实验六 图形界面

实验日期： 2017 年 4 月 28 日 班级： 软件1501

学号（后四位）： 0314 姓名： 马亚卿 成绩：

**一．实验目的**

1. 掌握图形用户界面的设计方法
2. 掌握常用的构建用户界面的组件的用法
3. 掌握事件操作的原理
4. 能够对所设计的用户界面进行事件处理
5. 能够应用运算符解决实际小问题
6. 进一步熟悉Java的面向对象的编程思想

**二．实验题目（前2题任选1题，第3，4题任选1题）**

1.采用图形界面实现两个内容的交换，图形界面如下图1所示所示：



图1 内容交换

2. 采用图形界面设计如下图2所示的界面。并能够实现当点击“+”按钮时，标签给出点击次数，如图3所示

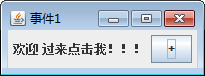


图2 开始界面

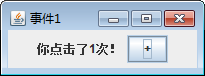


图3 点击“+”按钮后的界面

代码如下：

|  |
| --- |
| package shiyan6;  import java.awt.Container;  import java.awt.event.MouseAdapter;  import java.awt.event.MouseEvent;  import javax.swing.JButton;  import javax.swing.JFrame;  import javax.swing.JLabel;  import javax.swing.JPanel;  public class ClickTimes extends JFrame {  private JLabel jl;  private JPanel jp;  private JButton jb;  private int times = 0;  public ClickTimes(){  jl = new JLabel("欢迎过来点击我！");  jb = new JButton("＋");  jb.addMouseListener(new lis());  jp = new JPanel();  jp.add(jl);  jp.add(jb);  Container c = getContentPane();  c.add(jp);  setTitle("事件1");  setVisible(true);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setSize(400,200);  setLocation(400,300);  }  class lis extends MouseAdapter{  @Override  public void mouseClicked(MouseEvent e) {  times++;  jl.setText("鼠标点击了" + times + "次");  }  }      public static void main(String[] args) {  new ClickTimes();  }  } |

运行结果如图1-2

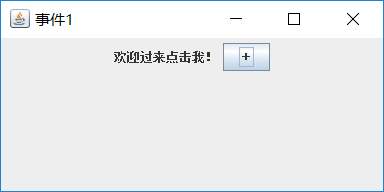


图1 开始界面

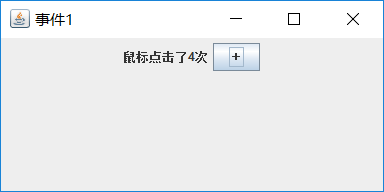


图2 点击‘+’按钮之后的界面

3．教材P220页，第3题，创建一个会员注册窗体，要求使用上JCheckBox,JRadioButton,JTextFiled,JTextArea等组件，并将用户的注册信息显示到文本区中。

