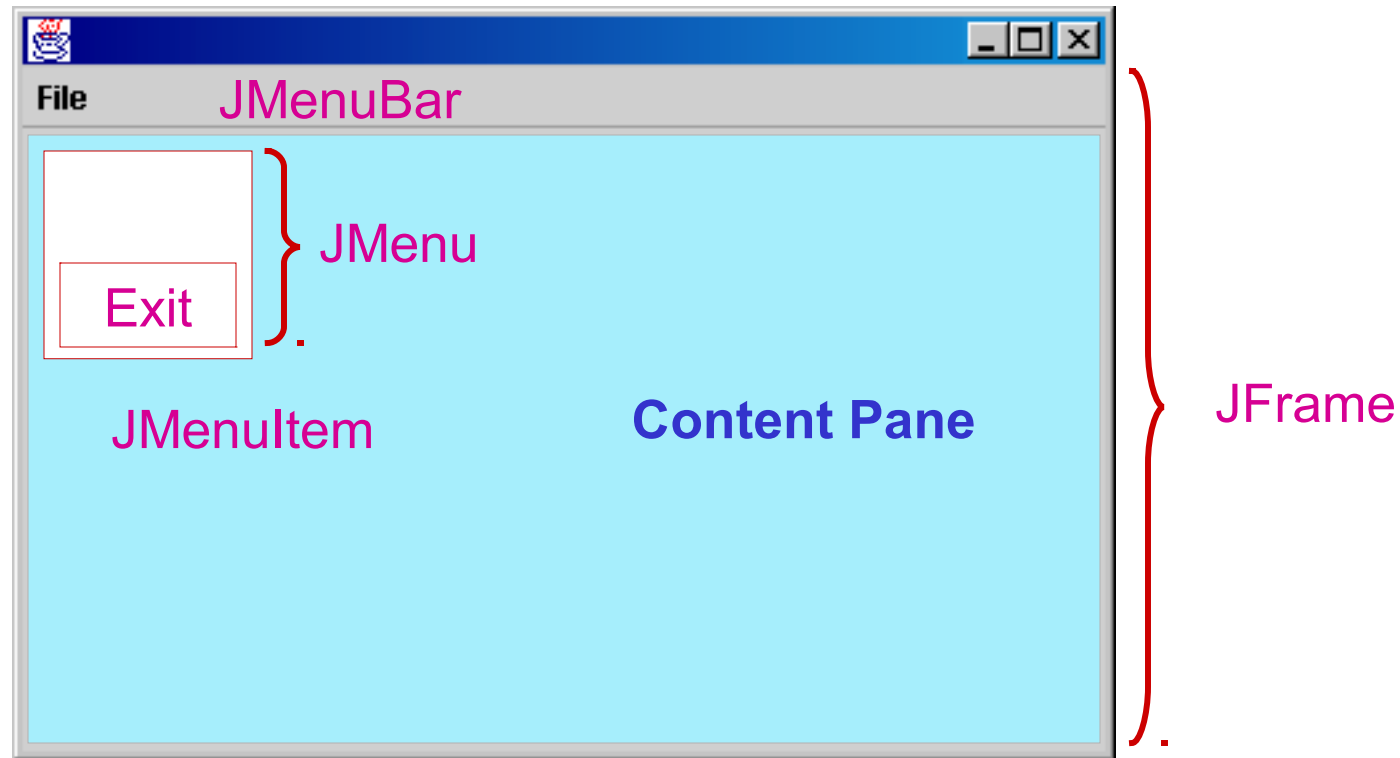
