108 《多媒体技术与应用》 实验报告 2008～2009学年第一学期 学 院 化学与分子工程学院 班 级 应材071 姓 名 张三 学 号 10079999 任课教师 李飞（周五3,4节） 计算机教学实验中心 2008年 10 月 《多媒体技术与应用》实 验 报 告 学号： 姓名： 班级： 实验名称：Photoshop综合实验：制作足球 实验地点： 所使用的工具软件及环境： 一、实验目的： 1．掌握Photoshop的形状工具； 2东然侠伙斜劫橇添梅渝仗酬翘答厂及脯辈采柠藏竣旁碰壶窥石毋憋丝甘湾野咎家违枉嗣厂攻猿前

**浙江理工大学**

**《Java程序设计》**

**实验报告**

**2024～2025学年第2 学期**

|  |  |
| --- | --- |
| **学 院** | **计算机科学与技术** |
| **班 级** | **23计科（）班** |
| **姓 名** | **盲灯** |
| **学 号** |  |
| **任课教师** | **孙麒 杨东鹤** |
| **上课时间** | **周四6-8 节** |

**《Java程序设计》实 验 报 告（一）**

|  |  |
| --- | --- |
| **实验名称**: Java开发环境的熟悉 | **实验地点**：10-413、414 |
| **实验类别：**基本 |  |
| **所使用的工具软件及环境：**   1. 环境：Windows 8.1;jdk 1.8; 2. 工具：Eclipse或其他Java IDE | |
| **实验目的：**   1. 安装和使用Java开发工具； 2. 学习使用Swing组件进行界面设计； 3. 理解事件处理机制，并能够编写简单的事件处理代码。 | |
| **实验内容：**  本次实验有两个任务，学生可以**选择其中一个**完成。选择完成计算器任务的学生将获得更高的分数。  **任务一：五分制计分转换程序**  编写一个图形界面程序，接受用户输入的一个0~100之间的分数，将其转化为1~5分的计分方法：   * 90~100：5 * 75~89：4 * 60~74：3 * 40~59：2 * 其他：1   要求：   1. 使用Swing组件设计用户界面。 2. 界面应包含一个输入框用于输入分数，一个按钮用于触发转换，以及一个标签用于显示转换后的分数。 3. 实现事件处理，当用户点击按钮时，程序应根据输入的分数显示相应的五分制分数。   **任务二：简单计算器程序**  编写一个图形界面程序，实现一个简单的计算器，支持基本的加减乘除运算。  要求：   1. 使用Swing组件设计用户界面。 2. 界面应至少包含数字按钮（0-9）、运算符按钮（+、-、\*、/）、等号按钮和一个显示结果的文本框（可增加更多计算运算按钮）。 3. 实现事件处理，当用户点击数字按钮和运算符按钮时，程序应能够正确显示输入的数字和运算符，并在点击等号按钮时显示计算结果。 | |
| **代码实现：**  package shiyan1;  import javax.swing.\*;  import java.awt.\*;  import java.awt.event.\*;  import java.math.RoundingMode;  import java.text.DecimalFormat;  import java.util.ArrayList;  public class ScientificCalculator extends JFrame {      private JTextField display;      private String memoryText = "";      private String currentInput = "0";      private String currentOperator = "";      private double firstOperand = 0;      private boolean startNewInput = true;  *// 历史记录列表*      private ArrayList<String> history = new ArrayList<>();      public ScientificCalculator() {          setTitle("Scientific Calculator");          setSize(500, 600);          setLocationRelativeTo(null);          setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);          initUI();      }      private void initUI() {  *// 顶部面板：显示制作信息*          JPanel topPanel = new JPanel(new BorderLayout());          JLabel makerLabel = new JLabel("Made by Huang Hongzhou", SwingConstants.CENTER);          makerLabel.setFont(new Font("微软雅黑", Font.BOLD, 16));          topPanel.add(makerLabel, BorderLayout.CENTER);          add(topPanel, BorderLayout.NORTH);  *// 显示区：只有一个 JTextField，用于显示记忆和当前输入，光标始终闪烁*          display = new JTextField();          display.setFont(new Font("Consolas", Font.BOLD, 24));          display.setHorizontalAlignment(JTextField.RIGHT);          display.setEditable(false);  *// 不允许手动编辑，所有内容由程序控制*          display.setText(currentInput);          add(display, BorderLayout.CENTER);  *// 按钮区：弄成5行5列*          String[] btnLabels = {                  "7", "8", "9", "/", "√",                  "4", "5", "6", "\*", "n√",                  "1", "2", "3", "-", "x²",                  "0", ".", "%", "+", "x³",                  "^", "=", "C", "Hist", "Exit"          };          JPanel buttonPanel = new JPanel(new GridLayout(5, 5, 5, 5));          for (String label : btnLabels) {              JButton btn = new JButton(label);              btn.setFont(new Font("Consolas", Font.BOLD, 18));              btn.addActionListener(new ButtonHandler());              buttonPanel.add(btn);          }          add(buttonPanel, BorderLayout.SOUTH);      }  *// 更新显示区：将记忆部分和当前输入部分拼接，并将光标置于文本末尾*      private void updateDisplay() {          display.setText(memoryText + currentInput);      }  *// 按钮事件处理*      private class ButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              String cmd = e.getActionCommand();  *// 数字或小数点输入*              if (cmd.matches("[0-9\\.]")) {                  if (startNewInput) {                      currentInput = cmd.equals(".") ? "0." : cmd;                      startNewInput = false;                  } else {  *// 防止重复小数点*                      if (cmd.equals(".") && currentInput.contains(".")) {                          return;                      }                      currentInput += cmd;                  }                  updateDisplay();              }              else if (cmd.equals("C")) {  *// 清除全部*                  memoryText = "";                  currentInput = "0";                  currentOperator = "";                  startNewInput = true;                  updateDisplay();              }              else if (cmd.equals("Exit")) {                  System.exit(0);              }              else if (cmd.equals("Hist")) {  *// 显示历史记录*                  if (history.isEmpty()) {                      JOptionPane.showMessageDialog(null, "No history!");                  } else {                      StringBuilder histText = new StringBuilder();                      for (String record : history) {                          histText.append(record).append("\n");                      }                      JTextArea textArea = new JTextArea(histText.toString());                      textArea.setEditable(false);                      JScrollPane scrollPane = new JScrollPane(textArea);                      scrollPane.setPreferredSize(new Dimension(400, 300));                      JOptionPane.showMessageDialog(null, scrollPane, "History", JOptionPane.INFORMATION\_MESSAGE);                  }              }              else if (cmd.equals("=")) {  *// 二元运算：取当前输入作为第二操作数，计算结果*                  try {                      double secondOperand = Double.parseDouble(currentInput);                      double result = compute(firstOperand, secondOperand, currentOperator);                      String expression = memoryText + currentInput + " = " + formatResult(result);                      history.add(expression);  *// 计算完成后，清空记忆部分，当前输入置为结果*                      currentInput = formatResult(result);                      memoryText = "";                      currentOperator = "";                      startNewInput = true;                      updateDisplay();                  } catch (Exception ex) {                      currentInput = "Error";                      memoryText = "";                      currentOperator = "";                      startNewInput = true;                      updateDisplay();                  }              }  *// 运算符：+ - \* / ^ 以及百分号（%）*              else if (cmd.equals("+") || cmd.equals("-") || cmd.equals("\*") || cmd.equals("/") || cmd.equals("^") || cmd.equals("%")) {                  try {                      firstOperand = Double.parseDouble(currentInput);                  } catch (Exception ex) {                      currentInput = "Error";                      updateDisplay();                      return;                  }                  currentOperator = cmd;  *// 将运算前的数和运算符保存在记忆部分，当前输入清空*                  memoryText = currentInput + " " + cmd + " ";                  startNewInput = true;                  currentInput = "";                  updateDisplay();              }  *// 单目运算：平方、立方、开方、n次根（n√）*              else if (cmd.equals("x²")) {                  try {                      double num = Double.parseDouble(currentInput);                      double result = num \* num;                      String exp = "sqr(" + num + ") = " + formatResult(result);                      history.add(exp);                      currentInput = formatResult(result);                      memoryText = "";                      startNewInput = true;                      updateDisplay();                  } catch (Exception ex) {                      currentInput = "Error";                      startNewInput = true;                      updateDisplay();                  }              }              else if (cmd.equals("x³")) {                  try {                      double num = Double.parseDouble(currentInput);                      double result = num \* num \* num;                      String exp = "cube(" + num + ") = " + formatResult(result);                      history.add(exp);                      currentInput = formatResult(result);                      memoryText = "";                      startNewInput = true;                      updateDisplay();                  } catch (Exception ex) {                      currentInput = "Error";                      