**密码生成器**

**344121138 陆楠**

package BitOperation;  
  
import java.awt.BorderLayout;  
import java.awt.Color;  
import java.awt.FlowLayout;  
import java.awt.GridLayout;  
import java.awt.LayoutManager;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JPanel;  
import javax.swing.JTextField;  
  
public class BuildSecret extends JFrame implements ActionListener{  
 // 1. 定义全局变量 全局对象  
 // 制作一个标签 放置标题  
 JLabel lblTitle = new JLabel("密码生成", JLabel.CENTER);  
   
 // 标签数组  
 JLabel [] lbl = new JLabel[7];  
 String [] lblStr = {"请输入源码1:", "请输入源码2:", "请输入源码3:", "请输入源码4:", "请输入源码5:", "请输入源码6:", "生成密码:" };  
  
 // 文本框数组  
 JTextField [] txt = new JTextField[7];  
  
   
 // 按钮数组  
 JButton [] btn = new JButton[3];  
 String [] btnStr = {"Create", "Clear", "Exit"};  
  
   
 // 面板数组  
 JPanel [] pnl = new JPanel[4];  
 Color [] c = {Color.BLUE, Color.pink, Color.green, Color.red};  
 LayoutManager [] lm = {  
 new BorderLayout(5, 5), // 东西南北中布局  
 new FlowLayout(FlowLayout.CENTER, 5, 5), // 自左向右 自上而下  
 new GridLayout(7, 2, 5, 5), // 7行2列表格  
 new FlowLayout(FlowLayout.CENTER, 5, 5) // 自左向右 自上而下  
 };  
 String[] posStr = {"North", "Center", "South"}; // 三个子面板的方位数组  
   
   
 // 2. 书写构造方法 为所有标签、文本框、按钮初始化  
 public BuildSecret() {  
 // 为4个面板初始化  
 for (int i = 0; i < pnl.length; i++) {  
 pnl[i] = new JPanel(lm[i]); // 为4个面板上分别设置4个布局  
 pnl[i].setBackground(c[i]); // 为4个面板上分别设置4个颜色  
 }  
   
 // 北面板上放置标题  
 pnl[1].add(lblTitle);  
   
 // 中面板上放置7个标签及7个文本框  
 for (int i = 0; i < lbl.length; i++) {  
 lbl[i] = new JLabel(lblStr[i], JLabel.CENTER); // 在7个标签上分别放置7个标题  
 txt[i] = new JTextField(20); // 初始化7个文本框  
 if(i == 6)  
 txt[i].setEditable(false);  
 }  
 pnl[2].add(lbl[0]);  
 pnl[2].add(txt[0]);  
 pnl[2].add(lbl[1]);  
 pnl[2].add(txt[1]);  
 pnl[2].add(lbl[2]);  
 pnl[2].add(txt[2]);  
 pnl[2].add(lbl[3]);  
 pnl[2].add(txt[3]);  
 pnl[2].add(lbl[4]);  
 pnl[2].add(txt[4]);  
 pnl[2].add(lbl[5]);  
 pnl[2].add(txt[5]);  
 pnl[2].add(lbl[6]);  
 pnl[2].add(txt[6]);  
   
 // 在南面板上放置3个按钮  
 for (int i = 0; i < btn.length; i++) {  
 btn[i] = new JButton(btnStr[i]); // 在三个按钮上分别放置3个标题  
 btn[i].addActionListener(this); // 为3个按钮添加监听器  
 pnl[3].add(btn[i]); // 在南面板依次放入按钮  
 }  
   
 // 在主面板上放置3个子面板  
 for (int i = 0; i < pnl.length - 1; i++) {  
 pnl[0].add(posStr[i], pnl[i + 1]); // 在主面板上依次放入子面板  
 }  
   
 // 在窗体网格上放置主面板  
 this.setContentPane(pnl[0]);  
 }  
   
  
 // 3. 新建整个程序的入口 调用超类方法显示整个窗口内容  
 public static void main(String [] args) {  
 BuildSecret mc = new BuildSecret();  
 mc.setTitle("密码生成系统"); // 窗口标题  
 mc.setBounds(700, 400, 400, 350); // 窗口大小  
 mc.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 mc.setVisible(true);  
 }  
   
 // 20021230左移2 非(19780403异或 19781101) (20010909或 20000607) 与 (20030114右移1)  
 // 4. 重写超类的抽象方法 实现程序的算法  
 @Override  
 public void actionPerformed(ActionEvent ae) {  
 // TODO Auto-generated method stub  
 // 与 或 非 异或 左移 右移  
 // (20021230 或 19780403) 与 (19781101左移2位) 或 (非20010909) 与 (20000607) 异或 (20030114右移1位)  
 if (ae.getSource() == btn[0]) {  
 int[] num = new int[6];  
 for (int i = 0; i < num.length; i++) {  
 num[i] = Integer.parseInt(txt[i].getText());  
 }  
  
 int sec = (num[0] | num[1]) & (num[2] << 2) | (~num[3]) & (num[4]) ^ (num[5] >> 1);  
  
 String secret = Integer.toString(sec);  
   
 txt[6].setText(secret);  
 }  
 else if (ae.getSource() == btn[1]) {  
 for (int i = 0; i < txt.length; i++) {  
 txt[i].setText(""); // 清空1个文本框  
 }  
 txt[0].requestFocus(); // 将光标移回第一个文本框  
 } else System.exit(0); // 退出程序   
 }  
  
}