

2009 - 2010

Tehnologii de Programare in Internet (TPI / RST)

Titulari curs: **Mihnea Magheti**, Eduard-Cristian Popovici

Suport curs: <http://discipline.elcom.pub.ro/tpi/>

Moodle: <http://electronica07.curs.ncit.pub.ro/course/category.php?id=3>

Continut curs TPI

1. Introducere in tehnologiile Internet

2. Introducere in tehnologiile desktop (SE) Java

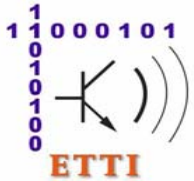
- 2.1. Elemente de baza. Tipuri de date referinta. Clase de biblioteca
- 2.2. Clase pentru fluxuri de intrare-iesire (IO)

3. Programarea la nivel socket in Java

- 3.1. Introducere in Protocolul Internet (IP) si stiva de protocoale IP
- 3.2. Socketuri flux (TCP) Java si programe multifilare (threads)
- 3.3. Socketuri datagrama (UDP) Java

4. Tehnologii Java de programare a aplicatiilor Web (EE) Java

- 4.1. Tehnologii client. Miniaplicatii Java (applet-uri)
- 4.2. Clase pentru interfete grafice cu utilizatorul (AWT, Swing)
- 4.3. Platforma Java EE. Arhitectura si tehnologiile implicate
- 4.4. Tehnologii server. Tehnologia Java Servlet
- 4.5. Tehnologia Java ServerPages (JSP)
- 4.6. Accesul la baze de date prin tehnologii Java (JDBC, Hibernate)
- 4.7. Tehnologii avansate (frameworks, componente EJB, Servicii Web)



Structura cursului



4. Tehnologii Java de programare a aplicatiilor Web

4.2. Clase pentru interfete grafice cu utilizatorul (AWT, Swing)

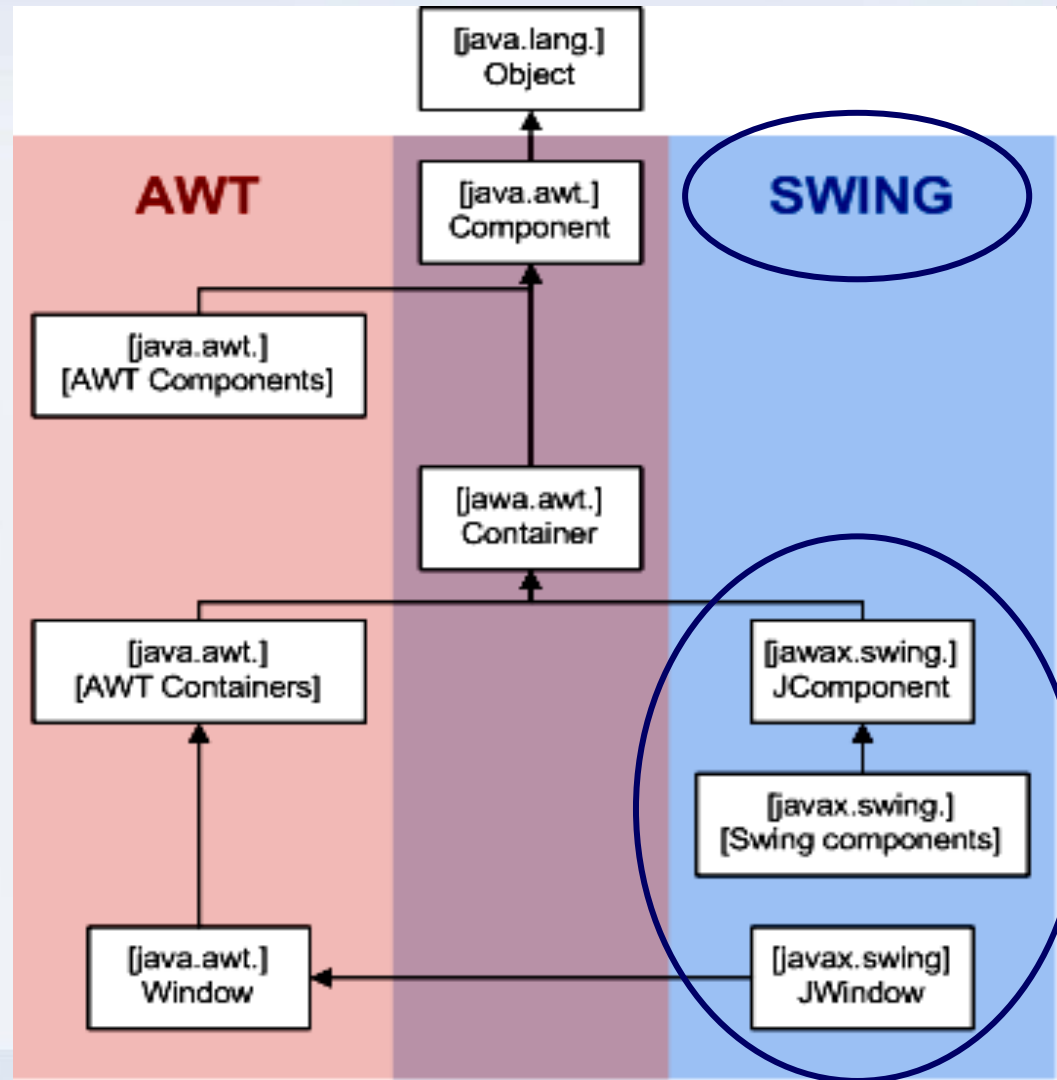
Tutorialul Java Swing:

<http://java.sun.com/docs/books/tutorial/uiswing/TOC.html>



4.2. Interfete grafice Java Swing

Ierarhiile de clase Java pentru interfete grafice

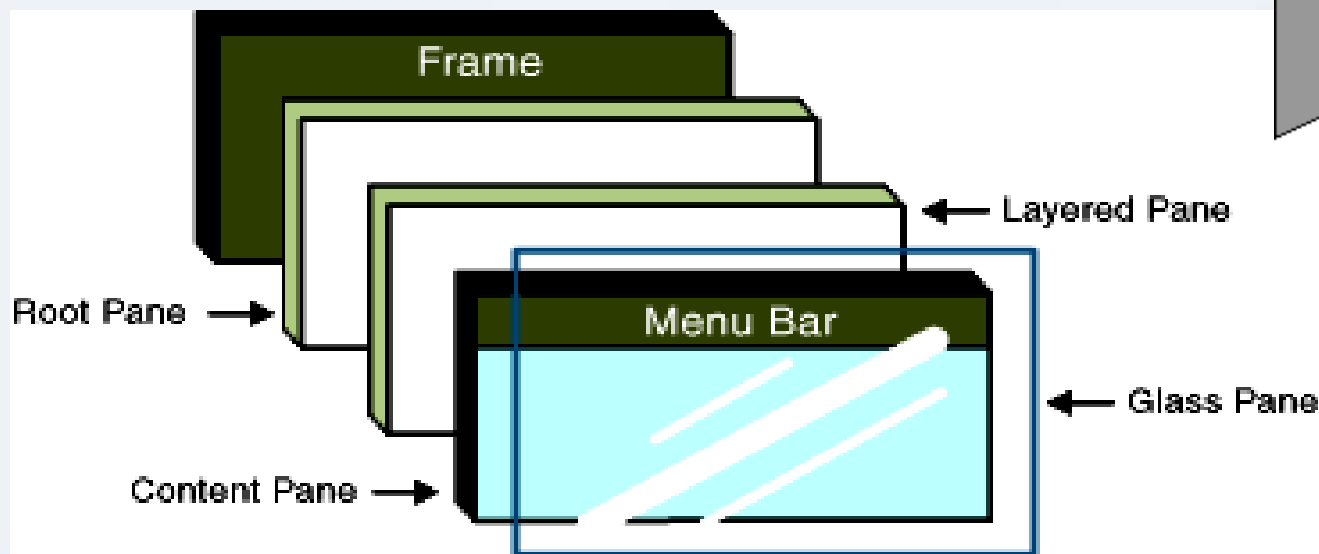
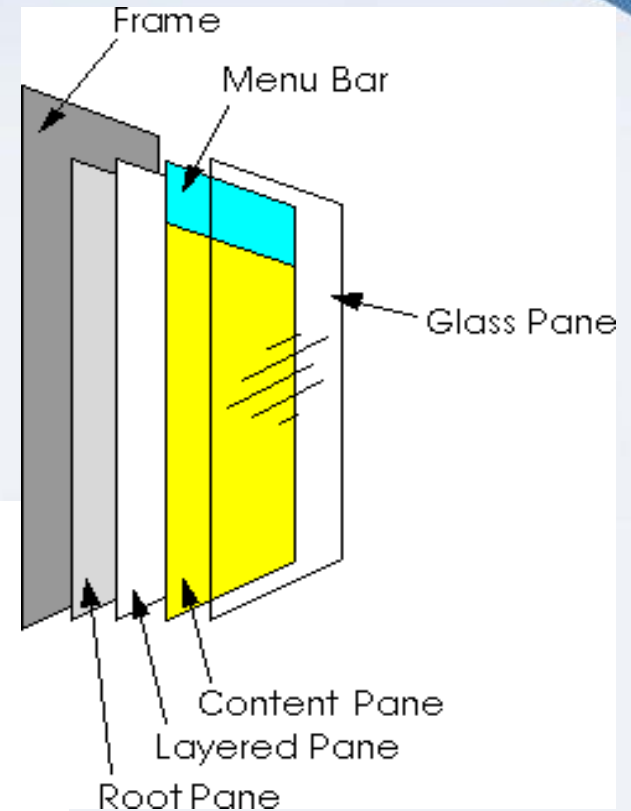


4.2. Interfete grafice Java Swing

Containere de nivel maxim (*top-level*)

Fiecare aplicație care utilizează componente Swing are cel puțin un “*top-level container*”

- care conține un container intermediar numit *Root Pane*
- iar acesta la randul lui este rădăcină a unei ierarhii de containere care conțin toate celelalte componente Swing

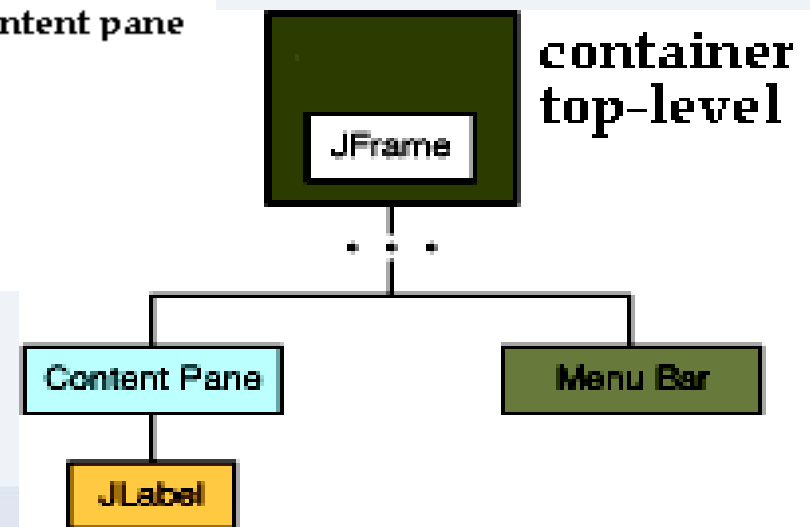
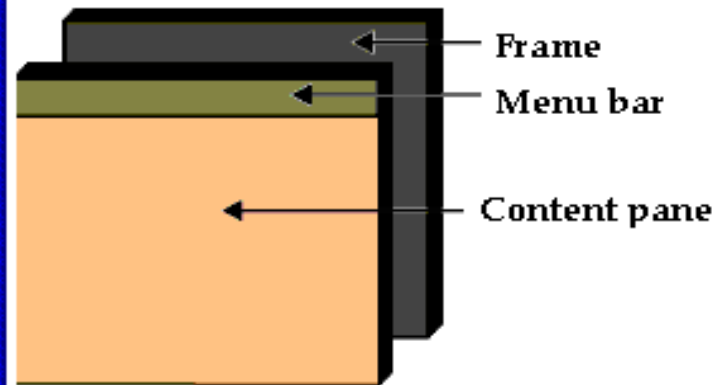


