山西工程技术学院

**实验报告**

（ 2024- 2025学年第一学期）

课程名称： Java程序设计

专业班级： 22计算机科学与技术一班

学 号： 2210708130

学生姓名： 郝泓毅

任课教师： 王晓霞

2024年9月

实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | **搭建Java的开发工具与Java程序的输入输出** | | | 指导教师 | 王晓霞 |
| 实验类型 | 验证型 | 实验学时 | 4 | 实验时间 |  |
| 1. **实验目的与要求**   了解JAVA虚拟机及JAVA运行系统；熟练掌握Java Applet程序和Java Application程序的编辑、编译和运行过程。 | | | | | |
| **二、实验环境**  操作系统：win-11  编程环境：visual stdio code | | | | | |
| **三、实验内容和步骤**  （1）搭建Java编程环境，练习Java Application和Java Applet的设计；  public class print{      public static void main(String [] args){          System.out.println("葡萄美酒夜光杯，欲饮琵琶马上催。\n" +                          "醉卧沙场君莫笑，古来征战几人回？\n");      }  }  实验结果：  import java.applet.Applet;  import java.awt.Graphics;  public class Myapplet extends Applet { | | | | | |
| public void paint(Graphics g) {          g.drawString("Hello, Applet!", 20, 20);      }  }  <html>      <body>          <APPLET CODE = "Myapplet.class" HEIGHT = 200 WIDTH = 300></APPLET>      </body>  </html>  实验结果： | | | | | |
| 1. **实验小结和思考** | | | | | |
| 实验成绩 |  | 批阅日期 |  | 批阅人 |  |

实验报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | **搭建Java的开发工具与Java程序的输入输出** | | | | | | 指导教师 | 王晓霞 |
| 实验类型 | 验证型 | 实验学时 | | 4 | | | 实验时间 |  |
| 1. **实验目的与要求**   了解JAVA虚拟机及JAVA运行系统；熟练掌握Java Applet程序和Java Application程序的编辑、编译和运行过程。 | | | | | | | | |
| **二、实验环境**  操作系统：win-11  编程环境：visual stdio code | | | | | | | | |
| **三、实验内容和步骤**  Java类的定义和使用；Java修饰符的使用；类的继承。编程实现时钟类；电话卡类，并测试。  参考代码：  时钟类  算法设计：  （1）声明一个名为Clock的时钟类：class Clock{ }  （2）添加成员变量时针、分针和秒针三个成员变量：int hour;int minute;int second;定义含三个参数的构造方法，与类名同名，将变量赋值给参数。  （3）定义方法prinTime()中实现对赋值的参数的输出。  （4）创建主函数类实现：Clock p1=new Clock(2,30,40);Clock p2=new Clock(4,50,10);Clock p3=new Clock(8,30,0);并通过调用prinTime()方法实现对算法的测试。  学生类  算法设计：  public class Clock {        int hour;      int minute;      int second;        public Clock(int hour, int minute, int second) {          this.hour = hour;          this.minute = minute;          this.second = second;      }      public void printTime() {          System.out.println("时间: " + hour + ":" + minute + ":" + second);      }      public static void main(String[] args) {          Clock p1 = new Clock(2, 30, 40);          Clock p2 = new Clock(4, 50, 10);          Clock p3 = new Clock(8, 30, 0);          p1.printTime();          p2.printTime();          p3.printTime();      }    }  （1）声明一个名为Student的学生类：class Student{ }  （2）添加成员变量姓名、学号和班级：int Id;String Name;String Banji;  （3）定义一个setter()的方法初始化成员变量：this.Id=1905000;this.Name=”李四”；this.Banji=”4班”；定义一个get()方法返回成员变量的值并输出到控制台。  （4）创建主函数类实现：Student s=new Student;通过调用setter()和get()对算法进行结果测试。  算法实现：（测试过程）  }  class Student {      int Id;      String Name;      String Banji;      public void setter() {          this.Id = 1905000;          this.Name = "李四";          this.Banji = "4班";      }      public void get() {          System.out.println("学号: " + Id);          System.out.println("姓名: " + Name);          System.out.println("班级: " + Banji);      }      public static void main(String[] args) {          Student s = new Student();          s.setter();          s.get();      }  }  实验结果： | | | | | | | | |
|  | | | | | | | | |
| 1. **实验小结和思考** | | | | | | | | |
| 实验成绩 |  | | 批阅日期 | |  | 批阅人 | |  |

