

**mitp**

Elisabeth Jung

# **Servlets und JavaServer Pages**

## **Das Übungsbuch**

### **Band III**

**Über 100 Aufgaben mit vollständigen Lösungen**

**Trainieren Sie Ihre Java-Kenntnisse**

**Learning by Doing anhand  
praktischer Übungen**

**Mit vollständigen und  
kommentierten Lösungen**

# Einleitung

Das Übungsbuch zu Servlets und JavaServer Pages soll dem Leser zeigen, wie Java-basierte Webapplikationen mit Servlets, JavaServer Pages, JavaBeans und JDBC (Java Database Connectivity) entwickelt werden können. Die erstellten Webapplikationen dienen dem Lernen des Programmierens von Servlets und JSP-Seiten und sollen dem Leser helfen, seine eigenen Webapplikationen zu erstellen, die im Internet all denjenigen, die einen Webbrowser besitzen, zur Verfügung gestellt werden können.

Für das Übersetzen und Ausführen der Programme wird auf den Tomcat-Container der Apache Software Foundation zurückgegriffen.

Die theoretischen Ausführungen, die in jedem der Unterkapitel den Aufgaben vorangestellt sind, basieren auf den Versionen Servlets 2.5, JSP 2.1, MySQL 5.1 und Tomcat 6 der Java Enterprise Edition (Java EE). Wie auch in den beiden anderen Übungsbüchern, *Java 6 Das Übungsbuch* Band I und Band II, sollen diese, zusammen mit den dazugehörigen Aufgaben, ähnlich wie in Lehrbüchern dazu dienen, die wichtigsten Begriffe aus der Servlet- und JSP-Programmierung zu veranschaulichen und eine Vielzahl von Klassen mit ihren Feldern und Methoden näher kennen zu lernen. Im Unterschied zu Lehrbüchern bleibt es dabei nicht nur bei Vorschlägen oder angegangenen Übungsaufgaben, sondern alle Aufgaben behandeln ausführlich die vorgegebenen Themen und bekommen einen vollständigen Lösungsvorschlag am Ende jedes Kapitels hinzugefügt. Die Lösungen werden vom ersten Ansatz bis zum Ergebnis vollständig durchgeführt und sowohl über ihre Formulierung als auch den darin enthaltenen Kommentaren vollständig erklärt. So kann es hilfreich sein, gerade bei den etwas umfangreicheren Webapplikationen, ein Durcharbeiten von Aufgaben wie anhand eines Lehrbuchs vorzunehmen.

Das Buch *Servlets und JavaServer Pages* soll ebenfalls den Umgang mit der Online-Dokumentation der Java-Servlet- und JSP-API erklären und so dem Leser behilflich sein, die dazugehörigen Standard-Klassen anhand von vielen Beispielen näher kennen zu lernen.

## Vorkenntnisse

Als Voraussetzung für die Arbeit mit diesem Buch gelten Grundlagen im Bereich der Programmierung und im Umgang mit dem Betriebssystem. Gleichzeitig sind grundlegende Kenntnisse aus der Programmiersprache Java im Umgang mit Klassen, Interfaces, Ein/Ausgaben, Threads, Exceptions und Errors erforderlich. Ele-

mente aus der grafischen Programmierung mit AWT und Swing, die in einigen der Aufgaben eingesetzt werden, sprechen für sich und machen nicht unbedingt weitere Kenntnisse aus diesem Bereich erforderlich. Es ist ein Übungsbuch, das zusätzlich zu den Lehrbüchern von Java zu den im Vorhinein angesprochenen Themen benutzt werden soll und kann.

In Kapitel 1 wird eine Wiederholung zu http (HyperText Transfer Protocol) und Netzwerkprogrammierung der Einführung von Servlet-Klassen vorangestellt. Die in den Aufgaben benutzten [X]HTML- und XML-Anweisungen genauso wie die Befehle aus der SQL-(Structured-Query-Language-)Datenbankabfragesprache werden ebenfalls an den Stellen, wo diese zum Einsatz kommen, im Detail beschrieben.

Die Java-6-Übungsbücher Band I und Band II sind keine Voraussetzung, aber die darin enthaltenen Definitionen und Erläuterungen von Begriffen, wie auch die Aufgaben selbst, können in vielen Fällen eine wichtige Hilfe bieten.