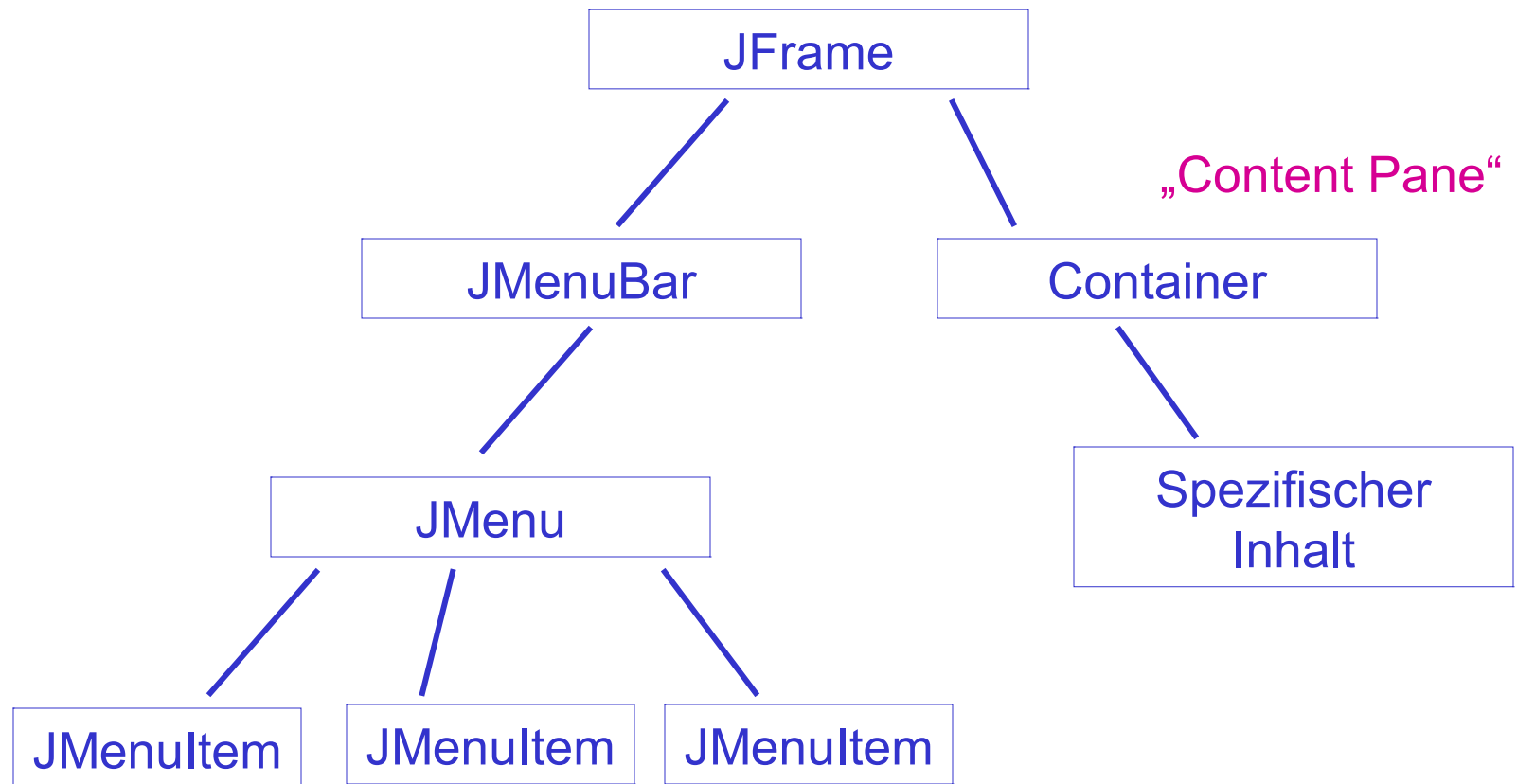


