

# Über dieses Buch

Stellen Sie sich dieses Buch wie ein Seminar vor. Dieses Kapitel ist dann Ihre Begrüßung und der erste Überblick. Sie haben Ihren Mantel abgelegt, an Ihrem Sitzplatz den üblichen Schreibblock vorgefunden und möchten nun wissen, welche Themen Sie erwarten und wo es in der Pause Kaffee gibt. Nun, Getränke kann ich Ihnen nicht anbieten, aber schauen Sie bitte im Downloadbereich unter [www.rheinwerk-verlag.de/4243](http://www.rheinwerk-verlag.de/4243) nach den Materialien zum Buch. Und jetzt zu Ihnen, und zu den Themen:

Dies ist ein Praxisbuch zu den Java EE-Technologien. Die Java EE-Spezifikation definiert alle Technologien und Dienste, die im Java EE-Standard enthalten sind. Für Sie als Java-Entwickler ist es wichtig zu wissen, welche dieser Technologien als Programmierschnittstelle für Ihre Geschäftsanwendung benötigt werden und welche davon die wertvollsten sind. Dieser Priorisierung habe ich im Buch Rechnung getragen. Um die richtigen Java EE-Technologien für Sie auszuwählen, habe ich Statistiken der verbreitetsten Stellenportale zugrunde gelegt und genau diejenigen Java EE-Technologien beschrieben, die Ihnen mit größter Wahrscheinlichkeit in der Praxis begegnen werden. Darüber hinaus gehe ich auf das »Drum und Dran« eines Java EE-Projekts ein. Beispielsweise erfahren Sie, wie Sie eine Java EE-Anwendung mit AMDD/XP entwerfen und mit Scrum planen.

Damit Sie die Übungen ausnahmslos nachvollziehen können, zeige ich Ihnen auch, wie Sie einen Java EE Server, eine relationale Datenbank und Ihre Entwicklungsumgebung einrichten und konfigurieren können. Denn all das werden Sie ebenso benötigen, wenn Sie als Mitarbeiter eines professionellen Java EE-Projekts eingesetzt werden.

Mein Ziel ist es, Sie hierbei an die Hand zu nehmen, Ihnen jeden einzelnen Schritt genau zu erklären und zu allen Techniken hilfreiche Beispiele und praktische Übungen anzubieten. Ich habe Wert darauf gelegt, dass Sie Schritt für Schritt Kenntnisse erlangen, die bausteinartig aufeinander gesetzt werden können. Gleichzeitig habe ich die Technologien so sortiert, dass sich die didaktisch sinnvollste Reihenfolge ergibt.

Fühlen Sie sich angesprochen? Dann sind Sie hier richtig, denn wir (die Fachlektoren, die Korrekturleser, die Sachverständigen und ich) sind davon überzeugt, dass Ihnen dieses Buch in Ihrem Arbeitsumfeld nützlich sein wird.

## Die Kapitel dieses Buches

In Kapitel 1 lege ich zu Beginn den Fokus auf den Java EE-Standard und die Java EE-Spezifikation. Ich erläutere, wofür die Java EE-Technologien erfunden wurden. Dieser Einstieg ist für Java EE-Neulinge wichtig, da sie hierdurch den Zweck des Java EE-Standards erfahren. Danach befassen wir uns mit den Java EE-Anwendungsszenarios und der Softwarearchitek-

tur von Java EE-Anwendungen. Zuletzt gebe ich Ihnen einen Rundumblick über die Java EE 8-Technologien.

In **Kapitel 2** richten Sie Ihre Entwicklungsumgebung ein. Hierbei installieren Sie Ihren eigenen Java EE Server und die Entwicklungsumgebung Eclipse. Danach programmieren Sie auch eine kleine Java EE-»Hallo Welt«-Anwendung, denn durch das aktive Programmieren wird die erste Lernhürde optimal überwunden. Keine Sorge: Hierbei werden Sie auf einfache Kenntnisse zurückgreifen. Außerdem erklären wir die einzelnen Schritte genau.