## Aufbau des Buches

In den Unterkapiteln jedes einzelnen Kapitels wird der Stoff, der in den Übungsaufgaben dieses Kapitels verwendet wird, kurz beschrieben. Danach folgen die Aufgabenstellungen der Übungen und am Ende der Kapitel Lösungsvorschläge.

Der Schwierigkeitsgrad von Aufgaben wird im Aufgabenkopf durch 1 bis 3 Sternchen gekennzeichnet:

- 1 Sternchen für besonders einfache Aufgaben, die auch von Anfängern leicht bewältigt werden können
- 2 Sternchen für etwas kompliziertere Aufgaben, die einen durchschnittlichen Aufwand benötigen
- 3 Sternchen für Aufgaben, die sich an geübte Programmierer richten und einen wesentlich höheren Aufwand oder die Kenntnis von speziellen Details erfordern

### Kapitel 1

Am Anfang des ersten Kapitels wird das Erstellen von statischen Webseiten mit HTML (HyperText Markup Language) und XHTML (Extensible HyperText Markup Language) erläutert und auf die Trennung der Präsentation, mit Hilfe von Cascading-Style-Sheets-(CSS-)Formatvorlagen, und der Strukturierung von derartigen Dokumenten hingewiesen.

XHTML ist eine Neuformulierung von HTML mit Hilfe von XML (Extensible Markup Language), eine Auszeichnungssprache, die zur Darstellung hierarchisch strukturierter Daten benutzt wird und als reduzierte Variante von SGML (Standard Markup Language) gegründet wurde. SGML ist die erste Auszeichnungssprache, aus der auch HTML abgeleitet wurde.

Es wird auf die unterschiedlichen Versionen von Auszeichnungssprachen eingegangen und die darin benutzten Möglichkeiten von Dokumententypdefinitionen.

Zur Illustration des Nachrichtenaustausches zwischen Webserver und Webbrowser werden die Struktur des http-Protokolls, die http-Methoden GET und POST und die Kommunikation über das http-Protokoll zwischen Webserver und Webbrowser vorgestellt.

Der Schwerpunkt des ersten Kapitels liegt im Erstellen von dynamischen Webseiten mit Servlets. So folgt der Beschreibung von Installation und Konfiguration des Tomcat-Containers die Beschreibung von Klassen und Schnittstellen der Servlet-API, deren Methoden vom Container automatisch aufgerufen werden.

Um Servlets auszuführen, müssen sie in Webapplikationen integriert werden und ihre Namen in einem dieser Webapplikation zugeordneten Deployment Descriptor (DD) konfiguriert werden. Dabei spielen die Namen und Pfadnamen sowie das Abbilden (Mapping) von Servlets auf URLs eine wichtige Rolle.

Wir beschäftigen uns in diesem Kapitel ausführlich mit der Definition von Formulardateien und dem Zusammenhang zwischen Formulardaten und Anfrage-(Request-)Parametern, der Wiedergabe von Request- und Response-Header-Einträgen im Browser, der Konfiguration von Initialisierungsparametern für Servlets und Webapplikationen und dem Umleiten und Weiterleiten von Client-Anfragen.

Gleichzeitig wird die Definition von Attributen im Zusammenhang mit Servlets und deren Gültigkeitsbereichen (Scopes) mit vielen anschaulichen Beispielen dokumentiert, sowie die Thread-Sicherheit bei parallelen Servlet-Ausführungen, die Protokollierung von Nachrichten in den Servlet-Log-Files und das Abfangen von Fehlermeldungen beschrieben und geübt.

Kapitel 1 beinhaltet auch das Aufbauen und Verfolgen von Client-Sitzungen (Session-Tracking) mit Cookies und URL-Rewriting und das Cookie-Handling mit Java 5 und Java 6.

Über einen vereinfachten Bücherkauf im Web mit einem Warenkorb wird in diesem Kapitel auf das Erzeugen einer eigenen Session für jeden Benutzer, das Setzen von Session-Attributen und deren Bedeutung für den Session-Gültigkeitsbereich aufmerksam gemacht. Die dazugehörigen Aufgaben werden in jedem der nachfolgenden Kapitel um die dort eingeführten Merkmale erweitert, um den Leser so Schritt für Schritt zu vollständigen Webapplikationen wie die aus Kapitel 4 zu führen, die nach der vorgegebenen MVC-Architektur aufgebaut sind.

