

HANSER

Christian Bauer, Gavin King

Java Persistence mit Hibernate

ISBN-10: 3-446-40941-6

ISBN-13: 978-3-446-40941-5

Vorwort

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/978-3-446-40941-5>
sowie im Buchhandel



Geleitwort zur zweiten Auflage

Als *Hibernate in Action* vor zwei Jahren erschien, wurde es nicht nur sofort als maßgebliches Buch über Hibernate akzeptiert, sondern auch als definitives Werk über das objekt-relationale Mapping.

In der Zwischenzeit hat sich die Persistenz-Landschaft durch das Release der Java Persistence API verändert, dem neuen Standard für objekt-relationales Mapping für Java EE und Java SE, das als Teil der Spezifikation von Enterprise JavaBeans 3.0 im Java Community Process entwickelt wurde.

Bei der Entwicklung der Java Persistence API hat die EJB 3.0-Expertengruppe in hohem Maße von der Erfahrung mit den objekt-relationalen Mapping-Frameworks profitiert, die in der Java Community bereits in Benutzung sind. Als ein führendes Framework hat Hibernate einen besonders signifikanten Einfluss auf die technische Richtung bei Java Persistence gehabt. Das lag nicht nur daran, dass Gavin King und andere aus dem Hibernate-Team an der Entwicklung des EJB 3.0-Standards mitgearbeitet haben, sondern zum großen Teil auch am direkten und pragmatischen Ansatz, den Hibernate bezüglich des objekt-relationalen Mappings und der Einfachheit, Klarheit und Leistungsfähigkeit seiner APIs genommen hat – und dem Reiz, der sich daraus für die Java Community ergab.

Zusätzlich zu ihrem Beitrag zu Java Persistence haben die Entwickler durch das in diesem Buch beschriebene Release (Hibernate 3) auch Hibernate beträchtlich weitergebracht. Dazu gehören der Support für Operationen mit großen Datensätzen, weitere und ausgefeiltere Mapping-Optionen (vor allem für den Umgang von Datenbanken aus Altsystemen), Datenfilter, Strategien zum Managen von Konversationen und die Integration mit Seam, dem neuen Framework für die Entwicklung von Webapplikationen mit JSF und EJB 3.0.

Java Persistence mit Hibernate ist von daher um einiges mehr als bloß die zweite Auflage von *Hibernate in Action*. Es bietet einen umfassenden Überblick über alle Möglichkeiten der Java Persistence API neben denen von Hibernate 3 sowie eine detaillierte Vergleichsanalyse beider APIs. Dieses Buch beschreibt, wie der Java Persistence Standard mit Hibernate implementiert wurde und wie man mit den Hibernate-Extensions für Java Persistence arbeitet.

Noch wichtiger ist, dass Christian Bauer und Gavin King bei ihrer Darstellung von Hibernate und Java Persistence die grundlegenden Prinzipien und Entscheidungen veranschaulichen und erklären, die sowohl beim Design als auch der Nutzung eines objekt-relationalen Mapping-Frameworks berücksichtigt werden müssen. Durch ihre Ausführungen über die dem ORM zugrunde liegenden Probleme bekommen die Leser ein fundiertes Verständnis über den effektiven Einsatz von ORM als Enterprise-Technologie.

Java Persistence mit Hibernate richtet sich von daher an eine breit gefächerte Zielgruppe von Entwicklern: Neueinsteiger beim Thema objekt-relationales Mapping bis hin zu erfahrenen Entwicklern, die mehr über die topaktuellen technologischen Innovationen in der Java-Community erfahren wollen und wissen möchten, was als Resultat dieser Arbeit geschehen ist und noch weiter auftauchen wird.

Linda DeMichiel

Specification Lead

Enterprise JavaBeans 3.0 und Java Persistence

Sun Microsystems



Geleitwort

Relationale Datenbanken stehen zweifellos im Zentrum moderner Unternehmen.

Während moderne Programmiersprachen wie Java™ eine intuitive, objektorientierte Ansicht von Business-Entities auf Applikationsebene bieten, sind die diesen Entities zugrunde liegenden Unternehmensdaten von ihrem Wesen her relational. Gegenüber den früheren navigationalen sowie den späteren OODB-Modellen besteht die größte Stärke des relationalen Modells überdies darin, dass es vom Design her von der programmatischen Manipulation und dem View auf Applikationsebene der Daten, mit denen es umgeht, intrinsisch keine Ahnung hat.