In jedem wenn auch noch so kleinen Projekt beginnt man mit der Planung und dem Entwurf. Genau das ist das Kernthema von **Kapitel 3**. Hier erfahren Sie, wie Sie Java EE-Projekte im professionellen Umfeld planen und wie Sie eine Java EE-Anwendung von Grund auf designen. Die Techniken hierzu nennen sich Scrum und AMDD/XP. Während dieser konzeptionellen Arbeit entstehen auf dem Whiteboard (oder auf dem Papier) sogenannte User-Stories, die Sie für die Erstellung von UI-Prototypen nutzen werden. Die UI-Prototypen werden mit HTML5 und CSS3 angefertigt und in Eclipse ausgeführt. Wir zeigen auch das sogenannte HTTP-Monitoring, mit dem Sie eine detaillierte Auskunft über die Kommunikation zwischen Client und Server erhalten.

In **Kapitel 4** veranschaulichen wir die grundlegende Low-Level-Programmierung mit Servlets.

**Kapitel 5** erklärt, wie Sie für die Präsentation der Geschäftsdaten JSPs programmieren.

In **Kapitel 6** installieren Sie die *Oracle Database* als relationale Datenbank und erzeugen in ihr die Datenbanktabellen für die Beispielanwendung des Buches. Hierbei werden Sie ANSI SQL-92 einsetzen.

In **Kapitel 7** greifen Sie mit der Low-Level-Technologie JDBC auf die Datenbanktabellen zu.

In **Kapitel 8** entfallen manuelle Kodierungen von eigenen JDBC-Anweisungen. Stattdessen werden die Geschäftsdaten als einfache Java-Klassen entworfen. Das Persistieren erfolgt hierdurch komfortabel über einen Automatismus des JPA-Frameworks.

**Kapitel 9** zeigt, wie Sie die Java EE-Anwendung, die im Buch bislang mit den althergebrachten Servlets und JSPs realisiert wurde, nun mit der High-Level-Technologie *Java Server Faces* programmieren. Besonders Eilige können die Servlet- und JSP-Programmierung überspringen und sich direkt in diesen Teil des Buches stürzen. Bedenken Sie aber, dass gute Kenntnisse in der Low-Level-Programmierung eine Voraussetzung dafür darstellen, die High-Level-Frameworks von Grund auf zu verstehen.

In **Kapitel 10** werden Sie *Enterprise JavaBeans* kennenlernen. Dabei wird die Programmierung von *Session Beans* und *Message Driven Beans* mit der Technologie *Java Message Service* erklärt.

**Kapitel 11** behandelt die WebServices-Technologien *JAX-WS* und *JAX-RS* sowie *JSON-P* und *JSON-B*, die das Einlesen und Erzeugen von JSON vereinfachen.

## Was setzt dieses Buch voraus?

Java EE 8- und Java-OpenSource-Technologien setzen fundierte Kenntnisse der *Java Standard Edition 8* (Java SE 8) voraus. Dieses Buch geht selbstverständlich nur in sehr begrenztem Umfang auf die Java-SE-Themen ein. Ich gehe davon aus, dass Ihnen die objektorientierte Programmierung mit Java SE keine Probleme bereitet. Aus der Sicht des Java EE-Entwicklers werden die Technologien der Java SE gewissermaßen als Untermenge des Java EE-Standards betrachtet.

## Das Betriebssystem

Ob Sie ein Unix-basiertes Betriebssystem (wie Linux, Solaris oder macOS) oder ein Microsoft-Betriebssystem (Windows) einsetzen, spielt eigentlich keine Rolle. Es ist nur wichtig, dass Sie sich mit Ihrem Betriebssystem auskennen. Für den Java EE-Entwickler sind Linux und Solaris von besonderer Bedeutung, da sie in der Produktion meistens die Zielplattform für den Java EE Server darstellen. Dies betrifft aber nicht den Arbeitsplatz des Entwicklers in den Unternehmen. Erstaunlicherweise werden Java EE-Entwickler fast immer an einen Microsoft-Windows-Rechner gesetzt. Dies liegt meistens an dem geringeren Aufwand für das Netzwerk- und Rechner-Pflegepersonal: Microsoft Windows ist pauschal gesagt »einfacher gestrickt«. In der Praxis sieht es dann so aus, dass der Java EE-Component-Developer, der seine Aufgabe auf einem Microsoft-Windows-Betriebssystem vollbracht hat, seinen Quelltext in ein Versionierungs-Repository über Subversion oder Git eincheckt, sodass sich auch der Java EE-Deployer daran erfreuen kann. Der Deployer checkt den Quelltext des Java-Developers auf einem Linux-Betriebssystem aus, kompiliert ihn und installiert die Java EE-Anwendung auf einem Server, der häufig entweder auf einem Linux- oder einem Unix-Betriebssystem basiert.

