

# Multimedia im Netz

## Wintersemester 2012/13

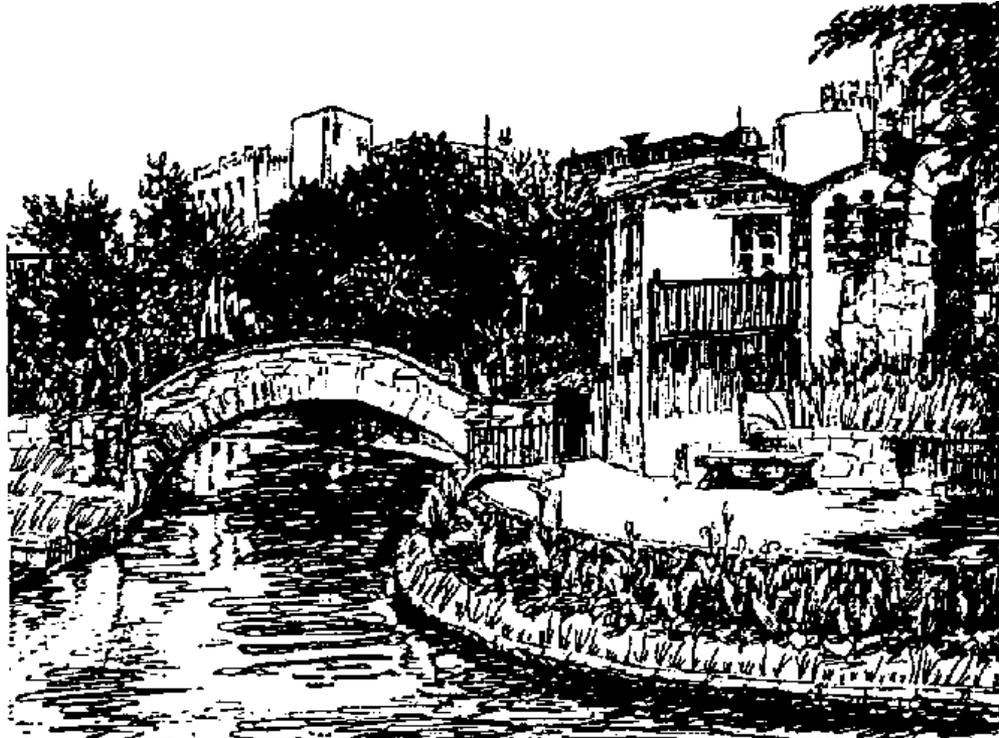
### Übung 09

# Klausur

- **Termin:** 20.02.2013; 18-20h
- Die Klausur ist Closed-Book (d.h., Hilfsmaterialien, wie zum Beispiel Skript, Ausdrucke, Musterlösungen, etc. sind nicht erlaubt).
- Die Anmeldung zur Klausur ist ab jetzt über UniWorx möglich.
- Anmeldeschluss: **13.02.2013; 12:00 Uhr**
- Bitte melden Sie sich bis spätestens **19.02.2013, 12:00 Uhr** über UniWorx von der Klausur ab, wenn Sie doch nicht daran teilnehmen möchten.

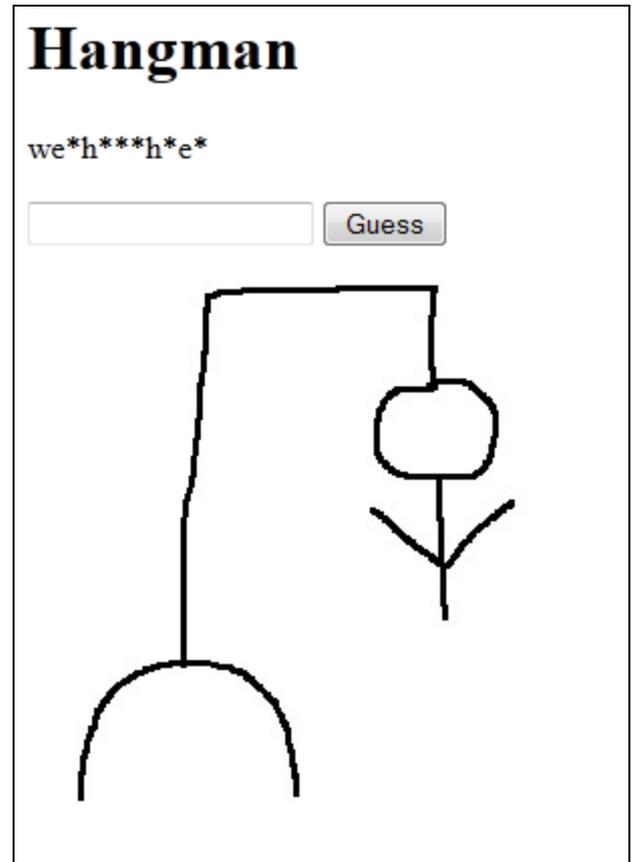
# Lösung zu Übungsblatt 07

# Steganographie



# Übungsblatt 08

- **Thema: JSP**
- Abgabe: 16.01.2013; 11:00 Uhr
- Abgabe erfolgt über UniWorx **und** die WAR-Datei des Spiels muss auf den JBOSS-Server hochgeladen werden
- Um die Zugangsdaten für den Server zu erhalten, kontaktieren Sie:  
[alina.hang@ifi.lmu.de](mailto:alina.hang@ifi.lmu.de)



