtonGroup BG1 = new ButtonGroup ( ) ;**  **// ButtonGroup BG2 = new ButtonGroup ( ) ;**  **Font KyrJL = new Font ( "幼圆" , Font.BOLD+Font.ITALIC , 50 ) ;**  **Font KyrJTF = new Font ( "新宋体" , Font.PLAIN , 40 ) ;**  **Font KyrJTA = new Font ( "新宋体" , Font.PLAIN , 30 ) ;**    **JMenuBar JMB1 = new JMenuBar ( ) ;**  **JF1.setJMenuBar ( JMB1 ) ;**  **JMB1.setLayout ( new BoxLayout ( JMB1 , BoxLayout.X\_AXIS ) ) ;**  **JMenu JM1 = new JMenu ( " 菜单 " ) ; // Action a 菜单属性由相应动作提供**  **JMenu JM2 = new JMenu ( " 作者留言 " ) ;**  **JMenuItem JMI1 = new JMenuItem ( " 收支明细 " ) ; // 平平无奇按钮控件**  **JMenuItem JMI2 = new JMenuItem ( " 登记收入 " ) ;**  **JMenuItem JMI3 = new JMenuItem ( " 登记支出 " ) ;**  **JMenuItem JMI4 = new JMenuItem ( " 退 出 " ) ;**  **JMenuItem JMI5 = new JMenuItem ( "暑假快乐!" ) ;**  **// JCheckBoxMenuItem JCBMI1 =new JCheckBoxMenuItem ( "多选" ) ;**  **// JRadioButtonMenuItem JRBMI1 = new JRadioButtonMenuItem ( " 单选1 " ) ;**  **// JRadioButtonMenuItem JRBMI2 = new JRadioButtonMenuItem ( " 单选2 " ) ;**    **class WindowMonitor implements WindowListener { // Adapterで**  **public void windowOpened(WindowEvent e) {**  **System.out.println ( " 开了 " ) ;**  **}**  **public void windowClosing(WindowEvent e) {**  **System.out.println ( " 关了 " ) ;**  **JF1.dispose () ;**  **}**  **public void windowClosed(WindowEvent e) {**  **// System.out.println ( " 关闭后执行 " ) ; // 多线程才能执行..**  **}**  **public void windowActivated(WindowEvent e) {**  **System.out.println ( " 选 " ) ;**  **}**  **public void windowDeactivated(WindowEvent e) {**  **System.out.print ( " 锁 " ) ;**  **}**  **public void windowIconified(WindowEvent e) {**  **System.out.print ( " 隐 " ) ;**  **}**  **public void windowDeiconified(WindowEvent e) {**  **System.out.print ( " 显 " ) ;**  **}**  **}**  **WindowMonitor WM = new WindowMonitor ( ) ;**  **JF1.addWindowListener ( WM ) ; // F**    **JF1.setSize ( 400 , 400 ) ;**  **JF1.setLocationRelativeTo ( null ) ; //setLocation ( 300 , 200 ) ;**  **JF1.setDefaultCloseOperation ( WindowConstants.EXIT\_ON\_CLOSE ) ;**  **JF1.setVisible (true) ;**  **JPanel JPn1 = new JPanel ( ) ; // 大块套娃面板**  **JPn1.setBackground ( Rainbow ) ; // 多用**    **// JPn1.setLayout ( null ) ; // .setBounds ( x , y , w ,h ) ;**  **// JPn1.setLayout ( new CardLayout ( 50 , 100 ) ) ;**  **// JPn1.setLayout ( new BoxLayout ( JPn1 , BoxLayout.Y\_AXIS ) ) ;**  **// JPn1.setLayout ( new FlowLayout ( FlowLayout.CENTER , 50 , 100 ) ) ; // JPanel デフォールト**  **// JPn1.setLayout ( new GridLayout ( 3 , 5 , xgap , ygap ) ) ; // マキシマムで**  **// JPn1.setLayout ( new BorderLayout ( 50 , 100 ) ) ; // JFrame、JDialog デフォールト**  **// PS : JPn1 ( Object , BorderLayout.CENTER ) ;**    **JLabel JL1 = new JLabel ( " 家庭记账 " , SwingConstants.CENTER) ;**  **// JPasswordField JPs1 = new JPasswordField ( " 请输入密码 " , 20 ) ;**  **// JPs1.setEchoChar ( '\_' ) ;**  **// JPs1.setText( " Please input Password ! " ) ;**  **// JTextField JTF1 = new JTextField ( " 请输入一行文本内容 " , 20 ) ;**  **// JTextArea JTA1 = new JTextArea ( " 请输入几行文本内容 " , 10 , 20 ) ;**  **// JTA1.append ( "追加内容" ) ;**  **// JButton JB1 = new JButton ( " 按钮 " ) ;**  **// JComboBox < String > JCBox = new JComboBox < > ( Choice ) ;**  **// JRadioButton JRB1 = new JRadioButton ( "是" , true ) ;**  **// JRadioButton JRB2 = new JRadioButton ( "否" , false ) ;**  **// JCheckBox JCB1 = new JCheckBox ( "①" , false ) ;**  **// JCheckBox JCB2 = new JCheckBox ( "②" , false ) ;**  **// JCheckBox JCB3 = new JCheckBox ( "③" , false ) ;**  **/\* JDialog JD1 = new JDialog ( JF1 , " 对话框 " , true ) ;**  **JD1.setSize ( 400 , 200 ) ;**  **JD1.setLocationRelativeTo ( null ) ;**  **// JD1.add ( JOptionPane ) ;**  **JD1.setVisible(true) ; // 可见性写最后**  **\*/**  **JL1.setFont ( KyrJL ) ; JMI5.setFont ( KyrJL ) ; // 字体设定**  **JM1.setFont ( KyrJTF ) ; JM2.setFont ( KyrJTF ) ;**  **JMI1.setFont ( KyrJTA ) ; JMI2.setFont ( KyrJTA ) ; JMI3.setFont ( KyrJTA ) ; JMI4.setFont ( KyrJTA ) ;**  **// JTF1.setFont ( KyrJTF ) ;**  **// JTA1.setFont ( KyrJTA ) ;**    **JF1.add ( JPn1 ) ; // 添来添去**  **JMB1.add ( JM1 ) ; JM1.add ( JMI1 ) ; // JM1.add ( JCBMI1 ) ;**  **JM1.add ( JMI2 ) ; JM1.add ( JMI3 ) ; JM1.add ( JMI4 ) ;**  **JMB1.add ( JM2 ) ; JM2.add ( JMI5 ) ; JM2.addSeparator () ; // 分离器——————**  **JPn1.add ( JL1 ) ; // JPn1.add ( JPs1 ) ; JPn1.add ( JTF1 ) ;**  **// JPn1.add ( JTA1 ) ; JPn1.add ( JB1 ) ; JPn1.add ( JCBox ) ;**  **// JPn1.add ( JRB1 ) ; JPn1.add ( JRB2 ) ; // JM1.add ( JRBMI1 ) ; JM1.add ( JRBMI2 ) ;**  **// BG1.add ( JRB1 ) ; BG1.add ( JRB2 ) ; // BG2.add ( JRBMI1 ) ; BG2.add ( JRBMI2 ) ;**  **// JPn1.add ( JCB1 ) ; JPn1.add ( JCB2 ) ; JPn1.add ( JCB3 ) ;**    **// ① ——————————**  **/\* class ItemMonitor implements ItemListener {**  **public void itemStateChanged ( ItemEvent IE ) {**  **if ( JItemSource.isSelected ( ) )**  **System.out.println ( " ON " ) ;**  **else**  **System.out.println ( " OFF " ) ;**  **}**  **}**  **ItemMonitor IM = new ItemMonitor ( ) ; // Just cαn be AnonymousClass**  **JItemSource.addItemListener ( IM ) ; // B/CBox/RB/CB 委托监听**  **\*/**  **// ② ——————————**  **class ActionMonitor02 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **int Times = 0 ;**  **while ( 1 != 0 ) {**  **int SelectWh = JOptionPane.showConfirmDialog ( null , " 当前账户余额 : " + Income [ Balance + Times ] + "\n 本次操作累计 : " + Times + "次 \n 确认登记吗？ " , " 登记收入 " , JOptionPane.OK\_CANCEL\_OPTION ) ; // o=y=yn**  **if ( SelectWh == JOptionPane.OK\_OPTION ) {**  **Object Select01 = JOptionPane.showInputDialog ( null , " 本次收入金额 : " , " 登记收入 " , JOptionPane.QUESTION\_MESSAGE , null , null , " 输入值 " ) ; // d = OC ERR 可为组合框**  **double Sel01 = Double.valueOf ( Select01.toString ( ) ) ;**  **if ( Select01 != null && Sel01 > 0 ) {**  **System.out.println ( " 本次收入金额 : " + Select01 ) ;**  **Income [ Balance + Times + 1 ] = (int) (Income [ Times ] + Sel01) ;**  **Object Select02 = JOptionPane.showInputDialog ( null , " 本次收入说明 : " , " 登记收入 " , JOptionPane.INFORMATION\_MESSAGE , null , null , " \_\_\_费 " ) ;**  **System.out.println ( " 本次收入说明 : " + Select02 ) ;**  **P1.Details [ P1.DLC + 1 ] = P1.Details [ P1.DLC ] + " 收 入 " + Income [ Balance + Times + 1 ] + " " + Select01 + " " + Select02 + "\n" ;**  **System.out.println ( P1.Details [ P1.DLC + 1 ] ) ;**  **P1.DLC ++ ;**  **Times ++ ;**  **}**  **else {**  **System.out.println ( " 用户输入错误内容 : " + Select01 ) ;**  **JOptionPane.showInputDialog ( null , " 莫得输入正确 " , " 警告信息 " , JOptionPane.ERROR\_MESSAGE , null , null , " 重新输入 " ) ; // d = OC ERR**  **}**  **}**  **else {**  **Income [ Balance ] = (int) (Income [ Balance + Times ]) ;**  **break ;**  **}**  **}**  **}**  **}**  **ActionMonitor02 AM02 = new ActionMonitor02 ( ) ;**  **JMI2.addActionListener ( AM02 ) ; // F Pn L Ps/TF TA B/CBox/RB/CB**    **class ActionMonitor03 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **int Times = 0 ;**  **while ( 1 != 0 ) {**  **int SelectWh = JOptionPane.showConfirmDialog ( null , " 当前账户余额 : " + Income [ Balance + Times ] + "\n 本次操作累计 : " + Times + "次 \n 确认登记吗？ " , " 登记支出 " , JOptionPane.OK\_CANCEL\_OPTION ) ; // o=y=yn**  **if ( SelectWh == JOptionPane.OK\_OPTION ) {**  **Object Select01 = JOptionPane.showInputDialog ( null , " 本次支出金额 : " , " 登记支出 " , JOptionPane.QUESTION\_MESSAGE , null , null , " 输入值 " ) ; // d = OC ERR 可为组合框**  **double Sel01 = Double.valueOf ( Select01.toString ( ) ) ;**  **if ( Select01 != null && Sel01 > 0 ) {**  **System.out.println ( " 本次支出金额 : " + Select01 ) ;**  **Income [ Times + 1 ] = (int) (Income [ Times ] - Sel01) ;**  **Object Select02 = JOptionPane.showInputDialog ( null , " 本次支出说明 : " , " 登记支出 " , JOptionPane.INFORMATION\_MESSAGE , null , null , " \_\_\_费 " ) ;**  **System.out.println ( " 本次支出说明 : " + Select02 ) ;**  **P1.Details [ P1.DLC + 1 ] = P1.Details [ P1.DLC ] + " 支 出 " + Income [ Balance + Times + 1 ] + " " + Select01 + " " + Select02 + " \n" ;**  **System.out.println ( P1.Details [ P1.DLC + 1 ] ) ;**  **P1.DLC ++ ;**  **Times ++ ;**  **}**  **else {**  **System.out.println ( " 用户输入错误内容 : " + Select01 ) ;**  **JOptionPane.showInputDialog ( null , " 莫得输入正确 " , " 警告信息 " , JOptionPane.ERROR\_MESSAGE , null , null , " 重新输入 " ) ; // d = OC ERR**  **}**  **}**  **else {**  **Income [ Balance ] = (int) (Income [ Balance + Times ]) ;**  **break ;**  **}**  **}**  **}**  **}**  **ActionMonitor03 AM03 = new ActionMonitor03 ( ) ;**  **JMI3.addActionListener ( AM03 ) ;**    **class ActionMonitor01 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **JOptionPane.showMessageDialog ( null , " 账户查询 : \n" + P1.Details [ P1.DLC ] , " -----当前收支明细记录----- " , JOptionPane.INFORMATION\_MESSAGE ) ; // null=JF Array/Comp/Icon**  **}**  **}**  **ActionMonitor01 AM01 = new ActionMonitor01 ( ) ;**  **JMI1.addActionListener ( AM01 ) ;**    **class ActionMonitor04 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **int SelectWh = JOptionPane.showOptionDialog ( null , " 要关闭吗？ " , " 退出 " , JOptionPane.YES\_NO\_OPTION , JOptionPane.QUESTION\_MESSAGE , null , null , null ) ; // 图标默认 op=CompString/Icon**  **if ( SelectWh == JOptionPane.YES\_OPTION ) {**  **JF1.setVisible(false);**  **}**  **}**  **}**  **ActionMonitor04 AM04 = new ActionMonitor04 ( ) ;**  **JMI4.addActionListener ( AM04 ) ;**  **// ③ ——————————**  **/\* class FocusMonitor implements FocusListener {**  **public void focusGained ( FocusEvent FE ) { // If Event = = > match Processor**  **System.out.println ( " 聚焦 " ) ;**  **}**  **public void focusLost ( FocusEvent FE ) {**  **System.out.println ( " 失焦 " ) ;**  **}**  **}**  **FocusMonitor FM = new FocusMonitor ( ) ;**  **JItemSource.addFocusListener ( FM ) ;**  **\*/**  **// ④ ——————————**  **/\* class KeyMonitor implements KeyListener {**  **public void keyPressed ( KeyEvent KE ) { // 瞬**  **System.out.println ( " 输入 : " ) ;**  **}**  **public void keyTyped ( KeyEvent KE ) { // 过程**  **System.out.println ( KE.getKeyChar ( ) ) ;**  **}**  **public void keyReleased ( KeyEvent KE ) { // 瞬**  **System.out.println ( " \_ " ) ;**  **}**  **}**  **KeyMonitor KM = new KeyMonitor ( ) ;**  **JItemSource.addKeyListener ( KM ) ;**  **\*/**  **// ⑤ ——————————**  **/\* class MouseMotionMonitor implements MouseMotionListener {**  **public void mouseMoved ( MouseEvent ME ) {**  **System.out.println ( " Hold " ) ;**  **}**  **public void mouseDragged ( MouseEvent ME ) {**  **System.out.println ( " Drag X : " + ME.getX ( ) + " Y : " + ME.getY ( ) ) ;**  **}**  **}**  **class MouseMonitor implements MouseListener {**  **public void mouseEntered ( MouseEvent Me ) {**  **System.out.print ( " GO " ) ;**  **}**  **public void mouseExited ( MouseEvent Me ) {**  **System.out.println ( " FULL COMBO " ) ;**  **}**  **public void mousePressed ( MouseEvent Me ) {**  **System.out.print ( " Tap " ) ;**  **}**  **public void mouseReleased ( MouseEvent Me ) {**  **System.out.print ( " FLick " ) ;**  **}**  **public void mouseClicked ( MouseEvent Me ) {**  **System.out.println ( " Perfect " ) ;**  **if (Me.getClickCount ( ) > 4) {**  **System.out.println ( Me.getClickCount ( ) + " Combo " ) ;**  **}**  **}**  **}**  **MouseMotionMonitor MM = new MouseMotionMonitor ( ) ;**  **JF1.addMouseMotionListener ( MM ) ;**  **MouseMonitor