## **Kapitel 2**

Dynamische Webseiten können nicht nur mit Servlets, sondern auch mit JavaServer Pages erstellt werden. Dafür werden in Kapitel 2 die Möglichkeiten der Einbettung von Java-Code in [X]HTML-Seiten beschrieben. Wir erläutern die spezifischen Elemente, die dazu in JSP-Seiten zum Einsatz kommen wie: JSP-Direktiven, -Deklarationen und -Ausdrücke und weisen auf die Unterschiede zwischen JSP-Seiten und JSP-Dokumenten (in denen eine XML-konforme Syntax benutzt wird) hin.

Die Gültigkeitsbereiche von Attributen spielen auch im Zusammenhang mit JavaServer Pages eine wichtige Rolle, genau wie die in JSP dem Programmierer zur Verfügung gestellten impliziten Objekte.

Des Weiteren befassen wir uns in diesem Kapitel mit der Definition von JavaBean-Klassen und JSP-Aktionen, die unter anderem die Einbindung von JavaBeans in JSP-Seiten gewährleisten.

Wir weisen ständig auf die Wichtigkeit des Erstellens von Scripting-freien JSP-Seiten hin und führen in diesem Zusammenhang die Expression Language (EL) und benutzerdefinierte Tags ein, die in einer Vielzahl von Aufgaben eingesetzt werden.

Eine große Aufmerksamkeit wird auch dem wichtigen Zusammenspiel zwischen dem Setzen von Attributen mit JavaBeans und benutzerdefinierten Tags und der Auswertung ihrer Werte mit der EL gewidmet.

Nachdem das Zusammenspiel von JSP-Aktionen, EL-Ausdrücken, JavaBeans und Tags eingeübt wurde, kann zum Erstellen von eigenen Tag-Libraries bzw. der Benutzung von Standard-Bibliotheken wie die JSTL (JSP Standard Tag Library) übergegangen werden. Diese Bibliothek beinhaltet eine Vielzahl von Tags, die anstelle von eigenen benutzerdefinierten Tags parallel zur EL bei der Entwicklung von JSP-Seiten eingesetzt werden können und vielen Anforderungen im Vorhinein genügen.

Die besten Erfahrungen beim Schreiben von JSP-Seiten können Sie machen, indem Sie versuchen, Ihre ersten JSP-Seiten, in denen für das Einbetten von Java-Code noch Scriptlets benutzt wurden, gezielt in Scripting-freie JSP-Seiten umzusetzen, um damit die gleichen Ergebnisse bzw. Anzeigen im Browser zu erreichen.

### Kapitel 3

Mit JDBC (Java Database Connectivity) können in Java relationale Datenbanksysteme wie MySQL, Oracle, Informix, DB2, Access etc. eingebunden werden. Den Beispielen aus diesem Buch wird in Kapitel 3 die Datenbank MySQL zugrunde gelegt, die im Internet kostenlos zu erhalten ist und mit Tomcat einwandfrei benutzt werden kann, nachdem der zugehörige JDBC-Treiber dafür installiert wurde.

Zum Erstellen einer Datenbank und Erfassen, Abfragen und Manipulieren von Daten werden die wichtigsten SQL-Befehle, die das Datenbanksystem MySQL unterstützt, in den Servlets und JSP-Seiten aus diesem Kapitel jedes Mal im Detail beschrieben und eingesetzt.

Im Abschluss zu den Ausführungen zu JDBC werden die Aufgaben aus Kapitel 1 und 2 für eine Buchbestellung im Internet mit den in diesem Kapitel gewonnenen Erkenntnissen erweitert. Dabei wird beim Aufbau des Buchkatalogs anstelle der Benutzung von Maps auf eine im Vorhinein zu diesem Zweck erstellte Datenbank zugegriffen.

Um die Daten eines Modells von der Anwendungslogik (Geschäftslogik) noch genauer zu trennen, soll der direkte Zugriff auf die Daten eines Datenbanksystems nicht von den Methoden der Anwendungsobjekte (Geschäftsobjekte), die als JavaBean-Instanzen erzeugt werden, durchgeführt werden, sondern von eigenständigen Java-Klassen, die als Zugriffsmodule für das Datenbanksystem dienen und die aus Tabellen eingelesene Spaltenwerte als Eigenschaften für die JavaBean-Objekte setzen.

