

14. Applets

Java-Beispiel:

TempApplet.java
TempApplet.html

Schwerpunkte

- Anwendungen und Applets
- Erstes Applet-Programm:
Temperaturberechnung

Zwei Arten von Java-Programmen

- **Applikationen / Anwendungen:**

Direkte Ausführung mit dem Java-Interpreter
(`java Temperature`) **bisher**

- **Applets:**

Start mit Hilfe eines Web-Browsers
(Firefox, Microsoft Internet Explorer) **neu**

Java-Applets durch Web-Browser starten?

→ Hypertext Markup Language **HTML** zur
Beschreibung von Webseiten erweitern

Im Detail:

- Kompilierte Java-Programme (Hallo.class) in Web-Seiten einbinden (analog wie Bilder, Links, Videos ...)
- Web-Browser kennt Java-Interpreter: ruft ihn auf
- Java-Applets können über das Netz geladen und direkt vom Browser ausgeführt werden
(.class-Programme
- nicht .java-Quellen)
- Applets laufen in graphischer Oberfläche der Web-Seite (z.B. Firefox-Oberfläche)

HTML-Dokumente: Bestandteile

- **Text**
(mit Hervorhebungen: Farben, Unterstrich, Dicke, Größe)

- **Verweise auf andere Dokumente:**

```
<a href="http://www.informatik.hu-berlin.de">
```

- **Bilder:**

```

```

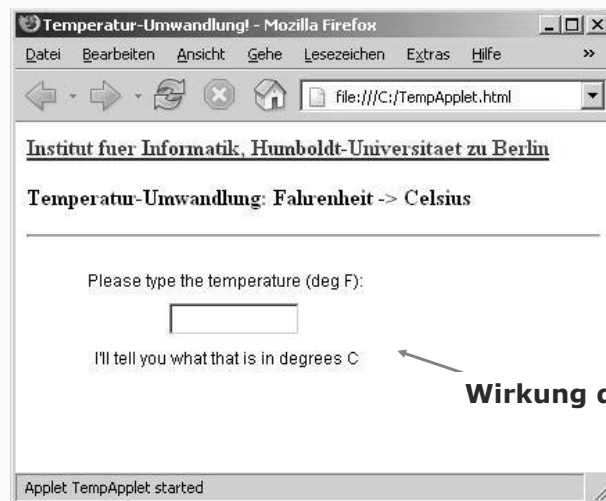
- **Programme:** laufen als Teil der Webseite

```
<applet  
CODE="TempApplet.class" WIDTH=300 HEIGHT=100>  
</applet>
```

Beispiel: TempApplet.html

```
<html>  
<head>  
<title>Temperatur-Umwandlung!</title>  
</head>  
<body>  
<P>  
<b>  
<a href=  
"http://www.informatik.hu-berlin.de">  
Institut fuer Informatik, Humboldt-Universitaet zu  
Berlin  
</a>  
</b>  
<P>  
<b>Temperatur-Umwandlung: Fahrenheit -> Celsius</b>  
<hr>  
<P>  
<applet CODE="TempApplet.class" WIDTH=300 HEIGHT=100>  
</applet>  
</body>  
</html>
```

