

MODULE 01, 02, 03 & 04

(Thực hành: 12 tiết, Tự học: 27 tiết)

Chương I. GUI – IO TRONG JAVA

Mục tiêu:

- Hiểu rõ được các khái niệm Containers, Components, Layout Manager, Events.
- Vận dụng thành thạo các đối tượng giao diện chứa (containers) các đối tượng khác trong ngôn ngữ lập trình Java (JFrame, JApplet, JPanel)
- Vận dụng thành thạo các đối tượng giao diện trong giao diện của ứng dụng (JLabel, JButton, JCheckBox, JRadioButton, JTextField, JTextArea, JPasswordField)
- Hiểu rõ được khái niệm kiểm soát sự kiện (event handling) trong ngôn ngữ lập trình Java.
- Nắm được mô hình ủy thác sự kiện (event delegation model) và các loại sự kiện (event) trong ngôn ngữ lập trình Java.
- Sử dụng thành thạo các thành phần giao diện (JScrollPane, JSlider, JProgressBar, JFormattedTextField, JEditorPane) các đối tượng hộp thoại (JOptionPane, InputDialog, OptionDialog, JDialog) trong ngôn ngữ lập trình Java.
- Sử dụng thành thạo các đối tượng thiết kế thực đơn (menu) cho giao diện ứng dụng (JMenuBar, JMenu, JMenuItem, JCheckBoxMenuItem, JRadioButtonMenuItem, JPopupMenu, JToolBar).
- Vận dụng thành thạo các khái niệm về luồng nhập xuất và sự tuần tự hóa đối tượng (serializable) trong ngôn ngữ lập trình Java.

Bài Tập 1. Thực hành containers – Hiện thị JFrame

```
package demo.gui;
import javax.swing.JFrame;
public class DemoJFrame extends JFrame{
    public DemoJFrame() {
        setTitle("Demo JFrame"); //Tiêu đề JFrame
        setSize(300, 200); //Kích thước của JFrame
        setDefaultCloseOperation(EXIT_ON_CLOSE); //Thoát chương trình khi click nút
exit
        setLocationRelativeTo(null); //Canh giữa màn hình
        setResizable(false); //không cho phép thay đổi kích thước
JFrame
        //các thuộc tính khác JFrame
    }
    public static void main(String[] args) {
        new DemoJFrame().setVisible(true); //Hiện thị JFrame
    }
}
```

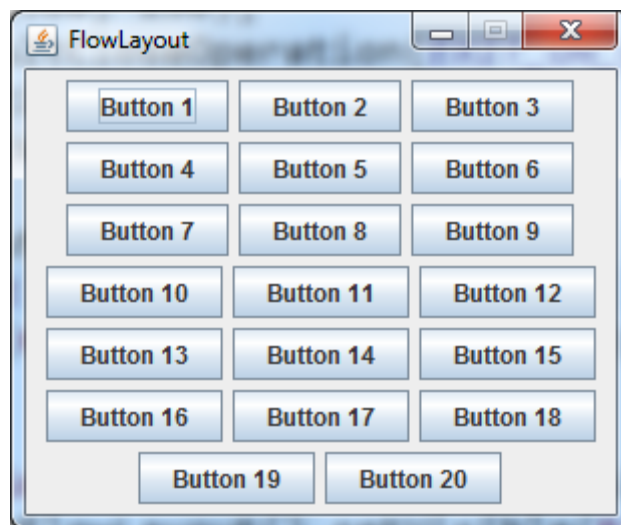
Bài Tập 2. Thực hành containers – Hiển thị JDialogs

```
package demo.gui;
import javax.swing.JDialog;
public class DemoJDialog extends JDialog{
    public DemoJDialog() {
        setTitle("Demo JDialog"); //Tiêu đề của hộp thoại JDialog
        setDefaultCloseOperation(DISPOSE_ON_CLOSE); //Đóng hộp
        thoại

        setSize(300, 200);
        setResizable(false);
        //Các thuộc tính khác
    }
    public static void main(String[] args) {
        new DemoJDialog().setVisible(true);
    }
}
```

