

Einführung in Java mit BlueJ

Arbeitsblatt 11.3.2

Button

Fachbegriffe:

Komponenten vom Typ Button erzeugen, Position, Breite und Höhe festlegen.

Abfragen, welcher Button geklickt wurde und entsprechend reagieren.

Konstruktor

Zur Erzeugung von Buttons gibt es zwei Konstruktoren.

Button()

Dieser Konstruktor erzeugt einen nicht beschrifteten Button, der sehr schmal aussieht.

Button(String text)

Hier wird ein beschrifteter Button erzeugt.

Wichtige Methoden

String getLabel()

Liefert den Beschriftungstext zurück.

void setLabel(String text)

Verändert den Beschriftungstext.

void setBounds(int x,int y,int breite,int hoehe)

Damit kann die Position, Breite und Höhe eines Buttons genau bestimmen. Allerdings muss dann der Befehl `setLayout(null)` in die `init()`-Methode des Applets eingebaut werden.

void setFont(Font f)

Hiermit kann eine bestimmte Schriftart gesetzt werden.

Abfragen der Buttons

Damit die Buttons abgefragt werden können, muss das Applet zunächst um einen ActionListener ergänzt werden:

```
public class ButtonTest extends Applet
    implements ActionListener
```

Die Buttons werden dann ganz normal erzeugt, allerdings muss jedem Button ein eigener ActionListener zugefügt werden:

```
b1.addActionListener(this);
b2.addActionListener(this);
```

Mit dem Schlüsselwort `this` wird der ActionListener des Buttons mit dem Applet selbst verknüpft.

Die Klasse `ButtonTest` stammt von der Klasse `Applet` ab, man sagt auch: `ButtonTest` erbt alle Attribute und Methoden von `Applet`. Eine dieser Methoden heißt `actionPerformed()`. In der Klasse `Applet` enthält diese Methode noch keinen Quelltext (schließlich konnten die Programmierer von `Applet` ja noch nicht wissen, was genau passieren soll, wenn später mal ein Benutzer auf einen Button klickt). In der Klasse `ButtonTest` muss diese Methode aber mit Quelltext gefüllt werden. Man sagt auch: Die Methode wird überschrieben. Man könnte jetzt irgendwelchen Quelltext in die Methode schreiben; er würde stur ausgeführt, wenn einer der beiden Buttons geklickt wird. Meistens will man aber jeden Button mit einer anderen Aktion verbinden. Daher kann man über `event.getSource()` abfragen, welcher Button geklickt wurde. Anschließend kann man eine bestimmte Aktion ausführen, im Quelltext von Abb. 1 wird zum Beispiel der String `label` verändert.

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

public class ButtonTest
    extends Applet
    implements ActionListener
{
    Button b1,b2;
    String label;

    public void init()
    {
        b1 = new Button("Anfang");
        b2 = new Button("Ende");

        setLayout(null);

        b1.setBounds(40,100,200,100);
        b2.setBounds(240,100,200,100);
        add(b1);
        add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
    }

    public void
    actionPerformed(ActionEvent event)
    {
        if (event.getSource() == b1)
            label = "Anfang geklickt";
        if (event.getSource() == b2)
            label = "Ende geklickt";
        repaint();
    }

    public void paint(Graphics g)
    {
        g.drawString(label,150,240);
    }
}
```

1 Quelltext für zwei abfragbare Buttons



2 Das Java-Applet