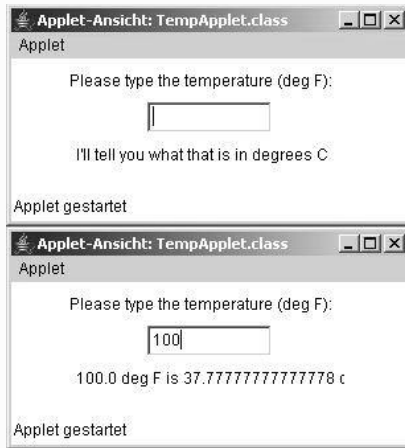
Firefox: TempApplet.html geladen



Firefox: TempApplet.html geladen

Appletviewer: aktiviert Applets direkt

> appletviewer TempApplet.html



Applet testen
- nicht gleich im
Browser das
gesamte HTML-File

Applet-Beispiel: TempApplet.java

3 Pakete

Mehrfach-
vererbung

2 Dialog-
elemente

2 Methoden

TempApplet.java

```

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

public class TempApplet extends Applet
    implements ActionListener {

    // Convert from Fahrenheit
    TextField tFahr;
    Label lCent;

    public void init() {
        // Create the TextField and the Label
        tFahr = new TextField(10);
        lCent = new Label("I'll tell you what
            that is in degrees C");

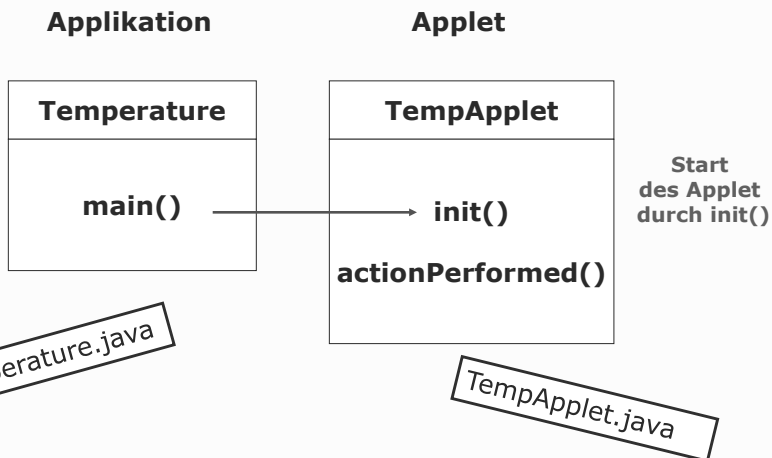
        // Lay out the three Components
        add(new Label("Please type the
            temperature (deg F): "));
        add(tFahr);
        add(lCent);

        // Register the Component Listener
        tFahr.addActionListener(this);
    }

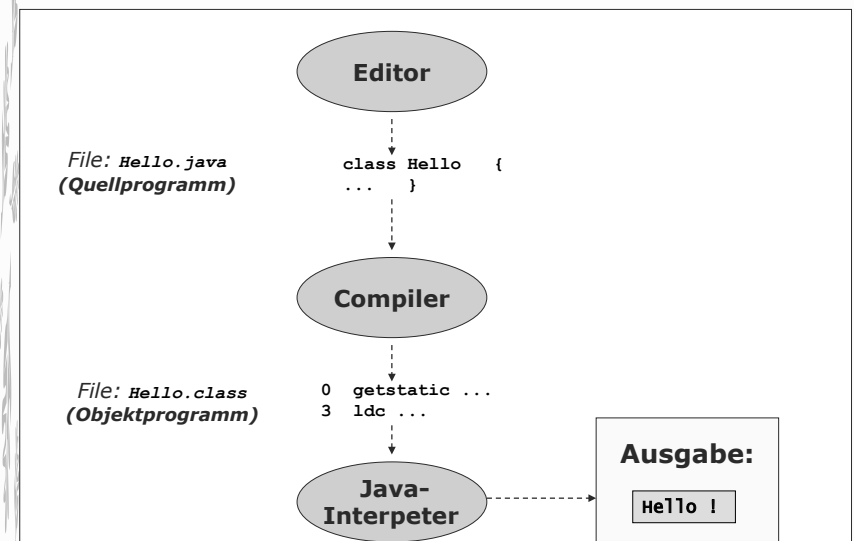
    // Respond to Action Event:
    // typing in the tFahr TextField
    public void actionPerformed(ActionEvent e) {

        double fahr = 0.0;
        cent = 0.0;
        fahr = Integer.parseInt(tFahr.getText());
        cent = 5.0 * (fahr - 32) / 9.0;
        lCent.setText(fahr + " deg F is " + cent + " deg C");
    }
    
```

