

Einstieg in die Informatik mit Java

Applets

Gerd Bohlender

Institut für Angewandte und Numerische Mathematik

- 1 Einführung
- 2 Sicherheits-Maßnahmen
- 3 Ereignisgesteuerte Programmierung
- 4 Ereignisse

- 1 Einführung
- 2 Sicherheits-Maßnahmen
- 3 Ereignisgesteuerte Programmierung
- 4 Ereignisse

Applets sind spezielle Javaprogramme mit graphischer Ein- und Ausgabe. Sie werden wie Texte oder Graphiken meist über das Internet verschickt und innerhalb eines Webbrowsers (z.B. Firefox, Opera oder Internet Explorer) lokal ausgeführt.

Es handelt sich also, wie der Name Applet schon vermuten lässt, nicht um komplette Anwendungsprogramme, sondern um Programmstücke, die für eine bestimmte Teilaufgabe zuständig sind.

- 1 Einführung
- 2 Sicherheits-Maßnahmen**
- 3 Ereignisgesteuerte Programmierung
- 4 Ereignisse

- Es erfolgt kein Zugriff auf lokale Dateien, außer wenn ein Verzeichnis ausdrücklich freigegeben ist.
- Daten werden nur mit dem Server ausgetauscht, von dem das Applet geladen wurde. Informationen an Dritte werden nicht versendet.
- Applets können keine Programme usw. starten.

Achtung

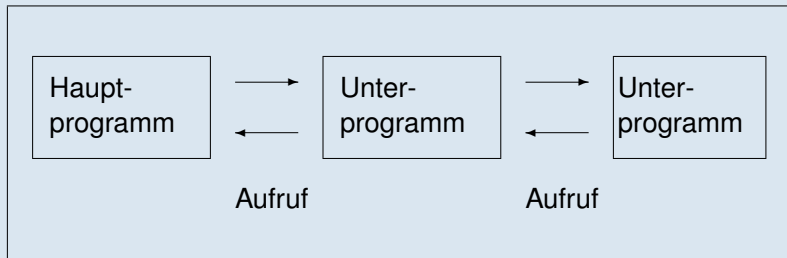
Trotzdem gibt es Sicherheitslücken! Applets können nach Beenden des Browsers weiterlaufen, Internetprotokolle können ausgespäht werden usw.

- 1 Einführung
- 2 Sicherheits-Maßnahmen
- 3 Ereignisgesteuerte Programmierung**
- 4 Ereignisse

Ereignisgesteuerte Programmierung

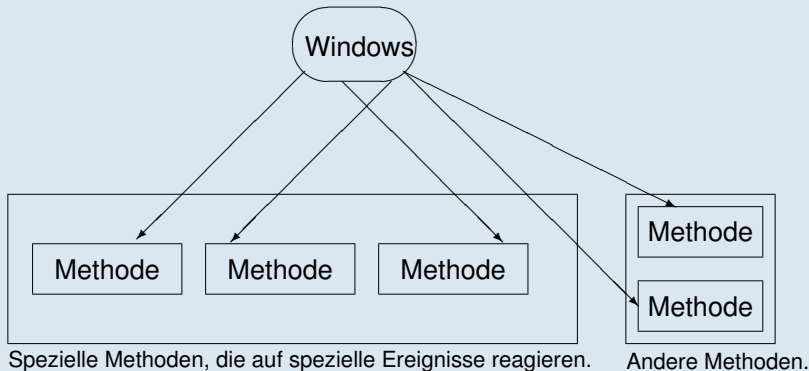
Applets besitzen kein Hauptprogramm `main`. Existiert es trotzdem, so wird es ignoriert. Methoden werden durch Ereignisse ausgelöst, das Hauptprogramm ist nun Windows bzw. der Web-Browser.

Klassische prozedurale Programmierung



Ereignisgesteuerte Programmierung

Ereignisgesteuerte Programmierung unter (X-/MS-) Windows



Achtung

Methoden eines Applets rufen sich i.allg. nicht explizit gegenseitig auf!

- 1 Einführung
- 2 Sicherheits-Maßnahmen
- 3 Ereignisgesteuerte Programmierung
- 4 Ereignisse**

Ereignisse

Die Klasse `Applet` enthält eine Reihe von Methoden, die bei bestimmten Ereignissen automatisch aufgerufen werden.

Der Lebenszyklus eines Applets sieht folgendermaßen aus.

