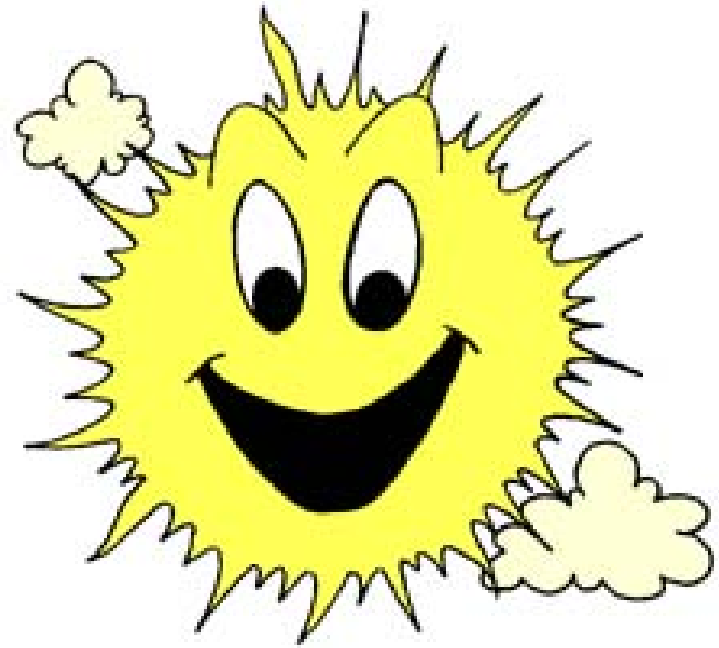
Wiederholung 4. Vorlesung



4.2 Ereignisbehandlung in Java

- eine Komponente kann ein **Ereignis** erzeugen
- ein Objekt muss **auf Ereignisse lauschen** (**Listener**)
- der aktive Komponenten (Button, Menu, ...) muss mitgeteilt werden, **wer auf Ereignisse** lauscht.
- Dem Listener muss mitgeteilt werden, **was zu tun ist, wenn das Ereignis eintritt (Aktion)**.

4.2.2 Listener-Klassen in Java

- Listener-Klassen in Java sind in der Regel **Interfaces** oder **abstrakte Klassen**
- sie geben die **Grundstruktur** vor und vereinbaren **Schnittstellen** (Methoden)
- **was** aber bei einem Ereignis zu tun ist, muss der Programmierer festlegen.
- daher **muss** die **Aktion**, die auf ein Ereignis folgen soll **implementiert werden**
- dies erfolgt durch die
  - **Implementierung** des **Interfaces** oder
  - **Ableitung** und Implementierung der **abstrakten Klasse**

4.2.2 Listener Beispiel (Erweiterung Hello2)

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

public class LDialog extends JFrame
{
   Container c;
   JPanel pCenter = new JPanel();
   JPanel pBottom = new JPanel();
   JLabel beschrift = null;
   JButton bRot  = new JButton( "Rot" );
   JButton bBlau = new JButton( "Blau" );
```

4.2.2 Listener Beispiel (2) Layout

```
public LDialog() {
      c = getContentPane();
      c.setLayout( new BorderLayout() );
      beschrift = new JLabel( "Text mit Farbe" );

      pCenter.add( beschrift );
      pCenter.setBackground( Color.GREEN );
      c.add( pCenter, BorderLayout.CENTER );

      pBottom.add( bRot );
      pBottom.add( bBlau );
      c.add( pBottom, BorderLayout.SOUTH );

      ButtonListener bl = new ButtonListener();
      bRot.addActionListener( bl );
      bBlau.addActionListener( bl );
}
```

4.2.2 Listener Beispiel (3) ActionListener

```
/* innere Klasse */
class ButtonListener implements ActionListener
{
   /* Action-Methode des Interface implementieren */
   public void actionPerformed( ActionEvent e )
   {
      if (e.getSource() == bRot )
         pCenter.setBackground( Color.RED );
      else if (e.getSource() == bBlau )
         pCenter.setBackground( Color.BLUE );
   }
}
```

4.2.2 Listener Beispiel (4) Listener hinzufügen

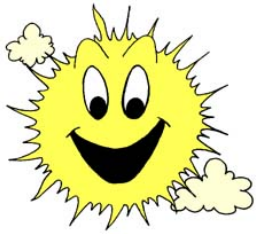
```
public LDialog() {
        c = getContentPane();
        c.setLayout( new BorderLayout() );
        beschrift = new JLabel( "Text mit Farbe" );

        pCenter.add( beschrift );
        pCenter.setBackground( Color.GREEN );
        c.add( pCenter, BorderLayout.CENTER );

        pBottom.add( bRot );
        pBottom.add( bBlau );
        c.add( pBottom, BorderLayout.SOUTH );