Weil der Java EE-Component-Developer aber meistens lediglich mit dem Betriebssystem Microsoft Windows in Berührung kommt, reichen diese Kenntnisse häufig aus. Andererseits werden immer häufiger virtuelle Maschinen auf den Windows-Rechnern genutzt, auf denen ein Linux-Derivat wie Ubuntu oder Mint betrieben wird. Deshalb werde ich die Beispiele aus dem Buch auch unter Linux testen, sodass die Screenshots und die Erläuterungen in manchen Fällen diese Variante zeigen. Achten Sie als Unix-Nutzer auf die Dateipfad-Unterschiede zwischen Unix- und Windows-Betriebssystemen. Als Unix-Nutzer werden Sie u. a. das C: entfernen und den Backslash durch einen Slash ersetzen. Auf einem Unix-System (also Linux, Solaris oder macOS) sollten Sie ferner darauf achten, dass Sie über ausreichende Benutzerrechte verfügen, um die in diesem Buch beschriebenen Vorgänge durchführen zu dürfen.

## Java Development Kit

Als *Java Development Kit* setzt Java EE 8 das JDK 8 ab dem Update 144 voraus. Java SE 9 wird nicht unterstützt. Dabei sollten Sie auch die Umgebungsvariablen `JAVA_HOME` und `PATH` angepasst haben. Die Umgebungsvariablen werden Sie in diesem Buch an der Schreibweise `[GROSSBUCHSTABEN_IN_ECKIGEN_KLAMMERN]` erkennen. Wenn zum Beispiel im Buch `[JAVA_HOME]/bin` steht, ist somit der Dateipfad (das Installationsverzeichnis von Java) und darunter das Unterverzeichnis `/bin` gemeint.

## Der Java EE Server

Während der Drucklegung dieses Buches stand nur ein einziger Java EE 8-Server zur Verfügung, der vollständig Java EE 8-konform war, und das ist *GlassFish 5*. Deshalb werden wir GlassFish 5 in den Kapiteln 1–10 einsetzen. Der GlassFish Server erfüllt die Voraussetzungen eines vollkonformen Java EE Servers aber auch deshalb optimal, weil er seit dem Jahre 2006 die Referenzimplementierung eines zertifizierten Java EE Servers ist. Weil die Oracle Corporation bereits seit November 2013 die kommerzielle Unterstützung des GlassFish Servers eingestellt hat, könnte der Payara Server eine gute Alternative für Unternehmen sein, die den GlassFish Server produktiv einsetzen und auf eine technische Unterstützung angewiesen sind. Denn die Lücke der fehlenden Unterstützung wurde durch das Unternehmen C2B2 Consulting im Jahre 2014 geschlossen. Payara ist ein neuer Java EE Server, der vom GlassFish Server abgeleitet (geforkt) ist und dabei gleichzeitig auch eine telefonische 24/7-Unterstützung anbietet. Dabei werden BugFixes regelmäßig eingearbeitet.

Obwohl andere Java EE 8-Server während der Entstehung dieses Buches noch nicht verfügbar waren, ist es aus heutiger Sicht so, dass die baldige Entwicklung folgender bedeutender Java EE 8-konformer Server mit Full Profile zu erwarten ist:

- ▶ Oracle WebLogic
- ▶ IBM WebSphere Application Server
- ▶ JBoss Application Server

Wenn Sie einen dieser Java EE Server in der Java EE 8-konformen Version verwenden, sollten die Beispielprogramme ebenso ausführbar sein, da die in diesem Buch eingesetzten Technologien unter den zertifizierten Java EE 8-Servern kompatibel sind.