实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | **图形用户界面的使用** | | | | | | 指导教师 | | 王晓霞 | |
| 实验类型 | 验证型 | | 实验学时 | | 4 | | 实验时间 | |  | |
| 1. **实验目的与要求**   了解Java 常用的布局策略及主要容器组件；学会图形界面的设计。 | | | | | | | | | | |
| **二、实验环境**  操作系统：win-11  编程环境：visual stdio code | | | | | | | | | | |
| **三、实验内容和步骤**  编程实现调色程序；四则计算器。包括：界面、控件、布局及菜单设计，各种控件的事件处理响应。重点在于图形界面元素的运用、JDK的事件处理机制的理解。    import javax.swing.\*;  import java.awt.\*;  import java.awt.event.\*;  public class Calculator extends JFrame {      private JLabel displayLabel;      private String currentInput = "";      public Calculator() {          setTitle("四则运算器");          setSize(400, 500);          setLocationRelativeTo(null);          setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);            JPanel panel = new JPanel();          panel.setLayout(new BorderLayout(10, 10));          displayLabel = new JLabel("0", SwingConstants.RIGHT);          displayLabel.setFont(new Font("Arial", Font.PLAIN, 40));          displayLabel.setBackground(Color.WHITE);          displayLabel.setOpaque(true);          displayLabel.setBorder(BorderFactory.createLineBorder(Color.GRAY, 1));          panel.add(displayLabel, BorderLayout.NORTH);          JPanel numberPanel = new JPanel();          numberPanel.setLayout(new GridLayout(5, 3, 5, 5));          String[] numbers = {"7", "8", "9", "4", "5", "6", "1", "2", "3", "0", ".", "="};          for (String number : numbers) {              JButton button = new JButton(number);              button.setFont(new Font("Arial", Font.PLAIN, 30));              button.setBackground(new Color(240, 240, 240));              button.setFocusPainted(false);              button.addActionListener(new NumberButtonHandler());              numberPanel.add(button);          }          panel.add(numberPanel, BorderLayout.CENTER);          JPanel functionPanel = new JPanel();          functionPanel.setLayout(new GridLayout(1, 4, 5, 5));          String[] operators = {"+", "-", "\*", "/"};          for (String operator : operators) {              JButton button = new JButton(operator);              button.setFont(new Font("Arial", Font.PLAIN, 30));              button.setBackground(new Color(255, 150, 100));              button.setFocusPainted(false);              button.addActionListener(new OperatorButtonHandler());              functionPanel.add(button);          }          JButton clearButton = new JButton("C");          clearButton.setFont(new Font("Arial", Font.PLAIN, 30));          clearButton.setBackground(new Color(255, 100, 100));          clearButton.setFocusPainted(false);          clearButton.addActionListener(new ClearButtonHandler());          functionPanel.add(clearButton);          panel.add(functionPanel, BorderLayout.SOUTH);          add(panel);      }      private class NumberButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              String buttonText = e.getActionCommand();              if (currentInput.equals("0")) {                  currentInput = buttonText;              } else {                  currentInput += buttonText;              }              displayLabel.setText(currentInput);          }      }      private class OperatorButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              String operator = e.getActionCommand();              if (!currentInput.isEmpty()) {                  currentInput += " " + operator + " ";                  displayLabel.setText(currentInput);              }          }      }      private class EqualButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              try {                  String result = evaluateExpression(currentInput);                  displayLabel.setText(result);                  currentInput = result;              } catch (Exception ex) {                  displayLabel.setText("错误");                  currentInput = "";              }          }      }      private class ClearButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              currentInput = "";              displayLabel.setText("0");          }      }      private String evaluateExpression(String expression) {          try {              javax.script.ScriptEngineManager manager = new javax.script.ScriptEngineManager();              javax.script.ScriptEngine engine = manager.getEngineByName("JavaScript");              return String.valueOf(engine.eval(expression));          } catch (Exception e) {              return "错误";          }      }      public static void main(String[] args) {          Calculator calculator = new Calculator();          calculator.setVisible(true);      }  }    } | | | | | | | | | | |
|  | | | | | | | | | | | | |
| 1. **实验小结和思考** | | | | | | | | | | | | |
| 实验成绩 | | |  | | 批阅日期 | |  | | 批阅人 | |  | |