8. Aufbau eines Rumpfprogramms

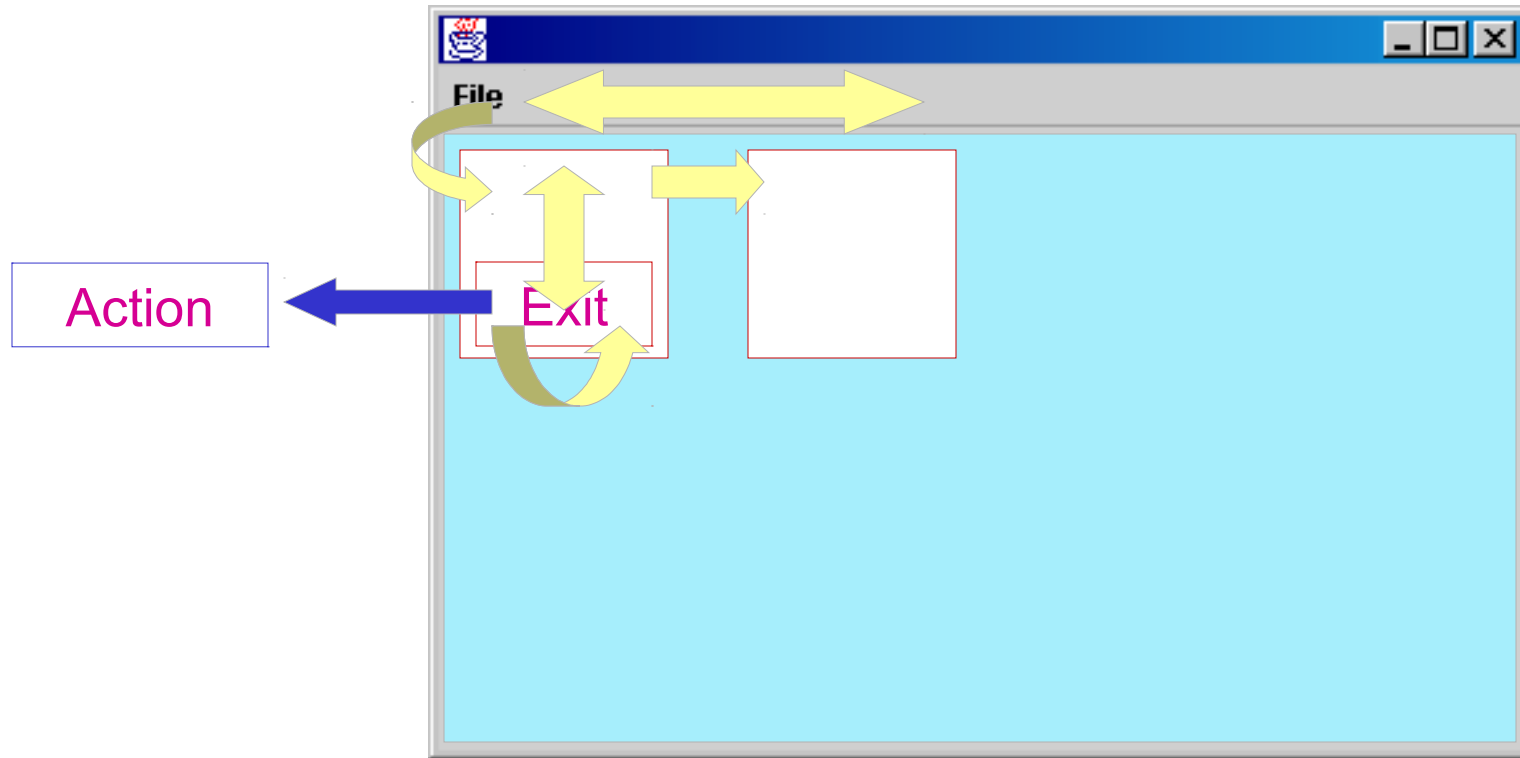
8.1 Rumpfprogramm



8.1 Rumpfprogramm



8.1 Rumpfprogramm



8.2 Programmaufbau

```
public class SimpleFrame extends JFrame {
    private JMenuBar menubar;
    private JMenu file_menu;
    private JMenuItem file_new;
    ...

    SimpleFrame () {
        nächste Folie
    }

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

8.2 Programmaufbau

```
SimpleFrame () {  
  
    getContentPane().setLayout ( new BorderLayout() );  
    menubar = new JMenuBar ( );           Erzeugen des Menubars  
    setJMenuBar ( menubar );             Methode des JFrame zum setzen  
                                         des MenuBar  
    file_menu = new JMenu ( "File" );     Erzeugen eines Menus  
    menubar.add ( file_menu );           Hinzufügen zum Menubar  
  
    file_new = new JMenuItem ( "New" );   Menu-Eintrag  
    file_menu.add ( file_new );           Hinzufügen zum Menu  
    file_new.addActionListener ( new NewAction( ) ); Aktion  
    getContentPane().add ( ... das Haupt-Pane ... );  
    pack ( );                             Layout berechnen  
    setVisible ( true );                  darstellen  
}
```

8.2 Programmaufbau

... Fortsetzung der Klasse Simple Frame ...

```
private class NewAction implements ActionListener
{

    public void actionPerformed ( ActionEvent e )
    {

        // new file: hier kommt die Anwendungslogik

    }

}
}
```

8.3 Class javax.swing.JFrame

Component → Container →
Window → Frame → JFrame

Beschreibung:

An extended version of `java.awt.Frame` that adds support for the JFC/Swing component architecture.

Konstruktoren:

```
JFrame ()
```

```
JFrame ( String title )
```


8.3 Class javax.swing.JFrame

Geerbte Eigenschaften/Methoden von Frame

```
String getTitle();          void setTitle ( String title );
int getState();            void setState ( int state );

static final int NORMAL
static final int ICONIFIED

boolean isResizable();
void setResizable ( boolean resizable );

java.awt.Image getIconImage();
void setIconImage ( java.awt.Image image );
```

8.3 Class javax.swing.JFrame

Geerbte Eigenschaften/Methoden von Window

