

Webbasierte Anwendungen SS 2015

Clientseitige Implementierungstechnologien: Java Applets

Dozentin: Grit Behrens
<mailto:grit.behrens@fh-bielefeld.de>

Lehrinhaltsübersicht der Vorlesungen zu WBA

1. Einführung in WBA
2. Wiederholung: Grundlagen des WWW, HTML und HTTP
3. **Clientseitige Implementierungstechnologien:** Javascript, DOM, Ajax, **Java-Applet**
4. Serverseitige Implementierungstechnologien: JSP, Java-Servlet
5. Anbindung von Datenbanken
6. WEB-Frameworks: JSF, Struts

Clientseitige Implementierungstechnologien (IV): Java Applet

- 1. Einführung**
- 2. Vergleich zu JavaScript und Java**
- 3. Laden eines Applets**
- 4. Lebenszyklus**
- 5. Einlesen von Parametern**
- 6. Abschlussbemerkungen zu Problem Client-Technologien**
- 7. Praktische Tips zur Netbeans IDE**
- 8. Literatur**
- 9. Ausblick**

Clientseitige Implementierungstechnologien (IV): Java Applet

- 1. Einführung**
2. Vergleich zu JavaScript und Java
3. Laden eines Applets
4. Lebenszyklus
5. Einlesen von Parametern
6. Abschlussbemerkungen zu Problem Client-Technologien
7. Praktische Tips zur Netbeans IDE
8. Literatur
9. Ausblick

Einführung in Java Applet

Java Applets aus der Sicht der Java-Programmierer:

* Quelle: Sven-Eric Panitz „Java will nur spielen“, Kap. 6 Spiele als Applet

90iger Jahre:

- „**Türöffner**“ der Programmiersprache Java
- **JVM** (Java Virtuelle Maschine) im **Mosaic-Browser** integriert
- erster **auf Webseiten** ausführbarer Code für **bewegte Bilder und Lauftexte**
- Java hatte lange Jahre mit dem **Vorurteil** zu kämpfen, es sei nur für kleine Browserprogramme gut und nicht ernsthaft einzusetzen.

Um 2000:

- Wurde in immer **größeren graphischen Browserapplikationen** eingesetzt, insbesondere erste Kartendarstellungen z.B. Map24

Ab 2004: Standardisierung der Browser verbessert sich DOM/JavaScript/Ajax gewinnt an Bedeutung und löst zum Teil Appletanwendungen ab (z.B. MAP24).

Heute:

- Applets im Desktopbrowser **verlieren immer mehr an Bedeutung**, die **Ladezeiten** sind oft zu lang; **Berechtigungen zum Download** der *.class – Dateien fehlen
- Microsoft –Produktpolitik: **Keine Java Virtual Maschine bei Erstinstallation**
- **Java-Liebhaber** haben aber die Möglichkeit, **Scriptsprachen zu umgehen**
- Verwendung oft in **Intranetanwendungen** oder kleinen redundanten Visualisierungen
- **ComeBack auf Java Embedded Plattformen**, z.B. RaspberryPi wird standardmäßig mit Java und Browser geliefert, Steuerung über JavaApplets möglich...
 - <http://www.forum-raspberrypi.de/Thread-tutorial-java-installation-von-java>
 - <https://www.chrisnewland.com/raspberry-pi-java-applets-iceweasel-openjdk-icedtea-plugin-hardfloat-295>

Einführung in Java Applet

Java Applets aus der Sicht des Web-Engineering*:

- Java Rich Client als eine Technologieauswahl für Webpräsentation mit clientseitiger Interaktion neben:
 - HTML-Form (jede Ereignisbearbeitung mit Serververbindung)
 - JavaScript
 - Ajax

Aktionselemente	Ereignis-Bearbeitung
HTML-Formular	Server-Request-Response (nach submit)
JavaScript	Client-Reaktion (ohne submit)
Java Rich Client (Applet + Java-Klassen)	Client-Reaktion (ohne submit)
Ajax	Asynchrone Interaktion

JavaScript, Java Rich Client und Ajax ermöglichen clientseitige Verarbeitung von Ereignissen. Die Zahl der Serververbindungen wird reduziert.

* Quelle: Günter Bauer „Architekturen für Web-Anwendungen“

Einführung in Java Applet

Eigenschaften von Java Applet I:

- Java-Klassen
- clientseitige Komponenten
- werden über das Netz geladen und auf dem Client-Rechner ausgeführt (mobiler Code)
- erfordern keine lokalen Installationen auf Clientseite
- einfacher Update möglich durch Bereitstellung neuer Version auf dem Server
- hohe Interaktivität durch Nutzung von ausgereiften Bibliotheken mit GUI-Komponenten (awt, swing)
- hochwertige Präsentation durch Verwendung von 2D-Graphik-Bibliothek
- Netzwerk –Kommunikation mit serverseitigen Applikationen u.a. über HTTP (auch Socketverbindungen oder Nutzung von RMI-Bibliotheken)

Einführung in Java Applet

Eigenschaften von Java Applet II:

- **Ausführung in *sicherer Browserumgebung*** (ohne besondere Zusatzprivilegien):
 - Kein Zugriff auf das Dateisystem des Clientrechners
 - keine Netzwerkverbindung zu anderem als Serverrechner des Applets selbst
 - Keine Nutzung von Bibliotheken anderer Programmiersprachen
 - Ausführen von Programmen auf Zielrechner (Client) nicht möglich
- **Deklaration ist in HTML-Code eingebettet : auf Platzierung beschränkt**
- **Als externe Komponenten Bestandteil von zusammengesetzten Dokumenten**
- **Probleme:**
 - **Ladezeiten** für class-Code des eigentlichen Applets sowie alle selbst implementierten Klassen, die zusätzlich genutzt werden
 - unterschiedliche Browserumgebungen
 - Firewalls und Proxies

Einführung in Java Applet

Kurzcharakteristik von Java Applets:

1. Java-Klassen
2. In HTML-Dokumente eingebettet
3. Sind GUI-Elemente
4. Sind interaktiv
5. Sind Komponenten
6. Werden in sicherer Browserumgebung ausgeführt

Anwendungen (viel clientseitige Interaktion ohne Serverkommunikation):

- interaktive Kartendarstellungen im WEB
- E-Learning Module mit graphischer Unterstützung
 - Rechner
 - Simulationen
- Ansteuerung von Geräten und Maschinen (Download des Codes meist aus Intranet auf JavaEmbedded Systems)

Clientseitige Implementierungstechnologien (IV): Java Applet

1. Einführung
- 2. Vergleich zw. JavaScript und Java**
3. Laden eines Applets
4. Lebenszyklus
5. Einlesen von Parametern
6. Abschlussbemerkungen zu Problem Client-Technologien
7. Praktische Tips zur Netbeans IDE
8. Literatur
9. Ausblick

Vergleich zw. Java Script und Java Applet

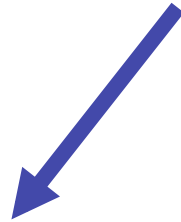


- eng mit HTML verknüpft
- Ausführen auf dem Client
- Scriptsprache
- Kommunikation mit serverseitigen Applikationen möglich (Ajax)



- externe Programmkomponenten
- unabhängig von HTML-Dokumenten (außer Position und Größe)
- separates Laden der Komponenten
- Ausführen auf dem Client
- höhere, objektorientierte Programmiersprache
- Kommunikation mit serverseitigen Applikationen möglich

Vergleich zw. Java Applik. und Java Applet



- Start durch Aufruf der main-Methode
- Ausführen durch Java-Interpreter (Aufruf aus Kommandozeile z.B. *java HelloWorld*)
- *start*-Klasse und beliebig weitere *.class* Dateien



- wird innerhalb der Browserapplikation ausgeführt
- Java-Interpreter des Browsers
- *start*-Klasse und beliebig weitere *.class* Dateien

