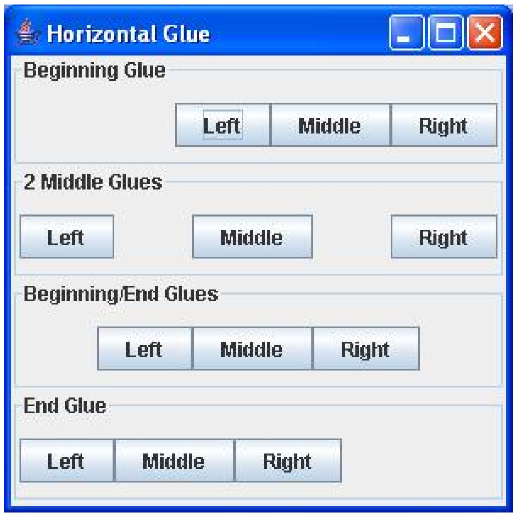
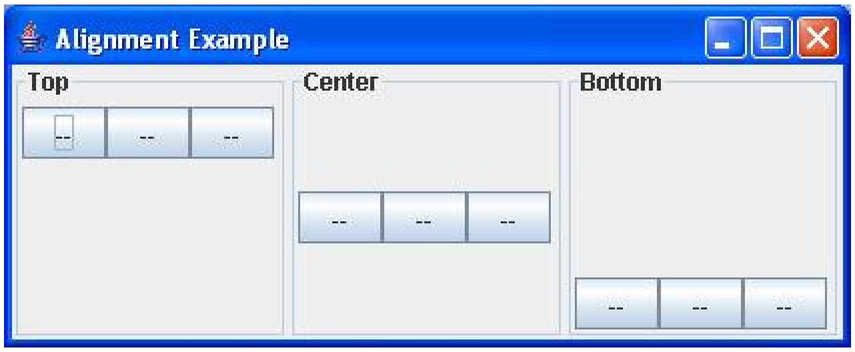
1. A sample of using GridLayout and BorderLayout



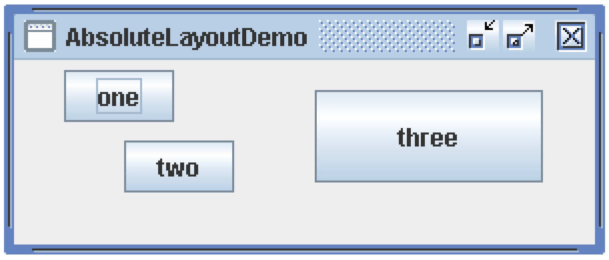
```
/*
Definitive Guide to Swing for Java 2, Second Edition
By John Zukowski
ISBN: 1-893115-78-X
Publisher: APress
*/

import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.GridLayout;

import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class GlueSample {
  public static void main(String args[]) {
    Box horizontalBox;
    JPanel panel;
    JFrame frame = new JFrame("Horizontal Glue");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    Container contentPane = frame.getContentPane();
    contentPane.setLayout(new GridLayout(0, 1));

    horizontalBox = Box.createHorizontalBox();
    horizontalBox.add(Box.createGlue());
    horizontalBox.add(new JButton("Left"));
    horizontalBox.add(new JButton("Middle"));
    horizontalBox.add(new JButton("Right"));
    panel = new JPanel(new BorderLayout());
    panel.add(horizontalBox);
    panel.setBorder(BorderFactory.createTitledBorder("Beginning Glue"));
    contentPane.add(panel);

    horizontalBox = Box.createHorizontalBox();
    horizontalBox.add(new JButton("Left"));
    horizontalBox.add(Box.createGlue());
    horizontalBox.add(new JButton("Middle"));
    horizontalBox.add(Box.createGlue());
    horizontalBox.add(new JButton("Right"));
    panel = new JPanel(new BorderLayout());
    panel.add(horizontalBox);
    panel.setBorder(BorderFactory.createTitledBorder("2 Middle Glues"));
    contentPane.add(panel);

    horizontalBox = Box.createHorizontalBox();
    horizontalBox.add(Box.createGlue());
    horizontalBox.add(new JButton("Left"));
    horizontalBox.add(new JButton("Middle"));
    horizontalBox.add(new JButton("Right"));
    horizontalBox.add(Box.createGlue());
    panel = new JPanel(new BorderLayout());
    panel.add(horizontalBox);
    panel
        .setBorder(BorderFactory
            .createTitledBorder("Beginning/End Glues"));
    contentPane.add(panel);

    horizontalBox = Box.createHorizontalBox();
    horizontalBox.add(new JButton("Left"));
    horizontalBox.add(new JButton("Middle"));
    horizontalBox.add(new JButton("Right"));
    panel = new JPanel(new BorderLayout());
    horizontalBox.add(Box.createGlue());
    panel.add(horizontalBox);
    panel.setBorder(BorderFactory.createTitledBorder("End Glue"));
    contentPane.add(panel);

    frame.setSize(300, 300);
    frame.setVisible(true);
  }
}
```