Bài Tập 3. Thực hành containers – Hiển thị JWindow

Bài Tập 4. Thực hành Layout Managers - FlowLayout



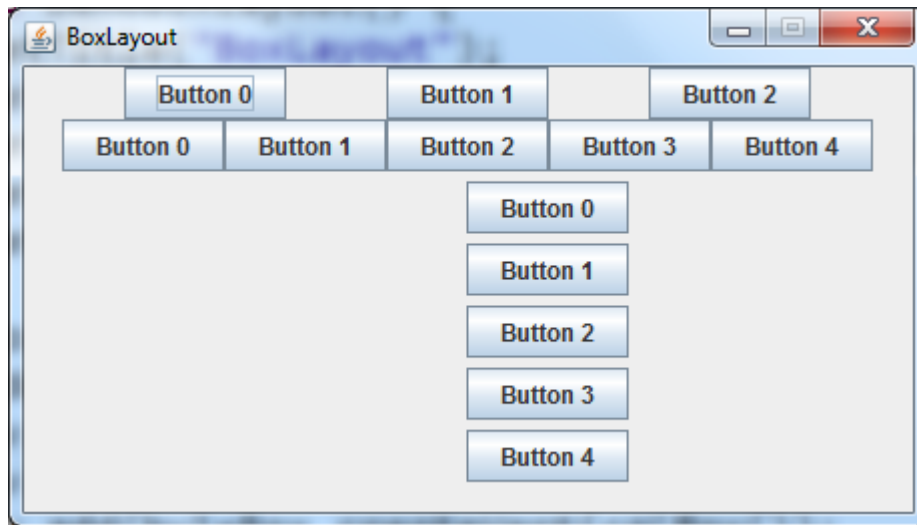
```
package demo.gui;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoFlowLayout extends JFrame{
    public DemoFlowLayout() {
        setTitle("FlowLayout");
        setSize(300, 250);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);
    }
}
```

```

        setLayout(new FlowLayout());
        for (int i = 1; i <=20; i++) {
            add(new JButton("Button "+i));
        }
    }
    public static void main(String[] args) {
        new DemoFlowLayout().setVisible(true);
    }
}

```

Bài Tập 5. Thực hành Layout Managers - BorderLayout



```

package demo.gui;
import javax.swing.*;
public class DemoBoxLayout extends JFrame{
    public DemoBoxLayout() {
        setTitle("BoxLayout");
        setSize(450, 250);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

        Box bv,bh1,bh2,bv1;
        add(bv=Box.createVerticalBox());
        bv.add(bh1=Box.createHorizontalBox());
        bv.add(bh2=Box.createHorizontalBox());
        bv.add(bv1=Box.createVerticalBox());

        for(int i=0;i<3;i++){
            bh1.add(Box.createHorizontalGlue());
            bh1.add(new JButton("Button "+i));
        }
    }
}

```

```

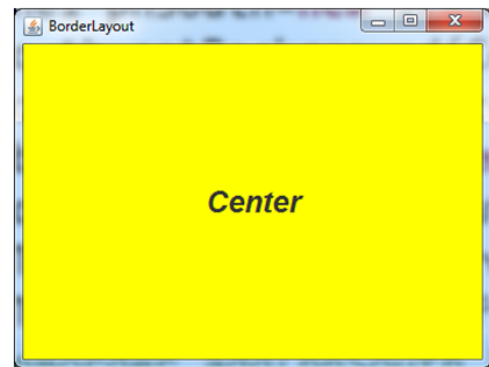
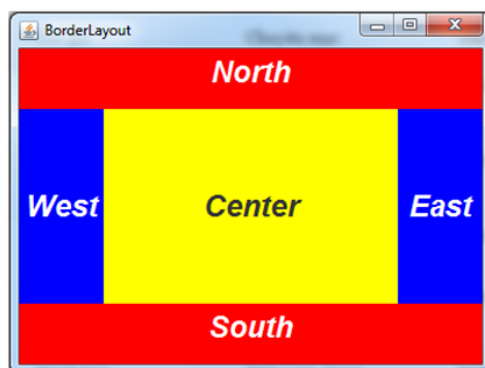
    }
    bh1.add(Box.createHorizontalGlue());

    for (int i = 0; i < 5; i++) {
        bh2.add(new JButton("Button "+i));
    }

    for (int i = 0; i < 5; i++) {
        bv1.add(Box.createVerticalStrut(5));
        bv1.add(new JButton("Button "+i));
    }
    bv1.add(Box.createVerticalStrut(5));
}
public static void main(String[] args) {
    new DemoBoxLayout().setVisible(true);
}
}

```

Bài Tập 6. Thực hành Layout Managers - BorderLayout.



```

package demo.gui;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoBorderLayout extends JFrame{
    private JButton
        bn=new JButton("North"),
        bs=new JButton("South"),
        be=new JButton("East"),
        bw=new JButton("West"),
        bc=new JButton("Center");

    public DemoBorderLayout() {
        setTitle("BorderLayout");
    }
}