```
void addWindowListener ( WindowListener listener );
```

```
void pack();
```

Geerbte Eigenschaften/Methoden von Component

Alle Basiseigenschaften, insbesondere:

```
void setVisible ( boolean visible );
```

8.3 Class javax.swing.JFrame

Eigenschaft: Content Pane

```
Container getContentPane();
```

Eigenschaft: Menubar

```
void setJMenuBar ( JMenuBar menubar );  
JMenuBar getJMenuBar();
```

8.3 Class javax.swing.JFrame

Eigenschaft: Default-Close-Operation

```
void setDefaultCloseOperation ( int hint );
```

Beschreibung:

Unlike a Frame, a JFrame has some notion of how to respond when the user attempts to close the window. The default behavior is to simply hide the JFrame when the user closes the window. To change the default behavior, you invoke the method `setDefaultCloseOperation(int)`. To make the JFrame behave the same as a Frame, use `setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE)`.

Konstante:

```
static int WindowConstants.DISPOSE_ON_CLOSE  
static int WindowConstants.HIDE_ON_CLOSE  
static int WindowConstants.DO_NOTHING_ON_CLOSE  
static int JFrame.EXIT_ON_CLOSE  
(siehe interface javax.swing.WindowConstants)
```

8.3 Menüklassen

Vorbemerkung

- Die Menüklassen werden hier noch nicht vollständig besprochen, sondern nur in Auszügen
- Eine detailliertere Beschreibung erfolgt später im Zusammenhang mit AbstractButton

Class javax.swing.JMenuBar

Konstruktoren

`JMenuBar ()`

Add-Methoden

`add (JMenu menu) ;`

8.3 Menüklassen

Class javax.swing.JMenu

Konstruktoren

```
JMenu(String text)
JMenu(String text,
        boolean tearOff)
```

Add-Methoden

```
add(Component component) ;
add(JMenuItem menuItem) ;
```

Bemerkungen

- Ist **tearOff** wahr, so kann man das Menü von der Fläche abziehen
- Da **JMenu** eine Subklasse von **JMenuItem** ist, kann man bei der add-Methode auch ein weiteres Menü angeben → kaskadierte Menüs

8.3 Menüklassen

Class javax.swing.JMenuItem

Konstruktoren

```
JMenuItem(String text)  
JMenu(String text,  
        Icon icon)
```

Methoden

```
setEnabled ( boolean enabled );
```

Action-Listener (siehe AbstractButton)

```
void addActionListener(ActionListener listener);  
void removeActionListener(ActionListener listener);
```

8.3 Menüklassen

Interface java.awt.event.ActionListener

Eine Methode für die Aktion

```
void actionPerformed(ActionEvent event) ;
```

Eventklasse java.awt.event.ActionEvent

```
int getModifiers() ;      Liefert eine Event-Maske aus den  
                           gedrückten Modifiern
```

```
Modifier:  ActionEvent.ALT_MASK  
           ActionEvent.CTRL_MASK  
           ActionEvent.META_MASK  
           ActionEvent.SHIFT_MASK
```


8.3 Menüklassen

Beispiel: MainFrame

MainFrameMenuBar

```
public class MainFrame extends JFrame {
    private MainFrameMenuBar menubar;
    public MainFrame() {
        menubar = new MainFrameMenuBar();
        setJMenuBar(menubar);
        pack();
    }
    public static void main(String[] args) {
        MainFrame frame = new MainFrame();
        frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
        frame.show();    (normal nicht anwenden, sondern
                          einen WindowListener verwenden)
    }
}
```

8.3 Menüklassen

Beispiel: `MainFrame`

`MainFrameMenuBar`

```
public class MainFrameMenuBar extends JMenuBar {
    private JMenu file,edit,help;
    private JMenuItem open,save,exit,cut,copy,paste,info;

    public MainFrameMenuBar() {
        file = new JMenu("File"); add(file);
        open = new JMenuItem("Open"); file.add(open);
        save = new JMenuItem("Save"); file.add(save);
        exit = new JMenuItem("Exit"); file.add(exit);

        ... analog für alle anderen Menüs ...
        exit.addActionListener(new ExitAction());
    }
    ...
}
```

8.3 Menüklassen

Beispiel: MainFrame

MainFrameMenuBar

```
private class ExitAction implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        System.exit(0);
    }
}
```