2. A sample of using GridLayout



```
import java.awt.Component;
import java.awt.Container;
import java.awt.GridLayout;

import javax.swing.BorderFactory;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class XAxisAlignY {
  private static Container makeIt(String title, float alignment) {
    String labels[] = { "--", "--", "--" };

    JPanel container = new JPanel();
    container.setBorder(BorderFactory.createTitledBorder(title));
    BoxLayout layout = new BoxLayout(container, BoxLayout.X_AXIS);
    container.setLayout(layout);

    for (int i = 0, n = labels.length; i < n; i++) {
      JButton button = new JButton(labels[i]);
      button.setAlignmentY(alignment);
      container.add(button);
    }
    return container;
  }

  public static void main(String args[]) {
    JFrame frame = new JFrame("Alignment Example");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    Container panel1 = makeIt("Top", Component.TOP_ALIGNMENT);
    Container panel2 = makeIt("Center", Component.CENTER_ALIGNMENT);
    Container panel3 = makeIt("Bottom", Component.BOTTOM_ALIGNMENT);

    Container contentPane = frame.getContentPane();
    contentPane.setLayout(new GridLayout(1, 3));
    contentPane.add(panel1);
    contentPane.add(panel2);
    contentPane.add(panel3);

    frame.setSize(423, 171);
    frame.setVisible(true);
  }
}
```

3. A sample of using absolute positioning technique to layout GUI (null layout)



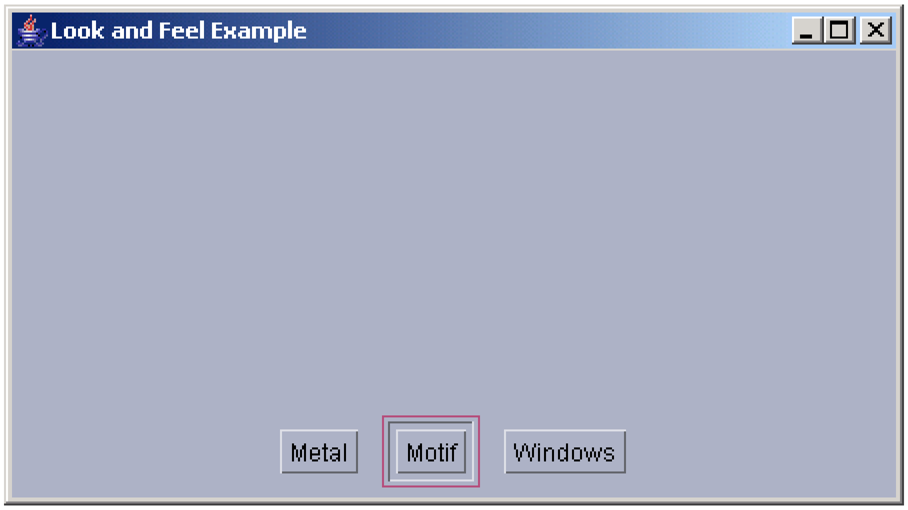
```
import java.awt.Container;
import java.awt.Dimension;
import java.awt.Insets;

import javax.swing.JButton;
import javax.swing.JFrame;

public class AbsoluteLayoutDemo {
  public static void addComponentsToPane(Container pane) {
    pane.setLayout(null);

    JButton b1 = new JButton("one");
    JButton b2 = new JButton("two");
    JButton b3 = new JButton("three");

    pane.add(b1);
    pane.add(b2);
    pane.add(b3);

    Insets insets = pane.getInsets();
    Dimension size = b1.getPreferredSize();
    b1.setBounds(25 + insets.left, 5 + insets.top, size.width, size.height);
    size = b2.getPreferredSize();
    b2
        .setBounds(55 + insets.left, 40 + insets.top, size.width,
            size.height);
    size = b3.getPreferredSize();
    b3.setBounds(150 + insets.left, 15 + insets.top, size.width + 50,
        size.height + 20);
  }

  /**
   * Create the GUI and show it. For thread safety, this method should be
   * invoked from the event-dispatching thread.
   */
  private static void createAndShowGUI() {
    //Make sure we have nice window decorations.
    JFrame.setDefaultLookAndFeelDecorated(true);

    //Create and set up the window.
    JFrame frame = new JFrame("AbsoluteLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //Set up the content pane.
    addComponentsToPane(frame.getContentPane());

    //Size and display the window.
    Insets insets = frame.getInsets();
    frame.setSize(300 + insets.left + insets.right, 125 + insets.top
        + insets.bottom);
    frame.setVisible(true);
  }

  public static void main(String[] args) {
    //Schedule a job for the event-dispatching thread:
    //creating and showing this application's GUI.
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        createAndShowGUI();
      }
    });
  }
}
```