Clientseitige Implementierungstechnologien (IV): Java Applet

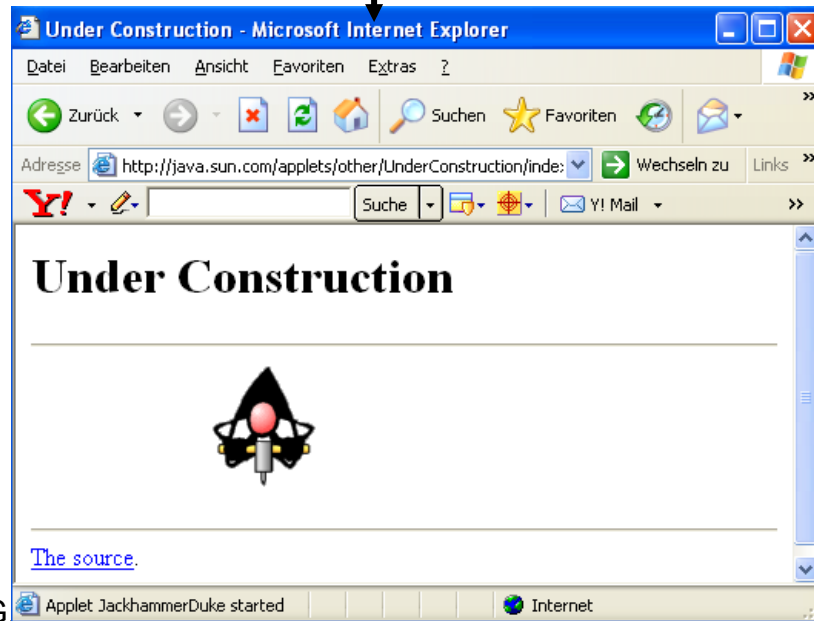
1. Einführung
2. Vergleich zu JavaScript und Java
- 3. Laden eines Applets**
4. Lebenszyklus
5. Einlesen von Parametern
6. Abschlussbemerkungen zu Problem Client-Technologien
7. Praktische Tips zur Netbeans IDE
8. Literatur
9. Ausblick

Laden eines Java Applets

http://someLocation/file.html

```
<html>
...
<object classid = java:MeinApplet.class...>
</object>
```

MeinApplet.class



1. Browser lädt URL

2. Browser lädt HTML – Datei

3. Browser lädt die Applet – Klasse
(.class-File)

4. Browser führt das Applet aus

Beispielprogramm Java-Applet

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>HTML Page for Learning Java Applets</title>
  </head>
  <body>
    <object classid="java:HelloWorldApplet.class"
      codetype=application/x-java-applet" width=300 height=50>
    </object>
  </body>
</html>
```

HTML-Dokument

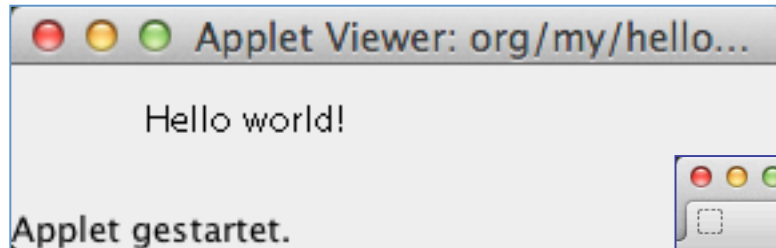
```
import java.awt.Graphics;

public class HelloWorldApplet extends java.applet.Applet {

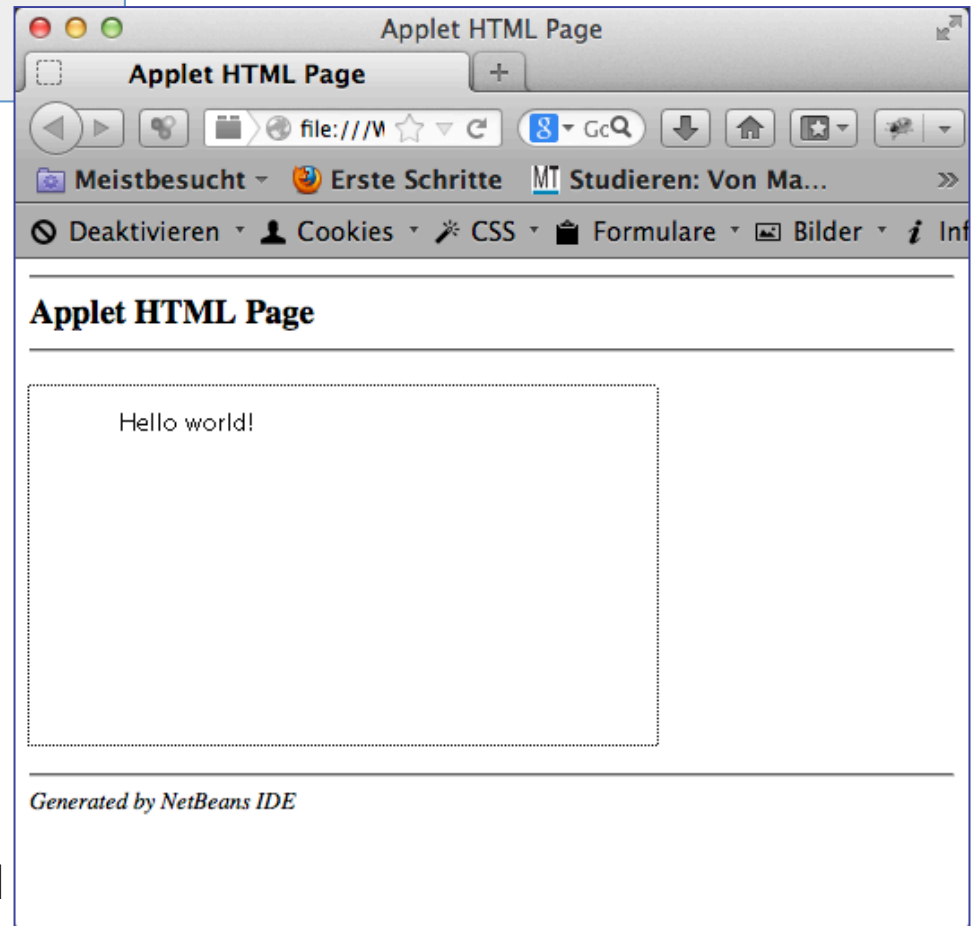
    public void paint(Graphics g) {
        g.drawString("Hello world!", 50, 25);
    }
}
```

Applet-File: HelloWorldApplet.java

Beispielprogramm Java-Applet



Appletviewer von Netbeans (8.0)



Mozilla FireFox V 25.01

Einbettung von Java-Applet in HTML alte Variante

```
<applet code="HelloWorldApplet" codebase="./build/classes"  
width=300 height=50>  
</applet>
```

Attribut	Beschreibung
<i>codebase</i>	URL des Applet-Klassenverzeichnisses
<i>width, height</i>	Breite und Höhe des Applet Bereiches
<i>code</i>	Instanzname des Applets

Aktuell: <applet> - Tag depreciated (dt. entwertet, veraltet)

Depreciated (seit HTML4.01, wird von Browsern aber meistens noch unterstützt):

```
<applet code="HelloWorldApplet" codebase="./build/classes"  
width=300 height=50>  
</applet>
```

neu mit <object> - Tag:

