

3 Wie geht es weiter?

3.1 Wie sollte man dieses Buch lesen?

3.1.1 Zu welchem Typ Leser gehören Sie?

Sie haben nun einen ersten Eindruck von Java gewonnen und wollen Ihr Wissen vervollständigen. Natürlich können Sie das Buch einfach Seite für Seite weiterlesen und dabei die Sprache Java und den Umgang mit ihrer umfassenden Klassenbibliothek erlernen. Vielleicht wollen Sie aber gar nicht bis Kapitel 39 auf Seite 887 warten, um zu erfahren, wie man ein Applet schreibt? Vermutlich wollen Sie auch nicht bis Kapitel 50 auf Seite 1189 warten, um den Umgang mit dem Debugger kennenzulernen? Auch kann es sein, daß Sie nicht an der Dateiein-/ausgabe interessiert sind und die Kapitel 18 auf Seite 405, 19 auf Seite 427 und 20 auf Seite 445 daher zunächst überspringen wollen. Je nach Ihren Vorkenntnissen und Absichten wird eine ganz bestimmte Vorgehensweise sinnvoll sein. Wir wollen in den nächsten Abschnitten ein paar Tips zum Lesen des Buchs und einige Hinweise zum Inhalt der einzelnen Kapitel geben.

Falls Sie zu dem Leserkreis gehören, der bereits einige Erfahrungen mit der Entwicklung von Programmen in einer konventionellen Hochsprache wie Pascal oder C hat, und Sie dieses Buch vor allem lesen, um auf den Java-Zug aufzuspringen, sind Sie hier goldrichtig. Lesen Sie das Buch einfach von vorne nach hinten und lernen Sie in jedem Kapitel ein wenig dazu. Einige Dinge – insbesondere in den vorderen Kapiteln – werden Ihnen vertraut vorkommen. Aber bereits ab Kapitel 7 auf Seite 143 werden Sie mit vielen Neuigkeiten konfrontiert, die den Reiz der Sprache ausmachen. Lesen Sie das Buch in aller Ruhe, nehmen Sie sich Zeit, die Beispiele nachzuvollziehen, und machen Sie eigene Experimente. Einige der Kapitel behandeln Spezialthemen, die nicht von jedem benötigt werden, und können beim ersten Lesen übersprungen werden. Sie werden am Ende des Buchs ein sicheres Verständnis für alle grundlegenden Belange der Java-Programmierung haben und können sich leicht in komplexere Themengebiete einarbeiten.

Falls Sie bereits weitreichende Programmiererfahrung in einer objektorientierten Sprache wie C++ oder SmallTalk haben, werden Sie sich am Anfang unterfordert fühlen. Die in Kapitel 4 auf Seite 87 bis Kapitel 6 auf Seite 123 eingeführten Grundlagen kommen Ihnen bekannt vor, die ab Kapitel 7 auf Seite 143 behandelte objektorientierte Programmierung kennen Sie im Schlaf, und die in Kapitel 13 auf Seite 269 vorgestellten Techniken zur Entwicklung größerer Programme sind ebenfalls nichts Neues für Sie. Aber kennen Sie sich auch mit Multithreading, Kryptographie, dem Abstract Windowing Toolkit und den Java Foundation Classes aus? Wissen Sie, wie grafische Animationen entwickelt werden, wie

man Objekte im Netz verteilt oder auf SQL-Datenbanken zugreift? Kennen Sie Java-Konzepte wie Serialisierung, Reflection oder Beans? Wenn nicht, dann werden auch Sie in diesem Buch interessante und wissenswerte Neuigkeiten finden.

Falls Sie zu dem Leserkreis gehören, der hauptsächlich Applets entwickeln will, verspüren Sie wahrscheinlich keine Lust, viele Kapitel darauf zu warten, das erste Applet vorgesetzt zu bekommen. Natürlich müssen Sie die Grundlagen der Sprache kennen, um Applets entwickeln zu können. Es spricht aber nichts dagegen, daß Sie sich bereits während der Lektüre der Grafikgrundlagen in den Kapiteln 23 auf Seite 497 bis 29 auf Seite 597 mit der Applet-Programmierung, die in Kapitel 39 auf Seite 887 und 40 auf Seite 909 beschrieben wird, vertraut machen. Die Unterschiede zwischen Applikationen und Applets halten sich in Grenzen, und sie werden genau erklärt.

Auch wenn Sie zu dem Typ Leser gehören, der Bücher nie am Stück liest, sondern nur dann zur Hand nimmt, wenn er nach einer Lösung für ein spezielles Problem sucht, kann dieses Buch für Sie nützlich sein. Die Gliederung erleichtert auch das Wiederfinden spezieller Themen. Verwenden Sie einfach das Inhaltsverzeichnis oder den umfassenden Index, um das Thema zu finden, an dem Sie gerade besonders interessiert sind. Zögern Sie nicht, die Beispielprogramme in die Tat umzusetzen und eigene Experimente durchzuführen. Es gibt viele Leser, die auf diese Weise am besten lernen.

Wenn Sie dagegen überhaupt keine Programmiererfahrung haben, wird die Lektüre des Buchs nicht einfach werden. An vielen Stellen werden Grundkenntnisse in Datenstrukturen, Algorithmen und der Entwicklung von Computerprogrammen vorausgesetzt. Die Kapitel mit den fortgeschrittenen Themen setzen darüber hinaus ein gutes Verständnis der einführenden Kapitel voraus. Sollten Ihnen diese Kenntnisse fehlen, versuchen Sie unbedingt, sie sich anzueignen. In Abschnitt 3.2 auf Seite 73 finden Sie Hinweise auf weiterführende Dokumentationen und Online-Ressourcen, die Ihnen dabei helfen können.