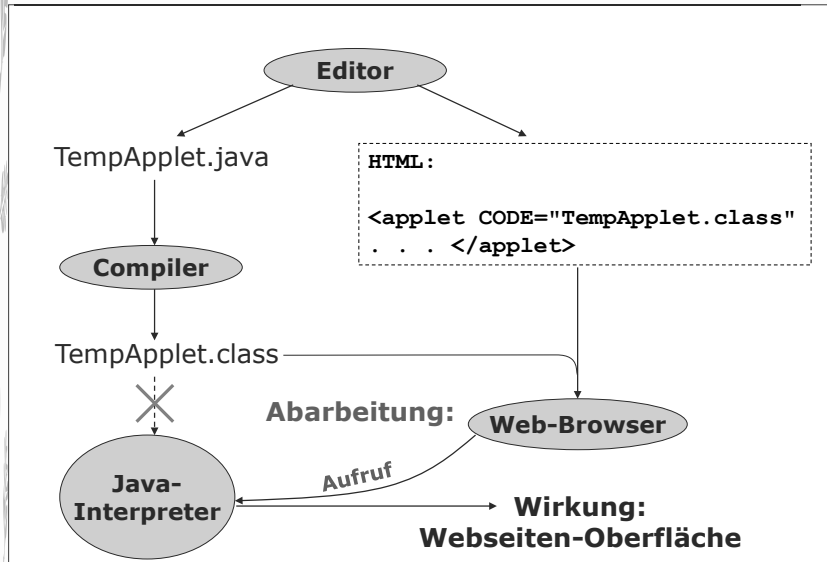
Vergleich: Applikation - Applet



Programmentwicklung: Anwendungen



Programmentwicklung: Applets



Erklärung zum Temperatur-Applet

TempApplet.java

1. Importe
2. API-Struktur: API - Pakete - Klassen
3. Klasse Applet
4. Initialisierung des Fensters
5. Reaktion auf Ereignisse
6. globale und lokale Variablen

Importe

TempApplet.java

drei Dienste benötigt:

- `import java.awt.*;` - Programmierung der graphischen Benutzeroberfläche (awt = abstract windowing toolkit)
- `import java.applet.*;` - Applet-Klasse u. a.
- `import java.awt.event.*;` - Ereignisbehandlung

```

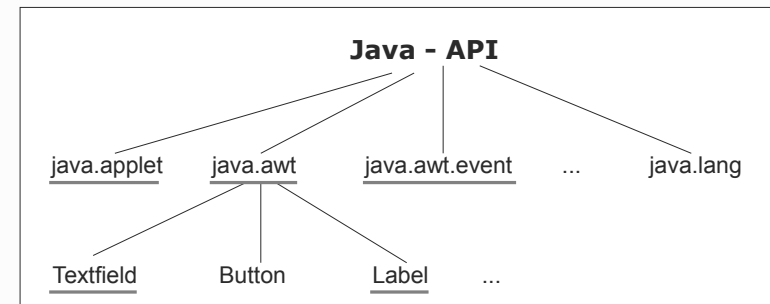
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class TempApplet extends Applet
implements ActionListener {

    // Convert from Fahrenheit to Centigrade
    TextField tFahr;
    Label lCent;

    public void init() { ... }
    public void actionPerformed (...){...}
}
    
```

Organisation des Java-API



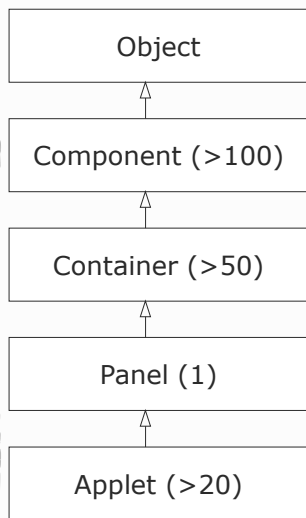
Pakete:
Sammlung von Klassen

Klassen:
Softwarekomponenten

Paket-Name (z. B. java.awt.event) spiegelt Directory-Namen wider:
→ /java/awt/event

API: Applets

(Anzahl der Methoden)