```
<OBJECT codetype="application/java"  
classid="java:HelloWorldApplet.class" codebase="./build/  
classes width="300" height="50"> </OBJECT>
```

Alle Varianten:

http://wiki.byte-welt.net/wiki/Einbinden_von_Applets_in_HTML-Dateien

Generische Inclusion mit dem <object> - Element

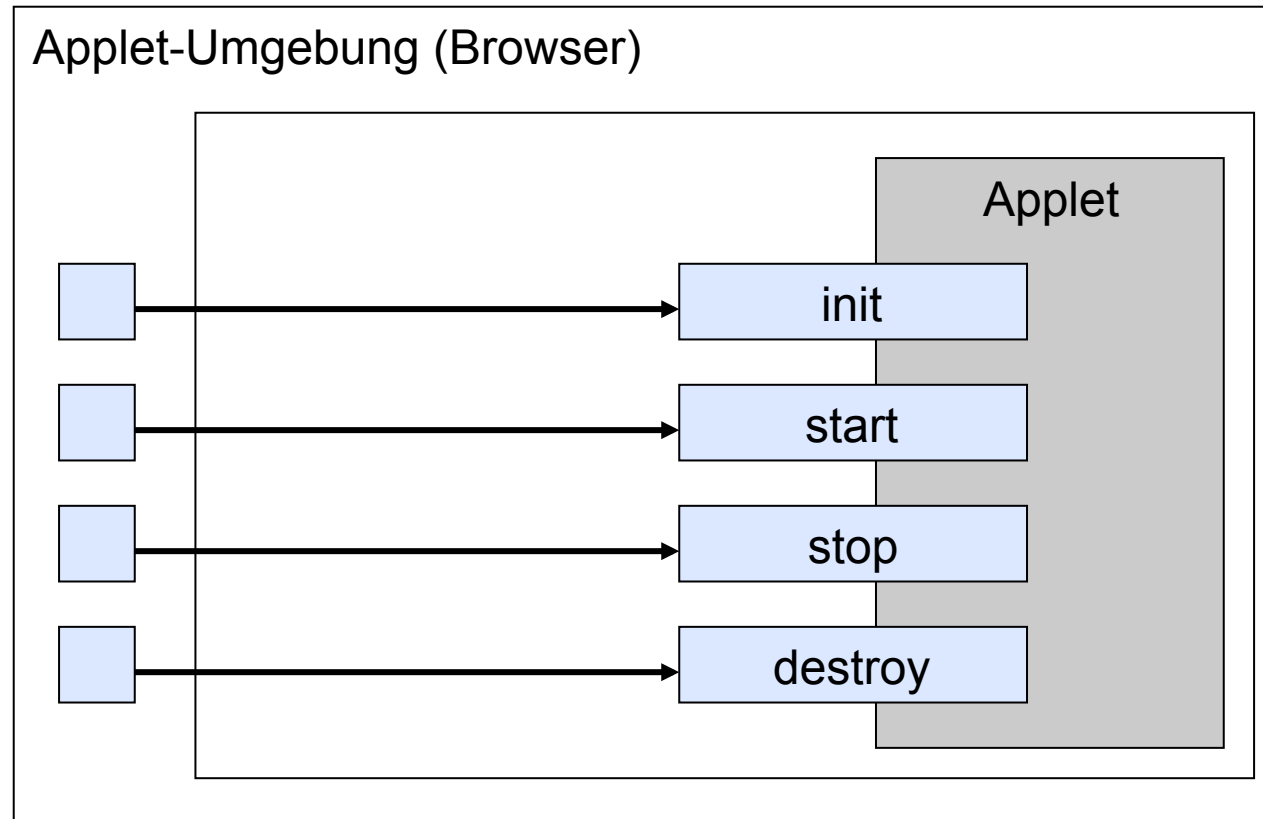
Einige Attribute:

• <i>classid</i>	Identifikator für Implementierung
• <i>codebase</i>	URI für archive, classid, data
• <i>data</i>	URI – referenziert Objektdaten
• <i>type</i>	content type für Daten
• <i>code type</i>	content type für Code
• <i>height</i>	Höhe
• <i>width</i>	Breite

Clientseitige Implementierungstechnologien (IV): Java Applet

1. Einführung
2. Vergleich zu JavaScript und Java
3. Laden eines Applets
- 4. Lebenszyklus**
5. Einlesen von Parametern
6. Abschlussbemerkungen zu Problem Client-Technologien
7. Praktische Tips zur Netbeans IDE
8. Literatur
9. Ausblick

Lebenszyklus von Java Applet(I)



Appletumgebung steuert Lebenszyklus des Applets durch Aufruf von 4 Methoden.

Lebenszyklus von Java Applet(II)

Methoden *init()*, *start()*, *stop()* und *destroy()* werden von Klasse *java.applet.Applet* geerbt und können überladen werden.

Methode	Beschreibung
<i>init()</i>	Initialisierung des Applets (enthält Code, welcher sonst im Konstruktor geschrieben wird, z.B. Laden von Grafiken, Initialisierung von Attributen, GUI-Objekte)
<i>start()</i>	Start der Ausführung des Applets (Kern des Applets, Aufruf auch bei Reload der HTML-Seite mit Applet)
<i>stop()</i>	Anhalten der Applet-Ausführung (Aufruf z.B. wenn vom Benutzer ein anderes Dokument in den Browser geladen wird, Anhalten einiger Threads)
<i>destroy()</i>	Löschen des Applets (Aufruf meist vor Entfernen des Applets aus dem Client – Cache, Endgültige Freigabe der Ressourcen)

Beispielprogramm Java-Applet : LifeCycle

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>HTML Page for Learning Java Applets</title>
    </head>
    <body>
      <applet code=„LifeCycle“ codebase=„./build/classes“
        width=300 height=20>
      </applet>
    </body>
  </html>
```

HTML-Dokument

Beispielprogramm Java-Applet : LifeCycle

```
import java.awt.Graphics;

public class LifeCycle extends java.applet.Applet {
    StringBuffer buffer;

    public void init() {
        buffer = new StringBuffer();
        addItem("initializing... ");
    }

    public void start() {
        addItem("starting... ");
    }

    public void stop() {
        addItem("stopping... ");
    }

    public void destroy() {
        addItem("preparing for unloading... ");
    }

    void addItem(String newWord) {
        System.out.println(newWord);
        buffer.append(newWord);
        repaint();
    }

    ... weiter
```

```
... weiter
public void paint(Graphics g) {
    //Draw a Rectangle around the applet's display area.
    g.drawRect(0, 0, size().width - 1, size().height - 1);

    //Draw the current string inside the rectangle.
    g.drawString(buffer.toString(), 5, 15);
}
}
```



LifeCycle.java

Interaktivität von Java - Applet

zeichnet sich aus durch :

- Erben der Lebenszyklus- Methoden von der Klasse Applet
- Erben der Methoden der Ereignisbehandlung aus den GUI-Klassen (z.B. java.awt.*):
 - Hilfsmethoden
 - action events
 - Event Handler

Clientseitige Implementierungstechnologien (IV): Java Applet

1. Einführung
2. Vergleich zu JavaScript und Java
3. Laden eines Applets
4. Lebenszyklus
- 5. Einlesen von Parametern**
6. Abschlussbemerkungen zu Problem Client-Technologien
7. Praktische Tips zur Netbeans IDE
8. Literatur
9. Ausblick

Einlesen von Parametern

- Übergabe von Parametern aus der HTML-Datei mit dem Parameterelement: **<param>**
- Appletmethode **getParameter()** liest die Argumente **immer als String** ein
- andere Argumenttypen müssen konvertiert werden :
 - z.B. für Integer:
int alter = Integer.parseInt(getParameter("alter"));
- Parameternamen sind nicht case-sensitiv (HTML-Eigenschaft)
- **Werte der Parameter sind case-sensitiv**

Einlesen von Parametern : Beispiel

```
import java.awt.Graphics;

public class Parameter extends java.applet.Applet {
    private String toDisplayVorname;
    private String toDisplayNachname;
    private String toDisplayAlter;
    private int alter;

    public void init() {
        String pVorname;
        String pNachname;
        String pAlter;

        pVorname = getParameter("vorname");
        if (pVorname == null){
            pVorname = "init_Vorname";
        }
        toDisplayVorname = "Vorname : " + pVorname;

        pNachname = getParameter("nachname");
        if (pNachname == null){
            pNachname = "init_Nachname";
        }
        toDisplayNachname = "Nachname : " + pNachname;
        // ...weiter...
```

```
//...weiter...
pAlter = getParameter("alter");
    if (pAlter == null){
        alter = 20;
    }
    else {
        alter = Integer.parseInt(pAlter);
    }
    toDisplayAlter = "Alter : " + alter;
}

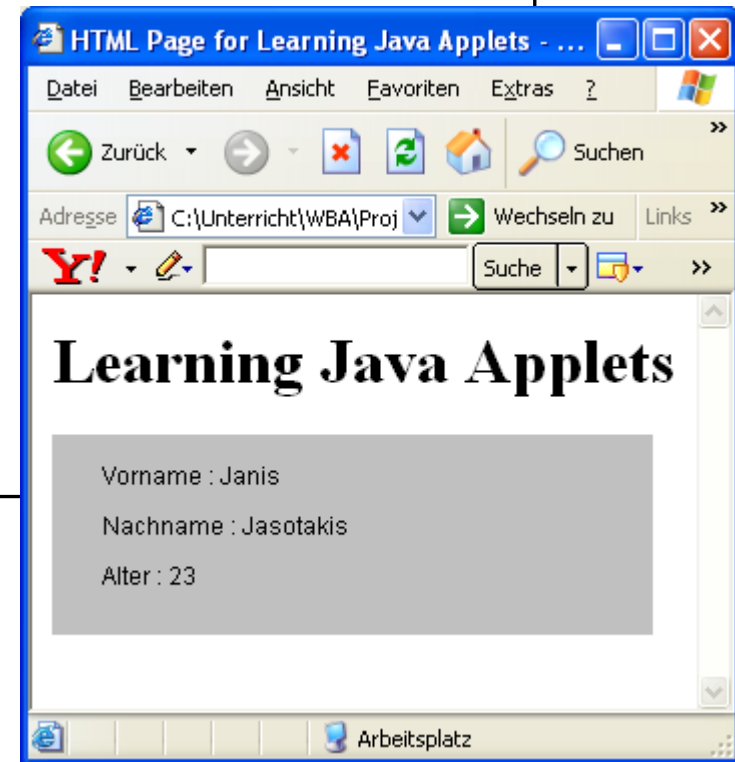
public void paint(Graphics g){
    g.drawString(toDisplayVorname,25,25);
    g.drawString(toDisplayNachname,25,50);
    g.drawString(toDisplayAlter,25,75);
}
}
```