实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | **数据库链接** | | | 指导教师 | 王晓霞 |
| 实验类型 | 验证型 | 实验学时 | 4 | 实验时间 |  |
| 1. **实验目的与要求**   学会用JDBC—ODBC桥链接进行数据库链接；学会用纯Java数据库驱动程序方式连接数据库；同时考虑异常处理。 | | | | | |
| **二、实验环境**  操作系统：win-11  编程环境：visual stdio code | | | | | |
| **三、实验内容和步骤**  使用JDBC-ODBC连接和本地纯JAVA数据库链接链接数据库，并查询student数据表的所有记录，将查询结果显示到屏幕上。    import javax.swing.\*;  import java.awt.\*;  import java.awt.event.\*;  public class Calculator extends JFrame {      private JLabel displayLabel;      private String currentInput = "";      public Calculator() {          setTitle("四则运算器");          setSize(400, 500);          setLocationRelativeTo(null);          setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);            JPanel panel = new JPanel();          panel.setLayout(new BorderLayout(10, 10));          displayLabel = new JLabel("0", SwingConstants.RIGHT);          displayLabel.setFont(new Font("Arial", Font.PLAIN, 40));          displayLabel.setBackground(Color.WHITE);          displayLabel.setOpaque(true);          displayLabel.setBorder(BorderFactory.createLineBorder(Color.GRAY, 1));          panel.add(displayLabel, BorderLayout.NORTH);          JPanel numberPanel = new JPanel();          numberPanel.setLayout(new GridLayout(5, 3, 5, 5));          String[] numbers = {"7", "8", "9", "4", "5", "6", "1", "2", "3", "0", ".", "="};          for (String number : numbers) {              JButton button = new JButton(number);              button.setFont(new Font("Arial", Font.PLAIN, 30));              button.setBackground(new Color(240, 240, 240));              button.setFocusPainted(false);              button.addActionListener(new NumberButtonHandler());              numberPanel.add(button);          }          panel.add(numberPanel, BorderLayout.CENTER);          JPanel functionPanel = new JPanel();          functionPanel.setLayout(new GridLayout(1, 4, 5, 5));          String[] operators = {"+", "-", "\*", "/"};          for (String operator : operators) {              JButton button = new JButton(operator);              button.setFont(new Font("Arial", Font.PLAIN, 30));              button.setBackground(new Color(255, 150, 100));              button.setFocusPainted(false);              button.addActionListener(new OperatorButtonHandler());              functionPanel.add(button);          }          JButton clearButton = new JButton("C");          clearButton.setFont(new Font("Arial", Font.PLAIN, 30));          clearButton.setBackground(new Color(255, 100, 100));          clearButton.setFocusPainted(false);          clearButton.addActionListener(new ClearButtonHandler());          functionPanel.add(clearButton);          panel.add(functionPanel, BorderLayout.SOUTH);          add(panel);      }      private class NumberButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              String buttonText = e.getActionCommand();              if (currentInput.equals("0")) {                  currentInput = buttonText;              } else {                  currentInput += buttonText;              }              displayLabel.setText(currentInput);          }      }      private class OperatorButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              String operator = e.getActionCommand();              if (!currentInput.isEmpty()) {                  currentInput += " " + operator + " ";                  displayLabel.setText(currentInput);              }          }      }      private class EqualButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              try {                  String result = evaluateExpression(currentInput);                  displayLabel.setText(result);                  currentInput = result;              } catch (Exception ex) {                  displayLabel.setText("错误");                  currentInput = "";              }          }      }      private class ClearButtonHandler implements ActionListener {          public void actionPerformed(ActionEvent e) {              currentInput = "";              displayLabel.setText("0");          }      }      private String evaluateExpression(String expression) {          try {              javax.script.ScriptEngineManager manager = new javax.script.ScriptEngineManager();              javax.script.ScriptEngine engine = manager.getEngineByName("JavaScript");              return String.valueOf(engine.eval(expression));          } catch (Exception e) {              return "错误";          }      }      public static void main(String[] args) {          Calculator calculator = new Calculator();          calculator.setVisible(true);      }  }    } | | | | | |