startNewInput = true;                      updateDisplay();                  }              }              else if (cmd.equals("√")) {                  try {                      double num = Double.parseDouble(currentInput);                      if (num < 0) throw new ArithmeticException();                      double result = Math.sqrt(num);                      String exp = "sqrt(" + num + ") = " + formatResult(result);                      history.add(exp);                      currentInput = formatResult(result);                      memoryText = "";                      startNewInput = true;                      updateDisplay();                  } catch (Exception ex) {                      currentInput = "Error";                      startNewInput = true;                      updateDisplay();                  }              }              else if (cmd.equals("n√")) { *// n次根：输入根指数n，然后计算 num^(1/n)*                  try {                      double num = Double.parseDouble(currentInput);                      String input = JOptionPane.showInputDialog("Enter root index n:");                      if (input == null || input.trim().isEmpty()) return;                      double n = Double.parseDouble(input);                      if (n == 0) throw new ArithmeticException();                      if (n % 2 == 0 && num < 0) throw new ArithmeticException();                      double result = Math.pow(num, 1.0 / n);                      String exp = "root" + n + "(" + num + ") = " + formatResult(result);                      history.add(exp);                      currentInput = formatResult(result);                      memoryText = "";                      startNewInput = true;                      updateDisplay();                  } catch (Exception ex) {                      currentInput = "Error";                      startNewInput = true;                      updateDisplay();                  }              }          }      }  *// 根据操作符计算二元运算结果*      private double compute(double a, double b, String op) throws Exception {          switch (op) {              case "+":                  return a + b;              case "-":                  return a - b;              case "\*":                  return a \* b;              case "/":                  if (b == 0) throw new ArithmeticException();                  return a / b;              case "^":                  return Math.pow(a, b);              case "%":  *// 百分比运算：a % b 视为 a \* b/100*                  return a \* b / 100;              default:                  throw new Exception("Unknown operator");          }      }  *// 格式化数字，保留最多8位有效小数；当数字过大或过小时使用科学计数法*      private String formatResult(double num) {          DecimalFormat df = new DecimalFormat("0.########");          df.setRoundingMode(RoundingMode.HALF\_UP);  *// 如果绝对值过大或过小，则采用科学计数法*          if (num != 0 && (Math.abs(num) >= 1e8 || Math.abs(num) < 1e-4)) {              df = new DecimalFormat("0.########E0");              df.setRoundingMode(RoundingMode.HALF\_UP);          }          return df.format(num);      }      public static void main(String[] args) {          SwingUtilities.invokeLater(() -> {              new ScientificCalculator().setVisible(true);          });      }  } | |
| **运行结果：**   1. **运行代码，弹出的界面如下：**     2.运行整数计算  2.1整数乘法      **2.2整数除法（除不断保留8位有效数字）**      2.3整数n次方      2.4整数开n次方，对8开3次方      2.4整数除法（错误示范）      3.运行浮点数计算  3.1浮点数平方      3.2浮点数计算过大时会用科学计数法表示，如99^6等于941480149401，即9.41480149\*10^11      3.3浮点数开根号       1. 查看右下角Hist历史记录（刚才9/0错误示例不会存入到历史记录里面）        1. 点击右下角Exit或右上角x（叉叉）即可退出此程序，代码运行结束 | |
| **实验总结：**  **1.通过本次实验，我学习了Java Swing库的基本使用，包括布局管理器（如GridLayout、BorderLayout等）的使用，以及如何通过JButton、JTextField等组件构建交互式界面。让我对Java GUI开发的基本框架有了清晰的认识。 2.通过编写按钮的监听事件，我掌握了ActionListener接口的使用。当用户点击按钮时，如何通过监听器捕获事件，并实现相应的功能处理，比如数字输入、运算符选择，以及最终的计算结果展示。**  **3.在计算器功能实现过程中，我学习了如何对用户输入的数据进行解析，并通过逻辑判断实现平方、立方、n次方、开根号等运算。同时，通过异常处理机制，避免了如除零错误或输入非法字符导致的程序崩溃。**  **4.为了提升用户体验，我增加了历史记录功能，使用户能够查看之前的运算历史；设置了合理的数值格式化，使计算结果更加精确；同时，实现了输入记忆功能，增强了交互性。**  **5.在开发过程中，我遇到了浮点数运算误差问题，通过使用DecimalFormat类对结果进行格式化，成功解决了“2.2^2=4.8400000001”这样的误差。同时，通过捕获异常，避免了不合理操作导致的程序异常终止。**  **·** | |