```

```

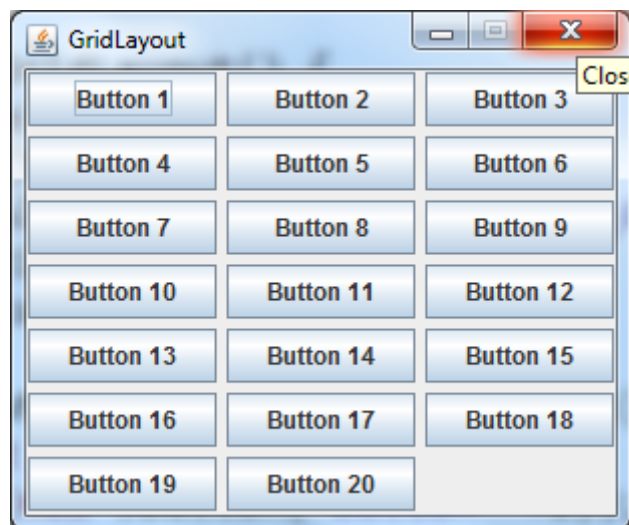
setSize(300, 200);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setLocationRelativeTo(null);
setResizable(false);

add(BorderLayout.NORTH, bn);
add(BorderLayout.SOUTH, bs);
add(BorderLayout.EAST, be);
add(BorderLayout.WEST, bw);
add(BorderLayout.CENTER, bc);

bn.setBackground(Color.red);
bn.setForeground(Color.white);
be.setBackground(Color.blue);
be.setForeground(Color.white);
//etc.
}
public static void main(String[] args) {
    new DemoBorderLayout().setVisible(true);
}
}

```

Bài Tập 7. Thực hành Layout Managers - GridLayout



```

package demo.gui;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
public class DemoGridLayout extends JFrame{
    public DemoGridLayout() {
        setTitle("GridLayout");
        setSize(300, 250);
    }
}

```

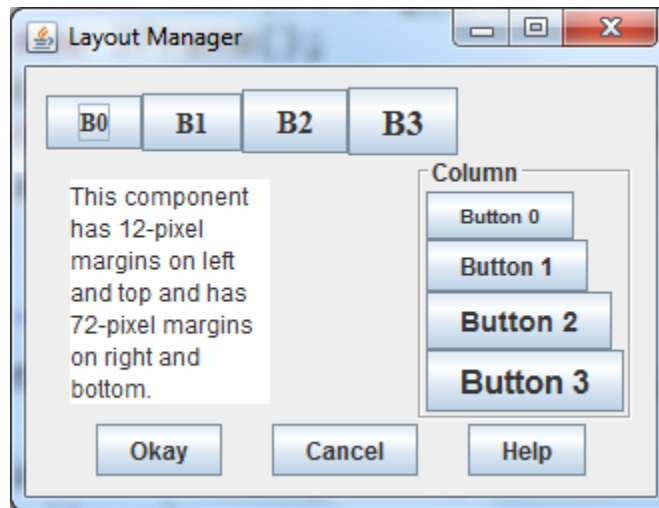
```

        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

        setLayout(new GridLayout(7,3,5,5)); //row=7, column=3,
h=5, v=5
        for (int i = 1; i <=20; i++) {
            add(new JButton("Button "+i));
        }
    }
    public static void main(String[] args) {
        new DemoGridLayout().setVisible(true);
    }
}

```

Bài Tập 8. Thực hành Layout Managers – Thiết kế giao diện sau:



```

package demo.gui;

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

public class BoxLayoutPane extends JPanel {
    public BoxLayoutPane() {
        this.setLayout(new BorderLayout());
        this.setBorder(new EmptyBorder(10, 10, 10, 10));
        Box row = Box.createHorizontalBox();
        for (int i = 0; i < 4; i++) {
            JButton b = new JButton("B" + i);
            b.setFont(new Font("serif", Font.BOLD, 12 + i * 2));