4. A program enables to change the face of it.

```
1.   import java.awt.*;
2.   import java.awt.event.*;
3.   import javax.swing.*;
4.
```

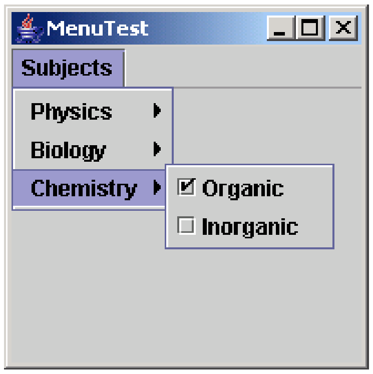
```
public class LookFeel extends JFrame implements ActionListener
{
    JButton btnMetal,  btnMotif, btnWindows;

    public LookFeel()
    {
        super("Look and Feel Example");

        getContentPane().setLayout(new BorderLayout());

        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
    });

        JPanel pnlLookFeel = new JPanel();

        btnMetal = new JButton("Metal");
        btnMetal.addActionListener(this);
        pnlLookFeel.add(btnMetal);

        btnMotif = new JButton("Motif");
        btnMotif.addActionListener(this);
        pnlLookFeel.add(btnMotif);

        btnWindows = new JButton("Windows");
        btnWindows.addActionListener(this);
        pnlLookFeel.add(btnWindows);

        getContentPane().add(pnlLookFeel, BorderLayout.SOUTH);

        setSize(450, 250);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)
    {

        String strLookFeel = null;

        if (ae.getActionCommand().equals("Metal"))
        {
            strLookFeel = "javax.swing.plaf.metal.MetalLookAndFeel";
        }
        else if (ae.getActionCommand().equals("Windows"))
        {
            strLookFeel = "com.sun.java.swing.plaf.windows.WindowsLookAndFeel";
    }
    else if (ae.getActionCommand().equals("Motif"))
    {
        strLookFeel = "com.sun.java.swing.plaf.motif.MotifLookAndFeel";
    }
    else
    {
        System.err.println("Unrecognized L&F request action: " + ae.getActionCommand());
        return;
    }

        try
    {
        UIManager.setLookAndFeel(strLookFeel);
            SwingUtilities.updateComponentTreeUI(this);
    }
    catch(UnsupportedLookAndFeelException ex1)
    {
        System.err.println("Unsupported LookAndFeel: " + strLookFeel);
    }
    catch(ClassNotFoundException ex2)
    {
        System.err.println("LookAndFeel class not found: " + strLookFeel);
    }
    catch(InstantiationException ex3)
    {
        System.err.println("Could not load LookAndFeel: " + strLookFeel);
    }
    catch(IllegalAccessException ex4)
    {
        System.err.println("Cannot use LookAndFeel: " + strLookFeel);
        }
  }

    public static void main(String args[])
  {
    LookFeel objLookFeel = new LookFeel();
  }
}
```

The result:



5. A frame with menu structure:

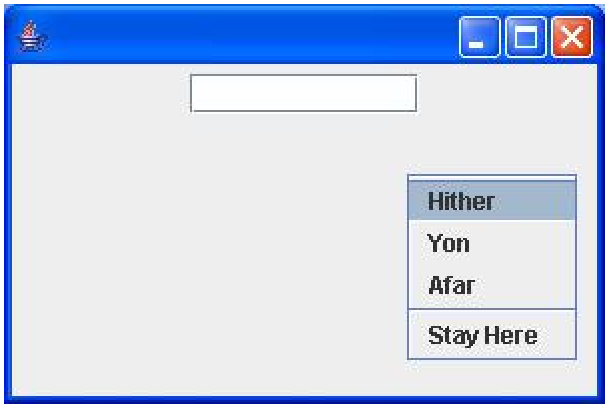
Subjects
- *Physics*
  - *Metaphysics*
  - *Astrophysics*
- *Biology*
  - *Microbiology*
  - *Biotechnology*
- *Chemistry*
  - *Organic*
  - *Inorganic*



```
import javax.swing.*;
import java.awt.*;

public class MenuTest extends JFrame
{
    MenuTest()
    {
        JMenuBar mb = new JMenuBar();
        JMenu subjects = new JMenu("Subjects");

        JMenu submenu1 = new JMenu("Physics");
        submenu1.add(new JCheckBoxMenuItem("Metaphysics"));
        submenu1.add(new JCheckBoxMenuItem("Astrophysics"));
        subjects.add(submenu1);

        submenu1 = new JMenu("Biology");
        submenu1.add(new JCheckBoxMenuItem("Microbiology"));
        submenu1.add(new JCheckBoxMenuItem("Biotechnology"));
        subjects.add(submenu1);
        submenu1 = new  JMenu("Chemistry");
        submenu1.add(new JCheckBoxMenuItem("Organic"));
        submenu1.add(new JCheckBoxMenuItem("Inorganic"));
        subjects.add(submenu1);

        mb.add(subjects);
        setJMenuBar(mb);
        setTitle("MenuTest");
        setSize(200,200);
        setVisible(true);
    }

    public static void main(String[] args)
    {
        MenuTest objMenuTest = new MenuTest();
    }
}
```

2. A sample about Popup menu

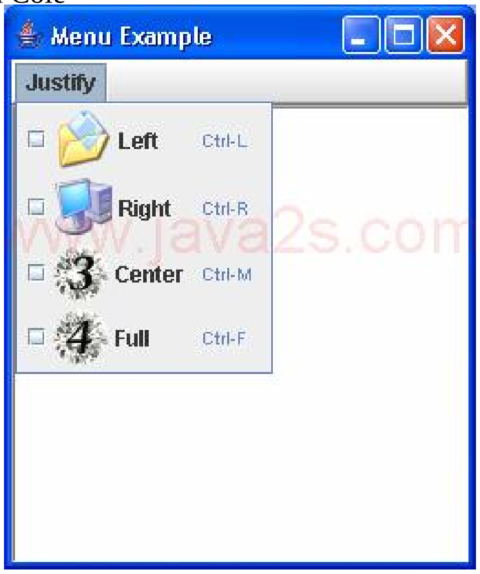
```
// : c14:Popup.java
// Creating popup menus with Swing.
// <applet code=Popup width=300 height=200></applet>
// From 'Thinking in Java, 3rd ed.' (c) Bruce Eckel 2002
// www.BruceEckel.com. See copyright notice in CopyRight.txt.

import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

import javax.swing.JApplet;
import javax.swing.JFrame;
import javax.swing.JMenuItem;
import javax.swing.JPopupMenu;
import javax.swing.JTextField;
```



```
public class Popup extends JApplet {
  private JPopupMenu popup = new JPopupMenu();

  private JTextField t = new JTextField(10);

  public void init() {
    Container cp = getContentPane();
    cp.setLayout(new FlowLayout());
    cp.add(t);
    ActionListener al = new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        t.setText(((JMenuItem) e.getSource()).getText());
      }
    };
    JMenuItem m = new JMenuItem("Hither");
    m.addActionListener(al);
    popup.add(m);
    m = new JMenuItem("Yon");
    m.addActionListener(al);
    popup.add(m);
    m = new JMenuItem("Afar");
    m.addActionListener(al);
    popup.add(m);
    popup.addSeparator();
    m = new JMenuItem("Stay Here");
    m.addActionListener(al);
    popup.add(m);
    PopupListener pl = new PopupListener();
    addMouseListener(pl);
    t.addMouseListener(pl);
  }

  class PopupListener extends MouseAdapter {
    public void mousePressed(MouseEvent e) {
      maybeShowPopup(e);
    }

    public void mouseReleased(MouseEvent e) {
      maybeShowPopup(e);
    }

    private void maybeShowPopup(MouseEvent e) {
      if (e.isPopupTrigger())
        popup.show(((JApplet) e.getComponent()).getContentPane(), e
            .getX(), e.getY());
    }
  }

  public static void main(String[] args) {
    run(new Popup(), 300, 200);
  }

  public static void run(JApplet applet, int width, int height) {
    JFrame frame = new JFrame();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add(applet);
    frame.setSize(width, height);
    applet.init();
    applet.start();
```

```
    frame.setVisible(true);
  }
} ///:~
```