Einigen Zuschriften entnahm ich, daß es den Lesern mitunter an elementaren Grundkenntnissen im Umgang mit dem Computer und seinem Betriebssystem mangelt. Die Leser dieser Gruppe sind dann gleich zu Beginn frustriert, weil etwa die Installation des JDK nicht klappt, der PATH nicht gesetzt wurde oder sie noch nie einen Editor gesehen haben. Falls Sie zu dieser Gruppe gehören, sollten Sie sich das fehlende Wissen unbedingt aneignen, sonst sind Mißerfolgsereignisse vorprogrammiert. In Abschnitt 2.3.7 auf Seite 65 werden einige Hinweise zu diesen elementaren Themen gegeben, aber ausreichend ist das nur, wenn schon gewisse Grundkenntnisse vorhanden sind.

Dieses Buch enthält an vielen Stellen Vorwärtsverweise auf Themen, die noch nicht behandelt wurden. Dadurch wird den Verflechtungen innerhalb der Themenbereiche Rechnung getragen, und man findet beim Nachschlagen schnell alle relevanten Textstellen. Wenn Sie beim Lesen auf einen Vorwärtsverweis stoßen, können Sie normalerweise zunächst warten, bis die Erklärung nachgereicht wird (oftmals schon kurze Zeit später). Sie können den Begriff natürlich auch sofort nachschlagen, aber aus didaktischen Gründen ist das meist nicht unbedingt notwendig.

3.1.2 Was ist der Inhalt der einzelnen Kapitel?

Die Kapitel 1 auf Seite 31 und 2 auf Seite 49 haben Sie bereits gelesen, Sie kennen also die wichtigsten Schlagworte zu Java und haben ein lauffähiges Entwicklungssystem. In diesem Kapitel erhalten Sie Informationen zum Inhalt des Buchs und bekommen Tips, wie Sie es am besten verwenden können. Auch Hinweise auf weiterführende Informationen finden Sie hier.

Die Kapitel 4 auf Seite 87, 5 auf Seite 107 und 6 auf Seite 123 beschäftigen sich mit den elementaren Eigenschaften der Sprache. Sie erklären die lexikalische Struktur von Java-Programmen und stellen Datentypen, Ausdrücke und Anweisungen vor. Wenn Sie bereits mit C oder C++ vertraut sind, werden Ihnen viele Dinge bekannt vorkommen. Dennoch gibt es einige elementare, aber wichtige Unterschiede, und auch als erfahrener C/C++-Programmierer sollten Sie nicht vollständig auf die Lektüre dieser Kapitel verzichten.

Kapitel 7 auf Seite 143 erklärt wichtige Grundlagen der objektorientierten Programmierung und erläutert Klassen und Objekte in Java. Auch die Definition und Verwendung von Methoden wird in diesem Kapitel behandelt. Kapitel 8 auf Seite 163 ist ebenfalls essentiell. Es beschreibt die Prinzipien der Vererbung, erläutert Attribute von Klassen und erklärt statische Membervariablen und Methoden. Schließlich werden abstrakte Klassen und Polymorphismus behandelt. Zu den wichtigsten Techniken der objektorientierten Programmierung in Java gehören Interfaces, die in Kapitel 9 auf Seite 183 beschrieben werden. Mit der Vorstellung einiger weiterführender Themen bildet Kapitel 10 auf Seite 203 den Abschluß der objektorientierten Programmierung. Während lokale und anonyme Klassen ebenso wie Wrapper-Klassen zum Handwerkszeug jedes Java-Programmierers gehören sollten, ist der ausführliche Teil über Design-Patterns optional und kann beim ersten Lesen ausgelassen werden.

Im nächsten Teil des Buchs werden weiterführende Spracheigenschaften und Grundlagen der Klassenbibliothek behandelt. Die Kapitel 11 auf Seite 245 bis 17 auf Seite 379 sind ebenso wichtig wie die Grundlagen der Sprache. Sie behandeln Strings, Exceptions und Packages, erklären die Collections und stellen die wichtigsten Utility-Klassen vor. Einzig Kapitel 15 auf Seite 321 könnte beim ersten Lesen übersprungen werden, denn dort wird

das mit dem JDK 1.2 eingeführte Collection-API vorgestellt. Die seit der Version 1.0 vorhandenen Collections sind dagegen Pflicht und werden in Kapitel 14 auf Seite 307 behandelt. Die nächsten vier Kapitel 18 auf Seite 405 bis 21 auf Seite 453 beschäftigen sich mit Dateizugriffen. Sie erläutern zeichen- und byteorientierte Streams, Random-Access-Dateien und den Zugriff auf Verzeichnisse. Den Abschluß des vierten Teils bildet Kapitel 22 auf Seite 465 mit der Behandlung des Multithreadings.

Die darauffolgenden Kapitel beschäftigen sich mit dem Abstract Windowing Toolkit und zeigen, wie man Java-Programme mit grafischer Oberfläche schreibt. Während Kapitel 23 auf Seite 497 bis 29 auf Seite 597 Grundlagen behandelt, die auch dann wichtig sind, wenn nur Swing-Programme geschrieben werden sollen, werden in den Kapiteln 30 auf Seite 621 und 32 auf Seite 675 ausschließlich AWT-Menüs und -Dialogelemente vorgestellt. Kapitel 31 auf Seite 645 erklärt den Umgang mit Layoutmanagern und ist sowohl für AWT- als auch für Swing-Programme wichtig. Den Abschluß dieses Teils bilden zwei Kapitel, in denen gezeigt wird, wie eigene Dialogelemente entwickelt werden können und wie man Bitmaps und Animationen einsetzt. Zwar sind sie weitgehend AWT-spezifisch, doch ist ihr Inhalt auch für entsprechende Swing-Programme und zum Verständnis der Java-Beans-Architektur vonnöten.