```

```

        row.add(b);
    }
    this.add(row, BorderLayout.NORTH);
    JPanel col = new JPanel();
    col.setLayout(new BoxLayout(col, BoxLayout.Y_AXIS));
    col.setBorder(new TitledBorder(new EtchedBorder(), "Column"));
    for (int i = 0; i < 4; i++) {
        JButton b = new JButton("Button " + i);
        b.setFont(new Font("sanserif", Font.BOLD, 10 + i * 2));
        col.add(b);
    }
    this.add(col, BorderLayout.EAST);

    Box buttonbox = Box.createHorizontalBox();
    buttonbox.add(Box.createHorizontalGlue());
    buttonbox.add(new JButton("Okay"));
    buttonbox.add(Box.createHorizontalGlue());
    buttonbox.add(new JButton("Cancel"));
    buttonbox.add(Box.createHorizontalGlue());
    buttonbox.add(new JButton("Help"));
    buttonbox.add(Box.createHorizontalGlue());
    this.add(buttonbox, BorderLayout.SOUTH);

    JTextArea textarea = new JTextArea();
    textarea.setText("This component has 12-pixel margins on left and
top"
        + " and has 72-pixel margins on right and bottom.");
    textarea.setLineWrap(true);
    textarea.setWrapStyleWord(true);

    Box fixedcol = Box.createVerticalBox();
    fixedcol.add(Box.createVerticalStrut(12));
    fixedcol.add(textarea);
    fixedcol.add(Box.createVerticalStrut(72));

    Box fixedrow = Box.createHorizontalBox();
    fixedrow.add(Box.createHorizontalStrut(12));
    fixedrow.add(fixedcol);
    fixedrow.add(Box.createHorizontalStrut(72));
    this.add(fixedrow, BorderLayout.CENTER);
}

public static void main(String[] a) {
    JFrame f = new JFrame();
    f.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {

```

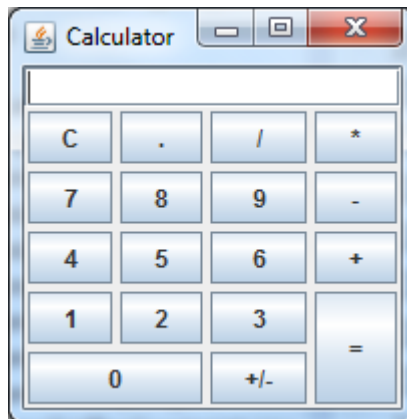
```

        System.exit(0);
    }
});
f.setTitle("Layout Manager");
f.setContentPane(new BoxLayoutPane());
f.pack();
f.setLocationRelativeTo(null);
f.setVisible(true);
}
}

```

Bài Tập 9. Thực hành Layout Managers - Sử dụng GridBagLayout và GridBagConstraints thiết kế các giao diện sau, thêm phần xử lý sự kiện khi tắt cửa sổ.

a.



```

package demo.gui;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class GridBagTest
{
    private String[] buttonText = { "C", ".", "/", "*", "7", "8",
    "9", "-", "4", "5", "6", "+", "1", "2", "3", "=", "0", "+/-" };
    private JButton[] button = new JButton[18];
    private int counter = 0;

    private void createAndDisplayGUI()
    {
        JFrame frame = new JFrame("Calculator");
    }
}

```



```

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JPanel contentPane = new JPanel();
contentPane.setLayout(new BorderLayout());

JPanel northPanel = new JPanel();
northPanel.setLayout(new BorderLayout(2, 2));
JTextField tfield = new JTextField();
northPanel.add(tfield, BorderLayout.CENTER);

JPanel centerPanel = new JPanel();
centerPanel.setLayout(new GridBagLayout());
GridBagConstraints gbc = new GridBagConstraints();
gbc.anchor = GridBagConstraints.PAGE_START;
gbc.fill = GridBagConstraints.BOTH;
gbc.weightx = 1.0;
gbc.weighty = 1.0;
gbc.insets = new Insets(2, 2, 2, 2);
for (int i = 0; i < button.length; i++){
    System.out.println("Button Text : " +
buttonText[i]);
    button[i] = new JButton(buttonText[i]);
}
for (int i = 0; i < 3; i++){
    for (int j = 0; j < 4; j++){
        gbc.gridx = j;
        gbc.gridy = i;
        centerPanel.add(button[counter++], gbc);
    }
}
gbc.gridx = 0;
gbc.gridy = 3;
centerPanel.add(button[counter++], gbc);
gbc.gridx = 1;
gbc.gridy = 3;
centerPanel.add(button[counter++], gbc);
gbc.gridx = 2;
gbc.gridy = 3;
centerPanel.add(button[counter++], gbc);
gbc.gridx = 3;
gbc.gridy = 3;
gbc.gridwidth = 1;
gbc.gridheight = 2;
centerPanel.add(button[counter++], gbc);
int count = counter;
System.out.println(button[--count].getText());