3. A sample about menu with icons



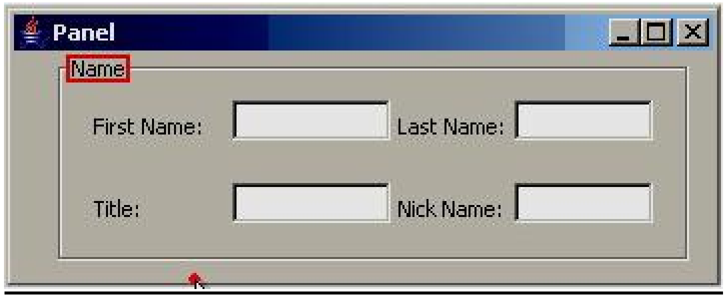
```
/*
Java Swing, 2nd Edition
By Marc Loy, Robert Eckstein, Dave Wood, James Elliott, Brian Cole
ISBN: 0-596-00408-7
Publisher: O'Reilly
*/
// CheckBoxMenuItemExample.java
// A quick demonstration of checkbox menu items.
//

import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ImageIcon;
import javax.swing.JCheckBoxMenuItem;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JTextPane;
import javax.swing.JToolBar;
import javax.swing.KeyStroke;
import javax.swing.border.BevelBorder;

public class CheckBoxMenuItemExample extends JPanel {
  public JTextPane pane;

  public JMenuBar menuBar;

  public JToolBar toolBar;

  public CheckBoxMenuItemExample() {
    menuBar = new JMenuBar();
    JMenu justifyMenu = new JMenu("Justify");
    ActionListener actionPrinter = new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        try {
          pane.getStyledDocument().insertString(
              0,
              "Action [" + e.getActionCommand()
                + "] performed!\n", null);
        } catch (Exception ex) {
          ex.printStackTrace();
        }
      }
    };
    JCheckBoxMenuItem leftJustify = new JCheckBoxMenuItem("Left",
        new ImageIcon("1.gif"));
    leftJustify.setHorizontalTextPosition(JMenuItem.RIGHT);
    leftJustify.setAccelerator(KeyStroke.getKeyStroke('L', Toolkit
        .getDefaultToolkit().getMenuShortcutKeyMask()));
    leftJustify.addActionListener(actionPrinter);
    JCheckBoxMenuItem rightJustify = new JCheckBoxMenuItem("Right",
        new ImageIcon("2.gif"));
    rightJustify.setHorizontalTextPosition(JMenuItem.RIGHT);
    rightJustify.setAccelerator(KeyStroke.getKeyStroke('R', Toolkit
        .getDefaultToolkit().getMenuShortcutKeyMask()));
    rightJustify.addActionListener(actionPrinter);
    JCheckBoxMenuItem centerJustify = new JCheckBoxMenuItem("Center",
        new ImageIcon("3.gif"));
    centerJustify.setHorizontalTextPosition(JMenuItem.RIGHT);
    centerJustify.setAccelerator(KeyStroke.getKeyStroke('M', Toolkit
        .getDefaultToolkit().getMenuShortcutKeyMask()));
    centerJustify.addActionListener(actionPrinter);
    JCheckBoxMenuItem fullJustify = new JCheckBoxMenuItem("Full",
        new ImageIcon("4.gif"));
    fullJustify.setHorizontalTextPosition(JMenuItem.RIGHT);
    fullJustify.setAccelerator(KeyStroke.getKeyStroke('F', Toolkit
        .getDefaultToolkit().getMenuShortcutKeyMask()));
    fullJustify.addActionListener(actionPrinter);

    justifyMenu.add(leftJustify);
    justifyMenu.add(rightJustify);
    justifyMenu.add(centerJustify);
    justifyMenu.add(fullJustify);

    menuBar.add(justifyMenu);
    menuBar.setBorder(new BevelBorder(BevelBorder.RAISED));

  }

  public static void main(String s[]) {
    CheckBoxMenuItemExample example = new CheckBoxMenuItemExample();
    example.pane = new JTextPane();
    example.pane.setPreferredSize(new Dimension(250, 250));
    example.pane.setBorder(new BevelBorder(BevelBorder.LOWERED));

    JFrame frame = new JFrame("Menu Example");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
    frame.setJMenuBar(example.menuBar);
    frame.getContentPane().add(example.pane, BorderLayout.CENTER);
    frame.pack();
    frame.setVisible(true);
  }
}
```