4.2. Interfete grafice Java Swing

Containere de nivel maxim (*top-level*)

Containerele de nivel maxim (*top-level cantainers*)

- ferestrele principale – **JFrame** in Swing (*Frame* in AWT)
- ferestrele de dialog – **JDialog** in Swing (*Dialog* in AWT)
- miniaplicatiile Web – **JApplet** in Swing (*Applet* in AWT)

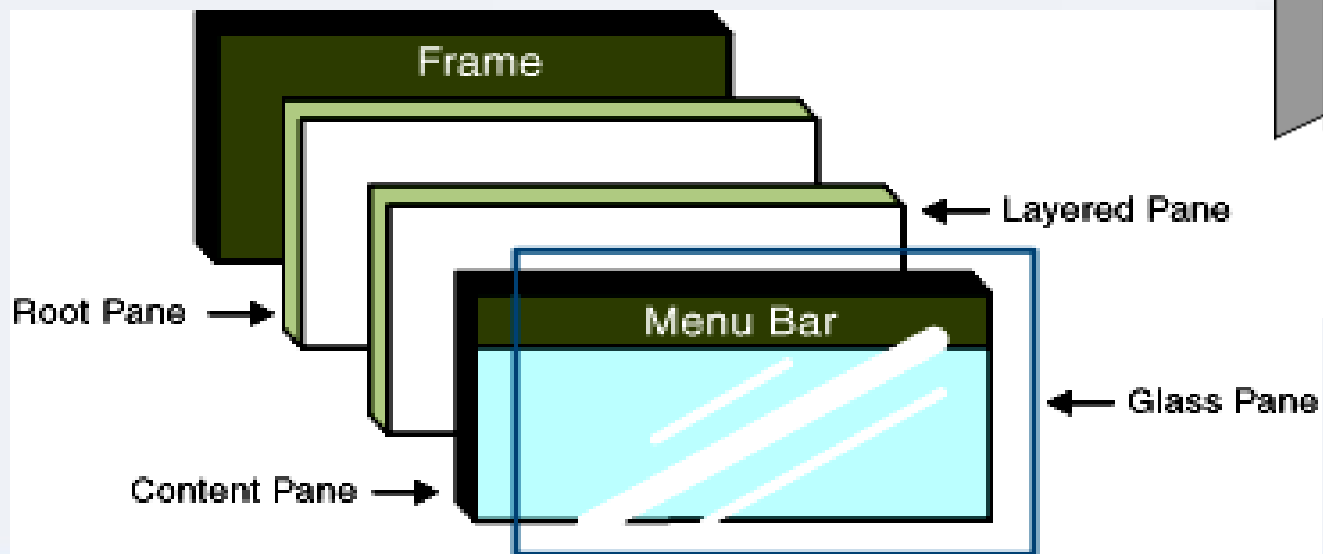
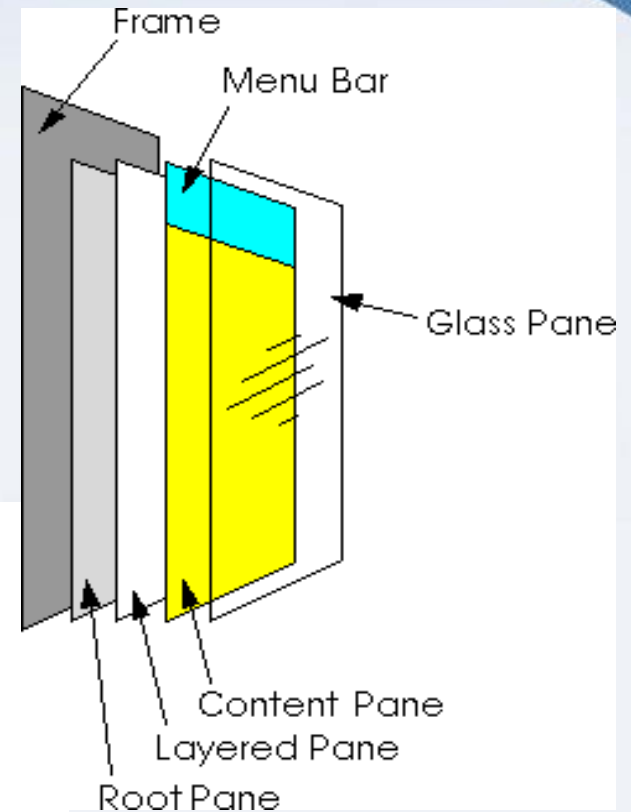


4.2. Interfete grafice Java Swing

Containere de nivel maxim (*top-level*)

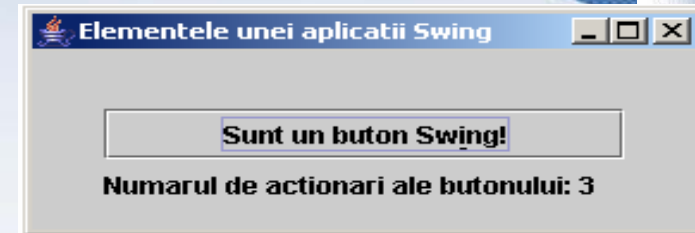
Containerul intermediar *Root Pane*

- contine bara de meniu (*Menu Bar*)
- si un container de continut numit *Content Pane*
- in care programatorul poate adauga de fapt ierarhia de containere care contin restul componentelor Swing



4.2. Interfete grafice Java Swing

Elementele unei aplicatii grafice Swing



// 1. Importul pachetelor Swing si AWT necesare

```
import javax.swing.*;           // Numele actual al pachetului Swing (JFC)
import java.awt.*;             // Numele pachetului AWT (necesar uneori)
import java.awt.event.*;       // Numele pachetului pentru tratarea
                                // evenimentelor, pentru interactivitate
```

```
public class ElementeAplicatieSwing {
```

```
    public static void main(String[] args) {
```

// 2. Optional: stabilirea aspectului (Java, Windows,...)

```
try { UIManager.setLookAndFeel(
        UIManager.getCrossPlatformLookAndFeelClassName());
} catch (Exception e) { }
```

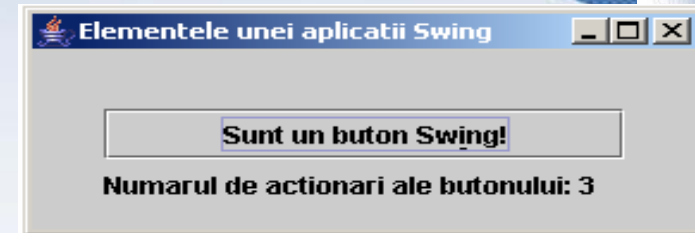
// 3. Stabilirea containerului de nivel maxim

// (JFrame, JApplet, JDialog)

```
JFrame frame = new JFrame("Elementele unei aplicatii Swing");
```


4.2. Interfete grafice Java Swing

Elementele unei aplicatii grafice Swing



```
// 4. Obtinerea panoului de continut intern cadrului  

// (container in care vor fi plasate componentele ferestrei)
```

```
Container container = frame.getContentPane();
```

```
// 5. Optional: Stabilirea modului de asezare (layout)  

// in panoul ContentPane (implicit BorderLayout)
```

```
container.setLayout(new BorderLayout() );
```

```
// 6. Crearea si configurarea componentelor grafice (controale)
```

```
// 6.1. Crearea unei etichete
```

```
final String textEticheta = "Numarul de actionari ale butonului: ";  

final JLabel eticheta = new JLabel(textEticheta + "0 ");
```

```
// 6.2. Crearea unui buton cu combinatie de taste atasata
```

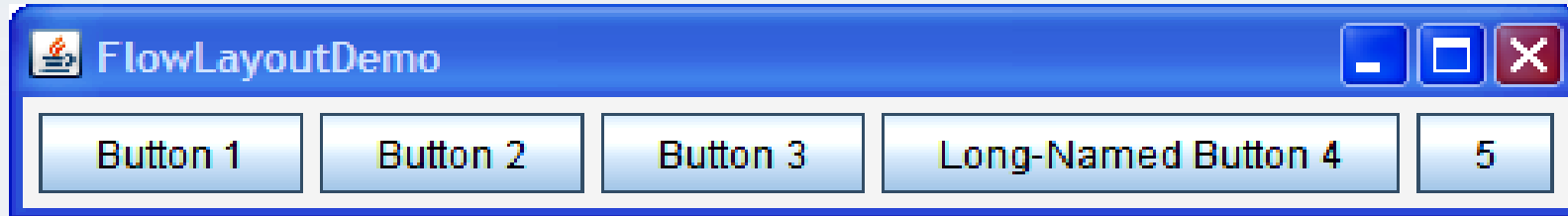
```
JButton buton = new JButton("Sunt un buton Swing!");  

buton.setMnemonic(KeyEvent.VK_I); // optional
```

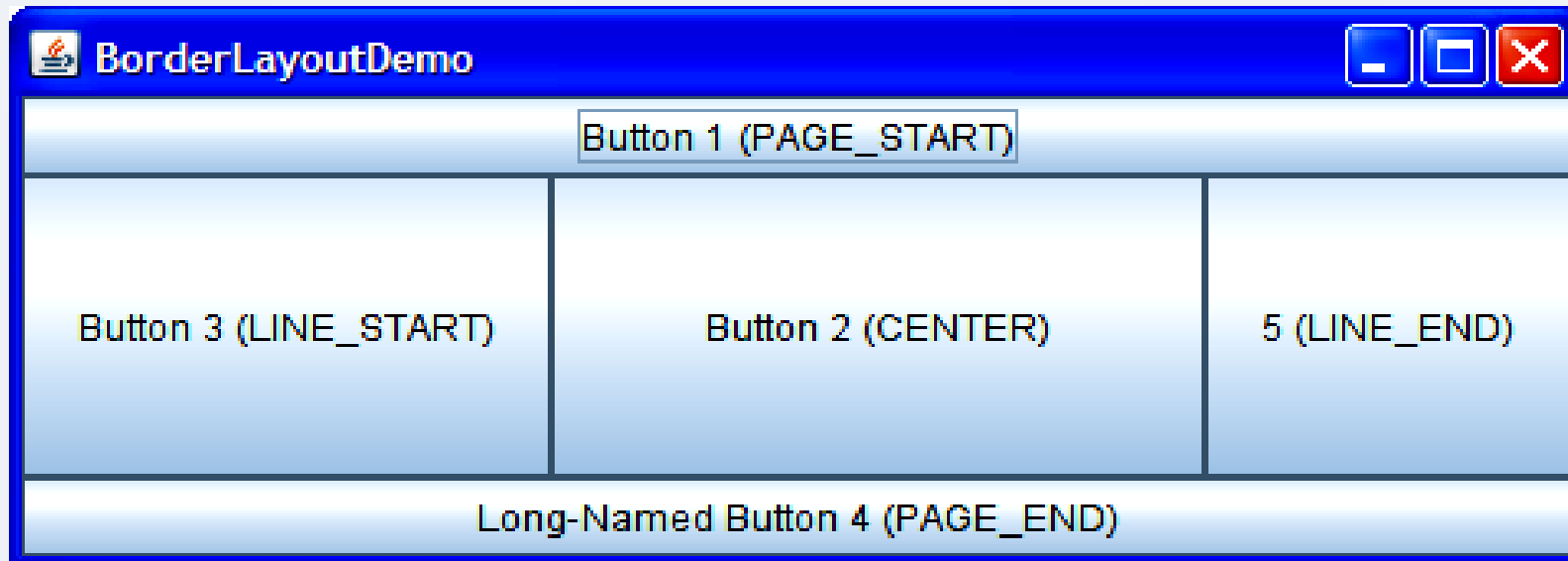
4.2. Interfete grafice Java Swing

Moduri de asezare a componentelor in containere (*Layouts*)