代码如下：

|  |
| --- |
| package shiyan6;  import java.awt.\*;  import java.awt.event.\*;  import javax.swing.\*;  import javax.xml.transform.Source;  public class VIPRegister extends JFrame implements ItemListener {  private JLabel[] info;  private JTextField[] infoText;  private JButton jbok;  private JButton jbcancel;  private JPanel jp1;  private JPanel jp2;  private JPanel jp3;  private JPanel jp4;  private JRadioButton jrbAccept;  private Font g;  public VIPRegister() {  g = new Font("宋体", Font.BOLD, 30);  /\* jp1 注册信息的各项种类 \*/  info = new JLabel[4];  info[0] = new JLabel("电话", SwingConstants.RIGHT);  info[1] = new JLabel("账号", SwingConstants.RIGHT);  info[2] = new JLabel("密码", SwingConstants.RIGHT);  info[3] = new JLabel("确认密码", SwingConstants.RIGHT);  infoText = new JTextField[4];  jp1 = new JPanel();  jp1.setLayout(new GridLayout(4, 1, 5, 5));  jp2 = new JPanel();  jp2.setLayout(new GridLayout(4, 1, 5, 5));  for (int i = 0; i < 4; i++) {  infoText[i] = new JTextField(15);  infoText[i].setFont(g);  info[i].setFont(g);  jp1.add(info[i]);  jp2.add(infoText[i]);  }  jp3 = new JPanel();  jp3.add(jp1);  jp3.add(jp2);  Font g1 = new Font("宋体", Font.BOLD, 20);  JCheckBox jcb1 = new JCheckBox("到期自动续费", true);  JCheckBox jcb2 = new JCheckBox("加入客户体验计划", true);  jrbAccept = new JRadioButton("我接受用户协议");  JRadioButton jrbRefuse = new JRadioButton("我拒绝用户协议", true);  jbok = new JButton("提交");  jbok.addActionListener(new lis());  jbok.setEnabled(false);  jbcancel = new JButton("重置");  jbcancel.addActionListener(new lis());  jcb1.setFont(g1);  jcb2.setFont(g1);  jrbAccept.setFont(g1);  jrbAccept.addItemListener(this);  jrbRefuse.setFont(g1);  jbok.setFont(g);  jbcancel.setFont(g);  ButtonGroup bg = new ButtonGroup();  bg.add(jrbAccept);  bg.add(jrbRefuse);  jp4 = new JPanel();  jp4.setLayout(new GridLayout(3, 2, 5, 5));  jp4.add(jcb1);  jp4.add(jcb2);  jp4.add(jrbAccept);  jp4.add(jrbRefuse);  jp4.add(jbok);  jp4.add(jbcancel);  Container c = getContentPane();  c.setLayout(new GridLayout(2, 1));  c.add(jp3);  c.add(jp4);  setTitle("请注册会员信息");  setVisible(true);  pack();  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setLocation(500, 300);  }  public class lis implements ActionListener {  @SuppressWarnings("deprecation")  @Override  public void actionPerformed(ActionEvent e) {  Object source = e.getSource();  boolean flag = true;  if (source == jbok) {  for (int i = 0; i < 4; i++) {  if (infoText[i].getText().length() == 0) {  JOptionPane.showMessageDialog(getParent(),  "请输入" + info[i].getText() + "信息！", "错误信息",  JOptionPane.WARNING\_MESSAGE);  infoText[i].requestFocus();  flag = false;  break;  }  }  if (infoText[0].getText().length() != 11  || !infoText[0].getText().matches("1[0-9]{10}")) {  flag = false;  JOptionPane.showMessageDialog(getParent(),  "请输入正确的电话号码！", "错误信息", JOptionPane.WARNING\_MESSAGE);  infoText[0].selectAll();  infoText[0].requestFocus();  }  if (infoText[1].getText().length() < 6  || infoText[1].getText().contains(" ")) {  flag = false;  JOptionPane.showMessageDialog(getParent(),  "账号是不包含空格的6位以上数字或字母", "错误信息",  JOptionPane.WARNING\_MESSAGE);  infoText[1].selectAll();  }  if (infoText[2].getText().length() < 6  || infoText[2].getText().contains(" ")) {  flag = false;  JOptionPane.showMessageDialog(getParent(),  "密码是不包含空格的6位以上数字或字母", "错误信息",  JOptionPane.WARNING\_MESSAGE);  infoText[2].selectAll();  }  if (infoText[2].getText().compareTo(infoText[3].getText()) != 0) {  flag = false;  JOptionPane.showMessageDialog(getParent(),  "两次密码不同，请重新输入！", "错误信息", JOptionPane.WARNING\_MESSAGE);  infoText[2].setText("");  infoText[3].setText("");  infoText[2].requestFocus();  }  if (flag==true) {  JDialog jd = new JDialog();  jd.setTitle("注册信息");  jd.setLocationRelativeTo(info[0]);  jd.setSize(300, 300);  Container cd = jd.getContentPane();  JTextArea jta = new JTextArea(4, 1);  jta.setEditable(false);  jta.setFont(g);  cd.setVisible(true);  cd.setLayout(new FlowLayout());  cd.add(jta);  JButton jb = new JButton("确定");  jb.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  jd.setVisible(false);  }  });  cd.add(jb);  jta.append("您的注册信息为：\n");  jta.append("账号：" + infoText[1].getText() + "\n");  jta.append("密码：" + infoText[2].getText() + "\n");  jta.append("请牢记您的信息");  jd.show();  }  } else {  for (int i = 0; i < 4; i++) {  infoText[i].setText("");  }  infoText[0].requestFocus();  }  }  }  public static void main(String[] args) {  new VIPRegister();  }  @Override  public void itemStateChanged(ItemEvent e) {  if (jrbAccept.isSelected()) {  jbok.setEnabled(true);  } else {  jbok.setEnabled(false);  }  }  } |