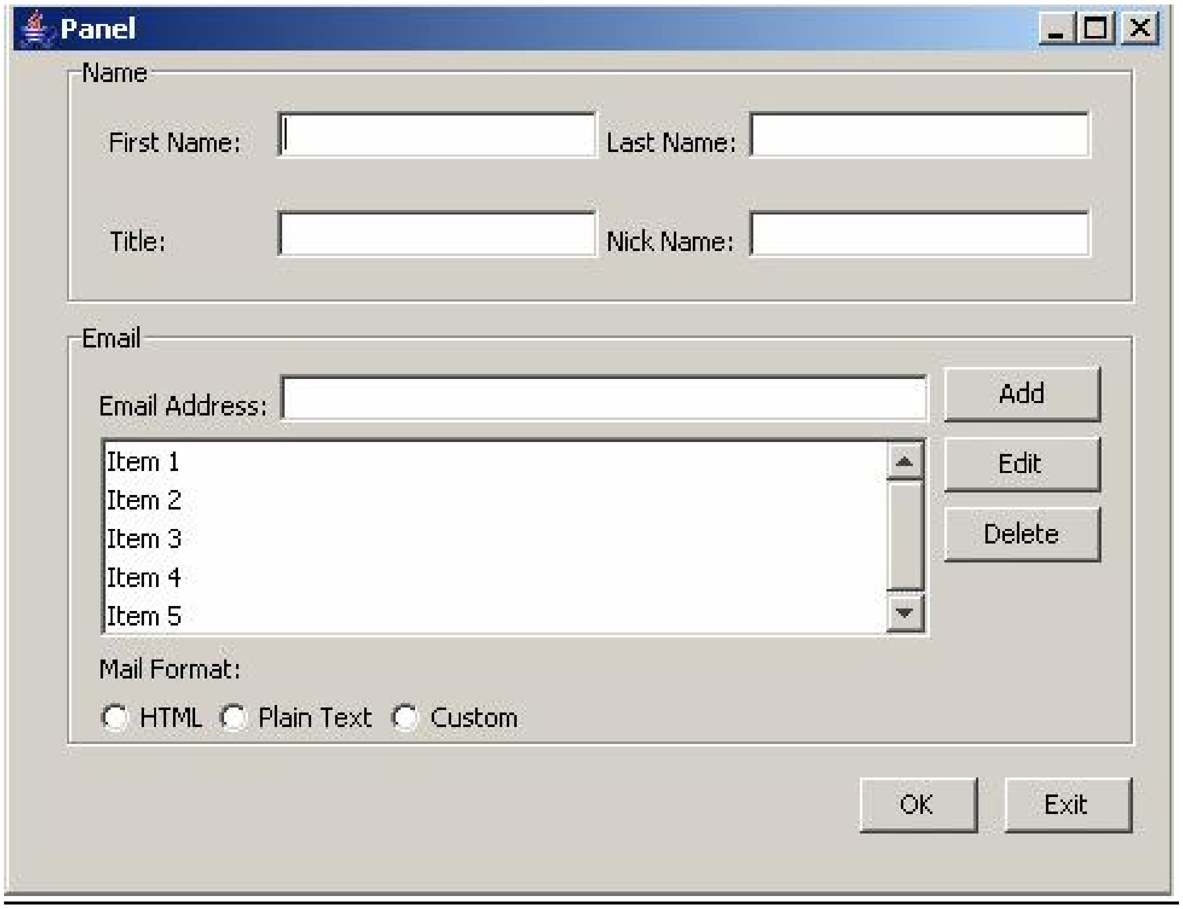
Do It Yourself

2.1. Do workshop 3, 4

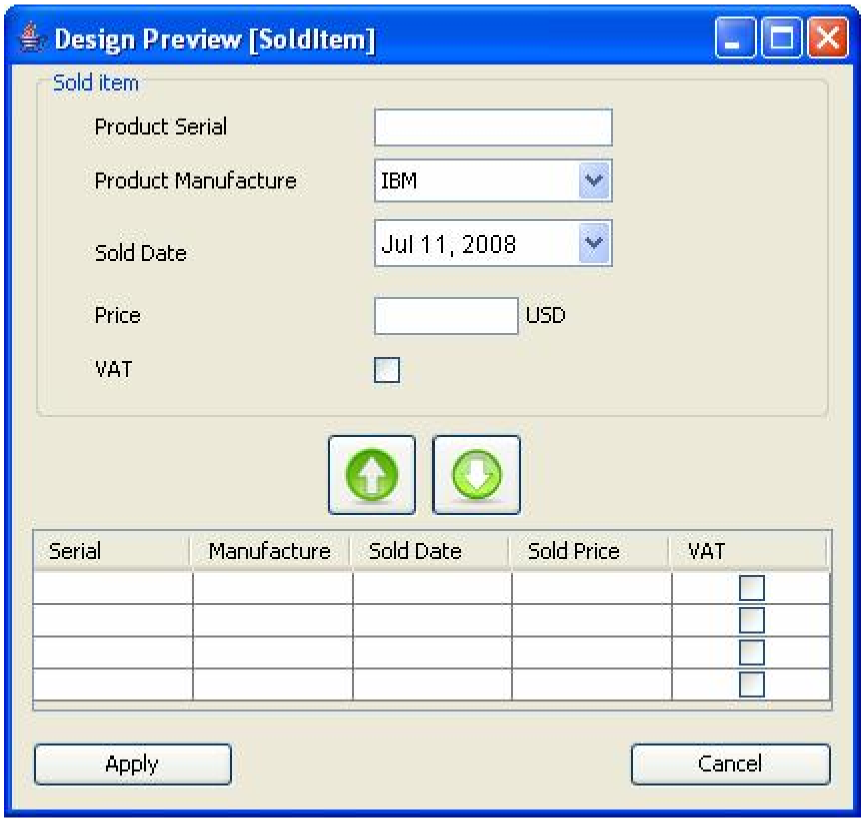
2.2. Create this form by two ways not using any tool and using a tool like Netbeans



2.3. Create another form