Teil 6 des Buchs befaßt sich mit der Programmierung von grafischen Oberflächen mit Swing. Während in Kapitel 35 auf Seite 749 Grundlagen erklärt und Unterschiede bzw. Parallelen zum AWT aufgezeigt werden, behandeln die nächsten drei Kapitel alle wichtigen Dialogelemente des Swing-Toolkits. Zusammen mit den im vorigen Teil vermittelten Grundlagen stehen nach Ende dieses Teils ausreichend Kenntnisse zur Entwicklung eigener Swing-Applikationen zur Verfügung.

In den nachfolgenden Kapiteln 39 auf Seite 887 und 40 auf Seite 909 wird die Applet-Programmierung erläutert. Diese Kapitel können bei Bedarf auch vorgezogen werden, insbesondere Swing-Kenntnisse sind zu ihrem Verständnis nicht nötig. Die Grundlagen der Grafikprogrammierung aus Kapitel 5 sollten Sie allerdings gelesen haben.

Die beiden letzten Teile des Buchs behandeln eine Reihe weiterführender Themen. Die Kapitel 41 auf Seite 929 und 42 auf Seite 953 erweitern die Möglichkeiten, Daten persistent zu speichern. Sie erläutern das Serialisieren von Objekten und beschreiben den Zugriff auf relationale Datenbanken. In Kapitel 43 auf Seite 989 wird gezeigt, wie mit Hilfe des Reflection-APIs zur Laufzeit auf die Interna von Klassen und Objekten zugegriffen werden kann. In Kapitel 44 auf Seite 1015 wird die zum Erstellen eigenständiger Komponenten wichtige Java-Beans-Architektur vorgestellt.

Die nächsten beiden Kapitel 45 auf Seite 1059 und 46 auf Seite 1091 beschäftigen sich mit der Netzwerkprogrammierung. Während das erste die Grundlagen vorstellt und zeigt, wie

TCP/IP-Clients und -Server geschrieben werden, stellt das zweite verteilte Objektarchitekturen mit RMI vor. Den Abschluß dieses Teils bildet Kapitel 47 auf Seite 1107 mit der Vorstellung des Sicherheitskonzepts von Java und der Beschreibung signierter Applets. Hier werden auch wichtige kryptografische Grundlagen erläutert. In Kapitel 48 auf Seite 1141 wird das Sound-API vorgestellt, und in den letzten beiden Kapiteln des Buchs werden Hinweise zur Steigerung der Performance von Java-Programmen und zum Umgang mit den Hilfsprogrammen des JDK gegeben.

3.1.3 Wie geht es nun weiter?

Daß man Bücher unterschiedlich gliedern kann, ist kein Geheimnis. Daß die Beschreibung einer Programmiersprache mehr Aufwand erfordert als die eines Mobiltelefons oder Videorecorders, ist ebenfalls bekannt. Ist es nun besser, eine umfangreiche Programmiersprache zunächst aus der Vogelperspektive zu betrachten und nur die jeweils benötigten Themen rezeptartig nachzuschlagen? Um so mit geringstem Mitteleinsatz frühzeitig lauffähige (aber nicht notwendigerweise verstandene) Ergebnisse zu produzieren? Oder sollte man zunächst die Grundlagen der Sprache studieren und erst später die komplexeren Hilfsmittel auf der Basis eines soliden Grundlagenwissens einsetzen?

Wie die Kapitelaufteilung zeigt, wurde dieses Buch unter der Annahme geschrieben, daß der zweite Ansatz der sinnvollere ist. Platt ausgedrückt, stehen die einfachen Dinge weiter vorne und die schwierigen weiter hinten, und im Zweifelsfall ist diese Leserichtung durchaus sinnvoll. Das soll allerdings keinesfalls heißen, daß unter allen Umständen und für alle Leser das sequentielle Durchlesen die richtige Vorgehensweise ist. Wie erwähnt, mag es durchaus sinnvoll sein, zu einem anderen Ablauf zu kommen. Gerade beim Einsatz im geschäftlichen Umfeld hat man durch Termin- und Projektdruck nicht immer die Zeit, die Anwendung neuer Werkzeuge mit der nötigen Ruhe zu erlernen.

Letztendlich müssen Sie selbst entscheiden, welche der Kapitel Sie durchlesen wollen und in welcher Reihenfolge Sie das tun. Die vorgegebene Kapitelaufteilung und -reihenfolge mag Ihnen dabei ein Leitfaden sein. Wählen Sie die Technik, die Ihren Neigungen und Erwartungen am meisten entgegenkommt und am besten mit Ihren Beweggründen in Einklang zu bringen ist. Dann werden Sie den größtmöglichen Nutzen aus dem Buch ziehen.

3.2 Weiterführende Informationen

3.2.1 Die Dokumentation des JDK

Die Dokumentation zum JDK befindet sich auf der beigelegten CD-ROM. Sie liegt im Verzeichnis `\install\jdk14` und kann wie in Abschnitt 2.1 auf Seite 49 beschrieben installiert werden. Zusätzlich befinden sich diverse weiterführende Informationen, Dokumenta-

tionen und Spezifikationen zu verschiedenen Aspekten der Java-Programmierung im Verzeichnis `\java` der CD-ROM. Es empfiehlt sich, die Datei `\readme.txt` zu lesen, um einen Überblick über den Inhalt der CD-ROM zu bekommen.