So werden im Hinblick auf die in Kapitel 4 zu erstellenden Webapplikationen, die eine Wiederholung aller vorausgegangenen Begriffe und Erläuterungen darstellen, JavaBean-Klassen definiert, die gleichartige Eigenschaften mit den Spaltennamen von Datenbanktabellen definieren und über weitere Java-Klassen mit Hilfe von Objekten vom Typ dieser Klassen Zugriffe auf die Datenbanktabellen ermöglichen.

Alle Servlet- und JSP-Beispiele aus den ersten drei Kapiteln werden anhand einer Webapplikation mit dem Namen `java6uebungsbuch3` dem Leser zum Lernen und Testen zur Verfügung gestellt.

## Kapitel 4

In Kapitel 4 beschäftigen wir uns ausführlich mit der MVC-Architektur von Webapplikationen und deren Entwicklung von der so genannten MVC-1-Architektur, die beschreibt, dass Datenhaltung, Anwendungslogik und Präsentation in einer JSP-Seite anzusiedeln sind, zur MVC-2-Architektur, die sich schnell in der Programmierung von Webapplikationen durchgesetzt hat.

Die MVC-2-Architektur übergibt das Managen des gesamten Applikationsflusses an ein oder mehrere Servlets, die den Controller-Zweig einer Webapplikation bilden. Die Servlets lesen beim Starten Werte, die vom Benutzer als Anfrageparameter im Servlet-Aufruf übergeben wurden oder vom Viewer über das Einlesen von Formulardaten in Anfrageparameter hinterlegt wurden. Sie übergeben diese Werte im Konstruktoraufruf von Anwendungsobjekten vom Typ einer JavaBean und holen sich die angeforderten Ergebnisse aus dem Modell, indem sie Methoden von Datenbankzugriffmodulen aufrufen, um Modelldaten abzuändern und/oder auszuwerten. Die vom Modell abgeholte Antwort wird vom Servlet als Attribut für einen der Scopes (Gültigkeitsbereiche) der Webapplikation gesetzt, um anderen Komponenten den Zugriff darauf zu vereinfachen, wie z.B. JSP-Seiten, die für die Präsentation aufgerufen werden.

Diese Vorgänge werden anhand von typischen Webapplikationen wie Chatrooms und Webshops demonstriert, in denen parallel Programmcode in der Standard-JSP-Syntax und der XML-Syntax benutzt wird.

Ebenfalls im vierten Kapitel kommen wir auf das Deployment (die Verteilung) von Webapplikationen zurück, klären den Unterschied zwischen der physischen und logischen Verzeichnisstruktur einer Webapplikation und beschreiben, welche Art von Musterdefinitionen beim Mapping von Servlets auf URLs eingesetzt werden kann.

Wir befassen uns des Weiteren mit Sicherheitsvorkehrungen für Webapplikationen, die vier wichtige Merkmale umfassen: Authentifizierung, Autorisierung, Vertraulichkeit und Datenintegrität.

Ein weiteres Thema aus diesem Kapitel sind so genannte Filter, die einer Webapplikation hinzugefügt werden können, um eine http-Anfrage abzufangen und den in der entsprechenden Filter-Klasse enthaltenen Programmcode durchzuführen, bevor das Servlet oder die JSP-Seite, an die die Anfrage gerichtet wurde, ausgeführt werden. Sie können aber auch eine http-Anwort abändern, nachdem das Servlet mit seiner Arbeit fertig ist, aber bevor diese an einen Client zurückgegeben wird.

Für das Deployment einer Webapplikation kann eine `.war`-Datei benutzt werden, die den Namen der zukünftigen Applikation trägt und alle Verzeichnisse und Dateien für diese Applikation enthält. Die `.war`-Datei muss vor dem Starten von Tomcat in dessen Verzeichnis `webapps`, in dem alle Webapplikationen hinterlegt werden müssen, abgelegt werden. Dann wird Tomcat das Web-Archiv entpacken und mit der dazugehörigen Verzeichnisstruktur und dem Dateinamen als Wurzelverzeichnis für die neue Webapplikation unter `webapps` ablegen. Die Webapplikation `DeploymentSecurityFilterApp`, die Servlets und JSP-Seiten zu den Themen Verteilung, Filterdefinitionen und Sicherheitsvorkehrungen beinhaltet, wird nach diesem Prinzip aufgebaut.