```

```

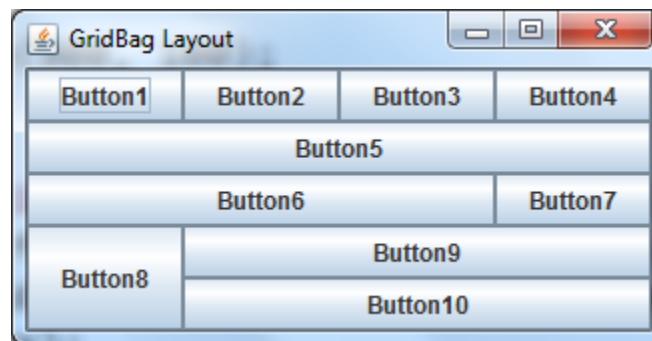
        gbc.gridx = 0;
        gbc.gridy = 4;
        gbc.gridheight = 1;
        gbc.gridwidth = 2;
        centerPanel.add(button[counter++], gbc);
        gbc.gridwidth = 1;
        gbc.gridx = 2;
        gbc.gridy = 4;
        centerPanel.add(button[counter++], gbc);

        contentPane.add(northPanel, BorderLayout.PAGE_START);
        contentPane.add(centerPanel, BorderLayout.CENTER);

        frame.setContentPane(contentPane);
        frame.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        frame.pack();
        frame.setLocationByPlatform(true);
        frame.setVisible(true);
    }
    public static void main(String[] args) {
        new GridBagTest().createAndDisplayGUI();
    }
}

```

b.



```

package demo.gui;

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

```

```

public class DemoGridBagConstraints extends JPanel {

    protected void makebutton(String name,
                               GridBagConstraints c) {
        JButton button = new JButton(name);
        gridbag.setConstraints(button, c);
        add(button);
    }

    public DemoGridBagConstraints() {
        GridBagLayout gridbag = new GridBagLayout();
        GridBagConstraints c = new GridBagConstraints();

        setFont(new Font("SansSerif", Font.PLAIN, 14));
        setLayout(gridbag);

        c.fill = GridBagConstraints.BOTH;
        c.weightx = 1.0;
        makebutton("Button1", gridbag, c);
        makebutton("Button2", gridbag, c);
        makebutton("Button3", gridbag, c);

        c.gridwidth = GridBagConstraints.REMAINDER; //end row
        makebutton("Button4", gridbag, c);

        c.weightx = 0.0; //reset to the default
        makebutton("Button5", gridbag, c); //another row

        c.gridwidth = GridBagConstraints.RELATIVE; //next-to-last in
row
        makebutton("Button6", gridbag, c);

        c.gridwidth = GridBagConstraints.REMAINDER; //end row
        makebutton("Button7", gridbag, c);

        c.gridwidth = 1; //reset to the default
        c.gridheight = 2;
        c.weighty = 1.0;
        makebutton("Button8", gridbag, c);

        c.weighty = 0.0; //reset to the default
        c.gridwidth = GridBagConstraints.REMAINDER; //end row
        c.gridheight = 1; //reset to the default
        makebutton("Button9", gridbag, c);
        makebutton("Button10", gridbag, c);
    }
}

```

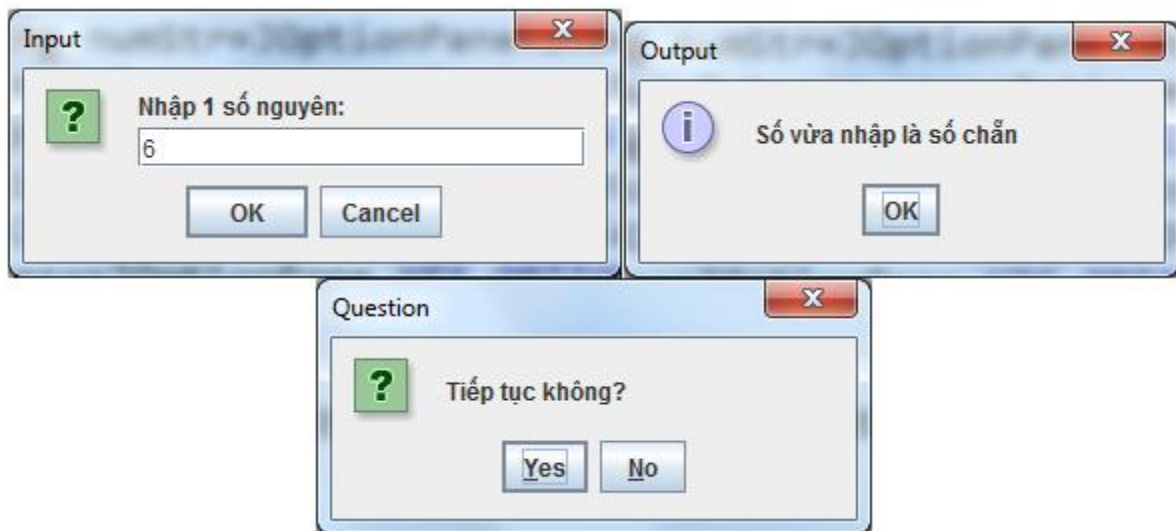
```

        setSize(500, 300);
    }

    public static void main(String args[]) {
        JFrame f = new JFrame("GridBag Layout");
        DemoGridBagConstraints ex = new DemoGridBagConstraints();
        f.add(ex);
        f.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        f.pack();
        f.setSize(f.getPreferredSize());
        f.setLocationRelativeTo(null);
        f.setVisible(true);
    }
}

