

## Übung „Netzbasierte Informationssysteme“ WS 2008/2009

---

### Übungsblatt 1

Ausgabe am 28.10.2008  
Abgabe bis spätestens 04.11.2008, 14 Uhr

Abgabe der Sourcen per Mail an: [paschke@inf.fu-berlin.de](mailto:paschke@inf.fu-berlin.de) (gepackt)  
Betreff: [CSW-NBI] Übung 1, <Name>, <Matrikelnummer>

#### Aufgabe A:

Die im Rahmen der Übung von Ihnen zu erstellenden Java-Komponenten und Web-Applikationen entwickeln Sie auf der integrierten Entwicklungsumgebung Eclipse©. **Gute Kenntnisse der Entwicklungsumgebung sind für einen späteren produktiven Einsatz derselben unumgänglich.** Machen Sie sich aus diesem Grunde zunächst mit Ihrer neuen Umgebung vertraut, indem Sie die nachfolgend aufgeführten vier Tutorials in Eclipse: ( Help → Help Contents → ... ) absolvieren:

- 1. Installieren sie Java und Eclipse auf ihrem Rechner**
- 2. Eclipse:**  
( Workbench User Guide → Getting Started → Basic Tutorial )
- 3. SVN:**  
( Subclipse – Subversion Eclipse Plugin → Getting Started → ... )  
( Subclipse – Subversion Eclipse Plugin → Tasks → Edit, Update, Commit )  
( Subclipse – Subversion Eclipse Plugin → Reference → Team Menu → ... )
- 4. Ant deployment**  
( Workbench User Guide → Getting Started → Ant & external tools tutorial )
- 5. Java Development User Guide**  
( Java Development User Guide → Getting Started → Basic tutorial → ... )  
( Java Development User Guide → Getting Started → Project Configuration → ... )

#### Aufgabe B:

Machen Sie sich mit den grundlegenden Prinzipien des *Java Web Application Development & Deployment* unter *Servlet API Specification 2.2* kompatiblen Servlet Containern am Beispiel der *Tomcat Servlet Engine* vertraut.

Hierzu lesen Sie bitte die Punkte *Introduction*, *Deployment Organization*, *Source Organization*, *Development Processes* und *Example Application* des *App. Dev. Guide* unter

 <http://jakarta.apache.org/tomcat/tomcat-5.0-doc/appdev/index.html>.

Installieren Sie Tomcat auf ihrem Rechner. Installieren und konfigurieren Sie die Tomcat Unterstützung in Eclipse.

## Aufgabe 1 - Wurzelziehen (2 Punkte):

Anhand eines Beispiels lernen Sie nun die Vorgehensweise des Deployens einer Webanwendung in die Tomcat Servlet Engine kennen. Die benötigten Sourcen finden Sie auf der Vorlesungshomepage.

