

Objektorientierte Software-Entwicklung

Priv.-Doz. Dr. Rolf Hennicker

08.01.2003

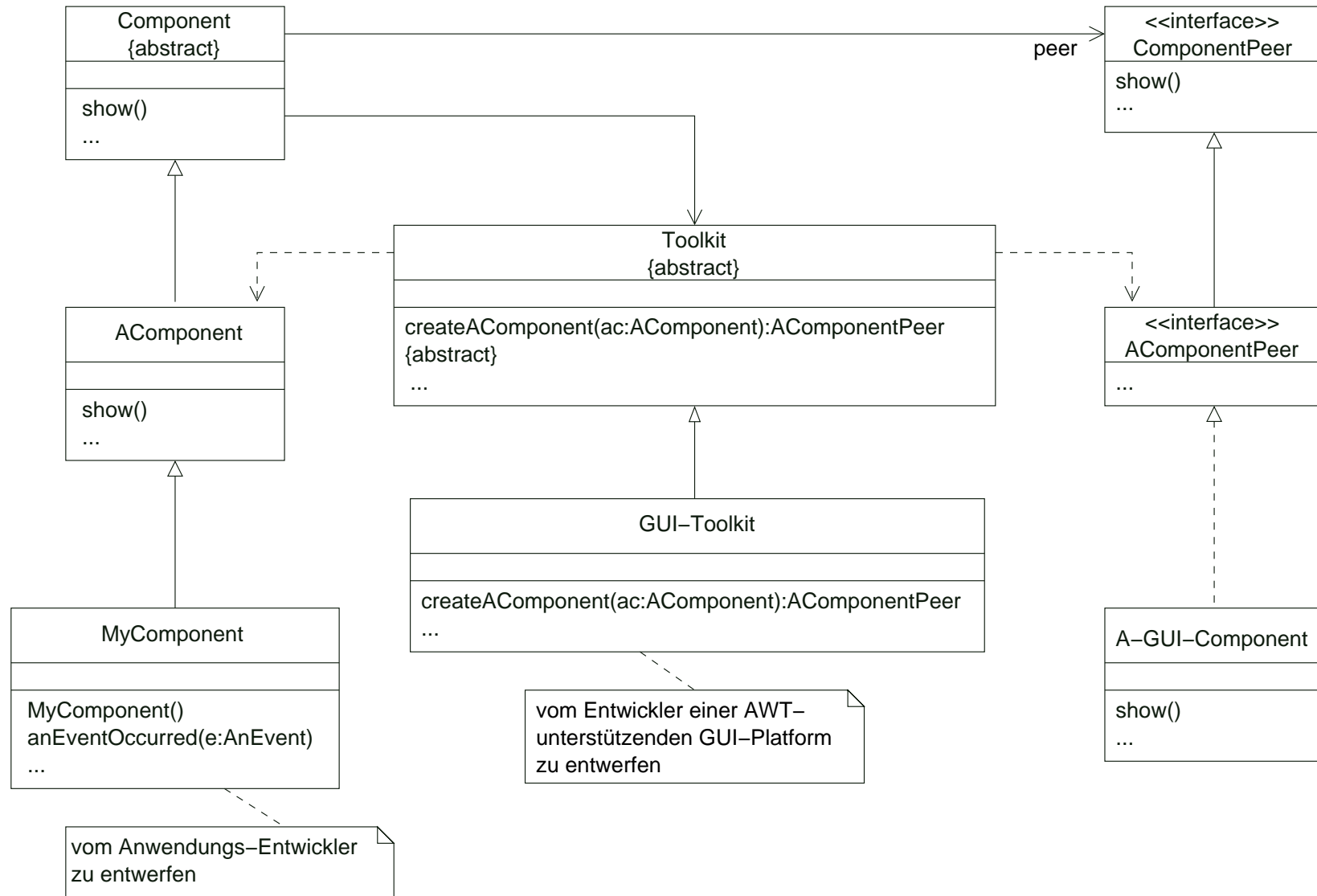


AWT und Swing

AWT (Abstract Window Toolkit) und Swing

- AWT und Swing bieten eine Klassenbibliothek zur Programmierung grafischer Benutzerschnittstellen (GUIs) für Java Programme (Pakete `java.awt`, `javax.swing`)
- *Grundidee*: plattformunabhängige Konstruktion von GUIs
- *Entwicklung*:
 - AWT 1.0
 - AWT 1.1 (neues Event-Handling)
 - Swing (ergänzt AWT und ersetzt die meisten AWT-Komponenten durch "Lightweight"-Komponenten)