Applet:
Java-Programm,
das von einem
Web-Browser
gestartet wird

```

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

public class TempApplet extends Applet
implements ActionListener {

    // Convert from Fahrenheit to Centigrade
    TextField tFahr;
    Label lCent;

    public void init() {...}
    public void actionPerformed (...) ...
}
  
```

nicht nur
2 Methoden

API-Klasse ,Applet'

```

java.applet
Class Applet
java.lang.Object
├── java.awt.Component
│   ├── java.awt.Container
│   │   ├── java.awt.Panel
│   │   └── java.applet.Applet
└── public class Applet
    extends Panel

An applet is a small program that is intended not to be run on its own, but rather
to be embedded inside another application.

The Applet class must be the superclass of any applet that is to be embedded
in a Web page or viewed by the Java Applet viewer.

Constructor Summary
Applet()
Creates a new Applet object.

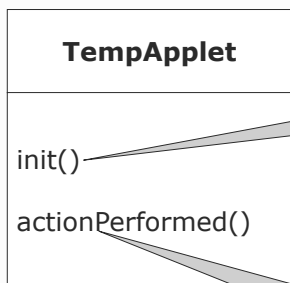
Method Summary
void destroy()
Called by the browser or applet viewer to inform this applet that it has been
loaded into the system.
AppletContext getAppletContext()
Determines this applet's context, which allows the applet to query and affect
the environment in which it runs.
AudioClip getAudioClip(URL url, String name)
Returns the AudioClip object specified by the URL and name arguments.
URL getDocumentBase()
Gets the URL of the document in which this applet is embedded.
String getParameter(String name)
Returns the value of the named parameter in the HTML tag.
void init()
Called by the browser or applet viewer to inform this applet that it has been
loaded into the system.
boolean isActive()
Determines if this applet is active.
void play(URL url, String name)
Plays the audio clip given the URL and name arguments.
void resize(int width, int height)
Requests that this applet be resized to the specified width and height.
void start()
Called by the browser or applet viewer to inform this applet that it should
begin execution.
void stop()
Called by the browser or applet viewer to inform this applet that it should
stop its execution.
  
```

An applet is a small program that is intended not to be run on its own, but rather to be embedded inside another application.

The Applet class must be the superclass of any applet that is to be embedded in a Web page or viewed by the Java Applet viewer.

void init()
Called by the browser or applet viewer to inform this applet that it has been loaded into the system.

Klasse Applet: Methodenaufrufe



Start des Programms:
analog: main() bei Applikationen

Reaktion auf Enter-Taste (Ereignis):
- Temperatur-Feld eingelesen
- umgerechneter Wert ausgegeben

Initialisierung des Fensters

Applet-Fenster auf Bildschirm ausgeben (Anfangszustand)

```

public void init ( ) {

    tFahr = new TextField (10);
    lCent = new Label (" I'll tell...");

    add(new Label ("Please ..."));
    add (tFahr);
    add (lCent);

    tFahr.addActionListener (this);
}
  
```

1. Erzeugt (noch unsichtbar):
- aktives Textfeld (Eingabe)
- Label-Textfeld (Ausgabertext)

Woher kommt add()? Oberklasse „Container“ (damit Instanzmethode von TempApplet)

2. Komponenten im Applet-Fenster platziert

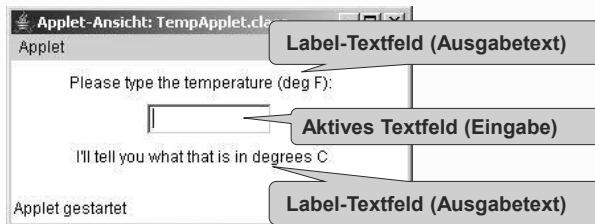
3. Textfeld tFahr soll 'beobachtet' werden: bei Enter-Taste soll actionPerformed(...) gerufen werden (Ereignisbehandlung)

Bestandteile einer graphischen Oberfläche: > 50 Klassen angeboten z. B.: TextField, Button, Label, Checkbox, Scrollbar ...