In vielen Anläufen hat man an einem Brückenschlag zwischen relationalen und objektorientierten Technologien gearbeitet oder versucht, die eine durch die andere zu ersetzen, doch die Kluft zwischen beiden ist eine der harten Tatsachen des heutigen Enterprise Computings. Hibernate stellt sich durch seinen Ansatz des objekt-relationalen Mappings (ORM) dieser Herausforderung, eine Brücke zwischen relationalen Daten und Java™-Objekten zu schaffen. Hibernate begegnet dieser Herausforderung auf sehr pragmatische, direkte und realistische Weise.

Wie Christian Bauer und Gavin King in diesem Buch demonstrieren, muss man es zur effektiven Nutzung der ORM-Technologie verstehen und konfigurieren können, wie die Vermittlung zwischen relationalen Daten und Objekten ausgeführt wird, wenn es sich nicht gerade um die allereinfachsten Enterprise-Umgebungen handelt. Das erfordert, dass sich der Entwickler sowohl mit der Applikation und deren Datenanforderungen als auch der Abfragesprache SQL, der relationalen Speicherstrukturen und dem Potenzial zur Optimierung, das die relationale Technologie bietet, auskennen muss.

Hibernate bietet nicht nur eine komplett funktionsfähige Lösung, die diesen Anforderungen aus dem Stand entspricht, sondern weist auch eine flexible und konfigurierbare Architektur auf. Hibernate wurde von seinen Entwicklern unter Berücksichtigung von Modularität, Austauschbarkeit von Elementen, Erweiterbarkeit und Anpassungsmöglichkeiten durch den Anwender designt. Als Folge davon ist Hibernate verdienterweise in den wenigen Jahren seit seiner Erstveröffentlichung rasch zu einer der führenden ORM-Technologien für Enterprise-Entwickler geworden.

Dieses Buch bietet einen umfassenden Überblick über Hibernate. Es stellt vor, wie man mit den Möglichkeiten und Mechanismen zum Mapping von Typen für das Modellieren von Assoziationen und Vererbung arbeitet, wie man Objekte mit der Hibernate-Abfragesprache effizient ausliest, wie man Hibernate zur Verwendung sowohl in gemanagten als auch nicht gemanagten Umgebungen konfiguriert und wie man die mitgelieferten Tools nutzt. Zusätzlich kommentieren die Autoren im Verlauf des Buches die zugrunde liegenden Probleme von ORM und die Entscheidungen für das Design von Hibernate. Durch diese Erkenntnisse bekommt der Leser ein weit reichendes Verständnis der effektiven Nutzung von ORM als Enterprise-Technologie.

Hibernate in Action ist der definitive Leitfaden für die Arbeit mit Hibernate und dem objekt-relationalen Mapping im heutigen Enterprise Computing.

Linda DeMichiel

Lead Architect, Enterprise JavaBeans

Sun Microsystems



Vorwort zur zweiten Auflage

Der Vorgänger dieses Buches, *Hibernate in Action*, begann mit einem Zitat von Anthony Berglas: „Nur weil man Zweige auch mit der Nase über den Boden schieben kann, heißt das nicht, dass das der beste Weg ist, um Feuerholz zu sammeln.“ Seitdem hat sich das Hibernate-Projekt weiterentwickelt und die Strategien und Konzepte, mit denen Entwickler an der Verwaltung von Informationen arbeiten, sind ausgefeilter geworden. Doch die fundamentalen Probleme sind weiterhin die gleichen: Jedes Unternehmen, mit dem wir täglich zusammenarbeiten, nutzt immer noch SQL-Datenbanken, und Java ist als erste Wahl für die Entwicklung von Enterprise-Applikationen in dieser Branche tief verwurzelt.

Doch die tabellarische Repräsentation der Daten in einem relationalen System unterscheidet sich immer noch fundamental von dem Netzwerk von Objekten in objektorientierten Java-Applikationen. Wir haben weiterhin mit der objekt-relationalen Unvereinbarkeit zu tun und müssen häufig die Erfahrung machen, dass Bedeutung und Kosten dieser Unvereinbarkeit unterschätzt werden.