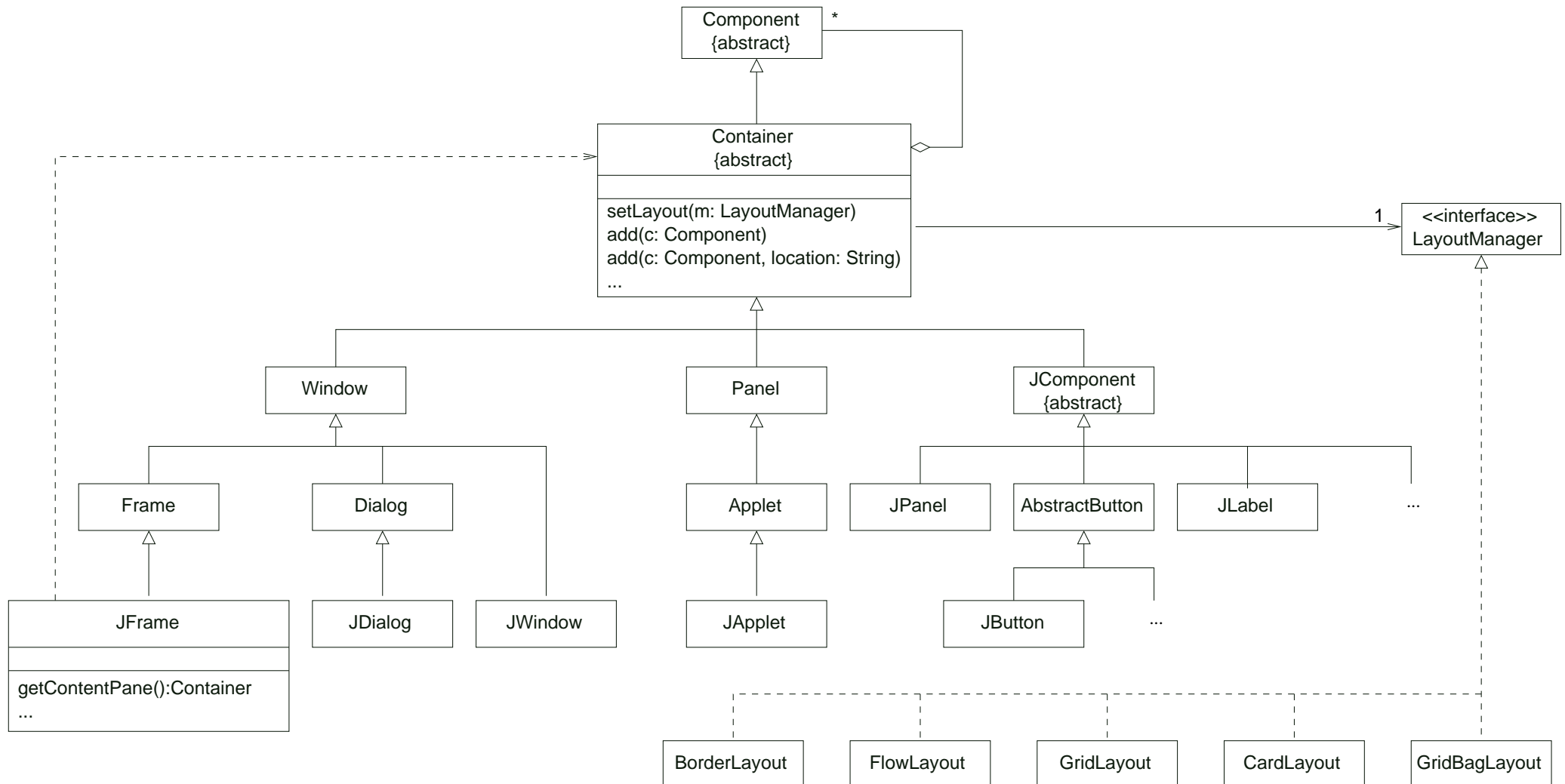
1. Grundkonzepte des AWT



Grundkonzepte des AWT (Zusammenfassung)

- AWT-Komponenten werden von einem Toolkit in entsprechende GUI-Komponenten einer speziellen Plattform ("native components") übersetzt.
- Die plattformspezifischen GUI-Komponenten müssen ein entsprechendes Peer-Interface (des AWT) implementieren. Das Peer-Interface beschreibt die Anforderungen an die GUI-Komponente.
- Soll eine spezielle GUI-Plattform AWT unterstützen, dann muss ein entsprechendes GUI-Toolkit implementiert werden. Die abstrakte Klasse Toolkit (des AWT) beschreibt die Anforderungen an das GUI-Toolkit.
- Der Anwendungs-Entwickler muss die zur Anwendung gehörigen (plattformunabhängigen) GUI-Komponenten entwerfen.