```

Bài Tập 10. Thực hành Dialog boxes – JOptionPane



```

package demo.gui;
import javax.swing.JOptionPane;
public class EvenOdd {
    public static void main(String[] args) {
        int qes;
        do{
            String numStr=JOptionPane.showInputDialog("Nhập 1 số nguyên:
");
            if(numStr!=null && !numStr.trim().equals("")){

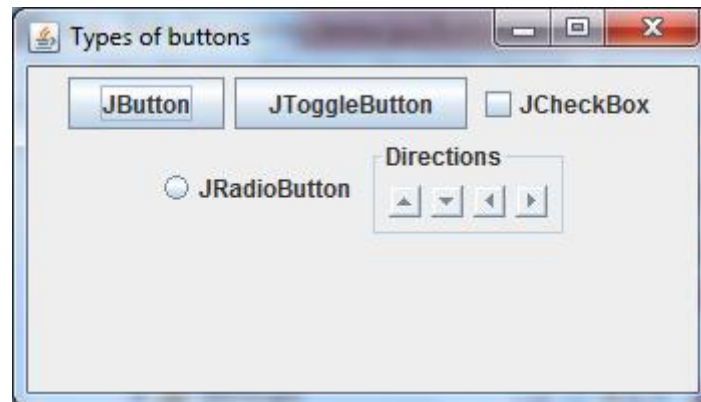
```

```

        try{
            int num=Integer.parseInt(numStr);
            String result="Số vừa nhập là số "+
                ((num%2==0) ? "chẵn" : "lẻ");
            JOptionPane.showMessageDialog(null,
result,
            "Output",JOptionPane.INFORMATION_MESSAGE);
        }catch(NumberFormatException ex){
            JOptionPane.showMessageDialog(null, "Nhập sai dữ
liệu");
        }
    }
    qes=JOptionPane.showConfirmDialog(null, "Tiếp tục
không?",
        "Question",JOptionPane.YES_NO_OPTION);
    }while(qes==JOptionPane.YES_OPTION);
}
}

```

Bài Tập 11. Thực hành về các components căn bản – Thiết kế giao diện (*types of buttons*)



```

package demo.gui;
import java.awt.FlowLayout;
import javax.swing.*;
import javax.swing.border.TitledBorder;
import javax.swing.plaf.basic.BasicArrowButton;
public class Buttons extends JFrame {
    private JButton jb = new JButton("JButton");
    private BasicArrowButton
up = new BasicArrowButton(BasicArrowButton.NORTH),
down = new BasicArrowButton(BasicArrowButton.SOUTH),
right = new BasicArrowButton(BasicArrowButton.EAST),

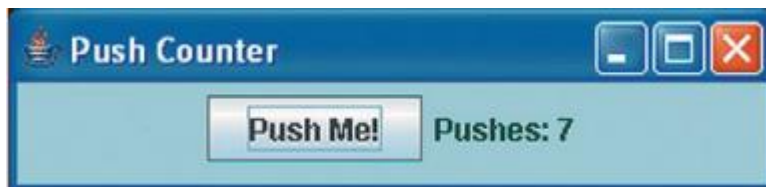
```

```

left = new BasicArrowButton(BasicArrowButton.WEST);
public Buttons() {
    setTitle("Types of buttons");
    setSize(350, 200);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setLayout(new FlowLayout());
    add(jb);
    add(new JToggleButton("JToggleButton"));
    add(new JCheckBox("JCheckBox"));
    add(new JRadioButton("JRadioButton"));
    JPanel jp = new JPanel();
    jp.setBorder(new TitledBorder("Directions"));
    jp.add(up);
    jp.add(down);
    jp.add(left);
    jp.add(right);
    add(jp);
}
public static void main(String[] args) {
    new Buttons().setVisible(true);
}
}

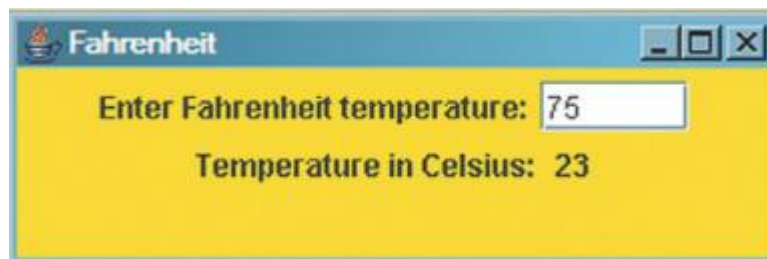
```