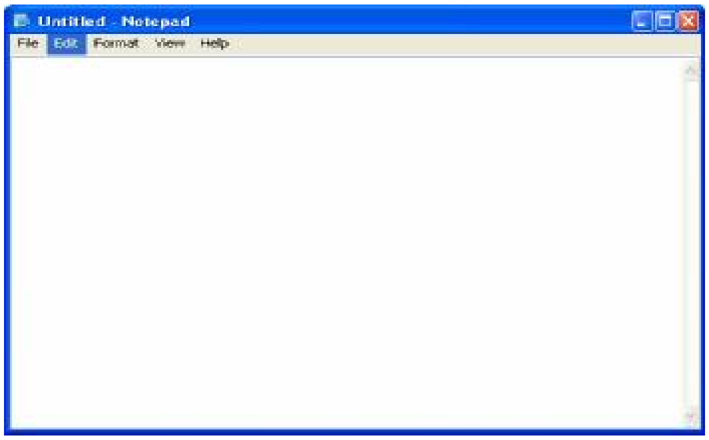


2.4. Continue creating this form



Serial	Manufacture	Sold Date	Sold Price	VAT
				☐
				☐
				☐
				☐

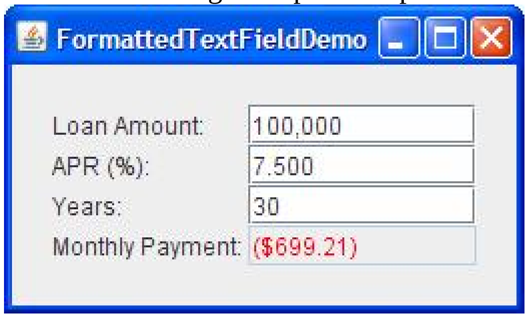
2.5. Create a window with menu system like the Notepad application