## Die Entwicklungsumgebung

Die Beispiele in diesem Buch sollten Sie mit einer integrierten Entwicklungsumgebung (*IDE*) programmieren. Die bekanntesten IDEs sind *Eclipse*, *NetBeans* und *IntelliJ*. Ich setze voraus, dass Sie bereits mit einer dieser IDEs gearbeitet haben. Gemessen an den Stellenanzeigen gehört *Eclipse* zu den gängigen Voraussetzungen, die ein Java EE-Mitarbeiter mitbringen muss. Deshalb wird in diesem Buch Wert auf dieses Know-how gelegt.

## Die relationale Datenbank

Die Java EE-Spezifikation sieht vor, dass Anwendungsdaten dauerhaft gespeichert werden. Die Persistenz erfolgt in der Regel in einer relationalen Datenbank. Seit jeher verwenden geschäftskritische Systeme die Datenbankmanagementsysteme *IBM DB2* und *Oracle Database*, weil diese von Beginn an für den Betrieb auf Großrechnern ausgelegt waren. Sowohl IBM DB2 als auch die Oracle Database liegen, was Hochperformance und Hochverfügbarkeit betrifft, weit vor der Konkurrenz. Beide Datenbankmanagementsysteme stehen sich in Zuverlässigkeit und Datendurchsatz in nichts nach. Deshalb ist es auch kein Wunder, dass Geldinstitute und Großindustrie fast ausschließlich mit diesen Datenbankmanagementsystemen arbeiten, wenn es um ihre monetären Belange geht. Weitere Konkurrenten wie *Informix*, *Sybase* oder *MS SQL Server* sind – gemessen an ihrem Marktvolumen – gegenüber DB2 und Oracle kaum nennenswert.

Ein weiteres sehr beliebtes Datenbankmanagementsystem ist das Open-Source-Produkt *MySQL*. Seit vielen Jahren gewinnt MySQL als Datenbankmanagementsystem Marktanteile. Als die *Oracle Corporation* das Unternehmen *Sun Microsystems* übernahm, ging auch die Weiterentwicklung von MySQL an den neuen Eigner über. Obgleich MySQL eine vielversprechende Zukunft vorausgesagt wird, ist es in der Finanzwelt und generell im Einsatzbereich geschäftskritischer Unternehmensanwendungen noch nicht angekommen.

Zur praktischen Arbeit mit einer Datenbank in diesem Buch habe ich mich für die Oracle Database entschieden, weil in den Stellenbörsen vor allem nach Entwicklern mit Oracle-Kenntnissen gesucht wird.

## Das durchgehende Programmierbeispiel

Viele der Programmierübungen dieses Buches haben einen sehr minimalen Charakter – so ähnlich wie die bewährten »Hallo Welt«-Programme. Andere wiederum sind Teil einer Java EE-Anwendung für einen Onlineshop. Diese Java EE-Anwendung wird im Buch immer wieder genutzt, um die Technologien auch im realen Umfeld zu zeigen. Dadurch lernen Sie praxisnah, wie in einem Java EE-Projekt gearbeitet wird.

Bei dem durchgehenden Programmierbeispiel »Onlineshop« bieten Kunden eigene Artikel an. Gleichzeitig können Kunden Artikel anderer Kunden erwerben. Ansonsten habe ich den Onlineshop so einfach wie nur möglich gehalten, denn die technologischen Feinheiten sind komplex genug, und wir sollten uns nicht in der Fachlichkeit verlieren. Beispielsweise besteht die Beispieldatenbank aus nur zwei Tabellen. Auf diese Weise geht der Fokus auf das Lernziel nicht verloren.

Beim Entwurf des Onlineshops lernen Sie darüber hinaus, wie Sie die sogenannten agilen Methoden nutzen, um ein Java EE-Projekt zu planen und die Java EE-Anwendung zu entwerfen.