Bài Tập 12. Thực hành JButton, JLabel, event listener – Khi người dùng nhấn nút “Push Me!”, đếm số lần được nhấn và hiển thị trong 1 JLabel.



Addison Wesley - Java Software Solutions, 7th Edition – page: 193 - 194

Bài Tập 13. Viết chương trình chuyển từ độ F sang độ C ($C = (F - 32) * 5 / 9$)



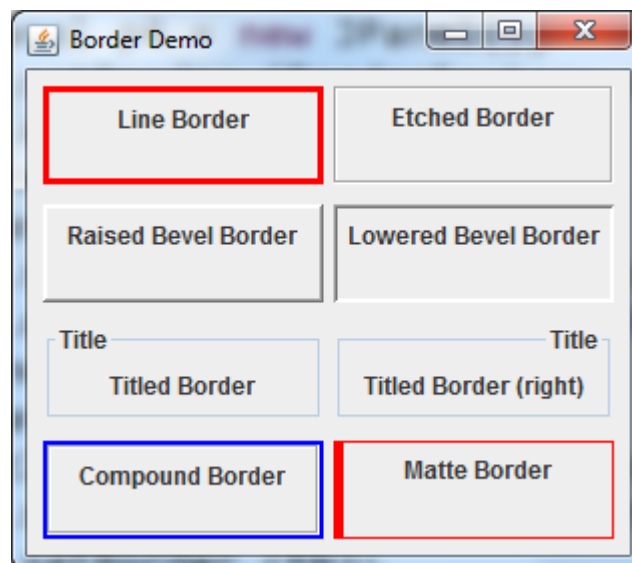
Bài Tập 14. Thêm một nút (JButton) vào form Temperature Conversion trên. Khi người sử dụng nhấn vào nút này, thì cũng chuyển đổi từ độ F sang độ C giống như ta nhấn trên JTextField.

Bài Tập 15. Thực hành về check boxes, event listener – Thiết kế giao diện gồm 1 JLabel và 2 check boxes như bên dưới.