Mm = new MouseMonitor ( ) ;**  **JF1.addMouseListener ( Mm ) ;**  **\*/**    **/\* .getHgap ( ) ; // 保留方法**  **.getVgap ( ) ;**  **FL.getAlignment ( ) ;**  **GL.getRows ( ) ;**  **GL.getColumns ( ) ;**  **CL.first/last/next/previous/remove ( Component ) ;**  **CL.addLayoutComponent/show ( Component , Name ) ;**    **JPs.char[] getPassword ( ) ;**  **JF.getContentPane ( ) ;**  **JD.getTitle ( ) ;**  **JTF.getSelectedText ( ) ;**  **JTF.getColumns ( ) ;**  **JTA.getText ( ) ;**  **JTA.getColumns ( ) ;**  **JTA.getRows ( ) ;**  **JTA.getColumnWidth ( ) ;**  **JTA.getLineCount ( ) ;**  **JRB.getText ( ) ;**  **JRB.isSelected ( ) ;**  **JRB.doClick ( ) ;**  **JCB.getText ( ) ;**  **JCB.isSelected ( ) ;**  **JCB.doClick ( ) ;**  **JCBox.getSelectedItem ( ) ;**  **JCBox.getItemCount ( ) ;**  **JCBox.addItem ( Object ) ;**  **JCBox.setSelectedItem ( Object ) ;**  **JCBox.removeItem ( Object ) ;**  **JCBox.insertItemAt ( Object , Index ) ;**  **\*/**  **}**  **}**  **七、总结**  **问题：明细的值带不走 过去以后 一出监听器就初始化**  **心得：勤于尝试 善于思考 全心全意 坚持不懈** | | |
| **项目二：简单文本编辑器** | **得分** |  |
| 1. **实验内容**   **界面颜色渐变 控件融入布局**  **按钮设置字体三种大小：10 30 50**  **按钮设置字体三种颜色：红黄蓝**  **滑动调节字体大小 范围：1-100**   1. **问题描述**   **两种方式简单设置多行文本内容样式**  **文档后台恢复**   1. **项目功能**   **不同文本均有默认字体**  **默认字体大小30 字体颜色红色**  **滑块默认位置50**  **显示自定义滑块的标签和轨道**   1. **项目分析**   **初始界面**    **小号蓝字**    **最大号文本 显示光标**     1. **项目设计**   **导入通配包→分组定义变量→想象草图→框架布局→添加控件→申请监听→补充方法设定→通配包换成对应包→反复检查是否符合项目报告要求→删掉多余的功能**   1. **系统实现**   **package Software;**  **import java.awt.Color;**  **import java.awt.Font;**  **import java.awt.GridLayout;**  **import java.awt.event.ActionEvent;**  **import java.awt.event.ActionListener;**  **import java.awt.event.MouseEvent;**  **import java.awt.event.MouseMotionListener;**  **import java.awt.event.WindowEvent;**  **import java.awt.event.WindowListener;**  **import java.util.Hashtable;**  **import javax.swing.BoxLayout;**  **import javax.swing.ButtonGroup;**  **import javax.swing.JComponent;**  **import javax.swing.JFrame;**  **import javax.swing.JLabel;**  **import javax.swing.JPanel;**  **import javax.swing.JRadioButton;**  **import javax.swing.JSlider;**  **import javax.swing.JTextArea;**  **import javax.swing.SwingConstants;**  **import javax.swing.WindowConstants;**  **@SuppressWarnings("serial")**  **public class Projc02 extends JFrame {**  **public static void main ( String [] args ) {**  **Color Rainbow1 = new Color ( 230 , 250 , 250 ) ; // 定义一部分常用的量**  **Color Rainbow2 = new Color ( 235 , 250 , 250 ) ;**  **Color Rainbow3 = new Color ( 240 , 250 , 250 ) ;**  **Color Rainbow4 = new Color ( 245 , 250 , 250 ) ;**  **// String Choice [] = { "a" , "b" , "c" } ;**  **JFrame JF1 = new JFrame (" 简单文本编辑器 ") ; // 整块**    **ButtonGroup BG1 = new ButtonGroup ( ) ;**  **ButtonGroup BG2 = new ButtonGroup ( ) ;**  **Font KyrJL = new Font ( " 幼圆 " , Font.PLAIN , 40 ) ;**  **Font KyrJTF = new Font ( " 幼圆 " , Font.PLAIN , 20 ) ;**  **Font KyrJTA = new Font ( " 新宋体 " , Font.BOLD+Font.ITALIC , 30 ) ;**  **Font KyrJTA1 = new Font ( " 新宋体 " , Font.BOLD+Font.ITALIC , 10 ) ;**  **Font KyrJTA2 = new Font ( " 新宋体 " , Font.BOLD+Font.ITALIC , 50 ) ;**  **Hashtable < Integer , JComponent > H1 = new Hashtable < Integer , JComponent > () ;**  **H1.put( 0 , new JLabel ( "Min" ) ) ;**  **H1.put( 20 , new JLabel ( "20" ) ) ;**  **H1.put( 40 , new JLabel ( "40" ) ) ;**  **H1.put( 60 , new JLabel ( "60" ) ) ;**  **H1.put( 80 , new JLabel ( "80" ) ) ;**  **H1.put( 100 , new JLabel ( "Max" ) ) ;**  **// JMenuBar JMB1 = new JMenuBar ( ) ;**  **// JF1.setJMenuBar ( JMB1 ) ;**  **// JMenu JM1 = new JMenu ( " 菜单 " ) ; // Action a 菜单属性由相应动作提供**  **// JMenuItem JMI1 = new JMenuItem ( " 普通 " ) ;**  **// JCheckBoxMenuItem JCBMI1 =new JCheckBoxMenuItem ( "多选" ) ;**  **// JRadioButtonMenuItem JRBMI1 = new JRadioButtonMenuItem ( " 单选1 " ) ;**  **// JRadioButtonMenuItem JRBMI2 = new JRadioButtonMenuItem ( " 单选2 " ) ;**    **class WindowMonitor implements WindowListener { // Adapterで**  **public void windowOpened(WindowEvent e) {**  **System.out.println ( " 开了 " ) ;**  **}**  **public void windowClosing(WindowEvent e) {**  **System.out.println ( " 关了 " ) ;**  **JF1.dispose () ;**  **}**  **public void windowClosed(WindowEvent e) {**  **// System.out.println ( " 关闭后执行 " ) ; // 多线程才能执行..**  **}**  **public void windowActivated(WindowEvent e) {**  **System.out.println ( " 选 " ) ;**  **}**  **public void windowDeactivated(WindowEvent e) {**  **System.out.print ( " 锁 " ) ;**  **}**  **public void windowIconified(WindowEvent e) {**  **System.out.print ( " 隐 " ) ;**  **}**  **public void windowDeiconified(WindowEvent e) {**  **System.out.print ( " 显 " ) ;**  **}**  **}**  **WindowMonitor WM = new WindowMonitor ( ) ;**  **JF1.addWindowListener ( WM ) ; // F**    **JF1.setSize ( 500 , 500 ) ;**  **JF1.setLocationRelativeTo ( null ) ; //setLocation ( 300 , 200 ) ;**  **JF1.setDefaultCloseOperation ( WindowConstants.EXIT\_ON\_CLOSE ) ;**  **JF1.setVisible (true) ;**  **JPanel JPN1 = new JPanel ( ) ; // 大块套娃面板**  **JPN1.setLayout ( new BoxLayout ( JPN1 , BoxLayout.Y\_AXIS ) ) ;**  **JPN1.setVisible (true);**  **JPanel JPn1 = new JPanel ( ) ;**  **JPn1.setLayout ( new GridLayout ( 1 , 4 , 0 , 0 ) ) ; // マキシマムで**  **JPn1.setVisible (true);**  **JPanel JPn2 = new JPanel ( ) ;**  **JPn2.setLayout ( new GridLayout ( 1 , 1 , 0 , 0 ) ) ;**  **JPn2.setVisible (true);**  **JPanel JPn3 = new JPanel ( ) ;**  **JPn3.setLayout ( new GridLayout ( 1 , 4 , 0 , 0 ) ) ;**  **JPn3.setVisible (true);**  **JPanel JPn4 = new JPanel ( ) ;**  **JPn4.setLayout ( new GridLayout ( 1 , 4 , 0 , 0 ) ) ;**  **JPn4.setVisible (true);**  **JPanel Pn1 = new JPanel( );JPanel Pn2 = new JPanel( );JPanel Pn3 = new JPanel( );JPanel Pn4 = new JPanel( );**  **JPanel Pn5 = new JPanel( );JPanel Pn6 = new JPanel( );JPanel Pn7 = new JPanel( );JPanel Pn8 = new JPanel( );**  **JPanel Pn9 = new JPanel( );JPanel Pn10 = new JPanel( );JPanel Pn11 = new JPanel( );JPanel Pn12 = new JPanel( );**  **Pn2.setBackground ( Rainbow1 ) ; // 多用**  **Pn3.setBackground ( Rainbow2 ) ;**  **Pn4.setBackground ( Rainbow3 ) ;**  **Pn6.setBackground ( Rainbow1 ) ;**  **Pn7.setBackground ( Rainbow2 ) ;**  **Pn8.setBackground ( Rainbow3 ) ;**  **Pn10.setBackground ( Rainbow4 ) ;**  **Pn11.setBackground ( Rainbow4 ) ;**  **Pn12.setBackground ( Rainbow4 ) ;**  **// JPn1.setLayout ( null ) ; // .setBounds ( x , y , w ,h ) ;**  **// JPn1.setLayout ( new CardLayout ( 50 , 100 ) ) ;**  **// JPn1.setLayout ( new FlowLayout ( FlowLayout.CENTER , 50 , 100 ) ) ; // JPanel デフォールト**  **// JPn1.setLayout ( new BorderLayout ( 50 , 100 ) ) ; // JFrame、JDialog デフォールト**  **// PS : JPn1 ( Object , BorderLayout.CENTER ) ;**    **JSlider JS1 = new JSlider ( SwingConstants.HORIZONTAL , 0 , 100 , 50 ) ;**  **JS1.setBackground ( Rainbow4 ) ;**  **JS1.setInverted(false);**  **JS1.setSnapToTicks(false);**  **JS1.setMajorTickSpacing( 20 );**  **// JS1.setMajorTickSpacing( 5 );**  **JS1.setPaintTicks(false);**  **JS1.setPaintLabels(true);**  **JS1.setPaintTrack(true);**  **JS1.setLabelTable(H1);**  **JLabel JL1 = new JLabel ( "设置大小" , SwingConstants.LEFT) ;**  **JLabel JL2 = new JLabel ( "字体颜色" , SwingConstants.LEFT) ;**  **JLabel JL3 = new JLabel ( "调节大小" , SwingConstants.LEFT) ;**  **JLabel JL4 = new JLabel ( "字体大小" , SwingConstants.LEFT) ;**  **//JPasswordField JPs1 = new JPasswordField ( " 请输入密码 " , 20 ) ;**  **//JPs1.setEchoChar ( '\_' ) ;**  **//JPs1.setText( " Please input Password ! " ) ;**  **//JTextField JTF1 = new JTextField ( " 请输入一行文本内容 " , 20 ) ;**    **//JTA1.append ( "追加内容" ) ;**  **//JButton JB1 = new JButton ( " 按钮 " ) ;**  **//JComboBox < String > JCBox = new JComboBox < > ( Choice ) ;**  **JRadioButton JRB1 = new JRadioButton ( " 小号 " , false ) ;**  **JRadioButton JRB2 = new JRadioButton ( " 中号 " , true ) ;**  **JRadioButton JRB3 = new JRadioButton ( " 大号 " , false ) ;**  **JTextArea JTA1 = new JTextArea ( " 请输入几行文本内容 " , 10 , 20 ) ;**  **JTA1.setForeground ( Color.RED ) ;**  **JRadioButton JRB4 = new JRadioButton ( " 黄色 " , false ) ;**  **JRadioButton JRB5 = new JRadioButton ( " 红色 " , true ) ;**  **JRadioButton JRB6 = new JRadioButton ( " 蓝色 " , false ) ;**  **JRB1.setBackground ( Rainbow1 ) ;**  **JRB2.setBackground ( Rainbow2 ) ;**  **JRB3.setBackground ( Rainbow3 ) ;**  **JRB4.setBackground ( Rainbow1 ) ;**  **JRB5.setBackground ( Rainbow2 ) ;**  **JRB6.setBackground ( Rainbow3 ) ;**  **//JCheckBox JCB1 = new JCheckBox ( "①" , false ) ;**  **//JCheckBox JCB2 = new JCheckBox ( "②" , false ) ;**  **//JCheckBox JCB3 = new JCheckBox ( "③" , false ) ;**    **// JOptionPane.showMessageDialog ( null , " Array/Comp/Icon " , " 确认信息 " , JOptionPane.INFORMATION\_MESSAGE ) ; // null=JF**  **// JOptionPane.showConfirmDialog ( null , " 确认吗？ " , " 默认问题 " , JOptionPane.DEFAULT\_OPTION ) ; // o=y=yn**  **// JOptionPane.showInputDialog ( null , " 选择吧！ " , " 没有图标 " , JOptionPane.PLAIN\_MESSAGE , null , null , " 可为组合框、输入值 " ) ; // d = OC ERR**  **// JOptionPane.showOptionDialog ( null , " 你想要？ " , " 警告信息 " , JOptionPane.YES\_NO\_CANCEL\_OPTION , JOptionPane.QUESTION\_MESSAGE , null , Choice , Choice [ 0 ] ) ; // 图标默认 op=CompString/Icon**  **/\* JDialog JD1 = new JDialog ( JF1 , " 对话框 " , true ) ;**  **JD1.setSize ( 400 , 200 ) ;**  **JD1.setLocationRelativeTo ( null ) ;**  **// JD1.add ( JItemSource ) ;**  **JD1.setVisible(true) ; // 可见性写最后**  **\*/**    **JL1.setFont ( KyrJL ) ; // 字体设定**  **JL2.setFont ( KyrJL ) ;**  **JL3.setFont ( KyrJL ) ;**  **JL4.setFont ( KyrJL ) ;**  **JRB1.setFont ( KyrJTF ) ;**  **JRB2.setFont ( KyrJTF ) ;**  **JRB3.setFont ( KyrJTF ) ;**  **JRB4.setFont ( KyrJTF ) ;**  **JRB5.setFont ( KyrJTF ) ;**  **JRB6.setFont ( KyrJTF ) ;**  **JTA1.setFont ( KyrJTA ) ;**    **JF1.add ( JPN1 ) ; // 添来添去**  **JPN1.add ( JPn1 ) ; JPN1.add ( JPn2 ) ; JPN1.add ( JPn3 ) ; JPN1.add ( JPn4 ) ;**  **// JMB1.add ( JM1 ) ; JM1.add ( JMI1 ) ; JM1.add ( JCBMI1 ) ; JM1.addSeparator () ; // 分离器——————**  **JPn1.add ( Pn1 ) ; JPn1.add ( Pn2 ) ; JPn1.add ( Pn3 ) ; JPn1.add ( Pn4 ) ;**  **JPn2.add ( JTA1 ) ;**  **JPn3.add ( Pn5 ) ; JPn3.add ( Pn6 ) ; JPn3.add ( Pn7 ) ; JPn3.add ( Pn8 ) ;**  **JPn4.add ( Pn9 ) ; JPn4.add ( Pn10 ) ; JPn4.add ( Pn11 ) ; JPn4.add ( Pn12 ) ;**  **// JPn1.add ( JPs1 ) ; JPn1.add ( JTF1 ) ;**  **// JPn1.add ( JTA1 ) ; JPn1.add ( JB1 ) ; JPn1.add ( JCBox ) ;**  **Pn1.add ( JL1 ) ; Pn2.add ( JRB1 ) ; Pn3.add ( JRB2 ) ; Pn4.add ( JRB3 ) ; // JM1.add ( JRBMI1 ) ; JM1.add ( JRBMI2 ) ;**  **BG1.add ( JRB1 ) ; BG1.add ( JRB2 ) ; BG1.add ( JRB3 ) ; // BG2.add ( JRBMI1 ) ; BG2.add ( JRBMI2 ) ;**  **Pn5.add ( JL2 ) ; Pn6.add ( JRB4 ) ; Pn7.add ( JRB5 ) ; Pn8.add ( JRB6 ) ;**  **BG2.add ( JRB4 ) ; BG2.add ( JRB5 ) ; BG2.add ( JRB6 ) ;**  **Pn9.add ( JL3 ) ; Pn10.add ( JL4 ) ; Pn11.add ( JS1 ) ;**  **// JPn1.add ( JCB1 ) ; JPn1.add ( JCB2 ) ; JPn1.add ( JCB3 ) ;**    **class ActionMonitor1 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **JTA1.setFont ( KyrJTA1 ) ;**  **}**  **}**  **ActionMonitor1 AM1 = new ActionMonitor1 ( ) ;**  **JRB1.addActionListener ( AM1 ) ;**  **class ActionMonitor2 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **JTA1.setFont ( KyrJTA ) ;**  **}**  **}**  **ActionMonitor2 AM2 = new ActionMonitor2 ( ) ;**  **JRB2.addActionListener ( AM2 ) ;**  **class ActionMonitor3 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **JTA1.setFont ( KyrJTA2 ) ;**  **}**  **}**  **ActionMonitor3 AM3 = new ActionMonitor3 ( ) ;**  **JRB3.addActionListener ( AM3 ) ;**  **class ActionMonitor4 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **JTA1.setForeground ( Color.YELLOW ) ;**  **}**  **}**  **ActionMonitor4 AM4 = new ActionMonitor4 ( ) ;**  **JRB4.addActionListener ( AM4 ) ;**  **class ActionMonitor5 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **JTA1.setForeground ( Color.RED ) ;**  **}**  **}**  **ActionMonitor5 AM5 = new ActionMonitor5 ( ) ;**  **JRB5.addActionListener ( AM5 ) ;**  **class ActionMonitor6 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **JTA1.setForeground ( Color.BLUE ) ;**  **}**  **}**  **ActionMonitor6 AM6 = new ActionMonitor6 ( ) ;**  **JRB6.addActionListener ( AM6 ) ;**  **class MouseMotionMonitor implements MouseMotionListener {**  **public void mouseMoved ( MouseEvent ME ) {**  **}**  **public void mouseDragged ( MouseEvent ME ) {**  **JTA1.setFont ( new Font ( " 新宋体 " , Font.BOLD+Font.ITALIC , JS1.getValue() ) ) ;**  **}**  **}**  **MouseMotionMonitor MM = new MouseMotionMonitor ( ) ;**  **JS1.addMouseMotionListener ( MM ) ;**  **// ① ——————————**  **/\* class ItemMonitor implements ItemListener {**  **public void itemStateChanged ( ItemEvent IE ) {**  **if ( JItemSource.isSelected ( ) )**  **System.out.println ( " ON " ) ;**  **else**  **System.out.println ( " OFF " ) ;**  **}**  **}**  **ItemMonitor IM = new ItemMonitor ( ) ; // Just cαn be AnonymousClass**  **JItemSource.addItemListener ( IM ) ; // B/CBox/RB/CB 委托监听**  **\*/**  **// ③ ——————————**  **/\* class FocusMonitor implements FocusListener {**  **public void focusGained ( FocusEvent FE ) { // If Event = = > match Processor**  **System.out.println ( " 聚焦 " ) ;**  **}**  **public void focusLost ( FocusEvent FE ) {**  **System.out.println ( " 失焦 " ) ;**  **}**  **}**  **FocusMonitor FM = new FocusMonitor ( ) ;**  **JItemSource.addFocusListener ( FM ) ;**  **\*/**  **// ④ ——————————**  **/\* class KeyMonitor implements KeyListener {**  **public void keyPressed ( KeyEvent KE ) { // 瞬**  **System.out.println ( " 输入 : " ) ;**  **}**  **public void keyTyped ( KeyEvent KE ) { // 过程**  **System.out.println ( KE.getKeyChar ( ) ) ;**  **}**  **public void keyReleased ( KeyEvent KE ) { // 瞬**  **System.out.println ( " \_ " ) ;**  **}**  **}**  **KeyMonitor KM = new KeyMonitor ( ) ;**  **JItemSource.addKeyListener ( KM ) ;**  **\*/**  **// ⑤ ——————————**  **/\***  **class MouseMonitor implements MouseListener {**  **public void mouseEntered ( MouseEvent Me ) {**  **System.out.print ( " GO " ) ;**  **}**  **public void mouseExited ( MouseEvent Me ) {**  **System.out.println ( " FULL COMBO " ) ;**  **}**  **public void mousePressed ( MouseEvent Me ) {**  **System.out.print ( " Tap " ) ;**  **}**  **public void mouseReleased ( MouseEvent Me ) {**  **System.out.print ( " FLick " ) ;**  **}**  **public void mouseClicked ( MouseEvent Me ) {**  **System.out.println ( " Perfect " ) ;**  **if (Me.getClickCount ( ) > 4) {**  **System.out.println ( Me.getClickCount ( ) + " Combo " ) ;**  **}**  **}**  **}**  **MouseMonitor Mm = new MouseMonitor ( ) ;**  **JF1.addMouseListener ( Mm ) ;**  **\*/**    **/\* .getHgap ( ) ; // 保留方法**  **.getVgap ( ) ;**  **FL.getAlignment ( ) ;**  **GL.getRows ( ) ;**  **GL.getColumns ( ) ;**  **CL.first/last/next/previous/remove ( Component ) ;**  **CL.addLayoutComponent/show ( Component , Name ) ;**    **JPs.char[] getPassword ( ) ;**  **JF.getContentPane ( ) ;**  **JD.getTitle ( ) ;**  **JTF.getSelectedText ( ) ;**  **JTF.getColumns ( ) ;**  **JTA.getText ( ) ;**  **JTA.getColumns ( ) ;**  **JTA.getRows ( ) ;**  **JTA.getColumnWidth ( ) ;**  **JTA.getLineCount ( ) ;**  **JRB.getText ( ) ;**  **JRB.isSelected ( ) ;**  **JRB.doClick ( ) ;**  **JCB.getText ( ) ;**  **JCB.isSelected ( ) ;**  **JCB.doClick ( ) ;**  **JCBox.getSelectedItem ( ) ;**  **JCBox.getItemCount ( ) ;**  **JCBox.addItem ( Object ) ;**  **JCBox.setSelectedItem ( Object ) ;**  **JCBox.removeItem ( Object ) ;**  **JCBox.insertItemAt ( Object , Index ) ;**  **\*/**  **}**  **}**   1. **总结**   **问题：组件数组需要指针调一下置空**  **心得：官网查了最新API文档 简单化操作 静态平淡才是真** | | |
| **项目三：多功能计算器** | **得分** |  |
| 1. **实验内容**   **一次输入 多重运算**  **原滋原味的使用体验**  **良心排版超精致护目界面 缓解视觉疲劳**  **普普通通的科学计算器 储存约五千万次计算**  **程序严谨 但仍有很大程度的优化空间**  **涵盖基本每一种情况下输出显示结果的真实实现 达到轻松计算**   1. **问题描述**   **基本要求的基础上追加一定程度的字符计算需求**  **启动界面默认空白 建议每次结束后清屏保持内存整洁**  **控制台实时反馈控件及其窗口框架信息**  **计算结果的数据类型为一般实型双精度浮点型数据类型**  **数值超出屏幕范围时将用科学计数法进行计数表述**  **本身无法计算的数会显示NaN**  **隐含一些符合人平时养成的特殊使用习惯的便捷规则设定**  **删减了避免混淆的初设操作**  **输入框不予具体设计 可复制、输入、拖拽、滚动、聚焦**   1. **项目功能**   **四则运算『+』/『-』/『×』/『÷』 输入符号前运算数值一、点击四则运算相应符号按钮『+』/『-』/『×』/『÷』、输入符号后运算数值二、点击等号按钮『=』后进行相应基本运算、输入框返回其数值计算结果。其中负号按钮『-』也兼具隐含了等同于正负转换按钮『+/-』的赋值功能、当输入框为空时点击负号按钮『-』输入框返回前置负号『-』然后输入框输入正数后即为负数数值**  **正负转换『+/-』 输入运算数值后点击按钮进行计算 1. 输入框输入正数后点击正负转换按钮『+/-』输入框返回前置负号『-』的正数转换负数的数值  2. 输入框输入零『0』后点击正负转换按钮『+/-』直接清除屏幕输入框 3. 输入框输入负数后点击正负转换按钮『+/-』输入框返回删除负号『-』 的负数转换正数的数值  三角函数『Sin』/『Cos』 输入运算数值后点击按钮进行计算、点击三角函数相应按钮『Sin』/『Cos』输入框返回数值三角函数计算结果  开方运算『√』 输入运算数值后点击按钮进行计算 1. 输入框输入正数后点击开方运算按钮『√』输入框返回数值开平方计算结果 2. 输入框输入零『0』后点击开方运算按钮『√』输入框直接返回数值开平方计算结果：零『0』 3. 输入框输入负数后点击开方运算按钮『√』输入框不做其他具体操作、直接返回数值本身  阶乘运算『X!』 输入运算数值后点击按钮进行计算 1. 输入框输入正数后点击阶乘运算按钮『X!』输入框返回数值求阶乘计算结果 2. 输入框输入零『0』后点击阶乘运算按钮『X!』输入框直接返回数值求阶乘计算结果：一『1』 3. 输入框输入负数（除负一『-1』以外的其他的负数）后点击阶乘运算按钮『X!』输入框不做其他具体操作、返回数值本身 4. 