Andererseits stehen uns nun eine Reihe von Tools und Lösungen zur Verfügung, um dieses Problem anzugehen. Wir sind mit dem Sammeln von Feuerholz fertig, und das Feuerzeug haben wir durch einen Flammenwerfer ersetzt!

Hibernate gibt es nun als drittes Major Release, und in diesem Buch beschreiben wir Hibernate 3.2. Verglichen mit älteren Hibernate-Versionen enthält das neue Major Release doppelt so viele Features – und dieses Buch ist beinahe doppelt so dick wie *Hibernate in Action*. Bei den meisten dieser Features geht es um solche, nach denen die Leser als Entwickler, die täglich mit Hibernate zu tun haben, gefragt haben. Wir haben so manches Mal gesagt, dass Hibernate die Lösung für 90 Prozent aller Probleme ist, mit denen ein Entwickler von Java-Applikationen fertig werden muss, wenn er eine Datenbank-Applikation erstellt. Durch die neueste Hibernate-Version geht dieser Wert in Richtung 99 Prozent.

Hibernate ist immer ausgereifter geworden, und die Anwendergruppe und die Community wachsen ständig. Vor diesem Hintergrund wurden die Java-Standards für Datenmanagement und die Entwicklung von Datenbank-Applikationen von vielen Entwicklern als unzureichend erachtet. Wir haben Ihnen in *Hibernate in Action* sogar geraten, nicht mit EJB 2.x Entity Beans zu arbeiten.

Und nun kommt EJB 3.0 und der neue Java Persistence Standard! Dieser neue Industriestandard ist für die Community der Java-Entwickler ein großer Schritt nach vorne. Damit werden ein leichtgewichtiges und vereinfachtes Programmiermodell und eine leistungsfähige objekt-relationale Persistenz definiert. Viele der Schlüsselkonzepte des neuen Standards wurden im Hinblick auf Hibernate und andere erfolgreiche Lösungen für die objekt-relationale Persistenz gestaltet. Die neueste Hibernate-Version implementiert den Java Persistence Standard.

Somit können Sie Hibernate nun sowohl als Allzweckwaffe und auch wie jeden anderen Java Persistence Provider mit oder ohne andere EJB 3.0-Komponenten und Java EE 5.0-Dienste einsetzen. Durch diese tiefe Integration von Hibernate mit einem solch reichhaltigen Programmiermodell können Sie bei Applikationen solche Funktionalitäten designen und implementieren, die vorher von Hand schwierig zu schaffen waren.

Wir haben dieses Buch geschrieben, damit Sie sowohl für Hibernate als auch Java Persistence (und allen anderen relevanten EJB 3.0-Konzepten) einen vollständigen und korrekten Leitfaden bekommen. Wir hoffen, dass Sie Freude daran haben, mit Hibernate zu arbeiten und dass diese Referenz-Bibel einen Stammpplatz auf Ihrem Schreibtisch bekommt.



Vorwort

*Nur weil man Zweige auch mit der Nase über den Boden schieben kann,
heißt das nicht, dass das der beste Weg ist, um Feuerholz zu sammeln.*

Anthony Berglas

Heute arbeiten viele Software-Entwickler mit Enterprise Information Systems (EIS). Diese Art von Applikation erstellt, verwaltet und speichert strukturierte Informationen und verteilt diese zwischen vielen Anwendern, die sich an unterschiedlichen Standorten befinden.

Zum Speichern von EIS-Daten gehört der intensive Einsatz von auf SQL basierenden Datenbankmanagementsystemen (DBMS). Jedes Unternehmen, mit dem wir im Laufe unseres Berufslebens zu tun hatten, arbeitet mit wenigstens einer SQL-Datenbank; die meisten Firmen sind in ihrem Business komplett von relationaler Datenbanktechnologie abhängig.