Im Abschluss des vierten Kapitels wird eine Webapplikation, die alle Regeln der MVC-2-Architektur respektiert, erstellt. Das heißt, dass die Datenbankzugriffe aus den Methoden der `JavaBean`-Klassen in eigene `Java`-Klassen ausgelagert werden und das Abändern und Auswerten von Modelldaten von Servlets durchgeführt wird. Die Servlets stellen den für die Präsentation zuständigen JSP-Seiten die Anwendungsobjekte als Attributwerte zur Verfügung, so dass diese, wenn überhaupt, nur lesend auf die `JavaBean`-Objekte zurückgreifen, um deren Eigenschaften im Browser anzuzeigen.

Gleichzeitig kommen mit diesem Beispiel Filter-Klassen zum Einsatz und für die neue Webapplikation werden im DD Sicherheitskontrollen eingebaut. Mit benutzerdefinierten Tags wird das Tagesdatum in JSP-Seiten eingeblendet.

Auch in die in Kapitel 4 erstellten Webapplikationen werden nicht alle denkbaren Funktionen integriert, damit deren gesamte Funktionalität im Sinne der MVC-Architektur einfacher erläutert werden kann. Es bleibt wie immer eine Herausforderung für den Leser, die von ihm gewünschten Erweiterungen einzubauen, sollten diese für einen Live-Einsatz bereitgestellt werden.

## Benötigte Software

Für die Durchführung der Übungen können Sie, wie in Kapitel 1 spezifiziert wird, die Enterprise Edition von Java verwenden, die unter <http://java.sun.com/javaae/technologies> zu finden ist und die APIs von Java EE 5 und Java EE 6 beinhaltet.

Sie können aber auch wie in diesem Buch vorgehen und dazu das aktuelle Java Development Kit der Java Standard Edition (Java SE 6 JDK 1.6) von der Java Homepage von Sun unter der Adresse <http://java.sun.com/javase/downloads/index.jsp> herunterladen.

Tomcat 6.0 mit Servlet 2.5/JSP2.1 kann ebenfalls kostenlos von <http://jakarta.apache.org> heruntergeladen werden. Installieren und konfigurieren Sie diese Software wie in Kapitel 1 beschrieben.

Die dazugehörige Dokumentation zur Servlet- und JSP-API finden Sie unter:

```
http://tomcat.apache.org/tomcat-5.5-doc/servletapi/overview-tree.html  
http://tomcat.apache.org/tomcat-5.5-doc/jspapi/overview-tree.html
```

MySQL 5.0 kann von <http://mysql.com> heruntergeladen werden und wie in Kapitel 3 beschrieben installiert und konfiguriert werden.

All diese Softwarekomponenten sind Betriebssystem-unabhängig. Den Beispielen aus diesem Buch liegt das Betriebssystem Windows XP zugrunde.

Für das Entwickeln von Java-Webapplikationen mit Servlets und JSP-Seiten können auch Entwicklungstools, wie z.B. NetBeans IDE (Integrated Development Environment), eingesetzt werden. Diese sind weder Voraussetzung noch Bestandteil dieses Buches. Die Java-Servlet-Klassen, HTML- und JSP-Seiten lassen sich grundsätzlich mit einem Texteditor wie z.B. TextEdit oder auch WordPad eingeben und können wie beschrieben mit dem Tomcat-Container in Webapplikationen eingebunden und ausgeführt werden.

## Website

Die Website zum Buch an der Adresse [www.it-fachportal.de/5603](http://www.it-fachportal.de/5603) beinhaltet den plattformunabhängigen Quellcode der Lösungsvorschläge. Diese Archivdatei enthält alle [X]HTML-Seiten, Java-Klassen, JSP-Seiten und -Dokumente und Bild-dateien in einer Verzeichnisstruktur, die mit der im Buch beschriebenen übereinstimmt. Sie können diese Verzeichnisse in das entsprechende Tomcat-Verzeichnis ziehen und mit einem Browser testen. Bei der Arbeit mit dem Datenbanksystem MySQL müssen die dazugehörigen Datenbanken und Tabellen erstellt werden, wie dies in den entsprechenden Aufgaben vermerkt ist.

Zusätzlich zu den Sourcecode-Dateien werden auf dieser Webseite die Screenshots aus den Programmausgaben und viele zusätzliche Bilddateien, die beim Durchführen von parallelen Tests mit Webbrowsern erstellt wurden, zum Download zur Verfügung gestellt.

Ich wünsche Ihnen viel Erfolg beim Programmieren von Webapplikationen mit Java Servlets und JavaServer Pages.

Elisabeth Jung