Pe o linie (*FlowLayout*) - mod de asezare **implicit**



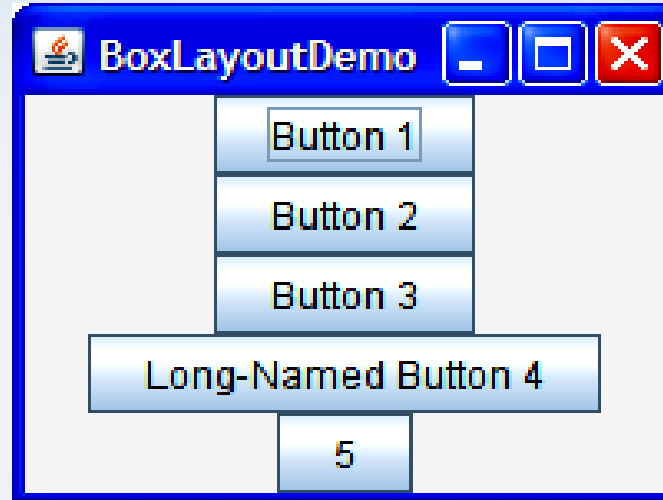
Relativ la margini (*BorderLayout*)



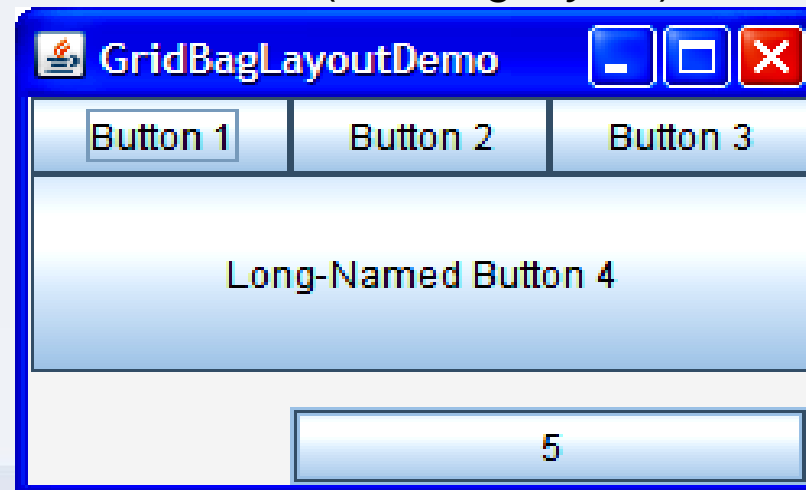
4.2. Interfete grafice Java Swing

Moduri de asezare a componentelor in containere (*Layouts*)

Pe o coloana (*BoxLayout*)



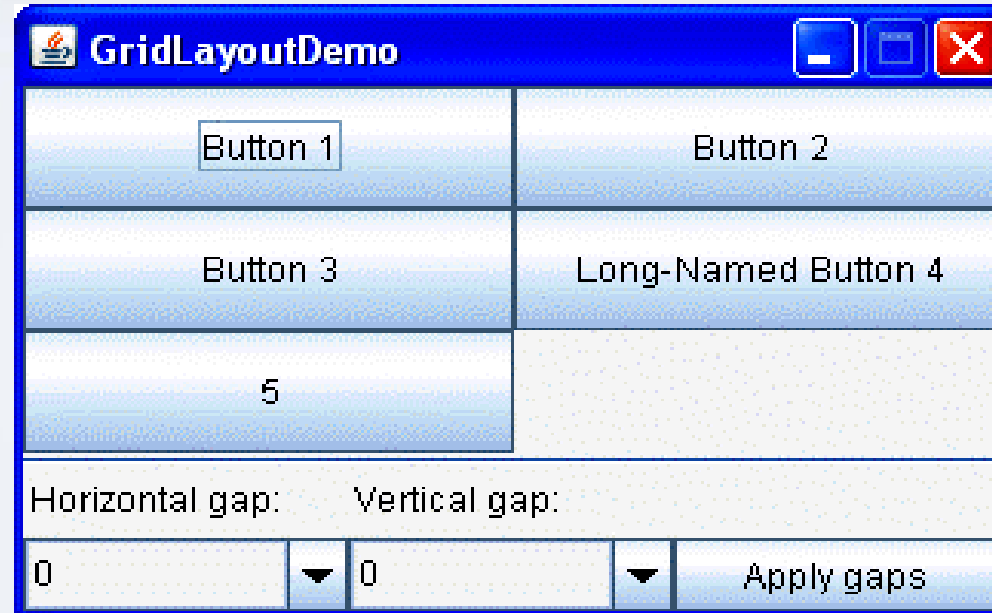
Grid pe coloane de latime diferita (*GridBagLayout*)



4.2. Interfete grafice Java Swing

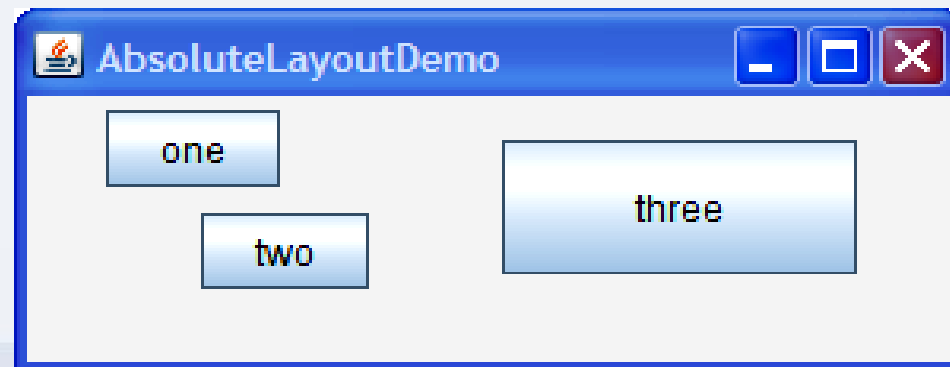
Moduri de asezare a componentelor in containere (*Layouts*)

Grid pe coloane de latime egala (*GridLayout*)



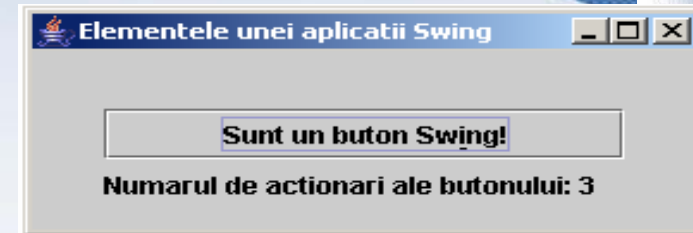
Asezarea fara LayoutManager

- utilizarea **pozitionarii absolute**



4.2. Interfete grafice Java Swing

Elementele unei aplicatii grafice Swing



```
// 6.3. Crearea si configurarea unui nou panou (care permite
//      gruparea spatiala, ierarhica, a componentelor)
JPanel panou = new JPanel();

// 6.3.1. Stabilirea spatiului care inconjoara continutul
panou.setBorder(BorderFactory.createEmptyBorder(
    30,    // sus
    30,    // in stanga
    10,    // jos
    30)); // in dreapta

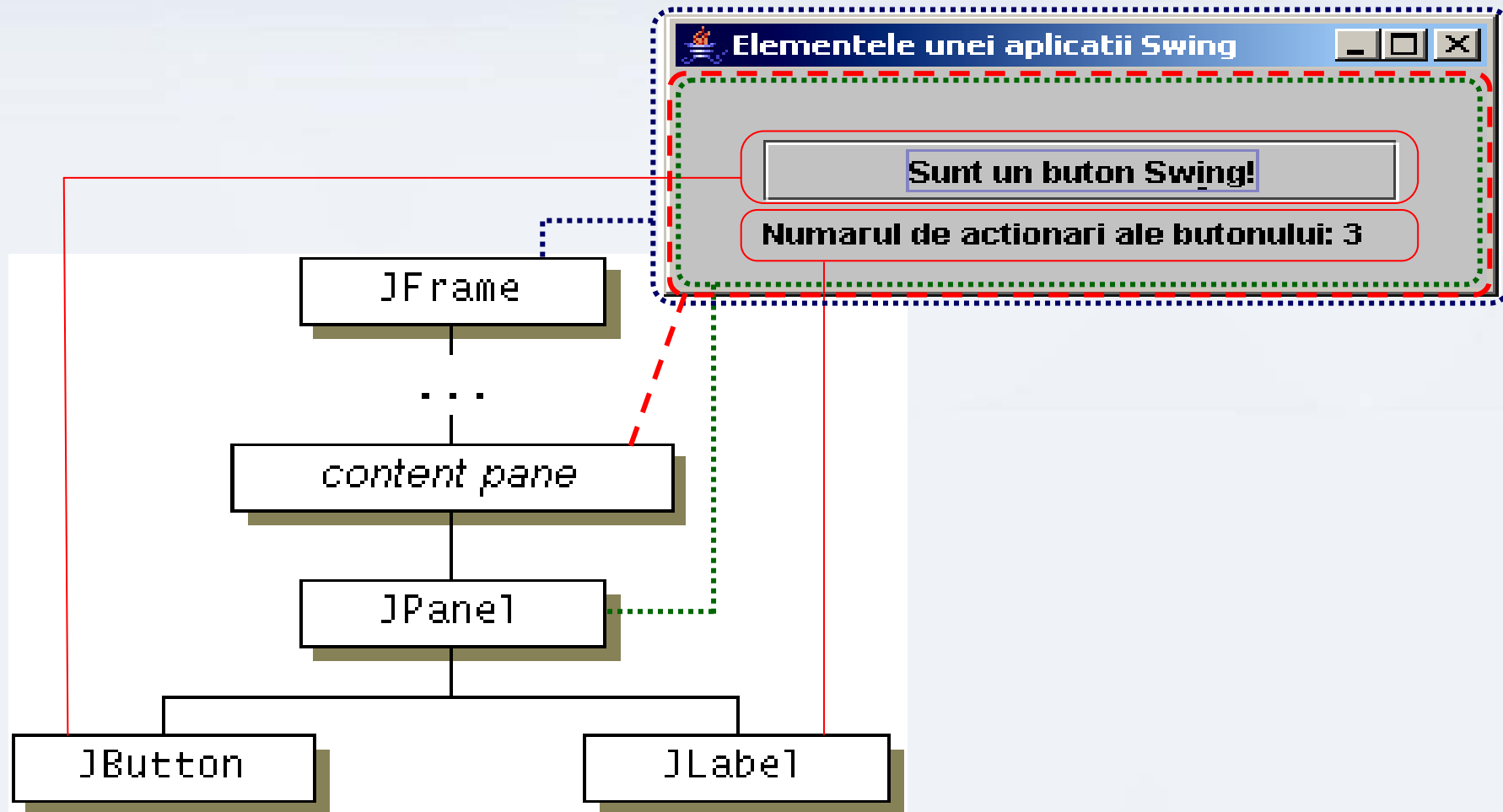
// 6.3.2. Stabilirea modului de asezare (layout) in noul panou
panou.setLayout(new GridLayout(0, 1));

// 6.3.3. Adaugarea componentelor in noul panou
panou.add(buton);
panou.add(eticheta);

// 7. Adaugarea in ContentPane a componentelor grafice
container.add(panou, BorderLayout.CENTER);
```