In den vergangenen fünf Jahren hat die breite Akzeptanz der Programmiersprache Java in der Softwareentwicklung zur Vorherrschaft des objektorientierten Modells geführt. Die Entwickler sind von den Vorteilen der Objektorientierung nun begeistert. Doch der Großteil der Firmen hat sich ebenfalls langfristig an teure relationale Datenbanksysteme gebunden. Dabei geht es nicht nur um die tiefe Verwurzelung bestimmter Produkte von Datenbankherstellern, sondern die funkelneuen objektorientierten Webapplikationen müssen auch mit den vorhandenen Legacy-Daten arbeiten können.

Allerdings unterscheidet sich die tabellarische Repräsentation der Daten in einem relationalen System fundamental vom Netzwerk der Objekte in objektorientierten Java-Applikationen. Dieser Unterschied hat zur sogenannten *objekt-relationalen Paradigmenunvereinbarkeit* geführt.

Traditionell werden die Bedeutung und die Kosten dieser Unvereinbarkeit unterschätzt, und die Tools für die Behebung dieser Unvereinbarkeit sind nicht zufrieden stellend. Nun geben Java-Entwickler der relationalen Technologie die Schuld an dieser Unvereinbarkeit, während Daten-Profis sie der Objekt-Technologie anlasten.

Mit dem Begriff *objekt-relationales Mapping* (ORM) werden die automatisierten Lösungen dieses Problems bezeichnet. Für Entwickler, die des lästigen Data Access Codes müde

sind, lautet die gute Nachricht, dass ORM nun ausgereift ist. Von den mit ORM-Middleware erstellten Applikationen kann man erwarten, dass sie kostengünstiger sind, eine bessere Performance haben und weniger herstellerspezifisch sind. Sie können besser mit Änderungen am internen Objektschema oder dem zugrunde liegenden SQL-Schema umgehen. Das Erstaunliche daran ist, dass all diese Vorteile für Java-Entwickler nun kostenfrei zur Verfügung stehen.

Gavin King begann Ende 2001 mit der Entwicklung von Hibernate, als er merkte, dass die damals beliebte Persistenzlösung CMP Entity Beans bei nicht-trivialen Applikationen mit komplexen Datenmodellen nicht skalieren konnte. Hibernate erblickte als unabhängiges, nicht-kommerzielles Open-Source-Projekt das Licht der Welt.

Das Hibernate-Team (einschließlich der Autoren dieses Buches) hat ORM auf die harte Tour erlernen müssen – das heißt, durch den Kontakt zu den Anwendern und der Implementierung dessen, was durch sie angefragt wurde. Als Resultat ist Hibernate eine praktische Lösung, betont die Produktivität der Entwickler und beansprucht die technische Vorherrschaft. Hibernate ist von Zehntausenden von Anwendern in Tausenden von produktiven Applikationen eingesetzt worden.

Als dem Hibernate-Team zeitlich immer mehr abverlangt wurde, wurde beschlossen, dass es für den zukünftigen Erfolg des Projekts (und die Bewahrung der geistigen Gesundheit von Gavin) nötig ist, dass sich professionelle Entwickler in Vollzeit mit Hibernate beschäftigen. Hibernate gehörte seit Ende 2003 zu jboss.org und hat nun einen kommerziellen Aspekt: Sie können von JBoss Inc. kommerziellen Support und Training erwerben. Doch ein kommerzielles Training sollte nicht der einzige Weg bleiben, Hibernate zu erlernen.

Es ist offensichtlich, dass viele, vielleicht sogar die meisten Java-Projekte vom Einsatz einer ORM-Lösung wie Hibernate profitieren – obwohl das vor ein paar Jahren vielleicht noch nicht so offensichtlich war! Im dem Maße, wie die ORM-Technologie immer mehr zum Mainstream wird, reicht die Produkt-Dokumentation wie das kostenlose User Manual von Hibernate einfach nicht mehr aus. Wir haben erkannt, dass die Hibernate-Community und neue Hibernate-Anwender ein umfassendes Buch brauchen – nicht nur, um die Entwicklung von Software mit Hibernate zu erlernen, sondern auch, um die objekt-relationale Unvereinbarkeit und die Beweggründe hinter dem Design von Hibernate verstehen und einschätzen zu können.