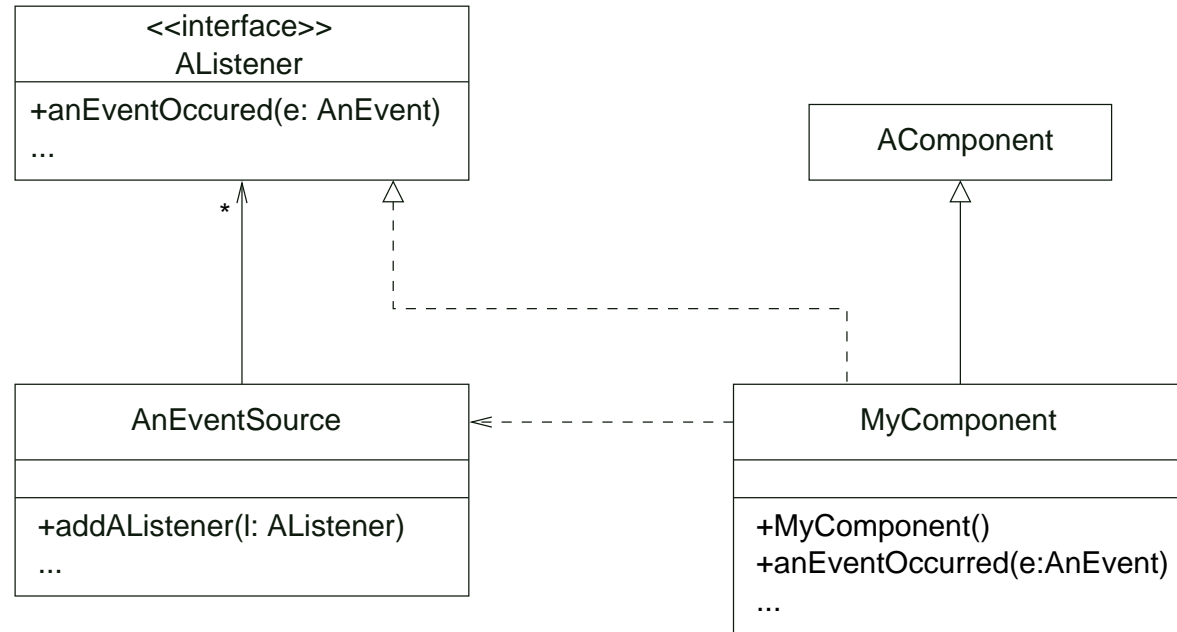
2. Komponenten-Hierarchie von AWT/Swing



Komponenten-Hierarchie (Zusammenfassung)

- Alle mit "J" beginnenden Klassen (und einige weitere) gehören zu Swing.
- In Swing werden unterschieden:
 - Heavyweight-Komponenten (JFrame, JDialog, JWindow, JApplet)
 - Lightweight-Komponenten (alle Spezialisierungen von JComponent)
- Heavyweight-Komponenten werden (wie AWT-Komponenten) in native Komponenten einer konkreten GUI-Plattform übersetzt.
- Heavyweight-Komponenten haben einen Container (Zugriff mit "getContentPane"), in dem die Lightweight-Komponenten gezeichnet werden.
- Jeder Container besitzt einen Layout-Manager.
- Mit "add" können neue Komponenten zu einem Container hinzugefügt werden (entsprechend des eingestellten Layout-Managers).