运行结果如图3-5

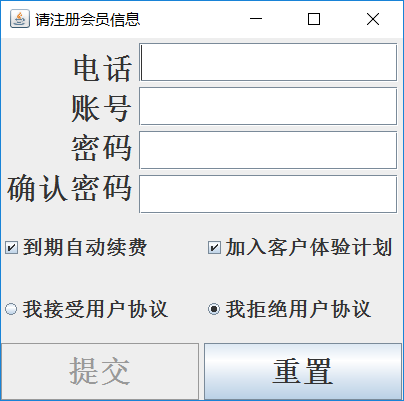


图3开始界面

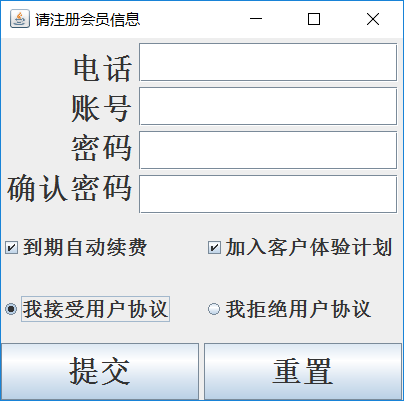


图4 点击‘我接受用户协议后’的界面

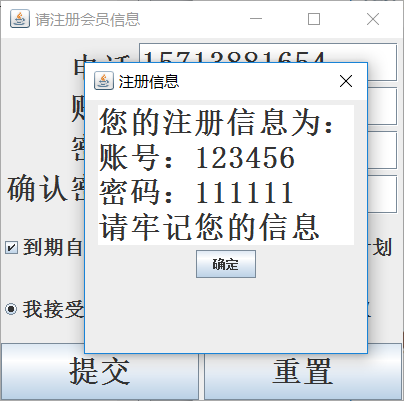


图5 提交信息后的界面

4． 利用相应的组件编辑一个文本编辑器，参照Windows记事本，可以对字体大小、字形（粗体，斜体）、颜色进行设置，可以对内容进行添加、删除、修改，并保存文本。

说明：（1）可以选用单选框、复选框、列表、下拉框或菜单任意一种或多种组合实现字体的设置

（2）根据自己的编程能力实现功能，可以只实现字体的设置，或者字体和内容的编辑，或者全部功能都实现。

（3）本题的目的是常用组件的使用。

**III 在本次实验中的问题和解决方法**

问题：各级的容器布局不合理。

解决办法：使用容器嵌套的方法优化布局。

**IV心得体会**

在做第三题的时候，刚开始只是在JFrame中进行布局了各个组件，但是每块大小一样，很不美观。后来经过思考，将注册信息分别进行分类，将类别一样且显示效果一样的放在同一个容器中，然后放入JFrame组件中。这样经过优化，美观了许多。而且思路也清晰了。

这次试验让我初步了解了图形用户界面的设计方法。提高了对事件的认识以及了解事件的处理方法。不仅如此，让我对面向对象编程有了深度的了解。

在看到自己的代码经过用户界面的效果展示出来之后，我感到了与以往不同的成就感，因为感觉自己的作品变成了立体的、可以触摸的一样。今后有了更大的动力学习编程。