Source der Applet-Datei

Einlesen von Parametern aus HTML : Beispiel

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>HTML Page for Learning Java Applets</title>
  </head>
  <body>
    <h1>Learning Java Applets</h1>
    <applet code="Parameter" codebase="./build/classes"
      width=300 height=100>
      <param name = "vorname" value = "Janis">
      <param name = "nachname" value = "Jasotakis">
      <param name = "alter" value = "23">
    </applet>
  </body>
```

HTML - Datei



Einlesen von Parametern mit JavaScript: Beispiel

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>HTML Page for Learning Java Applets</title>
    <script language="JavaScript">
      <!--
function setParams(){
  var tmp;
  tmp=parseInt(document.Uebergabe.alter.value);
  document.ParameterMitJavaScript.alterparam.value = tmp; }
      //-->
    </script>
  </head>

  <body>
    <h1>Learning Java Applets</h1>
    <applet code="ParameterMitJavaScript" name="ParameterMitJavaScript" codebase="./build/
classes" width=300 height=100>
      <param name= "alterparam" value="14" />
    </applet>
    <form name="Uebergabe" onsubmit="return setParams()">
      <input type="text" name="alter">
      <input type="submit" name="Senden" value="Senden">
    </form>
  </body>
```

Clientseitige Implementierungstechnologien (IV): Java Applet

1. Einführung
2. Vergleich zu JavaScript und Java
3. Laden eines Applets
4. Lebenszyklus
5. Einlesen von Parametern
- 6. Abschlussbemerkungen zu Problem Client-Technologien**
7. Praktische Tips zur Netbeans IDE 6.0
8. Literatur
9. Ausblick

Problem Client-Technologien

Problem Client - Technologien:

1. Clientseitige Interpretation des Codes notwendig
 - beim Client sind nicht immer die **aktuellsten Browserversionen** installiert (im Intranet vertretbar, nicht aber im Internet)
 - Entwicklung und Tests für unterschiedliche Browser notwendig
2. Sicherheitsbedenken – Konfiguration des Clients (Browser, Proxy, ...) kann Java-Interpretation, Java-Script (Aktive Scripting) ausschalten
3. Bei Einsatz von Client-Technologien muss Webapplikation auch ohne diese voll funktionsfähig sein : **redundante Funktionen** auf Serverseite

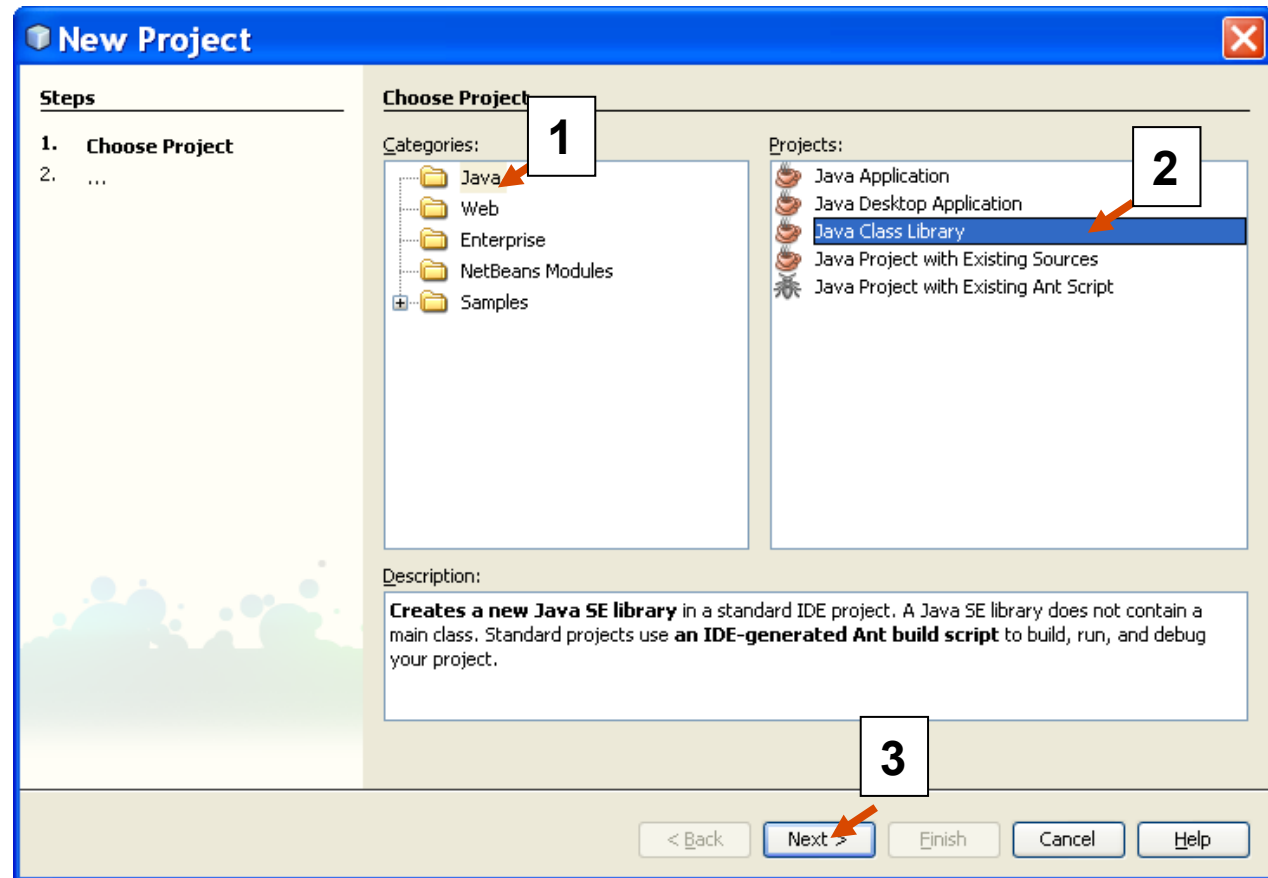
Clientseitige Implementierungstechnologien (IV): Java Applet

1. Einführung
2. Vergleich zu JavaScript und Java
3. Laden eines Applets
4. Lebenszyklus
5. Einlesen von Parametern
6. Abschlussbemerkungen zu Problem Client-Technologien
- 7. Praktische Tips zur Netbeans IDE**
8. Literatur
9. Ausblick