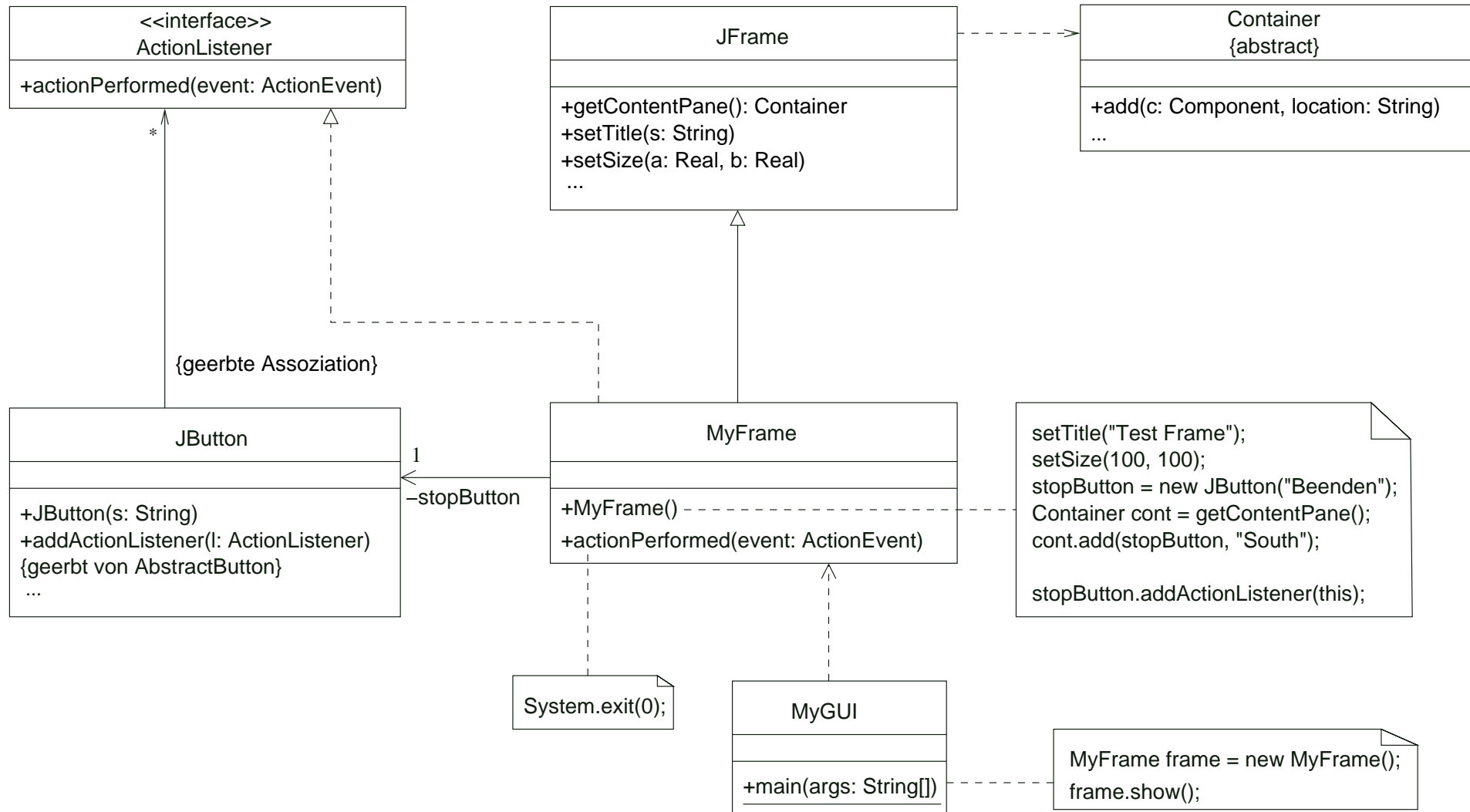
3. Ereignisbehandlung in AWT



Ereignisbehandlung (Zusammenfassung)

- In AWT werden 10 verschiedene Ereignisklassen unterschieden: KeyEvent, MouseEvent, ActionEvent, WindowEvent, ...
- Ist eine Komponente an Ereignissen eines bestimmten Typs interessiert, dann muss sie:
 1. sich bei der Komponente, in der ein solches Ereignis auftreten kann (AnEventSource) als "Listener" registrieren (addListener).
 2. die beim Eintritt eines solchen Ereignisses aufgerufene Operation (anEventOccured) der passenden Listener-Schnittstelle (AListener) implementieren.
- Listener-Schnittstellen sind z.B. KeyListener, MouseListener, ActionListener, WindowListener.
- Operationen von Listener-Schnittstellen sind z.B. actionPerformed(von ActionListener), windowClosing (von WindowListener).

4. GUI-Modellierung: Ein einfaches Beispiel





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

class GUIBeispiel {

    public static void main (String[] args) {
        MyFrame frame = new MyFrame();
        frame.show();
    }
}
```

```
class MyFrame extends JFrame implements ActionListener {

    private JButton stopButton;

    public MyFrame() {
        setSize(200,200);
        setTitle("TestFrame");
        stopButton = new JButton("Beenden");

        Container cont = getContentPane();
        cont.add(stopButton, "South");

        stopButton.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
}
```

Bemerkung

Damit das Programm auch ordnungsgemäß durch Schließen des Fensters beendet werden kann, ist die Operation `windowClosing` des `WindowListener`-Interfaces entsprechend zu implementieren.

GUI-Beispiel mit WindowCloser:

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

class GUIBeispielMitWindowCloser {

    public static void main (String[] args) {
        MyFrame frame = new MyFrame();
        frame.show();
    }
}
```

```
class MyFrame extends JFrame implements ActionListener {

    private JButton stopButton;

    public MyFrame() {
        setSize(200,200);
        setTitle("TestFrame");
        stopButton = new JButton("Beenden");

        Container cont = getContentPane();
        cont.add(stopButton, "South");

        stopButton.addActionListener(this);

        this.addWindowListener(new WindowCloser());
    }

    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
}
```

```
/* WindowCloser ist hier eine innere Klasse, die die AWT-Klasse WindowAdapter erweitert.  
WindowAdapter implementiert alle Operationen des WindowListener-Interface durch  
Dummy-Implementierungen. */
```

```
private class WindowCloser extends WindowAdapter {  
    public void windowClosing(WindowEvent event) {  
        System.exit(0);  
    }  
}
```