Wird die JDK-Dokumentation in das vorgeschlagene Installationsverzeichnis extrahiert, kann sie durch Aufruf der Datei `c:\jdk1.4\docs\index.html` mit einem HTML-Browser gelesen werden. Diese Einstiegsseite enthält Verweise auf alle anderen Teile der Dokumentation. Ein Großteil von ihnen wird zusammen mit dem JDK ausgeliefert, es gibt aber auch Online-Ressourcen, die auf dem JavaSoft-Server liegen. Die folgenden Hauptthemen sind als Verweise am oberen Rand der Startseite angeordnet und können direkt angesprungen werden:

Tabelle 3.1:
Inhalt der JDK-
Dokumentation

Name	Beschreibung
General Info	Allgemeine Infos zur Installation des JDK, Lizenzbestimmungen, Versionen, Fehlerhinweise
API & Language	Verweis auf die API-Dokumentation des JDK
Guide To Features	Überblick über alle großen Pakete des JDK, mit Hinweisen zur Architektur, Erläuterungen und teilweise Spezifikationen
Tool Docs	Dokumentation der Hilfsprogramme des JDK
J2RE & Plug-In	Verweis auf die Online-Dokumentation zum JRE und zum Java-PlugIn (siehe Abschnitt 40.4 auf Seite 922)
Demos/Tutorials	Aufruf der beigefügten Demo-Applets sowie Verweis auf eine Vielzahl von Online-Ressourcen zu Java

Die bei der täglichen Arbeit wichtigste Dokumentation ist die API-Dokumentation des JDK. Sie kann über den Link »API & Language« oder durch direkten Aufruf der Datei `c:\jdk1.4\docs\api\index.html` gestartet werden. Als *API* bezeichnet man das *Application Programming Interface*, also die Programmierschnittstelle einer Klasse, eines Pakets oder einer ganzen Bibliothek. Die API-Dokumentation des JDK gibt detaillierte Auskunft zu allen öffentlichen Paketen, Klassen, Methoden und Variablen. Sie wurde von den JDK-Entwicklern mit `javadoc` generiert und sieht seit dem JDK 1.2 etwa so aus.

Die drei Fenster haben folgende Bedeutung:

- ▶ Im linken oberen Fenster findet sich eine Liste aller Pakete des JDK. Ein Klick auf eines der Pakete stellt die Liste der zugehörigen Klassen im linken unteren Fenster dar.
- ▶ Das linke untere Fenster zeigt alle Klassen, Interfaces und Exceptions des aktuellen Pakets. Wird ein Element angeklickt, so erscheint auf der rechten Seite seine Beschreibung.

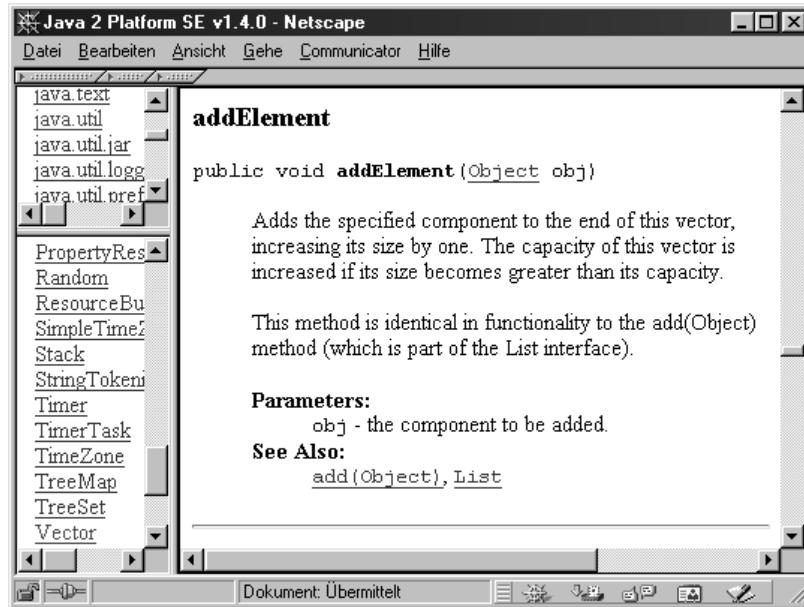


Abbildung 3.1: Die API-Dokumentation des JDK

- Auf der rechten Seite wird eine einzelne Klasse oder ein Interface beschrieben. Am oberen Rand werden Vererbungsinformationen angezeigt, darunter folgt die allgemeine Klassenbeschreibung. Anschließend kommt eine Linkeiste für Konstanten, Variablen, Konstruktoren und Methoden. Ein Klick auf eines dieser Elemente verzweigt zu dessen detaillierter Beschreibung. Abbildung 3.1 zeigt die Beschreibung der Methode addElement der Klasse Vector des Pakets java.util.

Die API-Dokumentation ersetzt zwar nicht die konzeptionelle Beschreibung der Java-Themen (das ist Aufgabe dieses Buchs), als Nachschlagewerk zu Details der Klassenbibliothek ist sie jedoch unentbehrlich. Der Umgang mit ihr sollte jedem Java-Entwickler in Fleisch und Blut übergehen. Manchmal bieten die Hilfesysteme der integrierten Entwicklungsumgebungen sogar noch komfortablere Möglichkeiten, auf die Dokumentation von Klassen und Methoden zuzugreifen.

Durch die alphabetische Anordnung der Pakete muß im linken oberen Fenster sehr häufig gescrollt werden. Einfacher wird es, wenn die wichtigsten und am häufigsten benötigten Pakete in der Liste nach oben verschoben werden. Wer rudimentäre HTML-Kenntnisse hat, kann dazu die Datei `c:\jdk1.4\docs\api\overview-frame.html` mit einem Texteditor öffnen und die gewünschten Zeilen editieren. Nützlich wäre es beispielsweise, die Pakete `java.lang`, `java.io`, `java.util`, `java.awt`, `java.awt.event`, `javax.swing`, `java.sql` und `java.net` an den Anfang der Liste zu setzen.