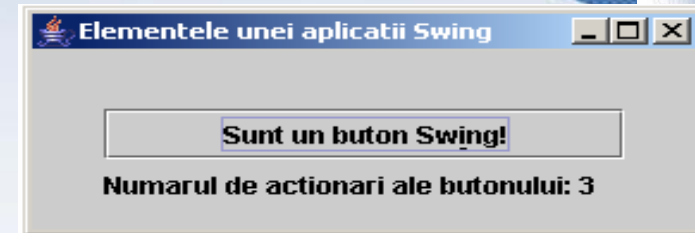
4.2. Interfete grafice Java Swing

Elementele unei aplicatii grafice Swing



4.2. Interfete grafice Java Swing

Elementele unei aplicatii grafice Swing



```
// 8. Crearea codului pentru tratarea evenimentelor
// (interactivitatii)
```

```
// 8.1. Atasarea unui obiect al unei clase anonime care
//      implementeaza interfata ActionListener, a carei metoda
//      actionPerformed() trateaza actionarea butonului
```

```
buton.addActionListener( new ActionListener() {
    int numActionari = 0;
    public void actionPerformed(ActionEvent e) {
        numActionari++;
        eticheta.setText(textEticheta + numActionari);
    }
});
```

4.2. Interfete grafice Java Swing

Elementele unei aplicatii grafice Swing

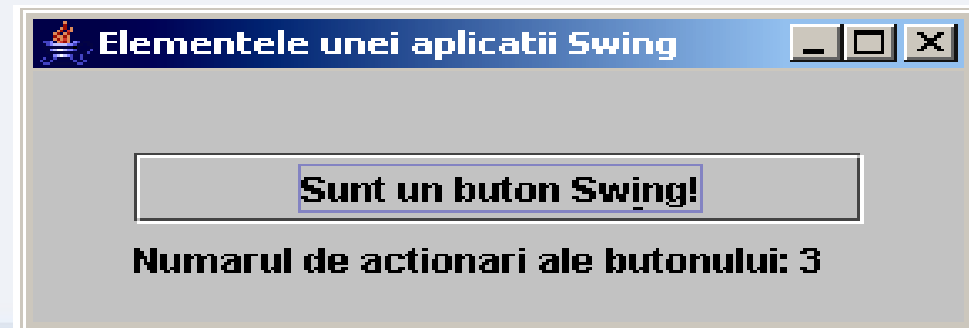
```
// 8.2. Stabilirea iesirii din program la inchiderea ferestrei
frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) { System.exit(0); }
});

// Alternativa, incepand cu Java 2, versiunea 1.3:
//      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// 9. Stabilirea dimensiunii ferestrei
//      - fie in functie de cea a componentelor:
frame.pack();

//      - fie prin impunerea dimensiunii ferestrei:
//      frame.setSize(400,      // latime
//                    300);    // inaltime

// 10. Afisarea ferestrei
frame.setVisible(true);
}
}
```

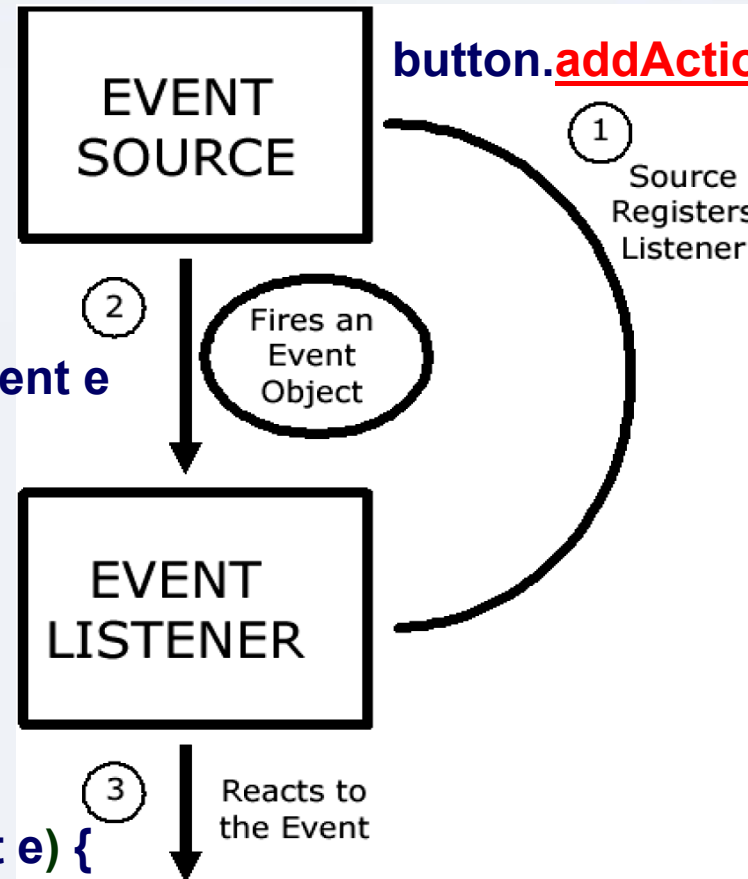


4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

Modelul folosit in **tratarea evenimentelor** (crearea interactivitatii) in interfetele grafice Swing

JButton button



ActionEvent e

```
// Aplicatia Swing listener
public class MyClass
implements ActionListener {
```

```
actionPerformed(ActionEvent e) {
```

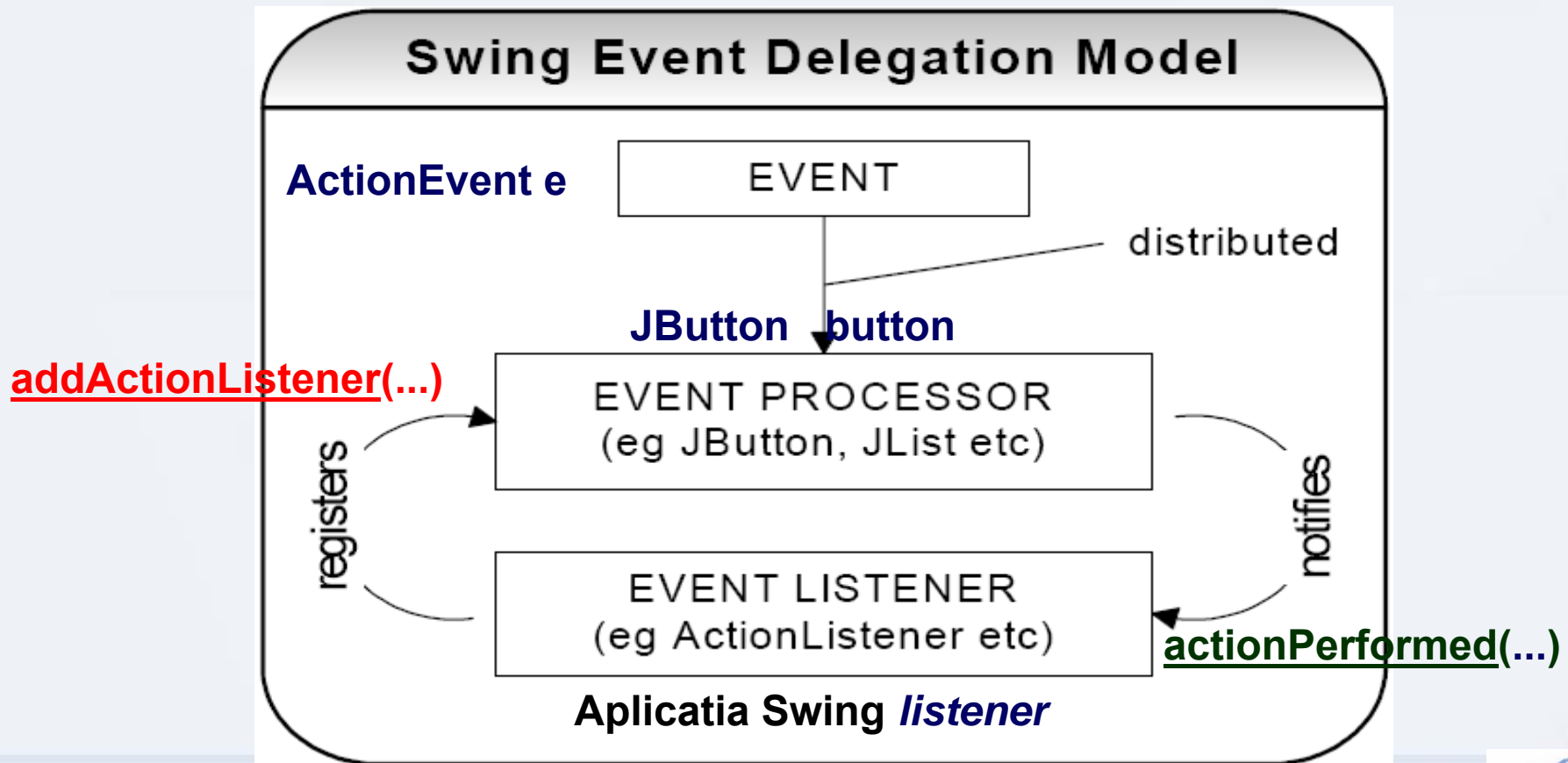
// Codul prin care aplicatia reactioneaza la apasarea butonului

4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

Modelul folosit in tratarea evenimentelor (crearea interactivitatii)

- modelul **delegarii evenimentelor** Swing

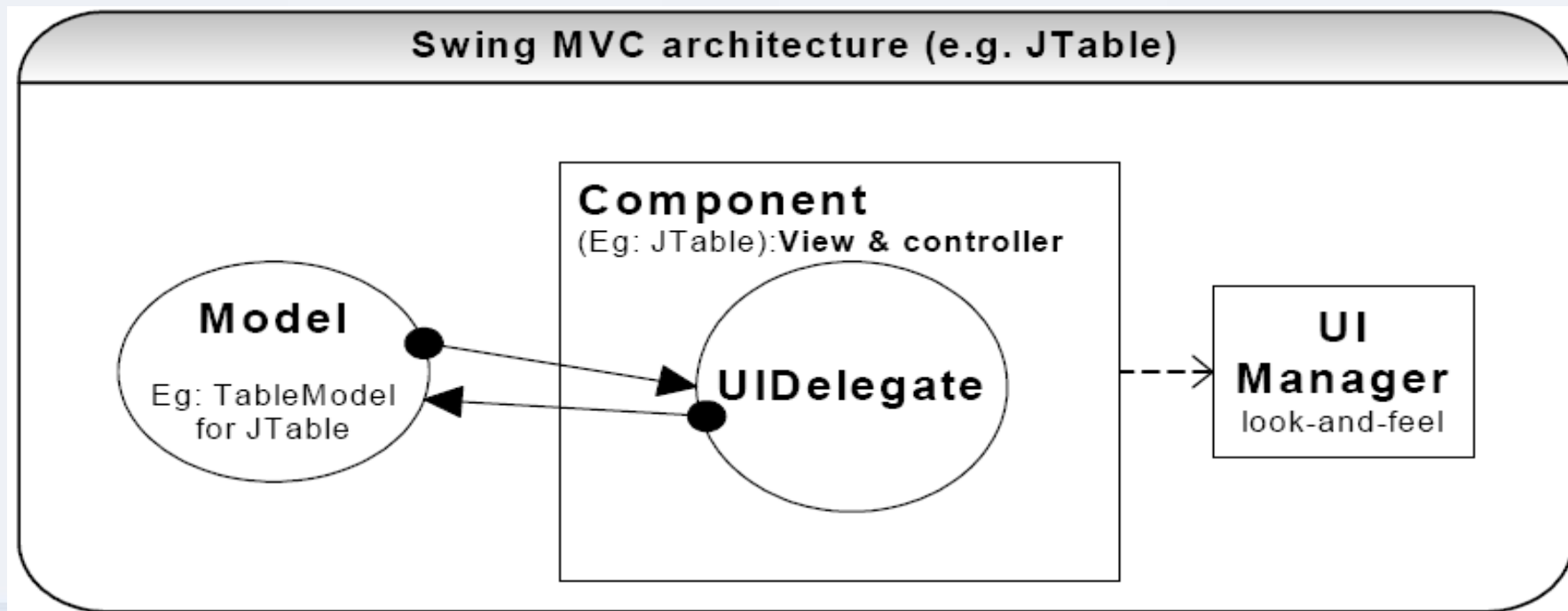


4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

Modelul folosit in tratarea evenimentelor (crearea interactivitatii)