Netbeans : Erstellen eines Java-Projektes

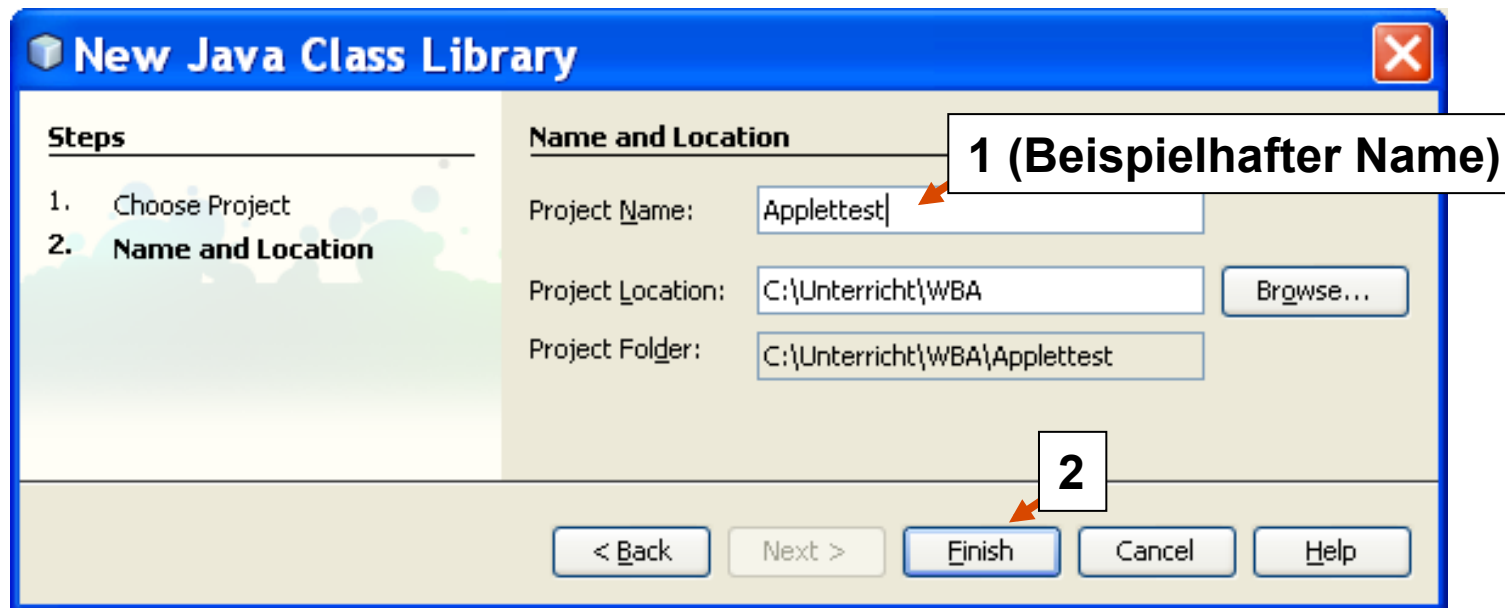
Genau so für Netbeans IDE 6.0, 8.0

Siehe auch - <https://netbeans.org/kb/docs/web/applets.html>



Auswahl : ->File ->New Project

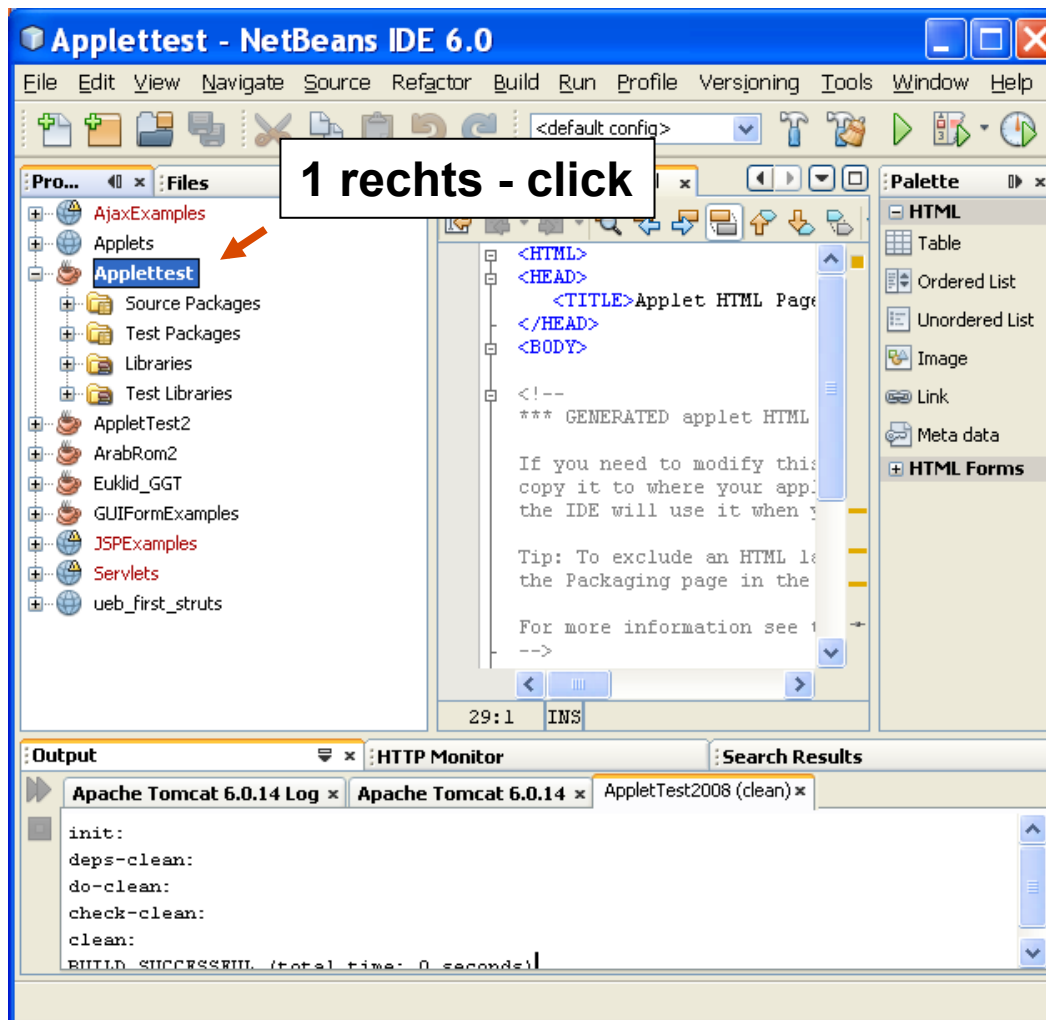
Netbeans : Erstellen eines Java-Projektes



Project wird erstellt.

Netbeans : Erstellen eines Java-Applets

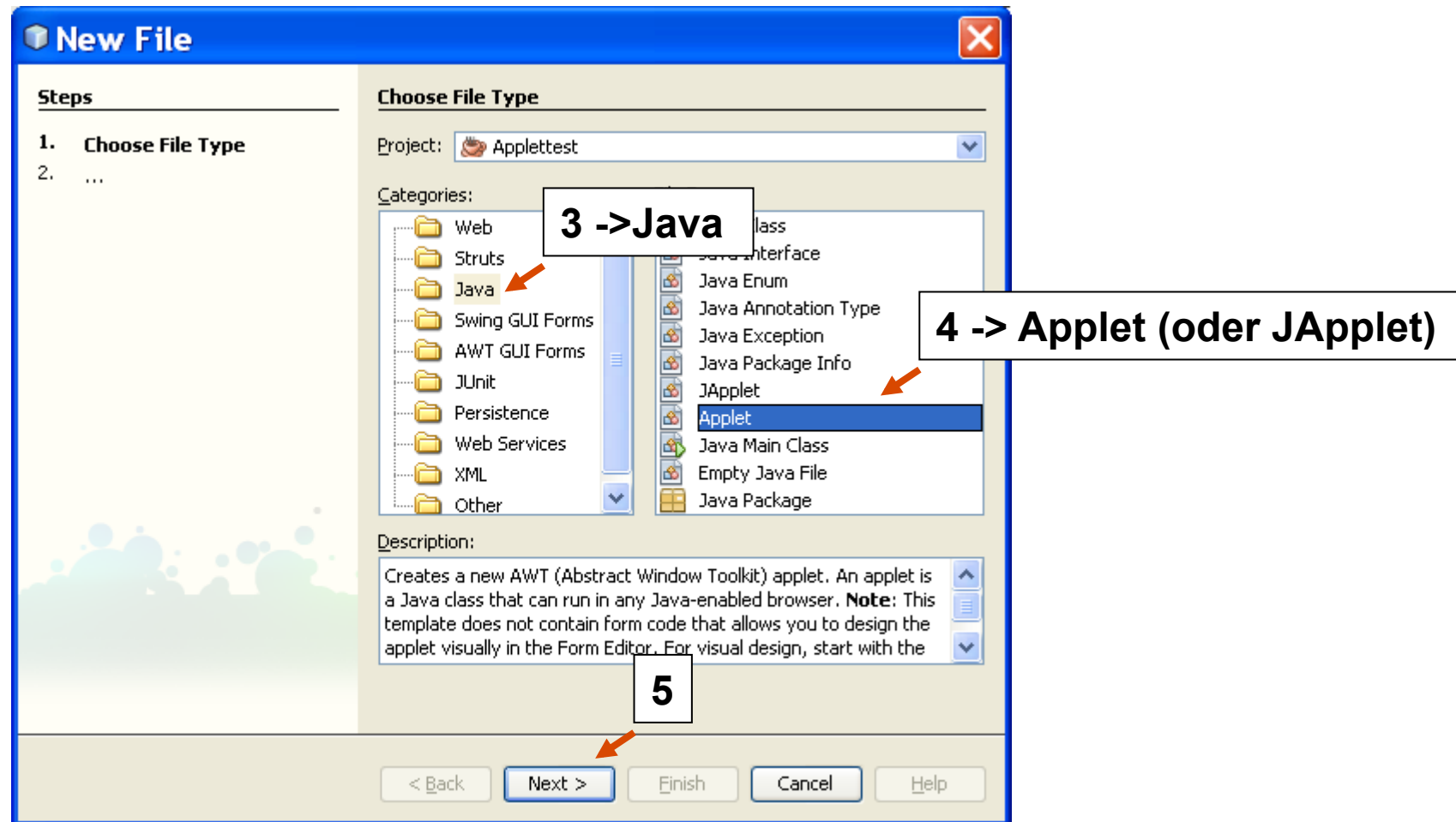
Rechts-Click „Projektname“ (z.B. Applettest) im Projektfenster