3.2.2 Informationen im Internet

Java ist die Sprache des Internets, und folglich gibt es unzählige Ressourcen im Internet, die sich in der einen oder anderen Weise mit Java beschäftigen. Leider veralten viele der Adressen fast ebenso schnell, wie sie erschienen sind, und ein Buch ist daher nur bedingt geeignet, sie aufzuzählen. Wir wollen uns auf einige der wichtigsten Adressen beschränken, die bei der Entwicklung von Java-Programmen nützlich sein können.

Usenet

Die offiziellen Usenet-Newsgroups zu Java beginnen mit dem Namen `comp.lang.java`. Hier gibt es eine ganze Reihe von Untergruppen zu speziellen Themen. Leider ist die Abgrenzung zwischen den einzelnen Untergruppen nicht immer klar, und es kommt regelmäßig zu Überschneidungen und Crosspostings. Tabelle 3.2 auf Seite 76 listet die Gruppen der `comp.lang.java`-Hierarchie auf.

Tabelle 3.2:
Die `comp.lang.java`-Hierarchie
im Usenet

Newsgroup	Inhalt
<code>news:comp.lang.java.3d</code>	Diskussionen über das Java-3D-API (Homepage auf http://www.j3d.org/)
<code>news:comp.lang.java.advocacy</code>	Allgemeine Diskussionen über Java
<code>news:comp.lang.java.announce</code>	Moderierte Newsgroup mit Ankündigungen und Vorstellungen von Neuentwicklungen. Wird kaum noch verwendet.
<code>news:comp.lang.java.api</code>	Das Application Programming Interface und die Klassenbibliothek. Die Gruppe ist veraltet und sollte nicht mehr verwendet werden.
<code>news:comp.lang.java.beans</code>	Die Komponentenarchitektur Beans
<code>news:comp.lang.java.corba</code>	Java, CORBA und Objektverteilung im Netz
<code>news:comp.lang.java.databases</code>	Datenbankprogrammierung mit JDBC. Die kurze Zeit vorhandene Gruppe <code>comp.lang.java.database</code> wird nicht verwendet.
<code>news:comp.lang.java.gui</code>	Programmierung von grafischen Oberflächen und Diskussion von GUI-Buildern
<code>news:comp.lang.java.help</code>	Allgemeine Quelle für Fragen aller Art, von der Installation bis zu Programmierproblemen
<code>news:comp.lang.java.machine</code>	Diskussionen um VMs und alles, was sich unterhalb der Sprachebene abspielt. Ersetzt die Gruppe <code>comp.lang.java.tech</code> .
<code>news:comp.lang.java.misc</code>	Veraltete Gruppe mit Diskussionen zu unterschiedlichen Themen. Sollte eigentlich nicht mehr verwendet werden.
<code>news:comp.lang.java.programmer</code>	Stark frequentierte Newsgroup zu allen möglichen Aspekten der Java-Programmierung
<code>news:comp.lang.java.security</code>	Diskussion von Sicherheitsaspekten
<code>news:comp.lang.java.setup</code>	Diskussion von Installationsaspekten. Ist veraltet und sollte durch <code>comp.lang.java.help</code> ersetzt werden.

Newsgroup	Inhalt
news:comp.lang.java.softwaretools	Diskussionen zu Tools, Werkzeugen und Entwicklungsumgebungen rund um Java
news:comp.lang.java.tech	Veraltete Gruppe zu technischen Fragestellungen. Wurde durch news:comp.lang.java.machine ersetzt.
news:comp.lang.javascript	Hier dreht sich alles um die Script-Sprache JavaScript. Diese Gruppe hat daher keinen direkten Bezug zu Java, soll aber der Vollständigkeit halber erwähnt werden.
news:de.comp.lang.java	Es gibt auch eine mittlerweile sehr stark frequentierte deutsche Newsgroup, in der alle Aspekte von Java diskutiert werden. Diese besitzt zudem eine Homepage http://www.dclj.de , auf der wichtige Dokumente direkt zur Verfügung gestellt werden.

Tabelle 3.2:
Die comp.lang.java-Hierarchie im Usenet (Forts.)

Meta-Ressourcen

Unter <http://java.sun.com/> oder <http://www.javasoft.com/> finden Sie den Java-Server von SUN bzw. SUNs JavaSoft Division. Hier sind Informationen aus erster Hand von den Entwicklern der Sprache zu finden. Dieser Server ist die erste Adresse, wenn es um Neuigkeiten, aktuelle Entwicklungen und Dokumentationen geht. Hier gibt es auch Links zu weiteren Meta-Ressourcen, die hier nicht erwähnt werden. Ein direkter Link auf die von SUN für Java zur Verfügung gestellten Entwicklungsumgebungen ist <http://java.sun.com/products/>. Unter der Adresse <http://java.sun.com/j2se/1.4/> gibt es alles rund um die aktuelle Java-Version 1.4.

Eine wichtige Adresse für Entwickler ist auch die der *Java Developer's Connection (JDC)* unter <http://developer.java.sun.com/>. Diese Seiten werden von SUN gepflegt, um eine zentrale Anlaufstelle für Java-Entwickler zur Verfügung zu stellen. Es gibt dort Diskussionsforen, Schulungsangebote, weitere Software und jede Menge nützliche Informationen. Wichtiges »Organ« der JDC ist der *JDC-Newsletter*. Dabei handelt es sich um einen Newsletter, der per E-Mail regelmäßig über aktuelle Neuerungen informiert. Der Zutritt zur JDC ist kostenlos, erfordert aber das Ausfüllen einer Registrierungsseite.