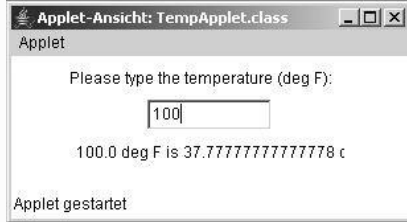
Wirkung von TempApplet

> appletviewer TempApplet.html

Nach init():



Nach actionPerformed(e):



Reaktion auf Ereignisse (Event) - Enter-Taste -

```
public void actionPerformed (...) {
    double fahr = 0.0;
    double cent = 0.0;

    fahr = Integer.parseInt(tFahr.getText());

    cent = 5.0 * (fahr - 32) / 9.0;

    lCent.setText (
        fahr + " deg F is " + cent + " deg C ");
}
```

Einlesen aus Textfeld tFahr und Transformation nach Integer

Ausgabe des Resultats in das Labelfeld lCent

GUI: als ADT/Objekte aufgefasst

zusammengesetzte Zeichenkette

Ereignisbehandlung: Woher kommt actionPerformed()?

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

Ereignisbehandlung

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

public class TempApplet extends Applet
    implements ActionListener {

    // Convert from Fahrenheit to Centigrade
    TextField tFahr;
    Label lCent;

    public void init() { ... }
    public void actionPerformed (...) { ... }
}
```

API-Interface „ActionListener“

java.awt.event
Interface ActionListener

All Superinterfaces:
EventListener
All Known Subinterfaces:
Action

All Known Implementing Classes:
AbstractAction, AWTEvent, BaseFileChooserUI, BaseFileChooserUI.Update, BaseInternalFrame, BaseScrollPaneUI, BaseScrollPaneUI.Keyboard, BaseScrollPaneUI.Mouse, DefaultCellEditor, EditorPane, DefaultEditorKit, DefaultEditorKit.InsertTab, DefaultFileChooserUI, DefaultEditorKit, StyledEditorKit, StyledEditorKit.Insertion