- forma de **arhitectura MVC** (*Model – View – Controller*)
- in care **subsistemul de prezentare** (*View*) si **subsistemul de control** (*Controller*) formeaza un **subsistem de tratare a evenimentelor prin delegare** (*UIDelegate*)

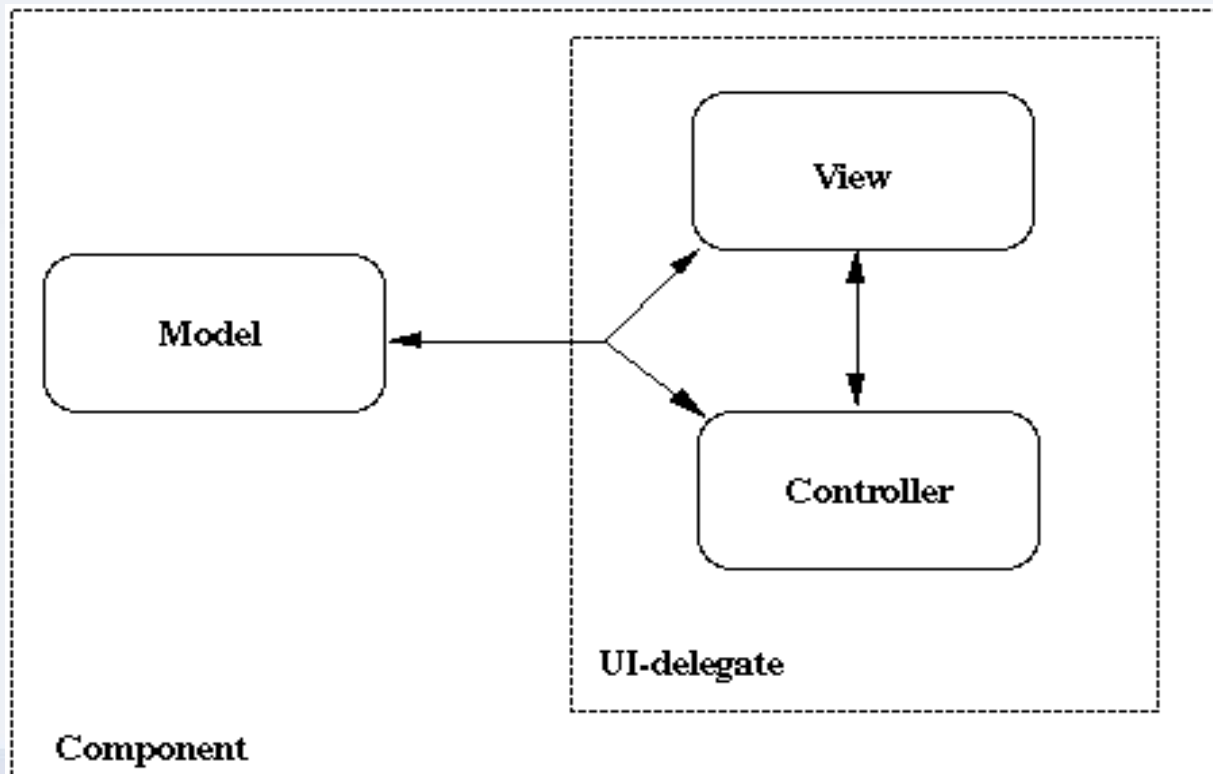


4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

Swing foloseste un **MVC simplificat**: numit **Model – UI-delegate**

- subsistemul **Model** mentine informatiile componentei
- subsistemul **UI-delegate** afiseaza componenta si reactioneaza la evenimentele care se propaga prin componenta

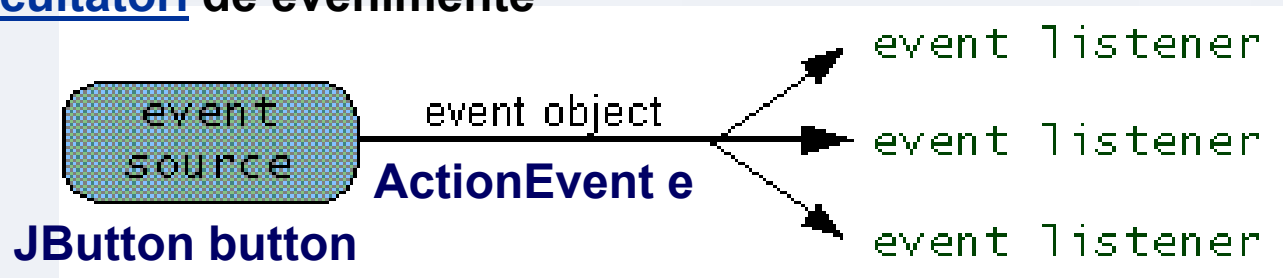


4.2. Interfete grafice Java Swing

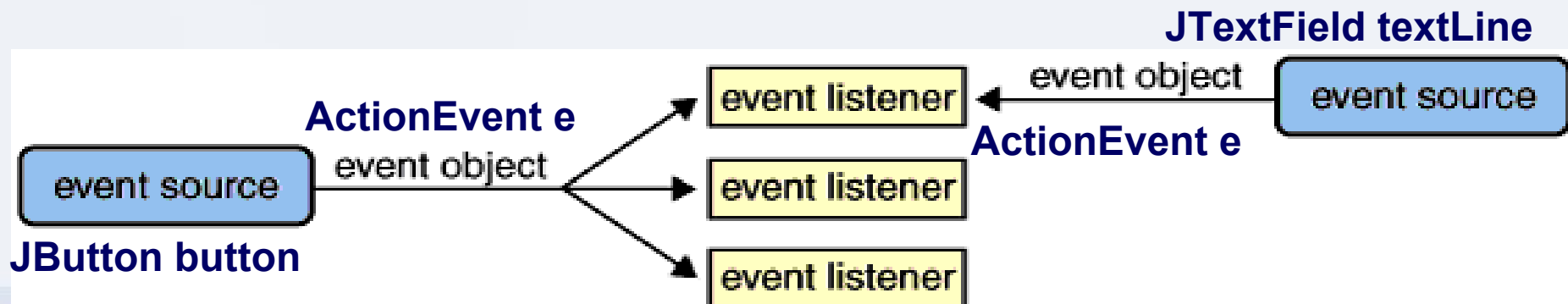
Tratarea evenimentelor in interfetele grafice Swing

Modelul folosit in tratarea evenimentelor (crearea interactivitatii)

- o sursa de evenimente poate trimite obiecte eveniment catre mai multi ascultatori de evenimente



- in plus, mai multe surse de evenimente pot trimite obiecte eveniment catre acelasi ascultator de evenimente



4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

Exemple de **surse** de evenimente si **ascultatoarele** de evenimente asociate lor

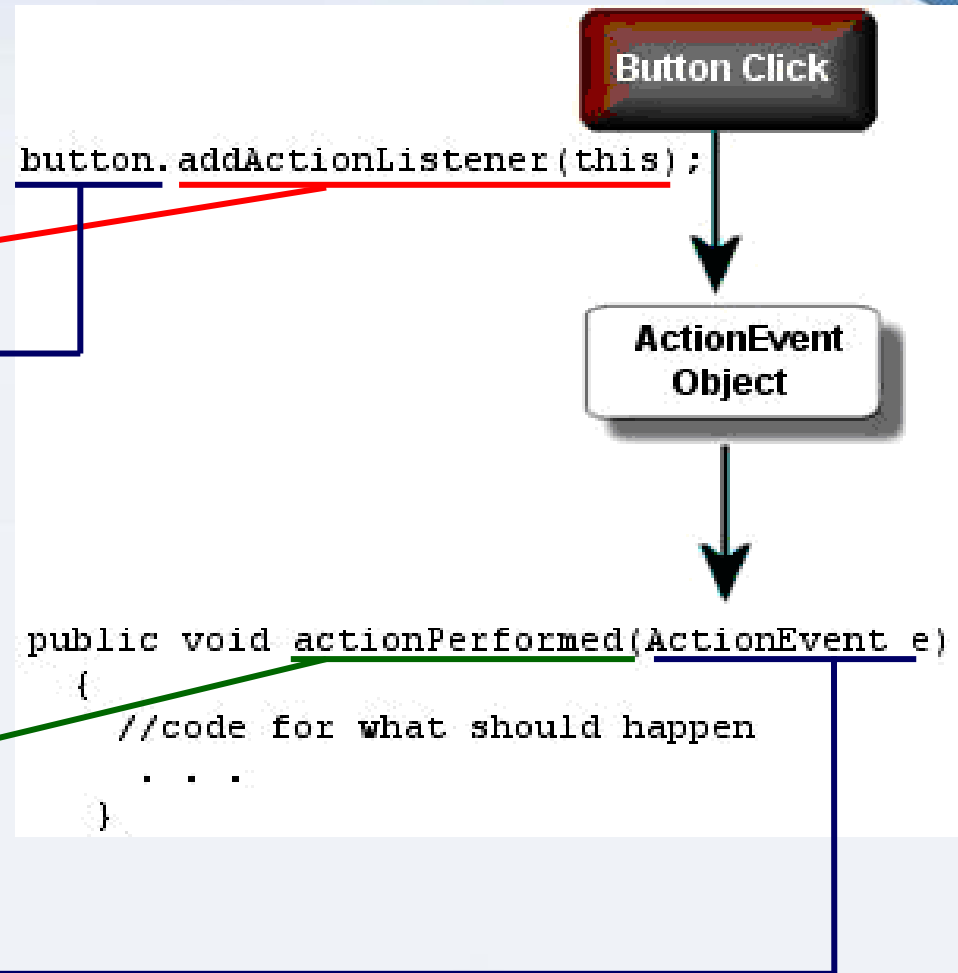
Act that results in the event	Listener type
User <u>clicks a button</u> , or <u>chooses a menu item</u> , or <u>presses Enter while typing in a text field</u>	ActionListener
User closes a frame (main window)	WindowListener
User presses a mouse button <u>while the cursor is over a component</u>	MouseListener
User moves the mouse <u>over a component</u>	MouseMotionListener
Component becomes visible	ComponentListener
Component gets the keyboard focus	FocusListener
Table or list selection changes	ListSelectionListener

4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

Ascultatorul de evenimente
 (ActionListener) trebuie:

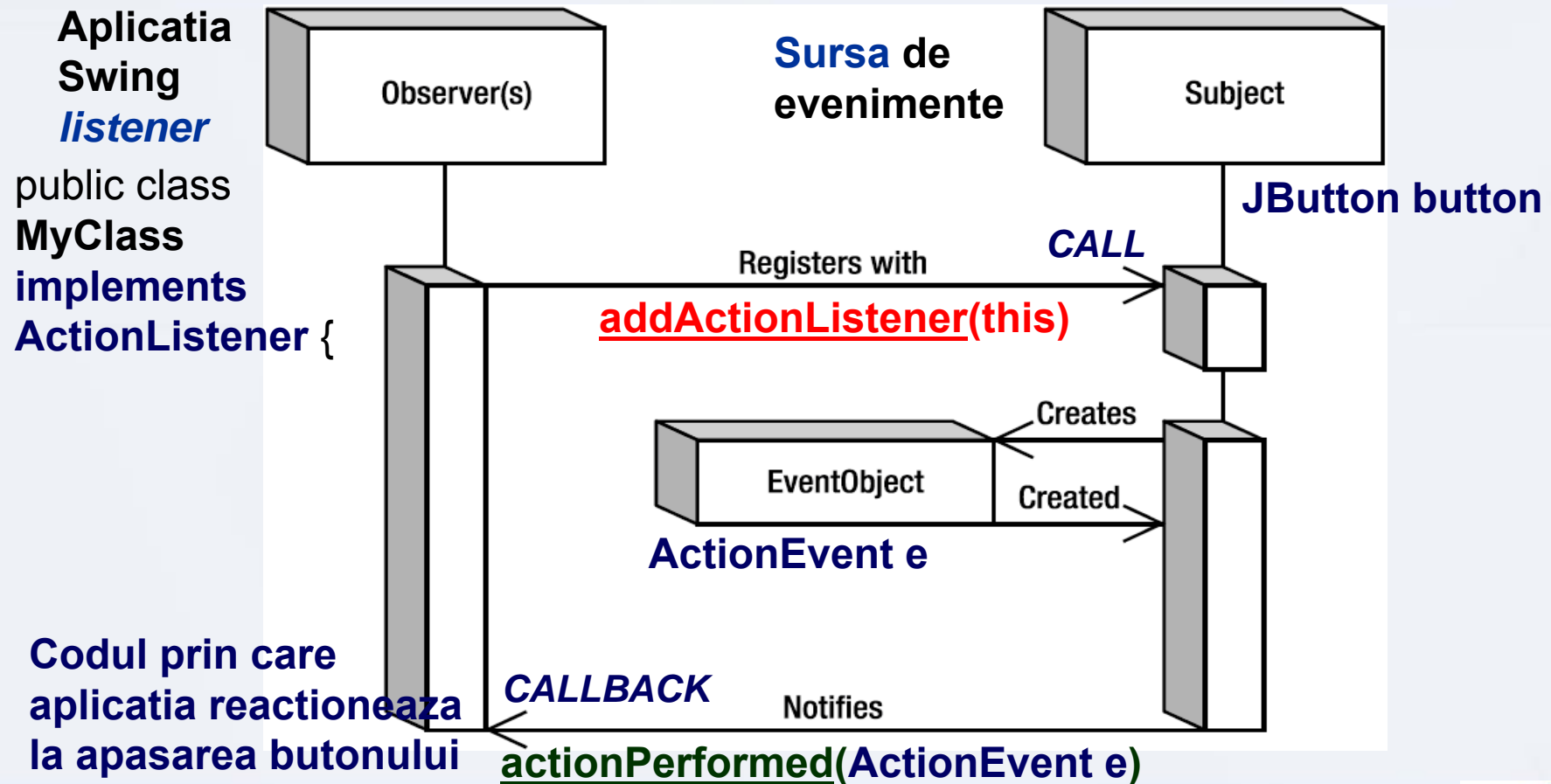
- sa se **INREGISTREZE**
 (addActionListener(this)) la
 sursa de evenimente (button)
- si sa ofere cel putin o metoda
 prin care
 - sursa de evenimente sa il
 poata **NOTIFICA** pe
 ascultatorul evenimentului
 (actionPerformed(...))
 - pasandu-i o referinta catre
 obiectul **eveniment**
 (ActionEvent e)



4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

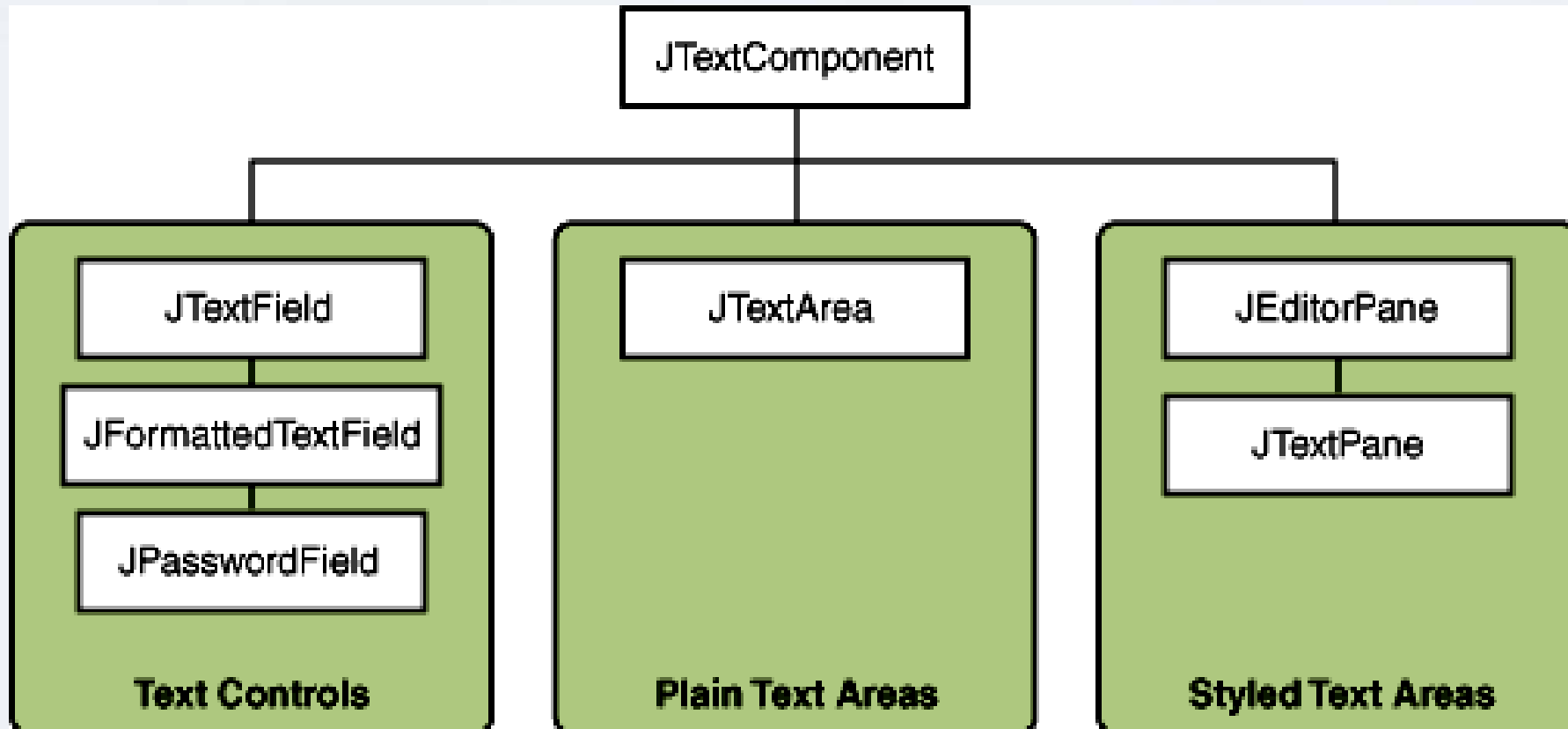
Acesta este un exemplu de pattern de proiectare *Observer* (cunoscut si ca *Publisher-Subscriber*, *REGISTER-NOTIFY*, *callback*, etc.):