Das Buch, das Sie nun in Händen halten, hat enorme Mühen beansprucht und uns über ein Jahr lang den Löwenanteil unserer Freizeit gekostet. Es war auch Quelle vieler hitziger Debatten und Lernerfahrungen. Wir hoffen, dass dieses Buch eine ausgezeichnete Anleitung für Hibernate ist (eine „Hibernate-Bibel“, wie ein Rezensent es genannt hat) und auch die erste umfassende Dokumentation der objekt-relationalen Unvereinbarkeit und ORM im Allgemeinen. Wir hoffen, dass Sie es hilfreich finden und dass Sie an der Arbeit mit Hibernate Freude haben.



Danksagungen

Dieses Buch wuchs von einer kleinen zweiten Auflage von „Hibernate in Action“ zu einem Band mit beachtlichem Umfang. Wir hätten das nicht geschafft, hätten uns nicht viele Leute geholfen.

Emmanuel Bernard hat als technischer Reviewer eine großartige Arbeit geleistet; wir danken ihm für die vielen Stunden, die er damit zugebracht hat, unsere Code-Beispiele zu editieren. Wir danken auch unseren anderen Reviewern: Patrick Dennis, Jon Skeet, Awais Bajwa, Dan Dobrin, Deiveehan Nallazhagappan, Ryan Daigle, Stuart Caborn, Patrick Peak, TVS Murthy, Bill Fly, David Walend, Dave Dribin, Anjan Bacchu, Gary Udstrand und Srinivas Nallapati. Ein spezieller Dank geht an Linda DeMichiel, die wie schon für die erste Auflage wieder ein Vorwort für unser Buch geschrieben hat.

Marjan Bace hat bei Manning, wo die englische Original-Ausgabe erschienen ist, wieder ein großartiges Produktionsteam zusammengestellt. Sydney Jones editierte unser rohes Manuskript und verwandelte es in ein echtes Buch. Tiffany Taylor, Elizabeth Martin und Andy Carroll fanden all unsere Tippfehler und machten das Buch lesbar. Dottie Marsico war verantwortlich für das Layout und sorgte dafür, dass das Buch toll aussieht. Mary Piergies koordinierte und organisierte den Produktionsprozess. Wir danken Euch allen für die Zusammenarbeit.



Über dieses Buch

Wir haben uns drei Dinge vorgenommen, als wir dieses Buch geschrieben haben. Sie können es daher wie folgt nutzen:

1. Als Tutorial für Hibernate, Java Persistence und EJB 3.0, das Sie durch die ersten Schritte mit diesen Lösungen führt.
2. Als Anleitung für alle grundlegenden und fortgeschrittenen Hibernate-Features für objekt-relationale Mapping (ORM), Objektverarbeitung, Abfragen (Querying), Performance-Optimierung und Applikationsdesign.
3. Als Referenz für eine vollständige und technisch genaue Definition der Funktionalitäten von Hibernate und Java Persistence.

Normalerweise sind Bücher entweder Tutorials oder Referenzbücher. Von daher hat diese Erweiterung auch ihren Preis. Wenn Hibernate für Sie neu ist, dann schlagen wir vor, dass Sie das Buch von vorne anfangen und die Tutorials in den Kapiteln 1 und 2 durcharbeiten. Wenn Sie schon mit älteren Versionen von Hibernate gearbeitet haben, sollten Sie die ersten beiden Kapitel kurz anlesen, um Ihr Gedächtnis aufzufrischen und einen Überblick zu bekommen, und dann gleich zum Kapitel 3 springen.

Wir werden Sie an entsprechender Stelle darauf hinweisen, wenn ein bestimmter Abschnitt oder ein Thema optional ist oder als Referenzmaterial dient, damit Sie diesen beim ersten Lesen guten Gewissens überspringen können.

Roadmap

Dieses Buch ist in drei Hauptteile gegliedert.

In Teil 1 führen wir in das grundlegende Problem der objekt/relationalen Persistenz ein, auch als *object/relational paradigm mismatch* bekannt, und erklären die Grundlagen des objekt-relationalen Mappings. Wir gehen mit Ihnen ein praktisches Tutorial durch, damit Sie Ihre ersten Schritte mit einem Hibernate-, Java Persistence- oder EJB 3.0-Projekt machen können. Wir schauen uns das Java-Applikationsdesign für Domänenmodelle an und die Optionen für die Erstellung von Metadaten für das objekt-relationale Mapping.