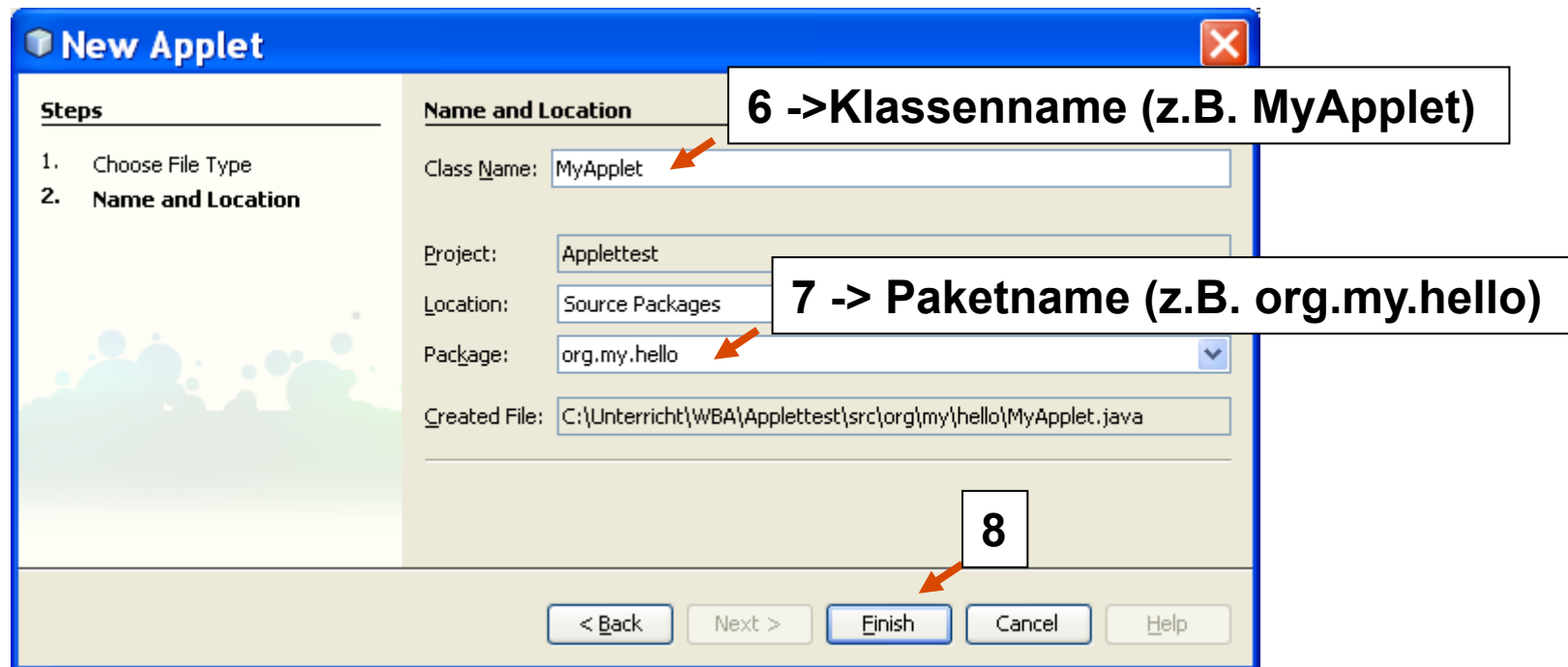
2. -> New -> Other

Netbeans : Erstellen eines Java-Applets

2. -> New -> Other



Netbeans : Erstellen eines Java-Applets



9. IDE erstellt Java-Applet – Schablone (z.B. MyApplet.java) , kann im Editor weiter programmiert werden.

Netbeans: Erstellen eines Java-Applets

z.B. für „MyApplet.java“

```
package org.my.hello;

import java.applet.Applet;
import java.awt.Graphics;
/**
 *
 * @author behrens
 */
public class MyApplet extends Applet {
    public void init() {
    }

    public void paint(Graphics g) {
        g.drawString("Hello applet!", 50, 25);
    }
}
```

Netbeans: Build und Run eines Applets

1. Rechts-Click Projektknoten (z.B. Applettest) -> Build
2. Applettest.jar wird erstellt

IDE Output-Window:

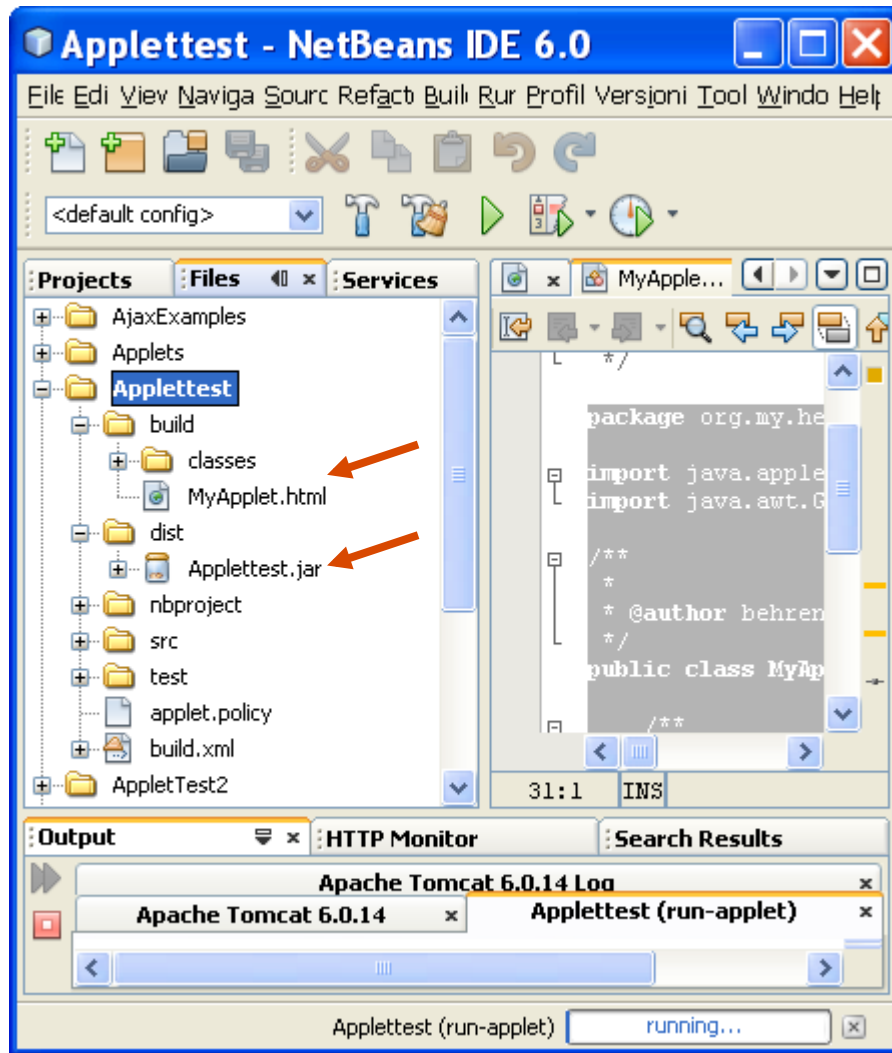
```
Created dir: C:\Unterricht\WBA\Applettest\build\classes  
Compiling 1 source file to C:\Unterricht\WBA\Applettest\build\classes  
compile:  
Created dir: C:\Unterricht\WBA\Applettest\dist  
Building jar: C:\Unterricht\WBA\Applettest\dist\Applettest.jar
```

3. Rechts-Click „MyApplet.java“ im Projekt-Fenster -> Run File
4. Applet – Viewer:



5. MyApplet.html –File wird erstellt im Build-Directory

Netbeans :Build und Run eines Applets



Im Files-Fenster:

Von netbeans generierte Dateien:

- Applettest
 - Verz. build „MyApplet.html“
 - Verz. dist „Applettest.jar“

Bemerkung:

Es muss meistens zweimal „build and run“ für die .class – Datei aufgerufen werden, bis das .html-file wirklich generiert und in der IDE angezeigt wird.