Self-study Samples

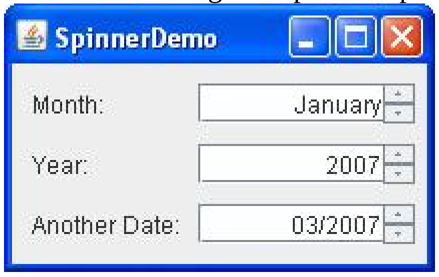
+ How to layout on java2s.com
http://www.java2s.com/Code/Java/Swing-JFC/Layout.htm

+ /tutorial/uiswing/examples/components/index.html#FormattedTextFieldDemo



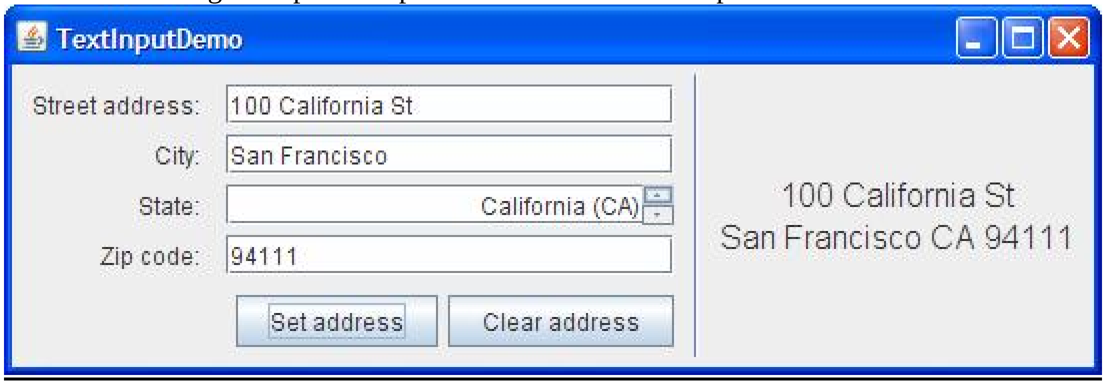
Loan Amount:	100,000
APR (%):	7.500
Years:	30
Monthly Payment:	($699.21)

+ /tutorial/uiswing/examples/components/index.html#SpinnerDemo



Month:	January
Year:	2007
Another Date:	03/2007

+ /tutorial/uiswing/examples/components/index.html#TextInputDemo

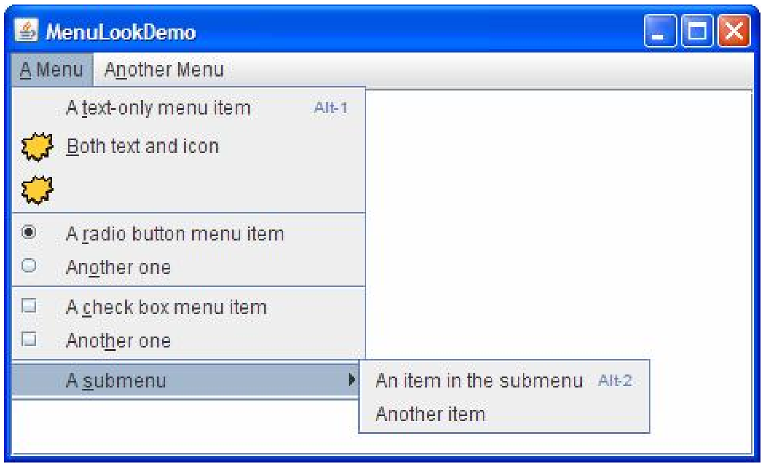


Street address:	100 California St
City:	San Francisco
State:	California (CA)
Zip code:	94111

100 California St
San Francisco CA 94111

+ Menu samples on java2s.com
http://www.java2s.com/Code/Java/Swing-JFC/Menu.htm

+ Tutorial about using menu in “java tutorial”
/tutorial/uiswing/components/menu.html



+ javapassion.com