In Teil 2 liegt der Schwerpunkt auf dem Mapping von Java-Klassen und -Eigenschaften zu SQL-Tabellen und -Spalten. Wir untersuchen alle grundlegenden und fortgeschrittenen Mapping-Optionen in Hibernate und Java-Persistenz mit XML-Mapping-Dateien und Java Annotations. Wir zeigen Ihnen, wie Sie mit Vererbung, Collections und komplexen Klassenzuordnungen umgehen. Zum Schluss besprechen wir die Integration mit Datenbank-schemata von Altsystemen und einige spezielle Mapping-Strategien, die besonders knifflig sind.

Teil 3 beschäftigt sich mit der Verarbeitung von Objekten und wie man mit Hibernate und Java Persistence Daten laden und speichern kann. Wir stellen die Programmierschnittstellen vor und wie Sie Anwendungen mit Transaktionen und Conversations erstellen sowie Abfragen schreiben. Später konzentrieren wir uns auf korrektes Design und Implementierung von mehrschichtigen Java-Applikationen. Wir besprechen die häufigsten Entwurfsmuster (Design-Patterns), die bei Hibernate verwendet werden, zum Beispiel die Data Access Object (DAO)- und EJB-Muster. Sie erfahren, wie Sie Ihre Hibernate-Applikation auf einfache Weise testen können und welche anderen Vorgehensweisen am besten sind, wenn Sie mit objekt-relationaler Mapping-Software arbeiten.

Zum Schluss stellen wir Ihnen in einem online verfügbaren Kapitel das JBoss Seam Framework vor, das viele Konzepte aus Hibernate weiterführt und Sie in die Lage versetzt, auf einfache Weise Web-Applikationen unter Verwendung des Conversations-Musters zu erstellen. Dieses Kapitel können Sie unter <http://downloads.hanser.de> herunterladen. Wir versprechen Ihnen, dass Sie dieses Kapitel interessant finden werden, auch wenn Sie nicht beabsichtigen, Seam einzusetzen.

Für wen ist dieses Buch gedacht?

Die Leser dieses Buches sollten über Grundkenntnisse in objektorientierter Softwareentwicklung verfügen und diese Kenntnisse praktisch genutzt haben. Um die Beispiele der Applikationen zu verstehen, sollten Sie mit der Programmiersprache Java und UML (Unified Modeling Language) vertraut sein.

Unsere Hauptzielgruppe besteht aus Java-Entwicklern, die mit SQL-basierten Datenbanksystemen arbeiten. Wir zeigen Ihnen, wie Sie durch den Einsatz von ORM die Produktivität wesentlich erhöhen können.

Wenn Sie ein Datenbankentwickler sind, kann dieses Buch als Einführung in die objekt-orientierte Softwareentwicklung dienen.

Als Datenbankadministrator werden Sie sich dafür interessieren, wie ORM sich auf die Performance auswirkt und wie Sie die Performance des SQL-Datenbank-Managementsystems und der Persistenzschicht aufeinander abstimmen können, um vorgegebene Performanceziele zu erreichen. Weil bei den meisten Java-Applikationen der Datenzugriff der Flaschenhals ist, widmet dieses Buch den Performanceproblemen besondere Aufmerksamkeit. Viele DBAs haben verständlicherweise eher ein schlechtes Gefühl bei der Vorstellung, die Performance einem Tool-generierten SQL-Code anzuvertrauen. Wir haben vor,

diese Befürchtungen zu mildern und auch die Fälle besonders zu erwähnen, bei denen Applikationen nicht mit einem von Tools verwalteten Datenzugriff arbeiten sollten. Möglicherweise sind Sie erleichtert, wenn Sie hören, dass wir ORM nicht als beste Lösung für jedes Problem betrachten.

Code-Konventionen

Dieses Buch bietet reichlich Beispiele, in denen alle Artefakte von Hibernate-Applikationen vorkommen: Java-Code, Hibernate-Konfigurationsdateien und XML-Mapping-Metadateien. Quellcode in Listings oder Text erscheint in einer festen Schriftbreite wie dieser, um ihn vom normalen Text zu unterscheiden. Weiterhin werden auch die Namen von Java-Methoden, Komponentenparametern, Objekteigenschaften sowie XML-Elementen und -Attributen in dieser Schriftart präsentiert.