1. Starten Sie den lokalen Tomcat auf Ihrem PC über die Eclipse-IDE
2. Überprüfen Sie die Funktionalität des Tomcats indem Sie sich die Startseite <http://localhost:8080> in Ihrem Browser anzeigen lassen. Falls Sie eine Fehlerseite bekommen, sollten Sie einen Restart der Tomcat Servlet-Engine versuchen.
3. Deployen Sie die statische HTML-page `startForm.html`
  - a. Wechseln Sie dazu in den `\webapps` Ordner in ihrem Tomcat
  - b. Legen Sie einen Unterordner `\aufgabe1` an, in diesem einen `\WEB-INF` Unterordner und in diesem wiederum einen `\classes` Unterordner
  - c. Kopieren Sie `startForm.html` in den zuerst erstellten Unterordner `\aufgabe1`
  - d. Restarten Sie den lokalen Tomcat über die Eclipse-IDE
  - e. Rufen Sie nun die soeben deployte, statische HTML-Seite im Browser auf: <http://localhost:8080/aufgabe1/startForm.html>
4. Klicken Sie auf den Button der `startForm.html` Seite – Sie sollten eine Fehlermeldung bekommen. Im Quellcode der HTML-Seite sehen Sie, dass in dem Formular ein Servlet aufgerufen wird, das noch nicht deployt ist.
5. Deployen Sie nun das Servlet `SqrtServlet.java`
  - a. Kopieren Sie den `web.xml` Deployment-Descriptor in `\aufgabe1\WEB-INF`
  - b. Die kompilierte `.class`-Datei von `SqrtServlet.java` muss noch in `\aufgabe1\WEB-INF\classes` kopiert werden. Sie müssen das `java`-Sourcefile mittels Ihrer lokalen Eclipse-Installation kompilieren. Legen Sie ein Eclipse Project für die Webanwendung `\aufgabe1` in den Konventionen der Apache-Group zur Source-Code Verwaltung an, wie Sie es bei der Durchführung der Tutorials aus Aufgabe B gelernt haben. Für diese Übung ist ab sofort das Einhalten dieser Konventionen bei der Anordnung von Webanwendungskomponenten Pflicht!
  - c. Rufen Sie nun erneut <http://localhost:8080/aufgabe1/startForm.html> auf. Machen Sie eine Eingabe und klicken Sie auf den Button. Nun sollten Sie eine Antwort des Servlets bekommen. Testen Sie auch gleich die Fehlerbehandlung des Servlets – was passiert bei ungültigen Formulareingaben wie z.B. „abc“ oder „-20“?
6. Eine andere Art, eine Webanwendung in die Servlet Engine zu Deployen, ist das Deployment mittels eines `.war` files. Vorteil dabei ist die vereinfachte Migration der Webanwendung. Deployen Sie nun auf diese Art eine neue Webanwendung namens `aufgabe1WAR`. Sie soll über die URL <http://localhost:8080/aufgabe1WAR/startForm.html> ansprechbar sein.
  - a. Erstellen Sie nun in Ihrem Workspace die neue Webanwendung `\aufgabe1WAR`. Legen Sie dazu in Ihrem Eclipse-Workspace eine entsprechende Verzeichnisstruktur an. Sie können sich an `\aufgabe1` orientieren.
  - b. Legen Sie das neue Servlet „`sqrtWARServlet`“ an das funktional dem `sqrtServlet` aus `aufgabe1` entspricht.
  - c. Ändern Sie für das Deployment die `web.xml` entsprechend ab – das neue Servlet soll über die URL <http://localhost:8080/aufgabe1WAR/startForm.html> ansprechbar sein.
  - d. Passen Sie den Aufruf des neuen `sqrtWARServlets` in der „`startForm.html`“ der `\aufgabe1WAR` an.
  - e. Machen Sie Eclipse den neuen Source-Folder `\aufgabe1WAR\src` bekannt: rechtsKlick auf das Projekt, Properties, Java Build Path, Source, Add Folder. Achten Sie darauf, dass Sie den richtigen Ordner adden – „`src`“! Ändern Sie den Ausgabeordner vom default-Ordner auf `\aufgabe1WAR\bin` um.
  - f. Kompilieren Sie das Eclipse-Projekt. Sollten sich Fehler ergeben so müssen Sie Ihre Sourcen sowie die Konfiguration des Projekts überprüfen.

- g. Sie haben die Teilkomponenten für die neue Webanwendung erstellt. Nun muss die neue Webanwendung den Konventionen entsprechend zusammengebaut werden. Wechseln Sie dazu auf Ihrem Laufwerk in den Workspace und darin in das Projekt. Hier finden Sie die gerade erstellte \aufgabe1WAR. Kopieren Sie den gesamten Folder \aufgabe1WAR in einen temporären Folder \temp\aufgabe1WAR.
- h. Passen Sie den temporären Folder an die Konventionen einer Webanwendung an – orientieren Sie sich dabei an der Webanwendung \webapps\aufgabe1. Alle nötigen Inhalte finden sich bereits im temporären Folder. Sie müssen nur die Ordnerstruktur ändern.
- i. Danach ist die neue Webanwendung fertig erstellt – man könnte Sie einfach in den /webapps Kontext des Tomcats kopieren und sie würde laufen (vgl. Schritte 3-5). Allerdings wollen wir nun ein einfach zu migrierendes Paket schnüren (ein .war-file) und danach direkt deployen. Öffnen Sie dazu eine Shell (Eingabeaufforderung) und wechseln Sie in das Verzeichnis \temp\aufgabe1WAR. Legen Sie nun das neue .war file mittels folgendem Befehl an:
 

```
jar -cf aufgabe1WAR.war WEB-INF *.html
```