Netbeans :Build und Run eines Applets

Bei Aufruf von „MyApplet.html“:



Java-Anwendungen müssen signiert werden, wenn sie im Browser laufen können sollen.

Signieren eines Applets mit *java keytool* und *jarsigner* (JDK-Tools)

RSA-Schlüsselpaar erzeugen:

```
keytool -genkey -keyalg rsa -alias HelloWorldApplet
```

Certificate Signing Request an Certificate Authority:

```
keytool -certreq -alias HelloWorldApplet -file HelloWorldApplet.csr
```

CA signiert und schickt public key zurück.

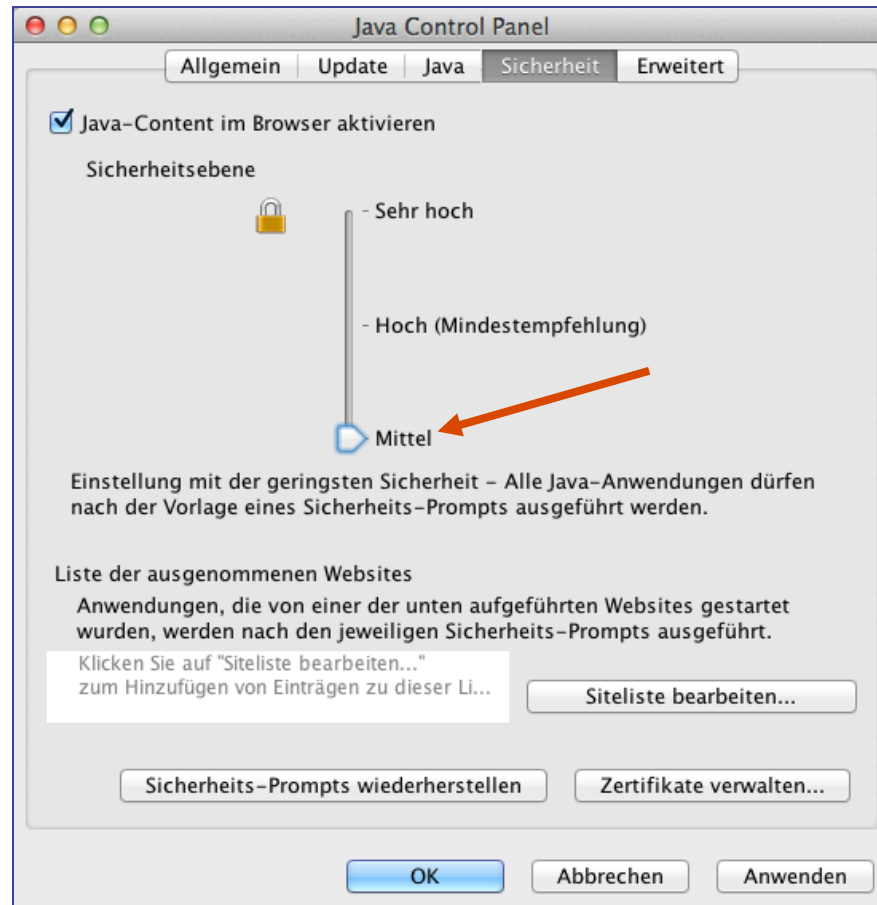
Importieren des public key in den keystore:

```
keytool -import -alias HelloWorldApplet -trustcacerts -file[Name der Public-key  
Datei von CA ]
```

Signieren des Applets mit dem Certificat mithilfe des jarsigners

```
jarsigner -verbose -signedjar HelloWorldApplet.jar unsigned_Applet.jar  
HelloWorldApplet
```

Java-Control Panel einstellen, um Applet auszuführen



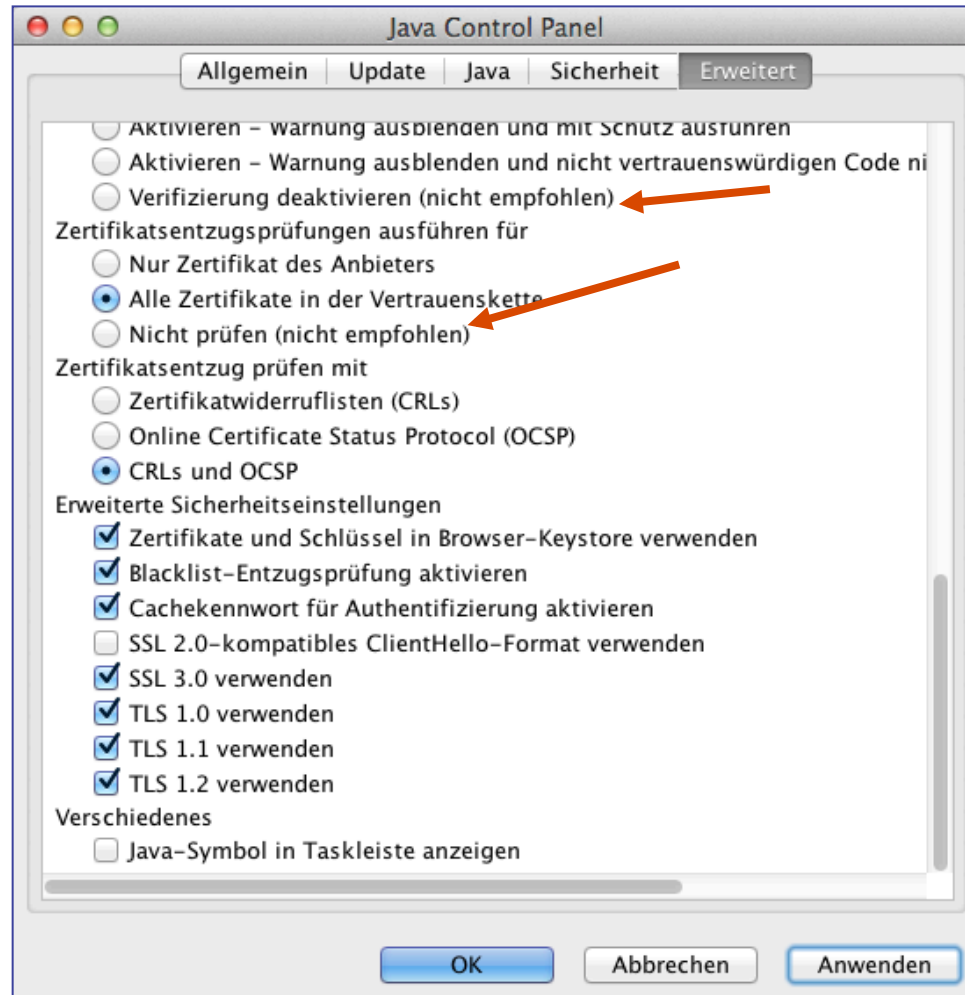
Java-Control-Panel aufrufen und **Mittlere Sicherheitsstufe** einstellen (default ist hoch)

Falls das nicht reicht, kann die Sicherheit für Entwicklungszwecke manuell einzeln weiter runter gesetzt werden.

Siehe weiter....