Java, HTML und XML können alle *verbose* (wortreich) sein. In vielen Fällen ist der Original-Quellcode (der online verfügbar ist) neu formatiert worden. Wir haben Zeilenumbrüche eingefügt und die Einrückungen umgearbeitet, um den verfügbaren Platz der Buchseiten auszunutzen. In seltenen Fällen hat auch das nicht gereicht und die Listings enthalten Markierungen zur Zeilenfortführung. Zusätzlich sind aus den Listings auch oft Kommentare im Quellcode entfernt worden, wenn der Code im Text beschrieben wird.

Anmerkungen zum Code begleiten einige der Quellcode-Listings und heben wichtige Konzepte besonders hervor. Gelegentlich verweisen nummerierte Gliederungspunkte auf Erklärungen, die nach dem Listing folgen.

Download des Quellcodes

Hibernate ist ein Open-Source-Projekt und wird unter der Lesser GNU Public License veröffentlicht. Anweisungen zum Download von Hibernate-Paketen (als Source oder Binary) finden Sie auf der Hibernate-Website www.hibernate.org.

Der Quellcode für alle Hello World- und CaveatEmptor-Beispiele dieses Buches ist bei <http://caveatemptor.hibernate.org/> unter einer kostenlosen (BSD-ähnlichen) Lizenz erhältlich. Auf dieser Website bekommen Sie auch den Code für das CaveatEmptor-Beispiel in verschiedenen Varianten, beispielsweise mit dem Schwerpunkt auf natives Hibernate, auf Java Persistence und JBoss Seam. Sie können sich den Code für die Beispiele dieses Buches auch von der Website des Manning-Verlages herunterladen, bei dem die englische Ausgabe dieses Buches erschienen ist (www.manning.com/bauer2).

Über die Autoren

Christian Bauer ist Mitglied des Entwicklerteams von Hibernate. Er arbeitet als Trainer, Consultant und Produktmanager für Hibernate, EJB 3.0 und JBoss Seam bei JBoss, einem Bereich von Red Hat. Gemeinsam mit Gavin King hat er *Hibernate in Action* verfasst.

Gavin King ist der Gründer der Hibernate- und JBoss Seam-Projekte und Mitglied der Expertengruppe für EJB 3.0 (JSR 220). Er leitet überdies Web Beans JSR 299, die sich mit Standardisierung beschäftigt und dabei Hibernate-Konzepte, JBoss Seam und EJB 3.0 einbezieht. Gavin arbeitet als Entwicklungsleiter bei JBoss, einer Sparte von Red Hat.

Author Online

Durch den Kauf von *Java Persistence mit Hibernate* sind Sie berechtigt, kostenlos auf ein privates, englischsprachiges Webforum bei Manning Publications (dem amerikanischen Originalverlag) zuzugreifen. Dort können Sie Kommentare über das Buch abgeben, technische Fragen stellen und sich Hilfe bei den Autoren und anderen Anwendern holen. Das Forum finden Sie auf www.manning.com/bauer2. Auf dieser Seite erhalten Sie Informationen, wie Sie nach der Registrierung ins Forum kommen, welche Art von Hilfe dort verfügbar ist sowie über die Etikette des Forums.

Über das vom Manning-Verlag bereitgestellte Forum finden Leser einen Platz, der informative Gespräche der Leser untereinander und zwischen Lesern und Autoren ermöglicht. Damit ist keine besondere Verpflichtung verbunden, in welchem Maße von Seiten der Autoren mit einer Mitarbeit zu rechnen ist, zumal deren Beitrag bei *Author Online* freiwillig (und unvergütet) bleibt. Wir schlagen vor, dass Sie versuchen, den Autoren ein paar anspruchsvolle Fragen zu stellen, auf dass ihr Interesse erhalten bleibe.

Das Forum *Author Online* und die Archive der früheren Diskussionen werden auf der Website des Manning-Verlags so lange zugänglich sein, wie das Buch aufgelegt wird.