 (Hinweis: Sehr empfehlenswert ist statt dieser händischen Generierung die Benutzung von Apache ANT für die Erstellung eines .war files; der jar-Befehl ist ebenso wie viele anderen Befehle ein Standardtask von ANT der direkt unterstützt wird; Referenz und Beispiele finden Sie z.B. unter <http://ant.apache.org/manual/CoreTasks/>; für diese Aufgabe ist jedoch die Verwendung von ANT noch optional).
- j. Nun Deployen Sie die neue Webanwendung aufgabe1WAR. Kopieren Sie dazu einfach das soeben erstellte aufgabe1WAR.war in das /webapps Verzeichnis auf Ihrem Netzlaufwerk. Rufen Sie die neue Webanwendung unter <http://localhost:8080/aufgabe1WAR/startForm.html> auf. (Hinweis: falls Sie bereits ein gleichlautendes .war-file deployed hatten, müssen Sie vor dem Kopieren Ihres neuen .war-files in /webapps den lokalen Ordner /webapps/aufgabe1 und alle darin enthaltenen Inhalte löschen; diese wurden automatisch beim deployen der alten .war-Datei angelegt und können vom Tomcat nicht überschrieben werden; evtl. müssen Sie zum Löschen der Dateien den Tomcat stoppen.)
- k. Löschen Sie der Übersicht halber das temp-Verzeichnis auf Ihrem Laufwerk. Es wird nicht mehr benötigt.

### Aufgabe 2 - BackwardString (6 Punkte):

Entwickeln Sie nun eine eigene Webanwendung „aufgabe2“. Diese Anwendung hat eine statische Seite „stringForm.html“ mit einem Formular, in das ein beliebiger String eingegeben werden kann. Nach dem Klicken auf einen Button wird ein Servlet „backwardServlet“ aufgerufen. Dieses liest den eingegebenen String aus, invertiert den String (bsp: „meinString“ wird zu „gnirtSniem“) und stellt dann das Ergebnis als html-response dar.

Entwickeln Sie Ihre Quellen in Eclipse in einem neuen Ordner \aufgabe2 den Konventionen entsprechend. Die Seite stringForm.html soll über /aufgabe2/stringForm.html aufrufbar sein, das Servlet über die URL /aufgabe2/backwardServlet.

Für diese und alle nachfolgenden Aufgaben soll das Deployment mittels eines ANT-Scripts build.xml erfolgen. Auf der Homepage finden Sie ein Beispielscript in dem Sie nur noch die aktuellen Kontexte anpassen müssen. Kopieren Sie dazu das Script in den \src-Folder in Ihren Workspace und passen Sie die Properties *build.context.name* und *project.aufgabe.name*.

Versuchen Sie vor dem Ausführen des Scripts die darin durchgeführten Teilschritte zu verstehen. Wie Sie sehen, sind die Schritte den manuellen Deployment-Schritten aus Aufgabe 1 sehr ähnlich.

Das Deployment mittels des Scripts lässt sich durchführen, indem Sie die build.xml in Eclipse in die ANT-View ziehen und danach einfach auf den deploy-Task doppelklicken.

Für die Entwicklung eigener ANT-Scripts findet sich eine gute Referenz zu allen unterstützten Tasks z.B. unter <http://ant.apache.org/manual/CoreTasks/>.

### Aufgabe 3 - CSWCalculator (7 Punkte):

In dieser Aufgabe soll ein einfacher geometrischer Dienst „aufgabe3“ erstellt werden. Im Gegensatz zu den bisherigen Aufgaben besteht dieser Dienst nur aus einem einzigen Servlet „CSWCalculator“, hat also keine statische html-Seite. Dennoch soll der Aufbau ähnlich sein – in einem Formular sollen eine positive Zahl (*Radius*) sowie eine *Berechnung* (*Flaeche* oder *Umfang*) eingegeben werden können (Drop-Down oder Checkbox, eine gute HTML-Referenz finden Sie z.B. unter <http://de.selfhtml.org>).

Wichtig ist bei allen Eingabefeldern, dass die Art der Eingabe überprüft wird – sollte die Eingabe keine positive Zahl sein, so brechen Sie die Berechnung ab und schicken eine entsprechende, aussagekräftige Meldung als Ergebnis zurück.

Bei konformer Eingabe berechnet das Servlet je nach Auswahl der Berechnung entweder den Flächeninhalt oder den Umfang des durch den Radius charakterisierten Kreises. Das Ergebnis der Berechnung wird an den Client zurückgegeben.

Überschreiben Sie für die Generierung der Formularseite die `doGet()`-Methode des Servlets. Die Berechnung sowie die Darstellung des Ergebnisses erfolgt wie gehabt in der `doPost()`-Methode. Der Dienst soll über `/aufgabe3/cswCalculator` aufgerufen werden können.

Das Deployment erfolgt analog zur Aufgabe 2 mittels eines angepassten ANT-Scripts.

Viel Erfolg!