1. Hausaufgaben machen
2. **Danach wieder zurück stellen!**

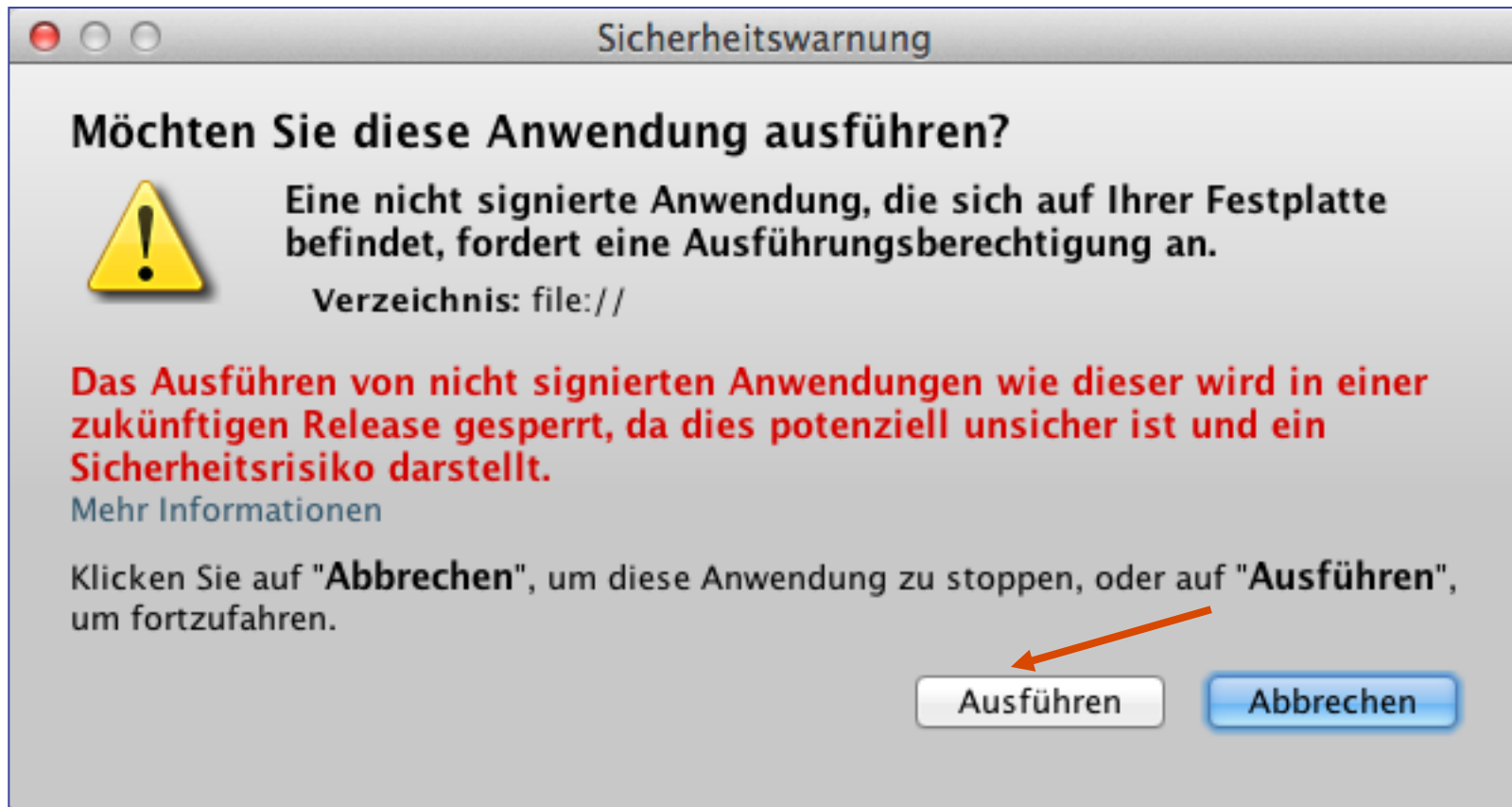
Java-Control Panel einstellen, um Applet auszuführen



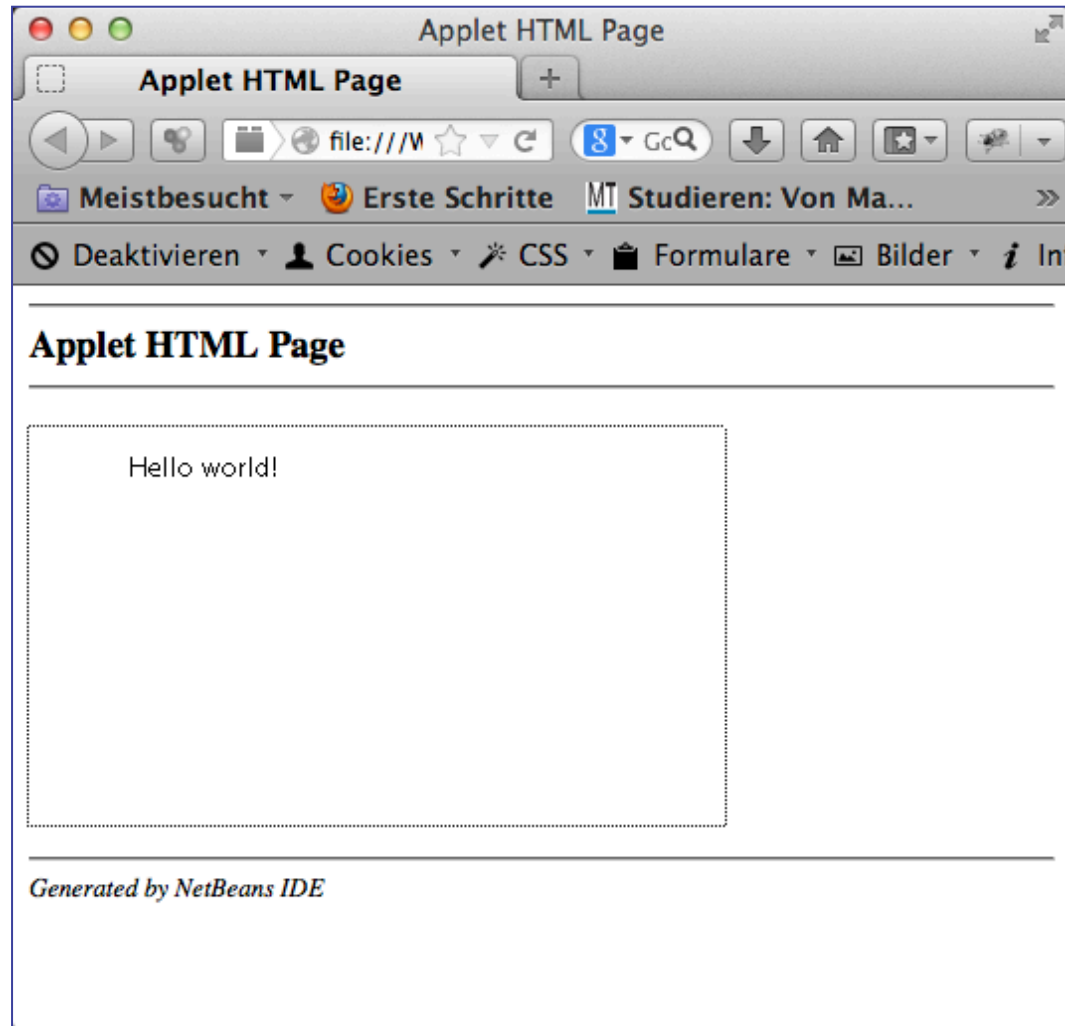
Java-Control-Panel:
erweiterte Einstellungen modifizieren

Java-Control Panel einstellen, um Applet auszuführen

Jetzt nur noch Warnung, bevor Applet ausgeführt wird:



Der Browser führt das Applet aus:



Self-Signieren eines Applets mit JDK-Tools

Tutorial mit Hinweisen für Einstellung der Java-Umgebung und das Erstellen von self-made keys:

<http://www.jade-cheng.com/uh/ta/signed-applet-tutorial/>



Literatur für Java - Applet

- **Heiko Wöhr „Webtechnologien“ , dpunkt.verlag, Heidelberg 2004**
- **Coy und Bornemann: „Java & Webapplikationen“, SPC TEIA Lehrbuch Verlag 2002**
- <http://java.sun.com/applets/>
- **Sven Eric Panitz „Java will nur spielen“, Vieweg+Teubner , Wiesbaden 2008 , Kap.6**
- **Günther Bauer „Architekturen für Web-Anwendungen“, Vieweg +Teubner, Wiesbaden 2009, Kap.5**
- <http://www.tutego.com/javabuch/>
- <http://www.heise.de/developer/artikel/Java-foerdert-nun-signierte-Applets-1840305.html>
- http://wiki.byte-welt.net/wiki/Einbinden_von_Applets_in_HTML-Dateien

Ausblick:

Einführung in Serverseitige Implementierungstechnologien