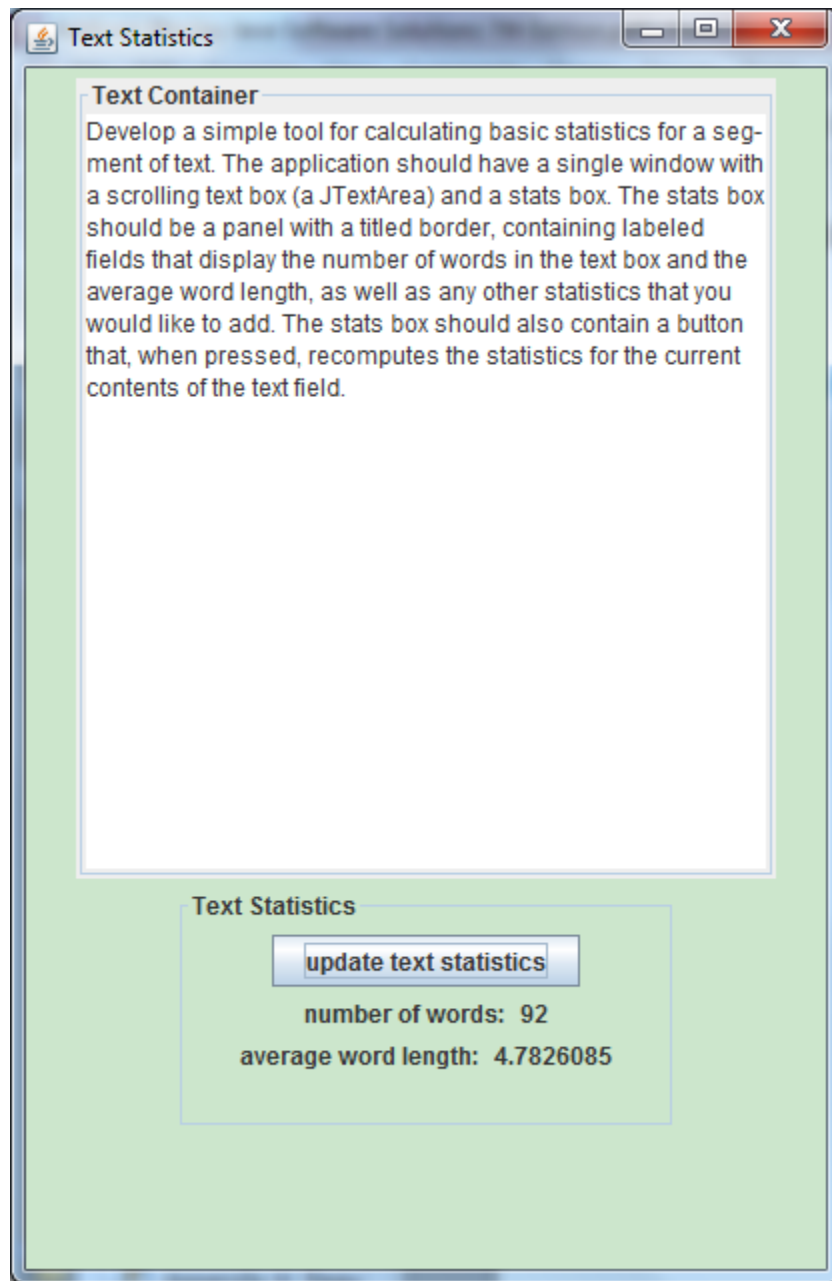


Khi chọn vào “Bold” nội dung trong văn bản tô đậm, tương tự cho các trường hợp khác.

Bài Tập 16. Thực hành về Border – Thiết kế giao diện sau:



Bài Tập 17. Thực hành JTextArea – Thiết kế và hiện thực chương trình: nhập vào 1 đoạn văn bản, thống kê tổng số từ, trung bình số ký tự của 1 từ, với giao diện sau:



```
package demo.gui;
//*****
*
//Demonstrates a graphical user interface and an event listener.
//*****
*
import java.awt.*;
import javax.swing.*;
```



```

import java.awt.event.*;
public class TextStatPanel
extends JPanel
implements ActionListener {
    private JTextArea textArea;
    private JButton updateButton;
    private JLabel[] stats;

    public static final String[] statLabels = {
        "number of words: ",
        "average word length: "
    };

    // -----
    // Constructor: Sets up the GUI.
    // -----

    public TextStatPanel() {
        init();
    }

    private void init() {
        textArea = new JTextArea("Paste any text here, I will update the
statistics.");
        textArea.setWrapStyleWord(true);
        textArea.setLineWrap(true);

        updateButton = new JButton("update text statistics");
        updateButton.addActionListener(this);

        JPanel statPane = new JPanel();
        statPane.setBorder(BorderFactory.createTitledBorder("Text
Statistics"));
        statPane.setPreferredSize(new Dimension(250, 120));
        statPane.setOpaque(false);
        int n = statLabels.length;

        statPane.add(updateButton);

        stats = new JLabel[n];
        for(int i=0; i<n; ++i) {
            JLabel l = new JLabel(statLabels[i]);
            statPane.add(l);
            stats[i] = new JLabel();
            statPane.add(stats[i]);
        }
    }
}

```

```

    }

    JScrollPane scroll = new JScrollPane(textArea);
    scroll.setPreferredSize(new Dimension(350, 400));
    scroll.setBorder(BorderFactory.createTitledBorder("Text
Container"));

    this.add(scroll);
    this.add(statPane);

    this.setBackground(new Color(.8f, 0.9f, 0.8f));
    this.setPreferredSize(new Dimension(400, 600));
    this.updateStatistics();
}

//
*****
// Represents a listener for button push (action) events.
//
*****
public void actionPerformed(ActionEvent event) {
    Object source = event.getSource();

    if (source == updateButton) {
        updateStatistics();
    }
}

private void updateStatistics() {
    String text = textArea.getText();
    String[] words = text.split(" ");
    float ave = (text.length()-
words.length+1.0f)/words.length;
    stats[0].setText(String.valueOf(words.length));
    stats[1].setText(String.valueOf(ave));
}

// -----
-----
// Creates and displays the main program frame.
// -----
-----
public static void main(String[] args) {
    JFrame frame = new JFrame("Text Statistics");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    TextStatPanel panel = new TextStatPanel();

```

```
        frame.getContentPane().add(panel);  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```