# Entwicklungsumgebung

- Um das Übungsblatt zu bearbeiten ist es sinnvoll, sich eine Entwicklungsumgebung einzurichten, in der man JSP-Anwendungen erstellen und testen kann
- Man benötigt dafür:
  - Programmiersprache JAVA
  - Applikationsserver JBOSS 7.1.1 AS  
(<http://www.jboss.org/jbossas/downloads/>)
  - Entwicklungsumgebung Eclipse (IDE für Java EE Developers)
  - Web Tools Platform (WTP) für Eclipse  
<http://www.eclipse.org/webtools/>
  - Browser

# Server in Eclipse einrichten

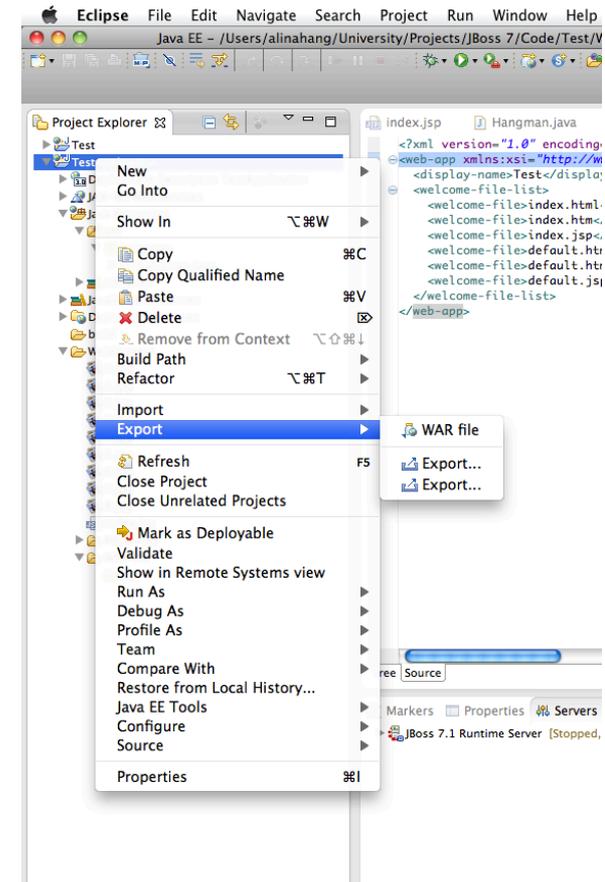
- Gehen Sie in die Einstellungen von Eclipse
- Unter dem Punkt Server > Runtime Environments, fügen Sie die JBOSS-Serverumgebung (7.1 Runtime) hinzu
- Wechseln Sie in Eclipse zur Server-View
- Erstellen Sie einen neuen Server
- Wählen Sie als Servertyp: JBOSS AS 7.1
- Folgen Sie den Anweisungen auf dem Bildschirm

# Erstes Projekt erstellen

- In Eclipse ein „Dynamic Web Project“ erstellen
- Target Runtime „Jboss 7.1“ auswählen
- web.xml-Datei automatisch erstellen lassen

# WAR-File exportieren

- Um für die Abgabe ein WAR-File des Projekts zu erstellen, gehen Sie wie folgt vor:
  - Rechtsklick auf das entsprechende Projekt
  - Export > WAR file auswählen



# JSP

- Java Server Pages
- Technik zur Generierung von dynamischen Webseiten
- Anweisungen können direkt in X(HTML), XML- oder beliebige Text-Dateien eingebettet werden
- Anweisungen werden zur Anfragezeit ausgeführt und dadurch dynamische Inhalte erzeugt
- Das Ergebnis ist ein XHTML-Dokument ohne Anweisungen

# JSP-Skriptelemente (1)

- Es gib verschiedene Arten von Skriptelementen:
  - JSP-Deklarationen
  - JSP-Skriptlets
  - JSP-Ausdrücke
  - JSP-Kommentare

# JSP-Skriptelemente (2)

- **Deklaration:** Hinzufügen von Attributen oder Methoden  
<%! Deklaration %>
- **Skriptlets:** Einbetten von Java-Anweisungen  
<% *Anweisungscode* %>
- **Ausdrücke:** zur Ausgabe von Werten in einer JSP-Seite  
<%= *JavaAusdruck* %>
- **Kommentare:** werden nicht zum Browser übertragen  
<%-- *Das ist ein Kommentar* --%>

# Beispiel

```
<html><head><title>Quadratzahl</title></head>
<body>
  <!-- Gibt die Quadratzahl von 5 aus -->
  <%! int zahl = 5; %>
  <p>
    Die Quadratzahl von <%= zahl %> ist:
    <% int ergebnis = zahl * zahl; %>
    <%= ergebnis %>
  </p>
</body>
</html>
```

# JSP-Direktiven

- steuern den Übersetzungsprozess und die Ausführung von JSP-Seiten
- 3 Arten von Direktiven:
  - **include**: Inhalte von anderen Dateien einfügen
  - **page**: Legt bestimmte Attribute für eine Seite fest
  - **taglib**: Bibliotheken zur Funktionserweiterung  
`<%@ include | page | taglib ... %>`

# Implizite Objekte

- ermöglichen den Zugriff auf wichtige Informationen:
  - request (javax.servlet.HttpServletRequest)
  - response (javax.servlet.ServletResponse)
  - session (javax.servlet.http.HttpSession)
  - out (javax.servlet.jsp.JspWriter)
  - pageContext (javax.servlet.jsp.PageContext)
  - ...
- Beispiel: **request.getParameter(ParameterName)**

# Sessions

- Sessioninformationen speichern  
`session.setAttribute(„variablenname“, new Instanz());`
- Sessioninformationen auslesen  
`session.getAttribute(„variablennamen“);`
- Session löschen und neue Session erstellen  
`session.invalidate();`  
`session = request.getSession(true);`