4.2. Interfete grafice Java Swing

Componente Swing de tip text

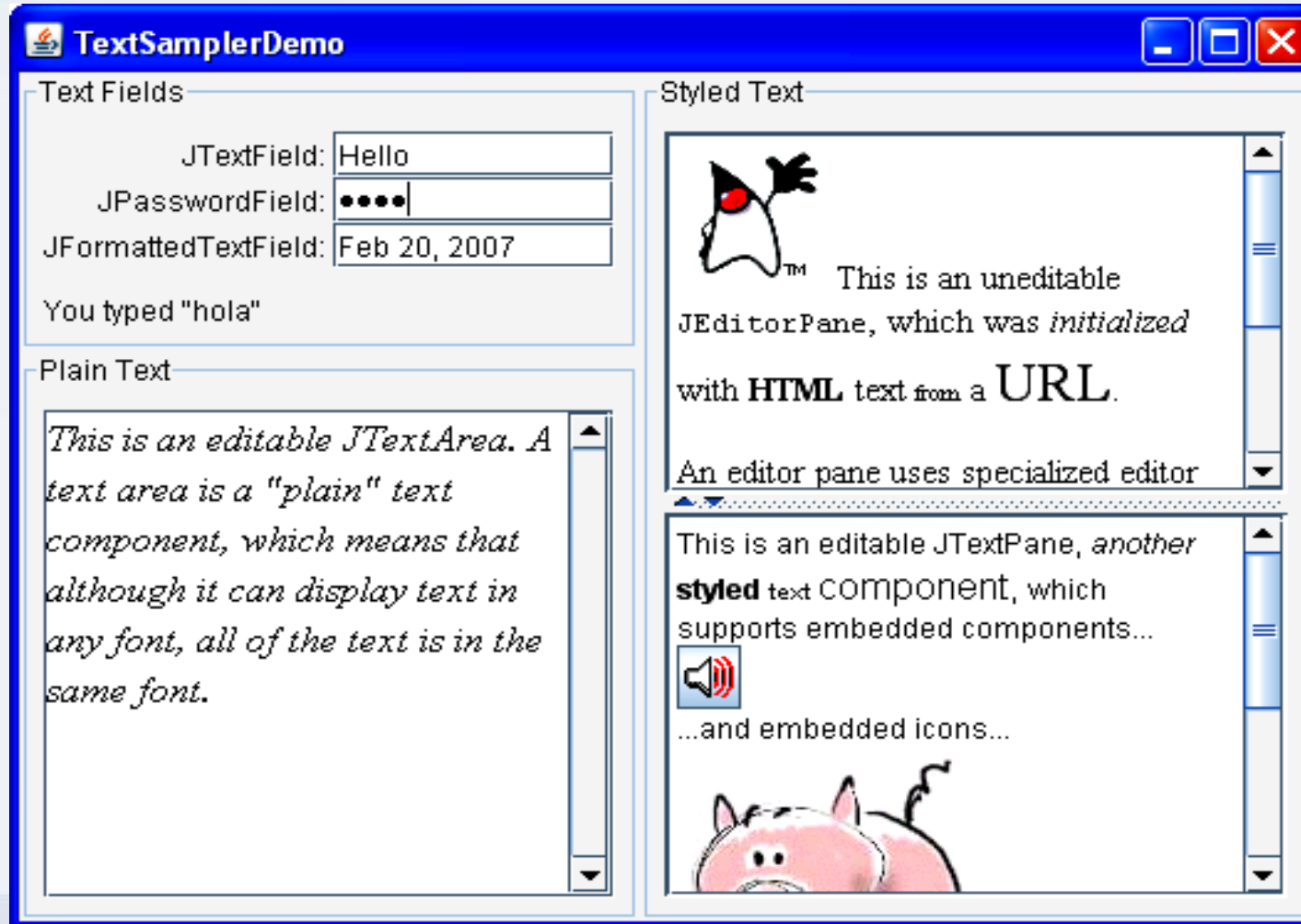
Ierarhia de clase Swing de tip text



4.2. Interfete grafice Java Swing

Componente Swing de tip text

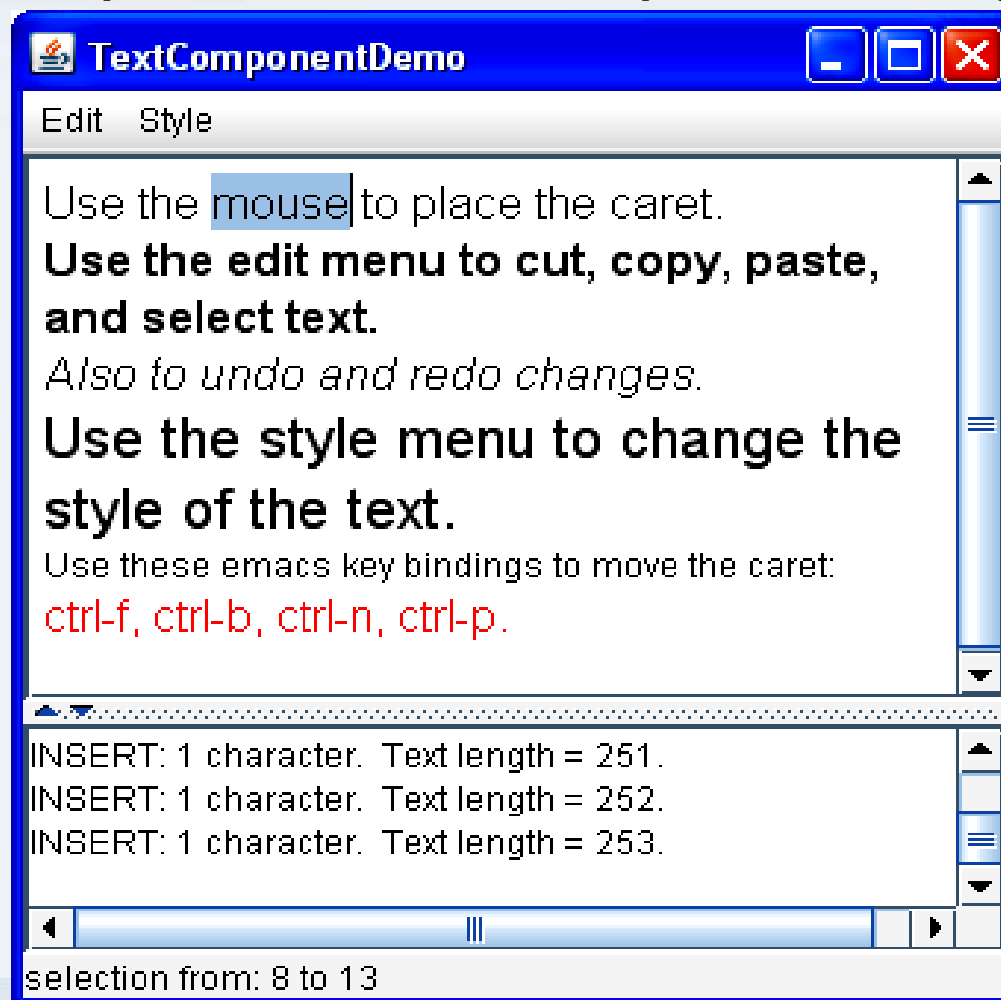
Ilustrarea diferitelor componente Swing de tip text



4.2. Interfete grafice Java Swing

Componente Swing de tip text

Caracteristici care pot fi modificate in componentele Swing de tip text



4.2. Interfete grafice Java Swing

Ferestre de dialog Swing

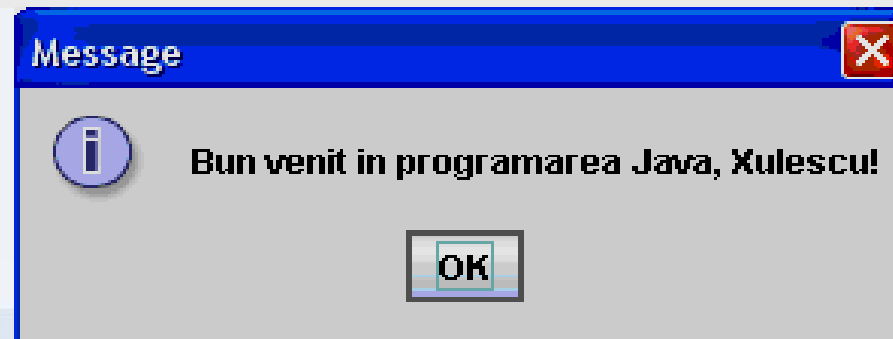
Pentru a obtine de la utilizator o valoare de tip sir de caractere (String):

```
String nume = JOptionPane.showInputDialog("Introduceti-va numele:");
```



Pentru a se afisa un mesaj catre utilizator se foloseste sintaxa:

```
JOptionPane.showMessageDialog(null,"Bun venit in lumea Java,Xulescu!");
```



4.2. Interfete grafice Java Swing

Ferestre de dialog Swing

Ambele situatii impun importul clasei `JOptionPane` din pachetul de clase de biblioteca grafica `javax.swing`, (sau a intregului pachet de clase de biblioteca `javax.swing`).

Pentru a se importa clasa `JOptionPane` din pachetul de clase Swing `javax.swing` se foloseste sintaxa:

```
import javax.swing.JOptionPane;
```

Pentru a se importa intreg pachetul de clase `javax.swing` se foloseste sintaxa:

```
import javax.swing.*;
```

4.2. Interfete grafice Java Swing

Modalitati de a crea containerul de nivel maxim (*top-level*)

1. Utilizarea unui obiect de tip JFrame

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

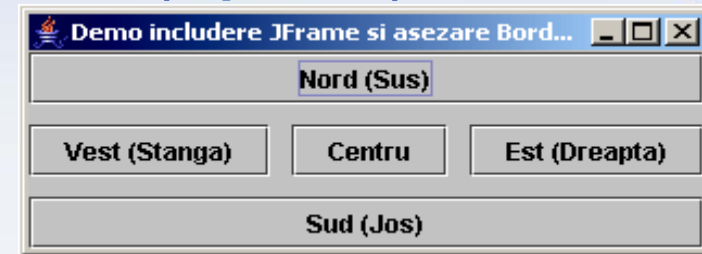
public class IncludereJFrame {
    public static void main(String[] args) {
        // Crearea obiectului cadru, cu titlu specificat
        JFrame cadru = new JFrame("Demo includere JFrame si asezare BorderLayout");

        // Obtinerea panoului de continut intern cadrului (container de componente)
        Container container = cadru.getContentPane();

        // Asezarea componentelor in panou (la 10 pixeli de marginea panoului)
        container.setLayout(new BorderLayout(10, 10));

        // Adaugarea a 5 butoane la panoul cadrului (ferestrei)
        container.add(new JButton("Est (Dreapta)", BorderLayout.EAST);
        container.add(new JButton("Sud (Jos)", BorderLayout.SOUTH);
        container.add(new JButton("Vest (Stanga)", BorderLayout.WEST);
        container.add(new JButton("Nord (Sus)", BorderLayout.NORTH);
        container.add(new JButton("Centru", BorderLayout.CENTER);

        // Stabilirea dimensiunii ferestrei
        cadru.pack();
        // Stabilirea vizibilitatii ferestrei (Atentie: implicit e false!)
        cadru.setVisible(true);
    }
}
```



4.2. Interfete grafice Java Swing

Modalitati de a crea containerul de nivel maxim (*top-level*)

2. Extinderea prin mostenire a clasei JFrame

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

public class ExtensieJFrame extends JFrame {
    public ExtensieJFrame() {
        // Obtinerea panoului de continut intern cadrului (container de componente)
        Container container = getContentPane();
        // Asezarea componentelor in panou (la 10 pixeli de marginea panoului)
        container.setLayout(new BorderLayout(10, 10));
        // Adaugarea a 5 butoane la panoul cadrului (ferestrei)
        container.add(new JButton("Est (Dreapta)"), BorderLayout.EAST);
        container.add(new JButton("Sud (Jos)"), BorderLayout.SOUTH);
        container.add(new JButton("Vest (Stanga)"), BorderLayout.WEST);
        container.add(new JButton("Nord (Sus)"), BorderLayout.NORTH);
        container.add(new JButton("Centru"), BorderLayout.CENTER);
    }
    public static void main(String[] args) {
        // Crearea obiectului cadru
        ExtensieJFrame cadru = new ExtensieJFrame();
        // Adaugarea titlului ferestrei
        cadru.setTitle("Demo extindere JFrame si asezare BorderLayout");
        // Compactarea componentelor
        cadru.pack();
        // Stabilirea vizibilitatii ferestrei (Atentie: implicit e false!)
        cadru.setVisible(true);
    }
}
```


4.2. Interfete grafice Java Swing

Modalitati de a crea containerul de nivel maxim (*top-level*)

3. Extinderea clasei JApplet (in cazul miniaplicatiilor Java)

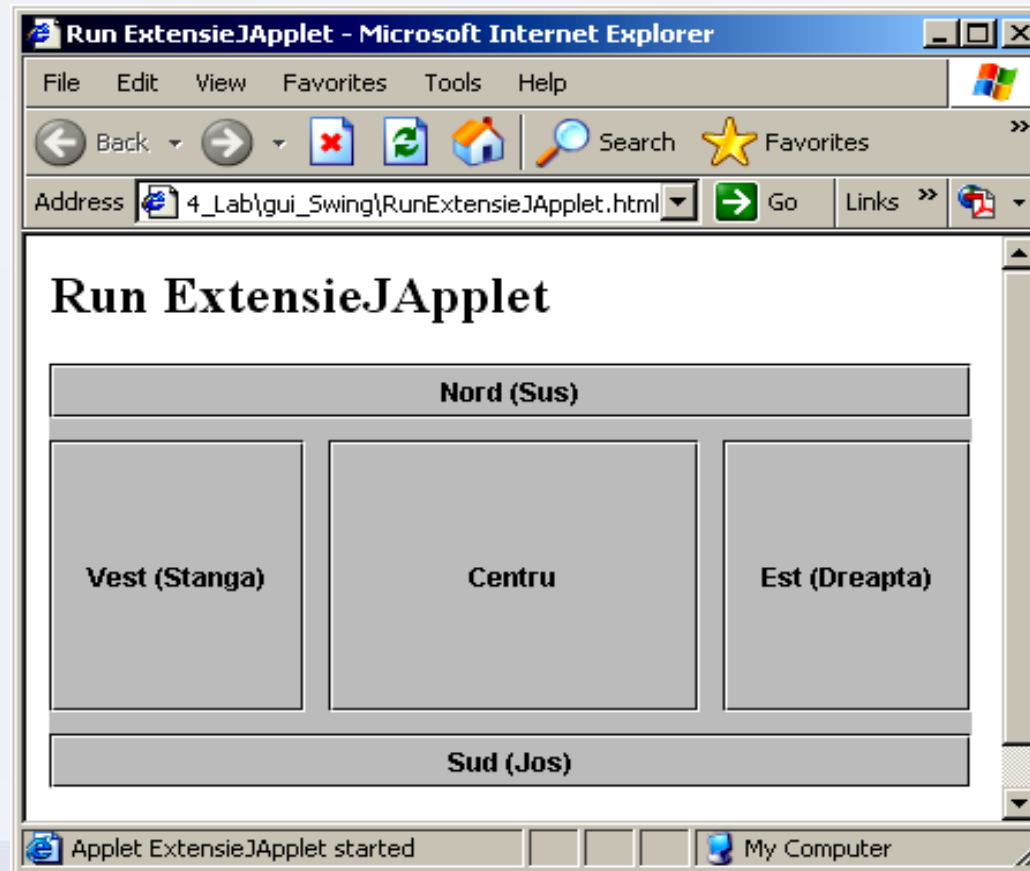
```
import java.awt.*;
import javax.swing.*;
/**
 * Se executa prin includerea intr-o pagina Web a unui tag HTML de genul:
 * <APPLET CODE = "ExtensieJApplet.class" WIDTH = 400 HEIGHT = 200 >
 * </APPLET>
 */
public class ExtensieJApplet extends JApplet {
    /**
     * Metoda de initializare a appletului. Apelata de browser la prima
     * utilizare a appletului, stabileste layout-ul (modul de dispunere a
     * componentelor in panoul de continut) si adauga componentele in panou.
     */
    public void init() {
        // Obtinerea panoului de continut intern cadrului (container de componente)
        Container container = getContentPane();
        // Asezarea componentelor in panou (la 10 pixeli de marginea panoului)
        container.setLayout(new BorderLayout(10, 10));
        // Adaugarea a 5 butoane la panoul appletului
        container.add(new JButton("Est (Dreapta)"), BorderLayout.EAST);
        container.add(new JButton("Sud (Jos)"), BorderLayout.SOUTH);
        container.add(new JButton("Vest (Stanga)"), BorderLayout.WEST);
        container.add(new JButton("Nord (Sus)"), BorderLayout.NORTH);
        container.add(new JButton("Centru"), BorderLayout.CENTER);
    }
}
```