Lebenszyklus eines Applets



Ereignisse und aufgerufene Methoden

Ereignisse	Aufgerufene Methode
(Neu-) Laden des Applet	<code>public void init() {...}</code>
(Neu-) Starten des Applets (nach Stop) Zurückwechseln zur Webseite (nach Wechsel auf andere Seite)	<code>public void start() {...}</code>
Verlassen der Webseite, Verkleinerung des Browsers zum Icon	<code>public void stop() {...}</code>
Beenden des Browsers, Entfernung des Applets wegen Speichermangels	<code>public void destroy() {...}</code>
Neuzeichnen einer Komponente	<code>public void paint(Graphics g) {...}</code>

Beispiele

Ereignisse, die einen Aufruf der Methode `paint` nach sich ziehen:

- Initialisierung des Applets,
- erneute Anzeige des Fensters,
- expliziter Aufruf im Quelltext.

Diese Methoden sind bereits in der Klasse `Applet` definiert, enthalten dort aber noch keinen Code.

Damit diese die gewünschte Funktionalität besitzen, müssen die Methoden in einer selbstdefinierten Subklasse überschrieben werden.

Üblicherweise werden in der Methode `init` nur einige Initialisierungen vorgenommen und in der Methode `paint` die eigentliche Graphik-Ausgabe implementiert.

Einige wenige Methoden zur graphischen Ausgabe in einem Applet sind in der Tabelle auf der nächsten Folie aufgelistet, eine ausführliche Beschreibung befindet sich in der Dokumentation zum JDK.

Achtung

Alle Methoden werden als Komponenten des Parameters `g` aufgerufen!

JDK Dokumentation

<http://java.sun.com/javase/6/docs/>

Methoden zur Ausgabe in einem Applet

Methoden	Wirkung
<code>drawString(s,x,y)</code>	gibt Zeichenkette <code>s</code> aus, Bezugspunkt <code>(x,y)</code> ist links unten.
<code>drawRect(x,y,b,h)</code>	gibt Rechteck mit Breite <code>b+1</code> und Höhe <code>h+1</code> aus, Bezugspunkt <code>(x,y)</code> ist links oben.
<code>fillRect(x,y,b,h)</code>	wie <code>drawRect</code> , Rechteck wird zusätzlich mit Farbe gefüllt.
<code>drawLine(x1,y1,x2,y2)</code>	zeichnet eine Linie von <code>(x1,y1)</code> nach <code>(x2,y2)</code>
<code>setColor(c)</code>	wählt Farbe <code>c</code> für nächste Ausgaben aus.

Achtung

`c` ist vom Typ `Color`. Instanzen der Klasse `Color` können beim Aufruf des Konstruktors entweder durch Angabe von `int`-Werten im Bereich 0 bis einschließlich 255 oder durch Angabe von drei `float`-Werten im Bereich von 0 bis einschließlich 1 gebildet werden!

Desweiteren existieren die gebräuchlichsten Farben als Konstanten!

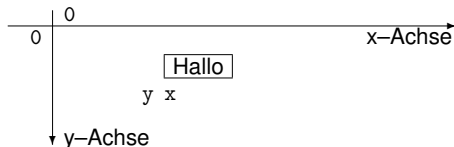
Beispiel

```
Color c1 = new Color (100, 20, 210);  
Color c2 = new Color (0.2f, 0.4f, 0.3f);  
Color c3 = Color.green;
```

In Java laufen die Bildschirmkoordinaten in Pixeln von links nach rechts bzw. von oben nach unten. Die Ausgabe:

```
g.drawString("Hallo", x, y);
```

erscheint damit wie folgt auf dem Bildschirm:



Achtung

Bei Wahl von $y=0$ bleibt diese Ausgabe unsichtbar, da der Text oberhalb des oberen Bildschirmrandes ausgegeben wird!

Beispiel: Applet und zugehörige html-Seite

```
import java.awt.Graphics;  
public class HalloWeltApp extends java.applet.Applet {  
    public void paint(Graphics g) {  
        g.drawString("Hallo Welt!", 40, 20);  
    }  
}
```

```
<HTML>  
<Applet Code="HalloWeltApp.class" Width=210 Height=50>  
</Applet>  
</HTML>
```

Achtung

HTML unterscheidet nicht zwischen Groß- und Kleinschreibung von Schlüsselwörtern! Dies gilt jedoch nicht für den Namen des Applets!

Der Aufruf der `html`-Seite erfolgt mit dem Programm `appletviewer` oder einem Webbrowser.

Das Paket `awt` bietet zusätzlich:

- Windows,
- Menus,
- Controls (z .B. Buttons, Scrollbars, Textfields usw.).

Applets werden i.allg. durch Ereignisse (Mausklicks, Mausbewegungen, Texteingaben usw.) gesteuert.