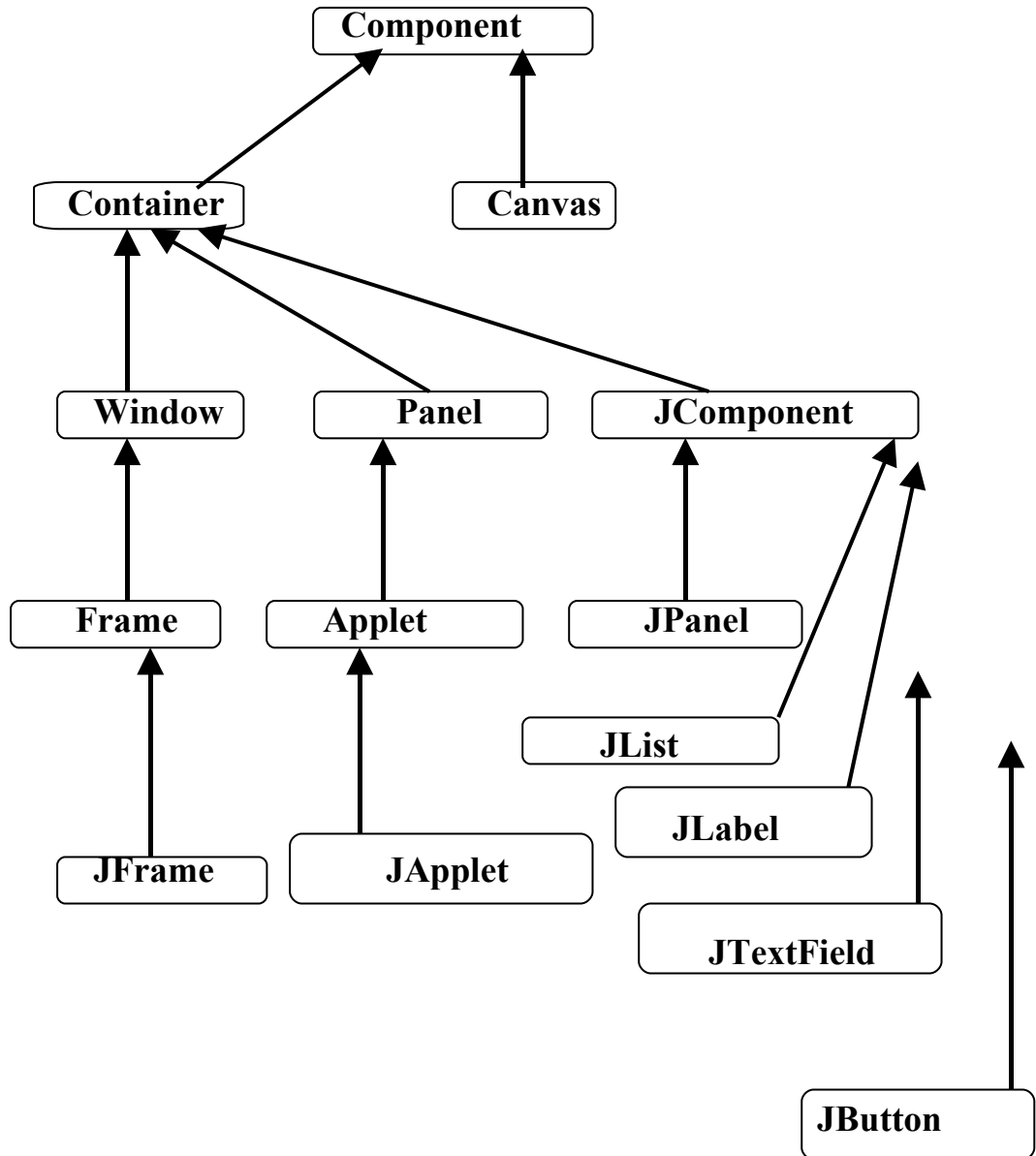


Paketi SWING klassihierarhia



SWING komponendid paiknevad paketi javax.swing. SWING komponendid on vaadeldavad paketi java.awt Vastavate komponentide alamkomponentidena. SWING komponendid on kirjutatud javas, seetõttu nimetatakse neid kergekaalulisteks.

SWING komponentide kasutamine

- **Kõigepealt impordime vastavad paketid ja klassid**

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

- **Kirjeldame klassi SwingUI**

```
Class SwingUI extends JFrame  
implements ActionListener  
{
```

Klass SwingUI pärineb klassist JFrame ning realiseerib liidest ActionListener. JFrame klass pärineb klassist Frame, mis kuulub graafilise kasutajaliidese AWT koosseisu. AWT komponendid on realiseeritud erinevate töökeskondade (Solaris, Windows, JavaOs jt.) jaoks “native”-koodis (tavaliselt assembleris). Swing komponendid laiendavad suhteliselt primitiivse AWT võimalusi ning on kirjutatud jas.

- **Klassi SwingUI muutujad:**

```
Jlabel text , clicked;  
Jbutton button , clickButton;  
Jpanel panel;  
private boolean _clickMeMode = true;
```

Muutuja boolean_clickMeMode on kättesaadav ainult klassi SwingUI meetoditele.

- **Kirjeldame konstruktormetodi.**

```
SwingUI () {  
  
Text = new JLabel(“Lihne programm”);  
Clicked = new JLabel ( “Vajutati nupule”);  
  
Button = new Jbutton ( “Vajuta !!!”);
```

```

//Lisame nupule sündmuste kuulaja
button.addActionListener(this);

clickButton = new JButton("Vajuta uuesti!!");

//Lisame nupule sündmuste kuulaja
clickButton.addActionListener(new NupuAction );

// Loome objekti panel

panel = new JPanel( );

//Määrame ära paigutusohjuri ja taustavärvi
panel.setLayout(new BorderLayout(1,1));
panel.setBackground(Color.white);

// Paigutame pealdise ja nuppu paneeli
getContentPane().add(panel);
panel.add(BORDERLAYOUT.CENTER,text);
panel.add(BORDERLAYOUT.SOUTH,button);
}

```

- **Main() – meetod**

```

public static void main (String[ ] args {

// Loome akna

    SwingUI frame = new SwingUI();
    frame.setTitle("Näide");

// Kood akna sulgemise kood
    WindowListener l = new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    };
    frame.addWindowListener(l);
// Akna kuvamise kood
    frame.pack();
    frame.setVisible(true);
}
}

```

- **Sündmuste töötlus**

```
class NupuAction implements ActionListener {  
  
    public void actionPerformed (ActionEvent event) {  
        Object source = event.getSource();  
        If (_clickMeMode) {  
            text.setText("Nupule vajutati");  
            button.setText("Vajuta uuesti");  
            _clickMeMode = false;  
        }  
        else {  
            text.setText("Lihtne programm");  
            button.setText("Vajuta !!!");  
            _clickMeMode = true;  
        }  
  
    }  
}
```

Nupusündmuste töötluseks kasutatakse

Nupusündmuste töötluseks kasutatakse *actionPerformed* meetodit, mis kirjeldatakse omaette klassis *NupuAction*. Akna sulgemiseks kasutatakse *windowClosing()* –meetodit , mille tühikirjeldus paineb liideses *WindowListener*. Kuna vastavas liideses on kokku kirjeldatud 7 erinevat aknaga seotud meetodit , siis tuleb kasutusele võtta vastav adaptertüüpi klass — *WindowAdapter*.