4.2. Interfete grafice Java Swing

Modalitati de a crea containerul de nivel maxim (*top-level*)

Pentru includerea unui applet intr-o pagina HTML in vederea vizualizarii ei se va adauga in corpul fisierului .html urmatorul cod:

```
<APPLET CODE = "ExtensieJApplet.class" WIDTH = 400 HEIGHT = 200 >
</APPLET>
```

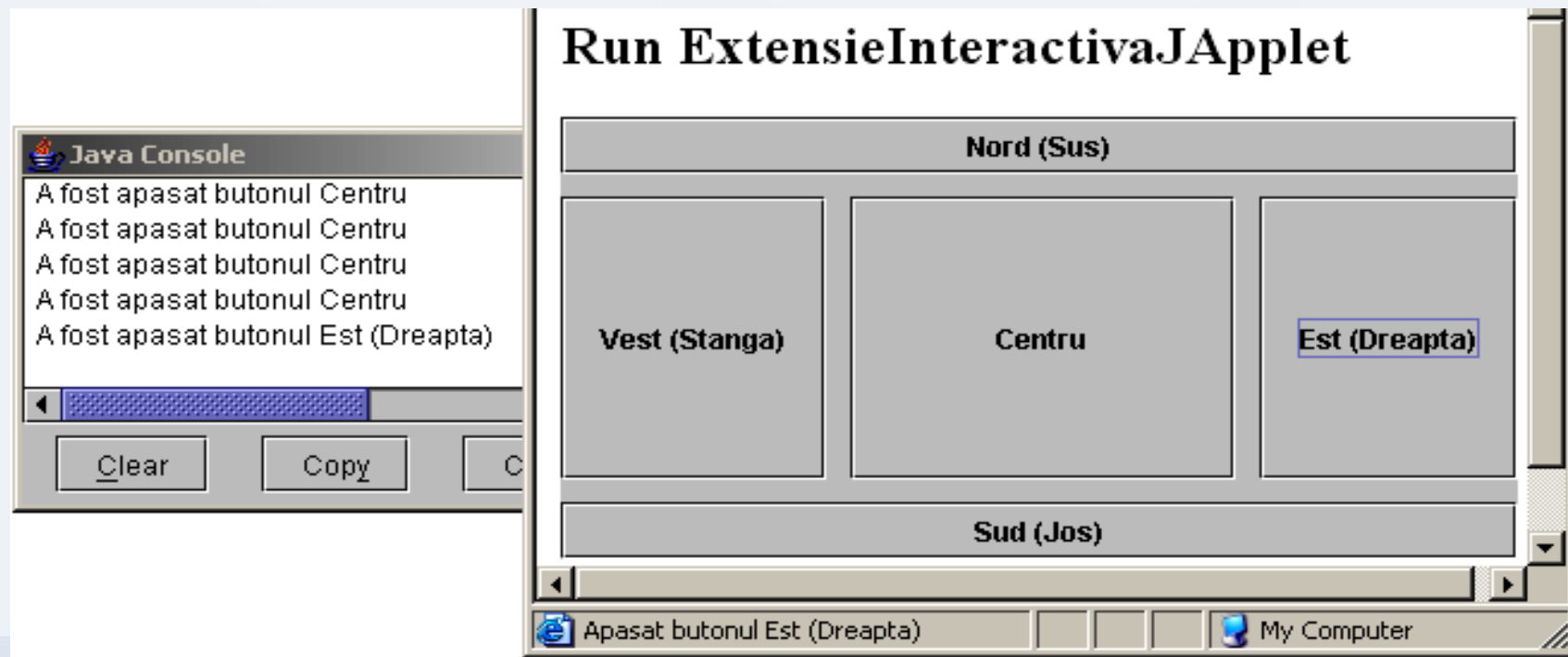


4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

Programul ExtensieInteractivaJApplet ilustreaza:

- crearea unui applet prin extinderea clasei JApplet, si
- tratarea evenimentului "actionare" (ActionEvent) pentru componentele buton (JButton) folosind implementarea interfetei de tratare a evenimentelor (ActionListener)



4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

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

public class ExtensieInteractivaJApplet extends JApplet {

    public void init() {

        // Obtinerea panoului de continut (content pane) creat de browser pentru
        // executia appletului (container in care vor fi plasate componentele)
        Container container = getContentPane();

        // Stabilirea layout-ului panoului, BorderLayout cu spatiu 10 pixeli
        container.setLayout(new BorderLayout(10, 10));

        // Adaugarea a 5 butoane la panoul appletului
        JButton b1 = new JButton("Est (Dreapta)");
        JButton b2 = new JButton("Sud (Jos)");
        JButton b3 = new JButton("Vest (Stanga)");
        JButton b4 = new JButton("Nord (Sus)");
        JButton b5 = new JButton("Centru");

        container.add(b1, BorderLayout.EAST);
        container.add(b2, BorderLayout.SOUTH);
        container.add(b3, BorderLayout.WEST);
        container.add(b4, BorderLayout.NORTH);
        container.add(b5, BorderLayout.CENTER);
    }
}
```

4.2. Interfete grafice Java Swing

Tratarea evenimentelor in interfetele grafice Swing

```
// Crearea unui obiect "ascultator" de "evenimente actionare"
// (pe care le trateaza)
ActionListener obiectAscultatorActionare = new ActionListener() {

    // Tratarea actionarii unui buton
    public void actionPerformed(ActionEvent ev) {

        // Mesaj informare in consola Java
        System.out.println("A fost apasat butonul " + ev.getActionCommand());

        // Mesaj informare in bara de stare
        showStatus("Apasat butonul " + ev.getActionCommand());
    }
};

// Inregistrarea "ascultatorului" de "evenimente actionare" la "sursele"
// de evenimente
b1.addActionListener(obiectAscultatorActionare);
b2.addActionListener(obiectAscultatorActionare);
b3.addActionListener(obiectAscultatorActionare);
b4.addActionListener(obiectAscultatorActionare);
b5.addActionListener(obiectAscultatorActionare);
}
```

4.2. Interfete grafice Java Swing

Utilizarea componentelor grafice Swing pentru lucrul cu text

Programul `EcouGrafic_Swing` ilustreaza utilizarea componentelor grafice Swing pentru lucrul cu text, de tip `JTextField` si `JTextArea`

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

// Ecou grafic bazat pe utilizarea JTextField si a JTextArea
public class EcouGrafic_Swing extends JFrame {

    // Intrare - linie de text grafica (JtextField)
    private static JTextField inTextGrafic;

    // Iesire - zona de text grafica (JtextArea)
    private static JTextArea outTextGrafic;

    // Initializari grafice
    public EcouGrafic_Swing() {

        // Stabilire titlu fereastră (JFrame)
        super("Ecou grafic simplu Swing");

        // Obtinerea panoului de continut intern cadrului (container de componente)
        Container containerCurent = this.getContentPane();

        // Asezarea componentelor in panou
        containerCurent.setLayout(new BorderLayout());
    }
}
```


4.2. Interfete grafice Java Swing

Utilizarea componentelor grafice Swing pentru lucrul cu text

```
// Zona de text non-editabila de iesire (cu posibilitati de defilare)
outTextGrafic = new JTextArea(5, 40);
JScrollPane scrollPane = new JScrollPane(outTextGrafic,
    JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
    JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
containerCurent.add("Center", scrollPane);
outTextGrafic.setEditable(false);

// Camp de text editabil de intrare
inTextGrafic = new JTextField(40);
containerCurent.add("South", inTextGrafic);

System.out.println("\nPentru oprire introduceti '.' si <Enter>\n");

// Inregistrarea "ascultatorului" de "evenimente actionare" la
// "obiectul sursa" intrare de text
inTextGrafic.addActionListener(ascultatorInText);

// Inregistrarea "ascultatorului" de "evenimente fereastră" la
// "sursa" (fereastră curenta)
this.addWindowListener(ascultatorInchidere);

pack();
show();

// Cerere focus pe intrarea de text din fereastră curenta
inTextGrafic.requestFocus();
}
```


4.2. Interfete grafice Java Swing

Utilizarea componentelor grafice Swing pentru lucrul cu text

```
// Crearea unui "ascultator" de "evenimente actionare", obiect al unei
// clase "anonime" care implementeaza interfata ActionListener
ActionListener ascultatorInText = new ActionListener() {

    // Tratarea actionarii intrarii de text (introducerii unui "Enter")
    public void actionPerformed(ActionEvent ev) {

        // Citirea unei linii de text din intrarea de text grafica
        String sirCitit = inTextGrafic.getText();

        // Pregatirea intrarii de text pentru noua intrare (golirea ei)
        inTextGrafic.setText("");

        // Scrierea liniei de text in zona de text grafica
        outTextGrafic.append("S-a introdus: " + sirCitit + "\n");

        // Conditie terminare program
        if (sirCitit.equals(new String(".")))
            System.exit(0);
    }
};

// Punctul de intrare in program
public static void main (String args[]) {

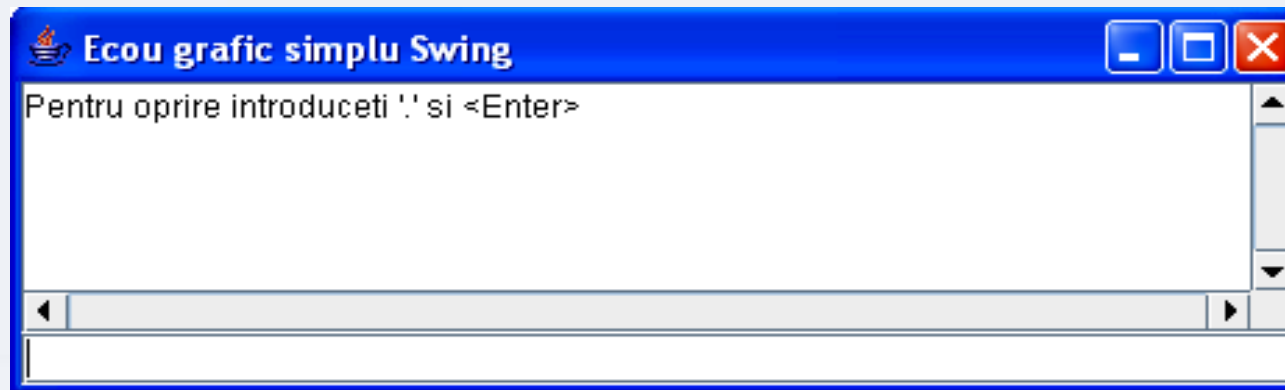
    EcouGrafic_Swing ecouGraficJTFJTA = new EcouGrafic_Swing();
}
}
```

4.2. Interfete grafice Java Swing

Utilizarea componentelor grafice Swing pentru lucrul cu text

Executia programului EcouGrafic_Swing

- initial



- dupa introducerea "Hello! <enter>"