      ButtonListener bl = new ButtonListener();
      bRot.addActionListener( bl );
      bBlau.addActionListener( bl );
}
```

Ende der Wiederholung



4.3 Varianten der Ereignisverabeitung

Bei der Ereignisverarbeitung kann man prinzipiell vier Varianten unterscheiden:

- Listener-Klasse als innere Klasse (Bsp: LDialog)
- Listener-Klasse als anonyme Klasse (Bsp: LDialog3)
- Container-Klasse wird selbst zur Listener-Klasse (Bsp: LDialog2)
- Listener-Klasse wird als separate Klasse realisiert (Bsp: LDialog4)

4.3.1 Listener als anonyme Klasse (LDialog3)

```
public LDialog3() {
    ...

    /* anonyme Klasse */
    ActionListener bl = new ActionListener() {
        public void actionPerformed( ActionEvent e )
        {
            if (e.getSource() == bRot )
                pCenter.setBackground( Color.RED );
            else if (e.getSource() == bBlau )
                pCenter.setBackground( Color.BLUE );
        }
    };

    bRot.addActionListener( bl );
    bBlau.addActionListener( bl );
}
```

4.3.1 Container als Listener (LDialog2)

```
public class LDialog2   extends JFrame
        implements ActionListener
{
  /* Action-Methode des Interface implementieren */
  public void actionPerformed( ActionEvent e ) {
     if (e.getSource() == bRot )
        pCenter.setBackground( Color.RED );
     else if (e.getSource() == bBlau )
        pCenter.setBackground( Color.BLUE );
   }
   public LDialog2()
   {
      …
      bRot.addActionListener( this );
      bBlau.addActionListener( this );
   }
…
}
```

4.3.1 Listener als separate Klasse (LDialog4)

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

public class ButtonListener implements
   ActionListener
{
    protected LDialog4 dlg;

    public ButtonListener( LDialog4 d ) {
        dlg = d;
    }

    public void actionPerformed( ActionEvent e )
    {
        // Aufruf der eigentlichen Aktion
        dlg.buttonActionPerformed( e );
    }
}
```

4.3.1 Listener als separate Klasse (2) (LDialog4)

```
public LDialog4() // Konstruktor
{
     ...

     ButtonListener bl = new ButtonListener( this );
     bRot.addActionListener( bl );
     bBlau.addActionListener( bl );
}

public void buttonActionPerformed( ActionEvent e )
{
      if (e.getSource() == bRot )
         pCenter.setBackground( Color.RED );
      else if (e.getSource() == bBlau )
         pCenter.setBackground( Color.BLUE );
}
```