Mitunter ebenfalls wichtig ist die – etwas euphemistisch als *Bug Parade* bezeichnete – Fehlerdatenbank des Java Development Kit. Hier werden alle bekannten Fehler gelistet und mit Beschreibung, Behebungs-Status und möglichen Workarounds beschrieben. Die Bug Parade kann unter <http://developer.java.sun.com/developer/bugParade/index.jshtml> erreicht und online nach Fehlern durchsucht werden. Registrierte Entwickler können neue Fehler eintragen oder zu bekannten Fehlern ihre Stimme abgeben – in der Hoffnung, dadurch die Behebung zu beschleunigen.

Auch in den großen Web-Verzeichnissen gibt es meist eigene Rubriken für die Programmiersprache Java. Yahoo stellt diese beispielsweise unter http://www.yahoo.com/Computers_and_Internet/Programming_Languages/Java zur Verfügung, und bei Google lautet die Adresse <http://directory.google.com/Top/Computers/Programming/Languages/Java/>.

In der Anfangszeit der Java-Entwicklung gab es eine ganze Reihe von Sites, die Unmengen an freien Java-Tools, -Applets und -Programmen oder frei zugänglichen Quellcode anboten. Viele von ihnen sind mittlerweile verschwunden, in einem anderen Dienst aufgegangen oder wurden kommerzialisiert. Einige Anlaufstellen sind <http://www.componentsource.com/>, <http://www.jguru.com/>, <http://www.sunsource.net/>, <http://www.jars.com/> oder das von SUN verwaltete Verzeichnis von Java-Lösungen <http://industry.java.sun.com/solutions/>. Frei zugängliche Java-Software und -Projekte gibt es unter anderem auf <http://www.gnu.org/software/software.html#Java>, <http://jakarta.apache.org/>, <http://sourceforge.net/> oder <http://freshmeat.net/>.

JavaLobby ist ein Zusammenschluß von Java-Enthusiasten, die das Ziel verfolgen, die Sprache zu verbreiten und für ein »100 % Pure Java« einzutreten. Die Homepage unter <http://www.javalobby.org/> bietet auch eine ganze Menge Verweise zu Java-Ressourcen und interessante Artikel rund um Java. Unter der Adresse <http://www.apl.jhu.edu/~hall/java/> verwaltet Marty Hall von der Johns Hopkins University eine interessante Liste von Java-Ressourcen mit Links zu FAQs, weiteren Dokumentationen, Beispielanwendungen, Entwicklungsumgebungen, Klassenbibliotheken und vielem anderen mehr.

FAQs

Eine Liste von Java-FAQs gibt es unter <http://www.faqs.org/faqs/computer-lang/java/>. Dort wird auch auf das sehr umfangreiche, aber nicht mehr ganz aktuelle FAQ von Peter van der Linden verwiesen, das unter <http://www.afu.com/javafaq.html> gefunden werden kann. Eine weitere Liste von Java-FAQs gibt es unter http://www.links2go.com/topic/java_faqs.

Von Roedy Green gibt es unter <http://mindprod.com/jgloss.html> ein Glossar, in dem viele Begriffe und Konzepte rund um Java erläutert werden.

Von SUN selbst gibt es ebenfalls ein FAQ, das unter <http://www.javasoft.com/products/jdk/faq.html> zu finden ist. Dort sind auch einige Metainformationen und firmenbezogene Informationen über Java zu finden.

Einige FAQs zur deutschen Java-Newsgroup sind unter <http://www.dclj.de/faq.html> zu finden.

Online-Magazine und Dokumentationen

Unter <http://www.sys-con.com/java/> ist die Onlineversion des *Java Developer's Journal* zu finden. Unter <http://www.javaworld.com/> findet sich die *Java World*, und die Adresse für den *Java Report Online* ist <http://www.javareport.com/>. Auch das nicht sprachgebundene *Dr. Dobb's Journal* hat eine Java-Rubrik unter <http://www.ddj.com/topics/java/>. Das auch in deutscher Sprache erhältliche *Java Spektrum* ist unter <http://www.sigs-datacom.de/sd/publications/js/index.htm> zu finden. Online steht es allerdings nur in Auszügen zur Verfügung.

Auf dem SUN-Server gibt es weitere Dokumentationen zu Java. Auf <http://java.sun.com/docs/books/> wird die *Java Series* vorgestellt, in der SUN zusammen mit Addison-Wesley eine große Zahl von Java-Büchern publiziert hat. Unter <http://java.sun.com/docs/books/jls/index.html> ist die Sprachspezifikation zu finden, und die Beschreibung der virtuellen Maschine findet sich unter <http://java.sun.com/docs/books/vmspec/index.html>.

3.2.3 Die HTML-Ausgabe

Beschreibung

Auf der CD-ROM befindet sich im Verzeichnis `\html` die HTML-Ausgabe des Buchs. Alternativ kann sie auch von <http://www.javabuch.de> oder <http://www.gkrueger.com> heruntergeladen werden. Sie enthält den kompletten Buchtext und eignet sich mit ihren Querweisen und Navigationshilfen ausgezeichnet als Nachschlagewerk. Die HTML-Ausgabe kann direkt von der CD-ROM aufgerufen oder lokal installiert werden. Beides ist in der beigefügten Dokumentation beschrieben.