public interface ActionListener extends EventListener

The listener interface for receiving action events. The listener's actionPerformed method is invoked when an action event occurs.

Since: 1.1
See Also: ActionListener, Tutorial: Java 1.1 Event Model

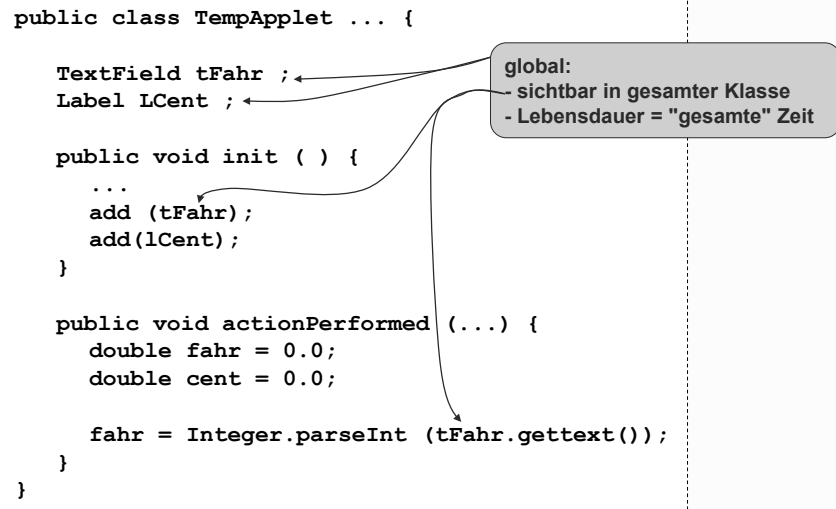
Method Summary

void actionPerformed(ActionEvent e)
Invoked when an action occurs.

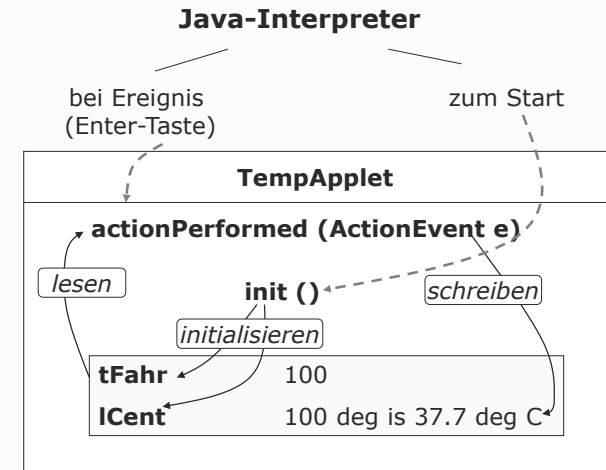
Method Detail

The listener interface for receiving action events. The class that is interested in processing an action event implements this interface, and the object created with that class is registered with a component, using the component's addActionListener method. When the action event occurs, that object's actionPerformed method is invoked.

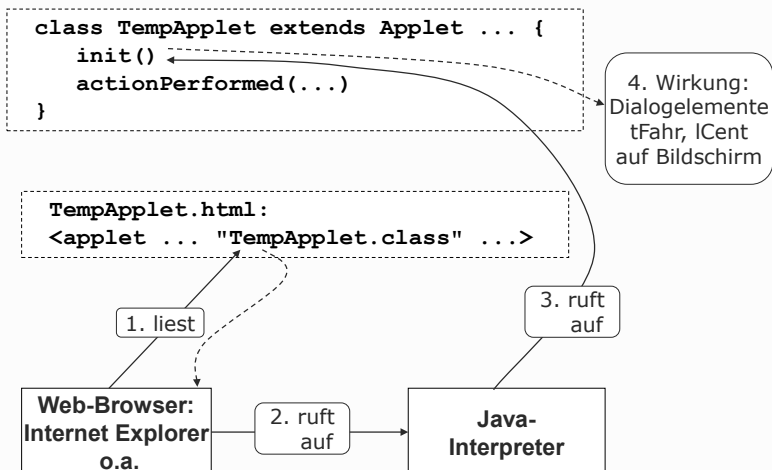
Globale Variablen der Klasse



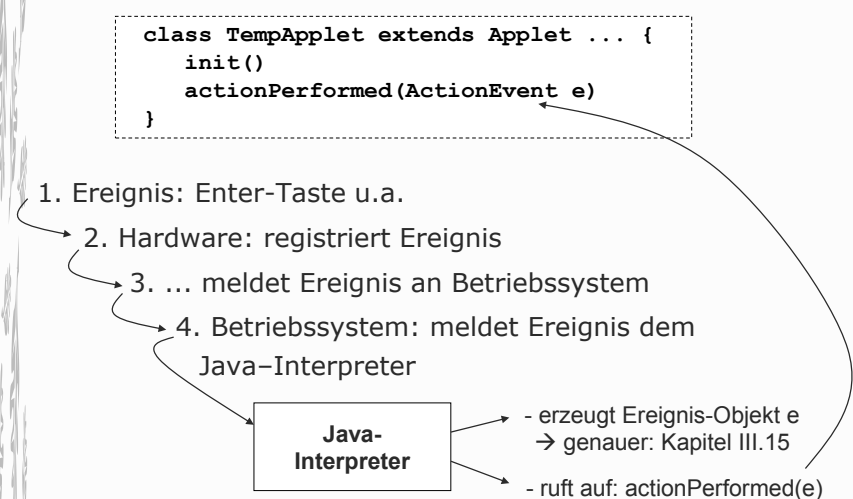
Objekt der Klasse TempApplet: Aufruf der Methoden



Applet-Abarbeitung: Start



Applet-Abarbeitung: Ereignis



Wie wichtig sind Applets?

Bell, Parr: Java for Students, Prentice Hall:

- ▶ "Because we see the Internet as being tremendously important, we focus on applets here. In addition, applets are simple to construct for the beginners. How to write applications, we explain in one of the last chapters of this book" (3rd edition, 2002)
- ▶ "In this book we concentrate on applications, because we believe that this is the main way in which Java is being used. We explain how to run applets in the appendix" (4th edition, 2005)