4.3.2 Verallgemeinerung ButtonListener ?

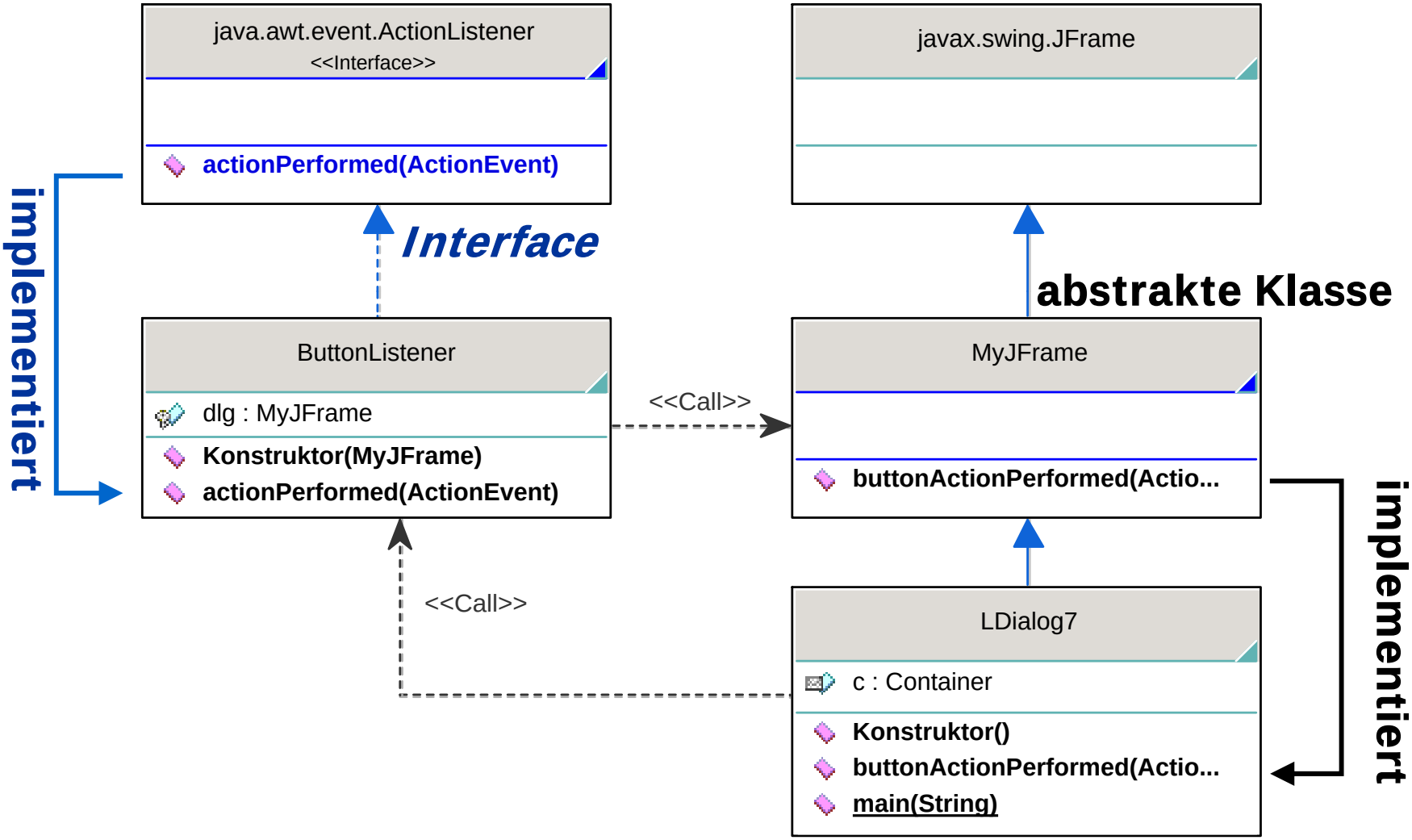
- Ist der ButtonListener aus dem Beispiel LDialog4 allgemein verwendbar?

  → Nein
- Was müsste man tun, um ihn allgemein für andere Klassen, welche von JFrame erben, verwenden zu können.

Lösung:

- Diese Dialoge sollten von einer **abstakten Klasse** erben, welche *buttonActionPerfomed(ActionEvent e)* definiert.
- Parameter im Konstruktor von ButtonListener in diesen Datentyp ändern.
- Implementierung der Lösung: **Übung zu Hause** (Lösung folgt mit LDialog7)

4.3.2 Bsp: Interface und abstrakte Klasse



4.3.3 Optimierung Listener

- mit *ActionEvent.**getSource()*** kann der Auslöser des Ereignisses bestimmt werden.
- Wir nutzen dies in den Bsp: LDialog – LDialog4
- Sollen **mehrere Schaltflächen** die **gleiche Aktion** auslösen, ist dieses Vorgehen **nicht optimal**, da alle Quellen abgefragt werden müssen.