Den bestmöglichen Nutzen ziehen Sie aus dem Buch, indem Sie die Übungen auf Ihrem Rechner ausprobieren, denn durch die selbständige Kodierung lernen Sie optimal, wie man die Java EE-Technologien einsetzt. Auf diese Weise entwickeln Sie sich Schritt für Schritt vom Java EE-Neuling zum Java EE-Profi. Dies ist der Weg, um anschließend überzeugend behaupten zu können, dass Sie die Technologien nicht nur verstehen, sondern Java EE-Anwendungen auch selbst programmieren können. Und genau das wird von Projektanbietern und Arbeitgebern verlangt.

Zu jedem Kapitel liegt ein entsprechendes Beispiel vor. Dieses Beispiel wird von Kapitel zu Kapitel immer wieder erweitert, sodass es häufig auch die Technologien des letzten Kapitels mitführt. Versuchen Sie, das Programm mit dem Buch fortführend weiterzuentwickeln. Fühlen Sie sich aber auch frei, die Quelltexte dieses Buches für Ihre eigenen Zwecke zu nutzen. Die Java-Gemeinde ist eine Open-Source-Gemeinschaft, die von der Begeisterung jedes einzelnen Java-Entwicklers lebt und sich ständig vergrößert. Diesem Gedanken gemäß dürfen auch die in diesem Buch gezeigten Quellen benutzt und weitergegeben werden.

## Code Conventions

Die Les- und Wartbarkeit von Quelltexten spielt bei kritischen Geschäftsanwendungen eine große Rolle. Eine komplexe Java EE-Anwendung eines Großkonzerns ist teuer und wird von vielen Entwicklern erwartet. In diesem Umfeld spielt es eine große Rolle, ob die Variablennamen oder Zeileneinschübe einheitlich gesetzt sind. Aus diesem Grund wird eine solche Kodierungskonvention auch in den meisten geschäftskritischen Projekten vorgegeben. In der Regel entsprechen solche Kodierungskonventionen der Vorgabe des ehemaligen Herstellers. Sun Microsystems, also der Erfinder von Java, hat die empfohlenen Kodierungskonventionen in folgendem Dokument festgelegt: <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>.

Die gute Nachricht ist, dass die Code Convention von Sun Microsystems auch in den gängigen IDEs (Eclipse, NetBeans, IntelliJ etc.) voreingestellt ist. Halten Sie diese Formatierungsregeln ein, denn dann müssen Sie sich im beruflichen Umfeld seltener umstellen. Auch in diesem Buch versuche ich, mich stets an diese vom Hersteller Sun Microsystems vorgegebenen Code Conventions zu halten. Ausnahmen stellen Listings dar, bei denen hierdurch eine eingerückte Zeile außerhalb des druckbaren Seitenrands ragt.

## Danksagung

Mein allergrößter Dank gehört den zuständigen Lektorinnen Anne Scheibe und Almut Poll sowie den Fachlektoren Björn Saxe und Nicolas Bost. Zusätzlich bedanke ich mich bei den zahlreichen Korrekturlesern und Bearbeitern vom Rheinwerk Verlag, die ich nie zu Gesicht bekomme und die dennoch fleißig im Hintergrund mitgearbeitet haben. Letztlich ist es das

Team des Rheinwerk Verlags als Ganzes, das durch seine besonders professionelle und sehr partnerschaftliche Art so ein großes Buch ermöglicht hat.

Die bedeutendste Stütze aber war meine Frau Susanne, die mich auch über die Dauer dieser Neuauflage erneut mit ihrer übermenschlichen Geduld begleitet hat.

**Alexander Salvanos, Bonn 2018**

## Die Gutachter

**Björn Saxe** ist als IT-Berater im öffentlichen Sektor tätig. Als Software Engineer konzipiert und implementiert er Anwendungssysteme mit Java EE und Spring.

**Nicolas Bost** ist Experte für die Architektur komplexer Systemlandschaften mit SAP-Beteiligung. Aktuell begleitet er für SAP als PaaS-Berater Unternehmen auf ihrem Weg in die Cloud.

Diese Leseprobe haben Sie beim  
 **edv-buchversand.de** heruntergeladen.  
Das Buch können Sie online in unserem  
Shop bestellen.

[Hier zum Shop](#)