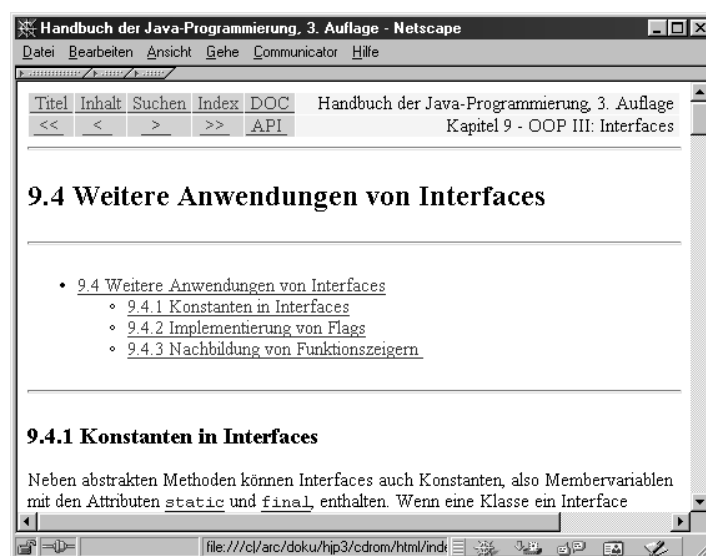


Abbildung 3.2:
Die HTML-
Ausgabe des
Buchs

Die HTML-Ausgabe sollte mit den gängigen aktuellen Browsern gelesen werden können. Getestet wurde sie mit den 4er und 6er Versionen des Netscape Navigator, mit Internet Explorer 4, 5 und 6 und mit Opera 3.5 (wegen fehlender JavaScript-Unterstützung und leicht abweichender Tabellenformatierung gibt es hier einige Einschränkungen). Die im Text verwendeten Farben wurden Web-konform gewählt und sollten auch auf LCD-Bildschirmen und 256-Farben-Displays gut lesbar sein. Als sinnvolle Mindestauflösung kann 800 * 600 Pixel angesehen werden, wenn die Schriftgröße im Browser nicht zu groß eingestellt ist.

Navigation mit der Maus

Es gibt eine Vielzahl von Navigationshilfen:

- ▶ Alle Seiten enthalten am oberen und unteren Ende eine Navigationsleiste, mit der folgende Seiten direkt angesprungen werden können:
 - "Titel": Titelseite
 - "Inhalt": Globales Inhaltsverzeichnis
 - "Suchen": Suchfunktion
 - "Index": Index
 - "DOC": Die Hauptseite der JDK-Dokumentation
 - "<<": Voriges Kapitel
 - "<": Voriger Abschnitt
 - ">": Nächster Abschnitt
 - ">>": Nächstes Kapitel
 - "API": Die API-Dokumentation zum JDK
- ▶ Das Inhaltsverzeichnis paßt normalerweise auf eine Seite und kann ohne Scrollen bedient werden.
- ▶ Alle Java-Bezeichner, Klassen-, Interface- und Methodennamen sind als Links realisiert und führen direkt in den Index.
- ▶ Internet-Links führen direkt zu der abgedruckten Ressource.
- ▶ Syntaxdiagramme haben auf der rechten Seite einen Link, der direkt auf die API-Dokumentation der betreffenden Klasse führt.
- ▶ Listings haben auf der rechten Seite einen Link, der direkt zu der abgedruckten Quell-datei führt. Bei Bedarf kann diese durch Aufruf des Browser-Menüs »Speichern unter« (meist durch `STRG` + `S`) zu erreichen) direkt als Datei gespeichert werden.

- ▶ Am Anfang jedes Kapitels und Abschnitts findet sich ein lokales Inhaltsverzeichnis.
- ▶ Verweise auf Abbildungen, Listings und Tabellen können als Links direkt angesprungen werden.
- ▶ Der Index enthält neben den Textverweisen bei Klassen-, Interface- und Methodennamen auch einen Verweis auf die zugehörige API-Dokumentation.
- ▶ Die Hauptseite des Index enthält nicht nur einen einzelnen Verweis auf den Anfang jedes Indexbuchstabens, sondern – je nach Anzahl der Einträge – auch auf Unterteilungen davon. Diese erleichtern das Auffinden von Fundstellen insbesondere bei Buchstaben wie »G« oder »S« mit Hunderten von Einträgen.

Navigation über die Tastatur

Es gibt eine limitierte Form der Tastaturbedienung, mit der wichtige Seiten ohne Zuhilfenahme der Maus angesprungen werden können. Auf den meisten Seiten stehen folgende Tastaturbefehle zur Verfügung:

- ▶ **[T]**: Titelseite
- ▶ **[Z]**: Globales Inhaltsverzeichnis
- ▶ **[S]**: Suchfunktion
- ▶ **[I]**: Index
- ▶ **[P]**: API-Dokumentation
- ▶ **[D]**: JDK-Dokumentation
- ▶ **[H]**: Voriges Kapitel
- ▶ **[J]**: Voriger Abschnitt
- ▶ **[K]**: Nächster Abschnitt
- ▶ **[L]**: Nächstes Kapitel
- ▶ **[A]**: Seitenanfang
- ▶ **[E]**: Seitenende

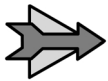
Auf der Hauptseite des Index kann der gewünschte Indexbuchstabe auch über die Tastatur eingegeben werden. Die zuvor beschriebenen Kürzel sind auf dieser Seite außer Kraft.

Einsatz von JavaScript

Die HTML-Ausgabe enthält hauptsächlich HTML-3.2-Code. Cascading Style Sheets oder ähnliche Erweiterungen wurden nicht verwendet. Ein Java-Applet wird nur für die Such-

funktion verwendet, und der Einsatz von JavaScript wurde so gering wie möglich gehalten. Die HTML-Ausgabe ist auch verwendbar, wenn JavaScript im Browser deaktiviert ist oder nicht unterstützt wird. In diesem Fall gibt es einige kleine Einschränkungen:

- ▶ Die Tastaturbedienung ist außer Kraft.
- ▶ Die eingebetteten Verweise auf die JDK- und API-Dokumentation lassen sich nicht konfigurieren. Sie funktionieren nur dann korrekt, wenn die JDK-Dokumentation im Unterverzeichnis `jdkdocs\` und die API-Dokumentation im Unterverzeichnis `jdkdocs\api\` innerhalb des Installationsverzeichnisses liegt.