4.3.3 Optimierung Listener (2) (LDialog5 u. 6)

Informatik

Lösung:

- Man kann Schaltflächen ein Kommando zuweisen, welches diese dann mit dem ActionEvent weitergeben:
- In der Dialogklasse:

```
JButton but = new Button( “Button Name“ );
but.setActionCommand(“tue was!“ );
```

- Mehreren Buttons oder Menu-Punkten kann gleiches Kommando zugewiesen werden
- Im ActionListener oder der entsprechenden Funktion erhält man aus dem `ActionEvent e` als `String` das Kommando mit `e.getActionCommand()` , das entsprechend ausgewertet werden kann, z.B:

```
if(e.getActionCommand().equals((“tue was!“))
    ...
```

4.3.3 Optimierung Listener (LDialog5)

```
public LDialog5() // Konstruktor
{
    // Buttons ein Komando zuweisen
    bRot.setActionCommand( "wechsle Rot!" );
    bBlau.setActionCommand( "jetzt Blau!" );

    ...
}

public void buttonActionPerformed( ActionEvent e )
{
  if (e.getActionCommand().equals("wechsle Rot!"))
      pCenter.setBackground( Color.RED );
  else if
   (e.getActionCommand().equals("jetzt Blau!“))
      pCenter.setBackground( Color.BLUE );
}
```

4.3.4 Menüs in Java (LDialog 6)

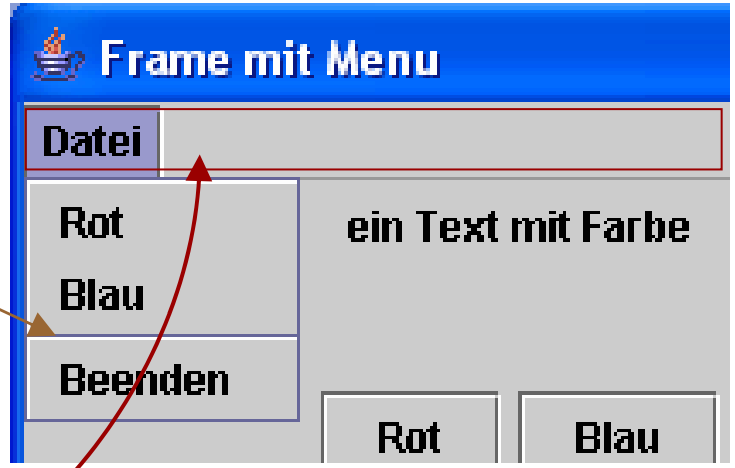
```
public class LDialog6 extends JFrame
{
   ...
   JMenuBar jMenuBar =
      new JMenuBar();
   JMenu jMenuFile =
      new JMenu( "Datei" );
   JMenuItem itemRot =
      new JMenuItem( "Rot" );
   JMenuItem itemBlau =
      new JMenuItem( "Blau" );
   JMenuItem itemQuit =
      new JMenuItem( "Beenden" );
```

4.3.4 Menüs in Java (2) (LDialog 6)

```
public LDialog6() { //Konstruktor
   ...
   // Menuleiste aufbauen
   jMenuFile.add(itemRot);
   jMenuFile.add(itemBlau);
   jMenuFile.addSeparator();
   jMenuFile.add(itemQuit);

   jMenuBar.add(jMenuFile);

   setJMenuBar(jMenuBar);
```



4.3.4 Menüs in Java (3) (LDialog 6)

```
public LDialog6() { //Konstruktor (Fortsetzung)
  ...
  // Buttons/Menus ein Komando zuweisen
   bRot.setActionCommand( "wechsle Rot!" );
   itemRot.setActionCommand( "wechsle Rot!" );
   bBlau.setActionCommand( "jetzt Blau!" );
   itemBlau.setActionCommand( "jetzt Blau!" );
   itemQuit.setActionCommand( "Fertig" );

   // ActionListener erzeugen
   ButtonListener bl = new ButtonListener();
   // Den Schaltflächen den Listener bekannt machen.
   bRot.addActionListener( bl );
   bBlau.addActionListener( bl );
   itemRot.addActionListener( bl );
   itemBlau.addActionListener( bl );
   itemQuit.addActionListener( bl );
```

4.3.4 Menüs in Java (4) (LDialog 6)

```
class ButtonListener implements ActionListener
{
   /**
    * einzige Methode von ActionListener
    * @param e ActionEvent. wird von Komponente an
    * den Listener gesendet.
    */
   public void actionPerformed( ActionEvent e ) {
     if (e.getActionCommand().equals("wechsle Rot!“))
         pCenter.setBackground( Color.RED );
     else if
       (e.getActionCommand().equals("jetzt Blau!"))
         pCenter.setBackground( Color.BLUE );
     else if (e.getActionCommand().equals("Fertig"))
         System.exit(0); // Anwendung beenden
   }
}
```