但当输入框输入零『-1』之后点击阶乘运算按钮『X!』输入框直接返回数值求阶乘计算结果：零『0』  倒数转换『1/X』 输入运算数值后点击按钮进行计算、点击倒数转换相应按钮『1/X』输入框返回数值倒数转换计算结果**  **零的倒数显示为无穷：Infinity  清除屏幕『AC』 输入框任意运算过程中点击清除屏幕按钮『AC』进行屏幕输入框的清除、输入框返回空白『』并终止输入框全部运算操作、对计算器程序内存进行一定程度的自动回收清除  次方运算『X^Y』 输入底数运算数值一、点击次方运算按钮『X^Y』、输入指数运算数值二、点击等号『=』后进行次方运算、输入框返回其数值次方运算计算结果**  **零的任意次方是零，任意数的零次方是一。**  **关于等号『=』 支持表达式连续计算**  **规定其具备一键清除非表达式内容的内载功能**   1. **项目分析**     **界面如图**    **仅以-4.2为例分别展示部分操作**    **具体功能详见操作演示**   1. **项目设计**   **草图→基层→控件→布局→界面美化→监听→算法优化→打点→改错→成品跑程序→反复观察试用→改错N次**   1. **系统实现**   **package Software;**  **import java.awt.BorderLayout;**  **import java.awt.Color;**  **import java.awt.FlowLayout;**  **import java.awt.Font;**  **import java.awt.GridLayout;**  **import java.awt.event.ActionEvent;**  **import java.awt.event.ActionListener;**  **import java.awt.event.WindowEvent;**  **import java.awt.event.WindowListener;**  **import javax.swing.JButton;**  **import javax.swing.JFrame;**  **import javax.swing.JMenuBar;**  **import javax.swing.JPanel;**  **import javax.swing.JTextField;**  **import javax.swing.WindowConstants;**  **@SuppressWarnings("serial")**  **public class Projc03 extends JFrame {**  **int OSU , Kami ;**  **double a , b , Kaidan ;**  **String X ;**  **public static void main ( String [] args ) {**  **Projc03 Pro = new Projc03 () ;**  **Pro.OSU = 0 ;**  **Pro.Kami = 0 ;**  **Pro.Kaidan = 1 ;**  **Pro.a = 0 ;**  **Pro.b = 0 ;**  **Pro.X = null ;**  **Color Rainbow1 = new Color ( 255 , 245 , 255 ) ; // 定义一部分常用的量**  **Color Rainbow2 = new Color ( 0 , 250 , 250 ) ;**  **Color Rainbow3 = new Color ( 150 , 250 , 150 ) ;**  **Color Rainbow4 = new Color ( 170 , 250 , 150 ) ;**  **Color Rainbow5 = new Color ( 100 , 250 , 150 ) ;**  **Color Rainbow6 = new Color ( 50 , 250 , 200 ) ;**  **JFrame JF1 = new JFrame (" 多功能计算器 ") ; // 整块**  **// ButtonGroup BG1 = new ButtonGroup ( ) ;**  **// ButtonGroup BG2 = new ButtonGroup ( ) ;**  **// Font KyrJL = new Font ( "幼圆" , Font.BOLD+Font.ITALIC , 50 ) ;**  **Font KyrJTF = new Font ( "新宋体" , Font.PLAIN , 100 ) ;**  **Font KyrB = new Font ( "新宋体" , Font.BOLD , 30 ) ;**    **JMenuBar JMB1 = new JMenuBar ( ) ;**  **JF1.setJMenuBar ( JMB1 ) ;**  **// JMenu JM1 = new JMenu ( " 菜单 " ) ; // Action a 菜单属性由相应动作提供**  **// JMenuItem JMI1 = new JMenuItem ( " 普通 " ) ;**  **// JCheckBoxMenuItem JCBMI1 =new JCheckBoxMenuItem ( "多选" ) ;**  **// JRadioButtonMenuItem JRBMI1 = new JRadioButtonMenuItem ( " 单选1 " ) ;**  **// JRadioButtonMenuItem JRBMI2 = new JRadioButtonMenuItem ( " 单选2 " ) ;**    **class WindowMonitor implements WindowListener { // Adapterで**  **public void windowOpened(WindowEvent e) {**  **System.out.println ( " 开了 " ) ;**  **}**  **public void windowClosing(WindowEvent e) {**  **System.out.println ( " 关了 " ) ;**  **JF1.dispose () ;**  **}**  **public void windowClosed(WindowEvent e) {**  **// System.out.println ( " 关闭后执行 " ) ; // 多线程才能执行..**  **}**  **public void windowActivated(WindowEvent e) {**  **System.out.println ( " 选 " ) ;**  **}**  **public void windowDeactivated(WindowEvent e) {**  **System.out.print ( " 锁 " ) ;**  **}**  **public void windowIconified(WindowEvent e) {**  **System.out.print ( " 隐 " ) ;**  **}**  **public void windowDeiconified(WindowEvent e) {**  **System.out.print ( " 显 " ) ;**  **}**  **}**  **WindowMonitor WM = new WindowMonitor ( ) ;**  **JF1.addWindowListener ( WM ) ; // F**    **JF1.setSize ( 700 , 700 ) ;**  **JF1.setLocationRelativeTo ( null ) ; //setLocation ( 300 , 200 ) ;**  **JF1.setDefaultCloseOperation ( WindowConstants.EXIT\_ON\_CLOSE ) ;**  **JF1.setVisible (true) ;**  **JPanel JPN1 = new JPanel ( ) ; // 大块套娃面板**  **JPN1.setLayout ( new BorderLayout ( 0 , 0 ) ) ; // JFrame、JDialog デフォールト**  **JPanel JPn1 = new JPanel ( ) ;**  **JPn1.setLayout ( new BorderLayout ( 40 , 40 ) ) ;**  **JPanel Pn1 = new JPanel ( ) ;**  **Pn1.setLayout ( new FlowLayout ( FlowLayout.CENTER , 500 , 0 ) ) ; // JPanel デフォールト**  **JPanel Pn2 = new JPanel ( ) ;**  **JPanel Pn3 = new JPanel ( ) ;**  **JPanel Pn4 = new JPanel ( ) ;**  **JPanel Pn5 = new JPanel ( ) ;**  **Pn2.setLayout ( new GridLayout ( 1 , 2 , 20 , 0 ) ) ;**  **JPanel P1 = new JPanel ( ) ;**  **JPanel P2 = new JPanel ( ) ;**  **P1.setLayout ( new GridLayout ( 4 , 3 , 0 , 0 ) ) ; // マキシマムで**  **P2.setLayout ( new GridLayout ( 4 , 3 , 0 , 0 ) ) ;**  **JPn1.setBackground ( Rainbow4 ) ;**  **Pn1.setBackground ( Rainbow1 ) ; // 多用**  **Pn2.setBackground ( Rainbow6 ) ;**  **Pn3.setBackground ( Rainbow3 ) ;**  **Pn4.setBackground ( Rainbow3 ) ;**  **Pn5.setBackground ( Rainbow3 ) ;**  **P1.setBackground ( Rainbow5 ) ;**  **P2.setBackground ( Rainbow2 ) ;**  **// JPn1.setLayout ( null ) ; // .setBounds ( x , y , w ,h ) ;**  **// JPn1.setLayout ( new CardLayout ( 50 , 100 ) ) ;**    **// JLabel JL1 = new JLabel ( " 标签 " , SwingConstants.CENTER) ;**  **// JPasswordField JPs1 = new JPasswordField ( " 请输入密码 " , 20 ) ;**  **// JPs1.setEchoChar ( '\_' ) ;**  **// JPs1.setText( " Please input Password ! " ) ;**  **JTextField JTF1 = new JTextField ( "" , 20 ) ;**  **JTF1.setSize(1000, 200);**  **// JTextArea JTA1 = new JTextArea ( " 请输入几行文本内容 " , 10 , 20 ) ;**  **// JTA1.append ( "追加内容" ) ;**  **String Controller1 [] = { "9" , "8" , "7", "6", "5", "4", "3", "2", "1", "0", "+/-", "." } ;**  **String Controller2 [] = { "÷" , "√" , "X!", "×", "Sin", "1/X", "-", "Cos", "AC", "+", "X^Y", "=" } ;**  **// JComboBox < String > JCBox = new JComboBox < > ( Choice ) ;**  **// JRadioButton JRB1 = new JRadioButton ( "是" , true ) ;**  **// JRadioButton JRB2 = new JRadioButton ( "否" , false ) ;**  **// JCheckBox JCB1 = new JCheckBox ( "①" , false ) ;**  **// JCheckBox JCB2 = new JCheckBox ( "②" , false ) ;**  **// JCheckBox JCB3 = new JCheckBox ( "③" , false ) ;**    **// JOptionPane.showMessageDialog ( null , " Array/Comp/Icon " , " 确认信息 " , JOptionPane.INFORMATION\_MESSAGE ) ; // null=JF**  **// JOptionPane.showConfirmDialog ( null , " 确认吗？ " , " 默认问题 " , JOptionPane.DEFAULT\_OPTION ) ; // o=y=yn**  **// JOptionPane.showInputDialog ( null , " 选择吧！ " , " 没有图标 " , JOptionPane.PLAIN\_MESSAGE , null , null , " 可为组合框、输入值 " ) ; // d = OC ERR**  **// JOptionPane.showOptionDialog ( null , " 你想要？ " , " 警告信息 " , JOptionPane.YES\_NO\_CANCEL\_OPTION , JOptionPane.QUESTION\_MESSAGE , null , Choice , Choice [ 0 ] ) ; // 图标默认 op=CompString/Icon**  **/\* JDialog JD1 = new JDialog ( JF1 , " 对话框 " , true ) ;**  **JD1.setSize ( 400 , 200 ) ;**  **JD1.setLocationRelativeTo ( null ) ;**  **// JD1.add ( JItemSource ) ;**  **JD1.setVisible(true) ; // 可见性写最后**  **\*/**    **// JL1.setFont ( KyrJL ) ; // 字体设定**  **JTF1.setFont ( KyrJTF ) ;**  **// JTA1.setFont ( KyrJTA ) ;**  **String Input [] = new String [ 100000000 ] ;**  **Input [Pro.OSU] = "" ;**  **for ( int Tool = 0 ; Tool < Controller1.length ; Tool ++ ) {**  **JButton JB = new JButton ( Controller1 [ Tool ] ) ;**  **class ActionMonitor100 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **if ( JB.getText() == "+/-" ) {**  **if ( Double.parseDouble ( Input [ Pro.OSU ] ) == 0) {**  **JTF1.setText ( Input [ Pro.OSU ] ) ;**  **}**  **if ( Double.parseDouble ( Input [ Pro.OSU ] ) > 0) {**  **Input [ Pro.OSU + 1 ] = "-" + Input [ Pro.OSU ] ;**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Pro.OSU ++ ;**  **}**  **else {**  **Input [ Pro.OSU + 1 ] = Input [ Pro.OSU - 1 ] ;**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Pro.OSU ++ ;**  **}**  **}**  **else {**  **Input [ Pro.OSU + 1 ] = Input [ Pro.OSU ] + JB.getText () ;**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Pro.OSU ++ ;**  **}**  **}**  **}**  **ActionMonitor100 AM100 = new ActionMonitor100 ( ) ;**  **JB.addActionListener ( AM100 ) ;**  **JB.setSize ( 50 , 50 );**  **JB.setFont(KyrB);**  **JB.setBorderPainted(false);**  **JB.setContentAreaFilled(true);**  **JB.setBackground(Rainbow1);**  **JB.setForeground(Rainbow5);**  **P1.add ( JB ) ;**  **}**  **for ( int Tool = 0 ; Tool < Controller2.length ; Tool ++ ) {**  **JButton JB = new JButton ( Controller2 [ Tool ] ) ;**  **class ActionMonitor100 implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **if ( JB.getText() == "AC" ) {**  **Input [ Pro.OSU + 1 ] = "" ;**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Pro.OSU ++ ;**  **}**  **if ( Input [ Pro.OSU ] == "" ) {**  **if ( JB.getText() == "-" ) {**  **Input [ Pro.OSU + 1 ] = Input [ Pro.OSU ] + JB.getText () ;**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Pro.OSU ++ ;**  **}**  **}**  **if ( Double.parseDouble ( Input [ Pro.OSU ] ) > 0 ) {**  **if ( JB.getText() == "√" ) {**  **Input [ Pro.OSU + 1 ] = Input [ Pro.OSU ] ;**  **JTF1.setText ( String.valueOf ( Math.sqrt ( Double.parseDouble ( Input [ Pro.OSU ] ) ) ) ) ;**  **Pro.OSU ++ ;**  **}**  **if ( JB.getText() == "X!" ) {**  **Input [ Pro.OSU + 1 ] = Input [ Pro.OSU ] ;**  **double Kaisu = Double.parseDouble ( Input [ Pro.OSU + 1 ] ) ;**  **while (Kaisu > 0 ) {**  **Pro.Kaidan = Pro.Kaidan \* Kaisu ;**  **Kaisu -- ;**  **}**  **Input [ Pro.OSU + 1 ] = String.valueOf ( Pro.Kaidan ) ;**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Pro.Kaidan = 1 ;**  **Pro.OSU ++ ;**  **}**    **}**  **if ( Double.parseDouble ( Input [ Pro.OSU ] ) == 0 ) {**  **if ( JB.getText() == "X!" ) {**  **Input [ Pro.OSU + 1 ] = "1" ;**  **JTF1.setText ( ( Input [ Pro.OSU + 1 ] ) ) ;**  **Pro.OSU ++ ;**  **}**  **if ( JB.getText() == "√" ) {**  **Input [ Pro.OSU + 1 ] = "0" ;**  **JTF1.setText ( ( Input [ Pro.OSU + 1 ] ) ) ;**  **Pro.OSU ++ ;**  **}**  **}**  **if ( Double.parseDouble ( Input [ Pro.OSU ] ) == -1 ) {**  **if ( JB.getText() == "X!" ) {**  **Input [ Pro.OSU + 1 ] = "0" ;**  **JTF1.setText ( ( Input [ Pro.OSU + 1 ] ) ) ;**  **Pro.OSU ++ ;**  **}**  **}**  **if ( JB.getText() == "Sin" ) {**  **Input [ Pro.OSU + 1 ] = Input [ Pro.OSU ] ;**  **JTF1.setText ( String.valueOf ( Math.sin ( Double.parseDouble ( Input [ Pro.OSU ] ) ) ) ) ;**  **Pro.OSU ++ ;**  **}**  **if ( JB.getText() == "1/X" ) {**  **Input [ Pro.OSU + 1 ] = String.valueOf ( 1 / Double.parseDouble ( Input [ Pro.OSU ] ) ) ;**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Pro.OSU ++ ;**  **}**  **if ( JB.getText() == "Cos" ) {**  **Input [ Pro.OSU + 1 ] = Input [ Pro.OSU ] ;**  **JTF1.setText ( String.valueOf ( Math.cos ( Double.parseDouble ( Input [ Pro.OSU + 1 ] ) ) ) ) ;**  **Pro.OSU ++ ;**  **}**  **if ( JB.getText() == "÷" || JB.getText() == "×" ||JB.getText() == "×" || JB.getText() == "-" || JB.getText() == "+" || JB.getText() == "X^Y" ) {**  **Input [ Pro.OSU + 1 ] = "" ;**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Pro.a = Double.parseDouble ( Input [ Pro.OSU ] ) ;**  **Pro.X = JB.getText() ;**  **Pro.OSU ++ ;**  **}**  **if ( JB.getText() == "=" ) {**  **Pro.b = Double.parseDouble ( Input [ Pro.OSU ] ) ;**  **if ( Pro.X == "÷" ) {**  **Input [ Pro.OSU + 1 ] = String.valueOf ( Pro.a / Pro.b ) ;**  **Pro.a = 0 ;**  **Pro.b = 0 ;**  **Pro.X = null ;**  **}**  **if ( Pro.X == "×" ) {**  **Input [ Pro.OSU + 1 ] = String.valueOf ( Pro.a \* Pro.b ) ;**  **Pro.a = 0 ;**  **Pro.b = 0 ;**  **Pro.X = null ;**  **}**  **if ( Pro.X == "-" ) {**  **Input [ Pro.OSU + 1 ] = String.valueOf ( Pro.a - Pro.b ) ;**  **Pro.a = 0 ;**  **Pro.b = 0 ;**  **Pro.X = null ;**  **}**  **if ( Pro.X == "+" ) {**  **Input [ Pro.OSU + 1 ] = String.valueOf ( Pro.a + Pro.b ) ;**  **Pro.a = 0 ;**  **Pro.b = 0 ;**  **Pro.X = null ;**  **}**  **if ( Pro.X == "X^Y" ) {**  **Input [ Pro.OSU + 1 ] = String.valueOf ( Math.pow ( Pro.a , Pro.b ) ) ;**  **Pro.a = 0 ;**  **Pro.b = 0 ;**  **Pro.X = null ;**  **}**  **else {**  **System.out.println( " 点不动哟 " );**  **}**  **JTF1.setText ( Input [ Pro.OSU + 1 ] ) ;**  **Input [ Pro.OSU + 1 ] = "" ;**  **Pro.OSU ++ ;**  **}**  **}**  **}**  **ActionMonitor100 AM100 = new ActionMonitor100 ( ) ;**  **JB.addActionListener ( AM100 ) ;**  **JB.setSize ( 50 , 50 );**  **JB.setFont(KyrB);**  **JB.setBorderPainted(false);**  **JB.setContentAreaFilled(true);**  **JB.setBackground(Rainbow1);**  **JB.setForeground(Rainbow2);**  **P2.add ( JB ) ;**  **if ( Tool == 8 || Tool == 11 ) {**  **JB.setBackground(Rainbow2);**  **JB.setForeground(Rainbow1);**  **}**  **}**    **JF1.add ( JPN1 ) ; JPN1.add ( JPn1 , BorderLayout.CENTER ) ;// 添来添去**  **JPn1.add ( Pn1 , BorderLayout.NORTH ) ;**  **JPn1.add ( Pn2 , BorderLayout.CENTER ) ;**  **JPn1.add ( Pn3 , BorderLayout.SOUTH ) ;**  **JPn1.add ( Pn4 , BorderLayout.WEST ) ;**  **JPn1.add ( Pn5 , BorderLayout.EAST ) ;**  **// JMB1.add ( JM1 ) ; JM1.add ( JMI1 ) ; JM1.add ( JCBMI1 ) ; JM1.addSeparator () ; // 分离器——————**  **// JPn1.add ( JL1 ) ; JPn1.add ( JPs1 ) ;**  **Pn1.add ( JTF1 ) ; Pn2.add ( P1 ) ; Pn2.add ( P2 ) ;**  **// JPn1.add ( JTA1 ) ; JPn1.add ( JCBox ) ;**  **// JPn1.add ( JRB1 ) ; JPn1.add ( JRB2 ) ; JM1.add ( JRBMI1 ) ; JM1.add ( JRBMI2 ) ;**  **// BG1.add ( JRB1 ) ; BG1.add ( JRB2 ) ; BG2.add ( JRBMI1 ) ; BG2.add ( JRBMI2 ) ;**  **// JPn1.add ( JCB1 ) ; JPn1.add ( JCB2 ) ; JPn1.add ( JCB3 ) ;**  **// ① ——————————**  **/\* class ItemMonitor implements ItemListener {**  **public void itemStateChanged ( ItemEvent IE ) {**  **if ( JItemSource.isSelected ( ) )**  **System.out.println ( " ON " ) ;**  **else**  **System.out.println ( " OFF " ) ;**  **}**  **}**  **ItemMonitor IM = new ItemMonitor ( ) ; // Just cαn be AnonymousClass**  **JItemSource.addItemListener ( IM ) ; // B/CBox/RB/CB 委托监听**  **\*/**  **// ② ——————————**  **/\* class ActionMonitor implements ActionListener {**  **public void actionPerformed ( ActionEvent AE ) {**  **System.out.println ( " 组件名称 ： " + AE.getActionCommand ( ) ) ;**  **System.out.println ( " 组件信息 ： " + AE.getSource ( ) ) ;**  **}**  **}**  **ActionMonitor AM = new ActionMonitor ( ) ;**  **JItemSource.addActionListener ( AM ) ; // F Pn L Ps/TF TA B/CBox/RB/CB**  **\*/**  **// ③ ——————————**  **/\* class FocusMonitor implements FocusListener {**  **public void focusGained ( FocusEvent FE ) { // If Event = = > match Processor**  **System.out.println ( " 聚焦 " ) ;**  **}**  **public void focusLost ( FocusEvent FE ) {**  **System.out.println ( " 失焦 " ) ;**  **}**  **}**  **FocusMonitor FM = new FocusMonitor ( ) ;**  **JItemSource.addFocusListener ( FM ) ;**  **\*/**  **// ④ ——————————**  **/\* class KeyMonitor implements KeyListener {**  **public void keyPressed ( KeyEvent KE ) { // 瞬**  **System.out.println ( " 输入 : " ) ;**  **}**  **public void keyTyped ( KeyEvent KE ) { // 过程**  **System.out.println ( KE.getKeyChar ( ) ) ;**  **}**  **public void keyReleased ( KeyEvent KE ) { // 瞬**  **System.out.println ( " \_ " ) ;**  **}**  **}**  **KeyMonitor KM = new KeyMonitor ( ) ;**  **JItemSource.addKeyListener ( KM ) ;**  **\*/**  **// ⑤ ——————————**  **/\* class MouseMotionMonitor implements MouseMotionListener {**  **public void mouseMoved ( MouseEvent ME ) {**  **System.out.println ( " Hold " ) ;**  **}**  **public void mouseDragged ( MouseEvent ME ) {**  **System.out.println ( " Drag X : " + ME.getX ( ) + " Y : " + ME.getY ( ) ) ;**  **}**  **}**  **class MouseMonitor implements MouseListener {**  **public void mouseEntered ( MouseEvent Me ) {**  **System.out.print ( " GO " ) ;**  **}**  **public void mouseExited ( MouseEvent Me ) {**  **System.out.println ( " FULL COMBO " ) ;**  **}**  **public void mousePressed ( MouseEvent Me ) {**  **System.out.print ( " Tap " ) ;**  **}**  **public void mouseReleased ( MouseEvent Me ) {**  **System.out.print ( " FLick " ) ;**  **}**  **public void mouseClicked ( MouseEvent Me ) {**  **System.out.println ( " Perfect " ) ;**  **if (Me.getClickCount ( ) > 4) {**  **System.out.println ( Me.getClickCount ( ) + " Combo " ) ;**  **}**  **}**  **}**  **MouseMotionMonitor MM = new MouseMotionMonitor ( ) ;**  **JF1.addMouseMotionListener ( MM ) ;**  **MouseMonitor Mm = new MouseMonitor ( ) ;**  **JF1.addMouseListener ( Mm ) ;**  **\*/**    **/\* .getHgap ( ) ; // 保留方法**  **.getVgap ( ) ;**  **FL.getAlignment ( ) ;**  **GL.getRows ( ) ;**  **GL.getColumns ( ) ;**  **CL.first/last/next/previous/remove ( Component ) ;**  **CL.addLayoutComponent/show ( Component , Name ) ;**    **JPs.char[] getPassword ( ) ;**  **JF.getContentPane ( ) ;**  **JD.getTitle ( ) ;**  **JTF.getSelectedText ( ) ;**  **JTF.getColumns ( ) ;**  **JTA.getText ( ) ;**  **JTA.getColumns ( ) ;**  **JTA.getRows ( ) ;**  **JTA.getColumnWidth ( ) ;**  **JTA.getLineCount ( ) ;**  **JRB.getText ( ) ;**  **JRB.isSelected ( ) ;**  **JRB.doClick ( ) ;**  **JCB.getText ( ) ;**  **JCB.isSelected ( ) ;**  **JCB.doClick ( ) ;**  **JCBox.getSelectedItem ( ) ;**  **JCBox.getItemCount ( ) ;**  **JCBox.addItem ( Object ) ;**  **JCBox.setSelectedItem ( Object ) ;**  **JCBox.removeItem ( Object ) ;**  **JCBox.insertItemAt ( Object , Index ) ;**  **\*/**  **}**  **}**   1. **总结**   **问题：异常系Bug比想象多**  **心得：熟悉的方法比自己实现要好 代码越短越好 多生成备注 记得保存嗷** | | |