Ist JavaScript aktiviert, kann die JDK-Dokumentation an einer beliebigen Stelle liegen. Damit die Links der HTML-Ausgabe korrekt funktionieren, muß in diesem Fall an der im Unterverzeichnis `html` liegenden JavaScript-Datei `hjp3lib.js` eine kleine Änderung vorgenommen werden. In den Variablen `jdkdocs` und `apidocs` in den Zeilen 27 und 28 muß nämlich der JDK-Dokumentationspfad korrekt gesetzt sein. Er ist standardmäßig auf `c:\jdk1.4\docs\` bzw. `c:\jdk1.4\docs\api\` eingestellt (passend für eine Windows-Standardinstallation) und sollte der eigenen Installation entsprechend verändert werden. Wenn alles korrekt eingestellt ist, müßten die Schaltflächen "DOC" und "API" am Anfang und Ende jeder Seite auf die Startseite der JDK- und API-Dokumentation verzweigen.

3.2.4 Die im Buch verwendete UML-Notation

Im Buch werden mitunter Grafiken verwendet, um die Beziehungen zwischen Klassen darzustellen. Wir wenden dazu eine leicht modifizierte Form von Klassendiagrammen an, wie sie auch in der *Unified Modeling Language* (kurz *UML*) verwendet werden. UML ist eine verbreitete Notation und Methodik für objektorientierte Analyse und Design. Mit ihrer Darstellung alleine könnte man leicht mehrere Bücher füllen. Wir wollen uns in diesem Buch auf die Basisnotation, die *Klassendiagramme*, beschränken.

Eine Klasse wird als graues Rechteck dargestellt, das in seinem Inneren den Namen der Klasse trägt. Mitunter hat es weitere Unterteilungen, in denen Methoden untergebracht sind, wenn diese für das Verständnis der Zusammenhänge von Bedeutung sind. Interfaces werden ebenfalls als Rechteck dargestellt (worum es sich dabei handelt, wird in Kapitel 9 auf Seite 183 erläutert), haben aber einen weißen Hintergrund. Zusätzlich wird über den Namen der Text »interface« geschrieben.

Abbildung 3.3 auf Seite 83 zeigt drei Klassen `Vector`, `String` und `MyOwnClass` und zwei Interfaces `Enumeration` und `Serializable`:

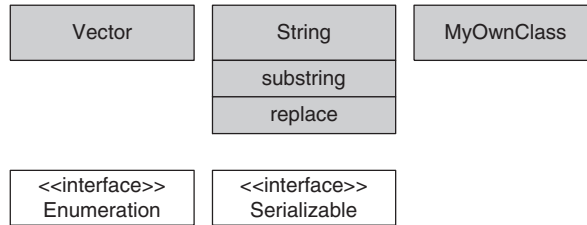


Abbildung 3.3: UML-Notation für Klassen und Interfaces

Klassen und Methoden können in Beziehungen zueinander stehen. Diese werden durch Verbindungslinien grafisch dargestellt. Bei einer Vererbungsbeziehung wird ein Pfeil von der abgeleiteten zur Basisklasse gezogen. Die Basisklasse steht in aller Regel über der abgeleiteten Klasse. Erben mehrere Klassen von einer Basisklasse, werden die Pfeile zur besseren Übersichtlichkeit zusammengefaßt. Die Implementierung eines Interfaces wird analog dargestellt, allerdings mit gestrichelten Linien.

Aggregation und Komposition wird durch eine Verbindungslinie dargestellt, die auf der Seite mit dem Container eine kleine Raute trägt. Wir unterscheiden dabei nicht zwischen den beiden Varianten. Aufrufbeziehungen werden als gestrichelte Pfeile mit Beschriftung dargestellt. Der Text beschreibt die Bedeutung des Aufrufs.

Abbildung 3.4 zeigt eine Basisklasse AbstractComponent, die das Interface Component implementiert. Aus AbstractComponent sind die drei Klassen ConcreteComponent1, ConcreteComponent2 und Container abgeleitet. Container ist Besitzer einer Sammlung von AbstractComponent-Objekten. ConcreteComponent2 verwendet die Klasse Cache:

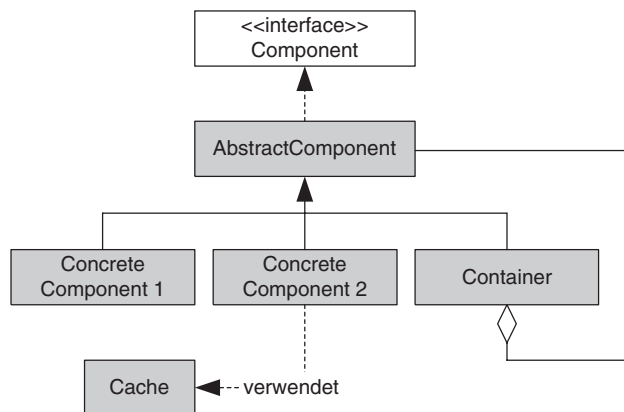


Abbildung 3.4: UML-Notation für Beziehungen zwischen Klassen und Interfaces

3.3 Zusammenfassung

In diesem Kapitel wurden folgende Themen behandelt:

- ▶ Hinweise zum Lesen des Buchs
- ▶ Übersicht über den Inhalt der einzelnen Kapitel
- ▶ Weiterführende Informationen
- ▶ Java-Ressourcen im Internet
- ▶ Die HTML-Ausgabe des Buchs
- ▶ Die im Buch verwendete UML-Notation