

Vorwort

Software ist heutzutage allgegenwärtig – sichtbar und unsichtbar. Es existiert kaum ein Bereich des Lebens, der nicht auf die ein oder andere Weise von Bit und Bytes durchdrungen ist. Angefangen bei der simplen Bezahlung im Lebensmittelgeschäft, wo softwarebasierte Kassensysteme zum Einsatz kommen, über das allgegenwärtige Smartphone hin zu WebShops und verteilten Unternehmensanwendungen zur Abbildung von Geschäftsprozessen. Mit dem Internet der Dinge (*engl. Internet of Things (IoT)*) wird sich dieser Trend verstärken und den sogenannten Digitalisierungsprozess unserer Gesellschaft weiter vorantreiben. Überall fließen Daten, die verarbeitet werden müssen, und dadurch erhöhen sich die Anforderungen an die Software. Die Folge sind immer größer und mächtiger werdende Softwaresysteme mit zunehmend komplexerem Charakter. Daraus ergeben sich neue Probleme und Herausforderungen, die es zu bewältigen gilt. Monolithische Systeme stoßen hier zunehmend an ihre Grenzen.

Um diesem Wandel in der Softwareentwicklung gerecht zu werden, hat sich in den letzten Jahren viel getan. Die agile Softwareentwicklung ist mittlerweile in vielen Unternehmen etabliert und Themen wie Configuration Management (CM) und Continuous Delivery (CD) sind gelebte Disziplinen. Und auch auf organisatorischer Ebene bringt die Stellenschaffung von DevOps als Erweiterung des agilen Entwicklungsprozesses einen großen, positiven Einfluss. Darüber hinaus zeigten der Aufstieg von Microservices und Self Contained Systems als Architekturmuster und insbesondere die Container-Technologie, dass sich die Art und Weise, wie Software entwickelt und veröffentlicht wird, ändert. Dies bringt größere Flexibilität, Produktivität und eine höhere Geschwindigkeit bei der Veröffentlichung von Software mit sich, aber auch Mehraufwände im Management und neue Anforderungen im Bereich des automatisierten Testens und Monitorings. Softwareentwicklung ist aktuell und auch in Zukunft weiterhin ein hochkomplexer Prozess, der in seiner Gesamtheit über den reinen Entwurf und die Implementierung hinausgeht.

Umso wichtiger ist die Reduzierung von Komplexität innerhalb solcher Softwaresysteme, um diese auch in den Phasen des Betriebs, der Wartung und Weiterentwicklung unter Kontrolle zu halten. Die Modularisierung, mit der sich das vorliegende Buch beschäftigt, ist dabei ein wichtiger Baustein. Und Java 9 bietet nun eine direkte Unterstützung für den Modularisierungsprozess.

In diesem Sinne viel Spaß beim Lesen des Buches und der Entwicklung von Software.

Guido Oelmann
Dortmund, 12. September 2017
www.Java-Akademie.de

Überblick über das Buch

Das Buch behandelt die Modularisierung mit Java. Neben der Darstellung des Prinzips der Modularisierung und warum dieses wichtig ist, wird konkret gezeigt, wie mit Java eigene Module erstellt werden können und wie eine Applikation basierend auf Modulen gebaut wird. Das neue modularisierte JDK wird vorgestellt und dessen Wichtigkeit für die Anwendung im Internet der Dinge. Daneben wird ein Vergleich zum bisherigen quasi Modularisierungsstandard OSGi gezogen. Es werden weitergehende Themen wie die Migration von Projekten hin zu Modulen und die Nutzung von Modulen im Kontext moderner Softwareentwicklung behandelt. Letzteres beschäftigt sich mit der Verwendung von Modulen im Umfeld von Microservices und der Container-Technologien wie Docker.

Alle Code-Beispiele und weitere Informationen finden sich unter:
<http://Java-Modularisierung.de>

Quellcodes des Buches

Für wen ist das Buch?

Das Buch richtet sich in erster Linie an Softwareentwickler, die die Modularisierung mit Java in eigenen Projekten nutzen wollen. Aber auch Softwarearchitekten können sich angesprochen fühlen, ist das Prinzip der Modularisierung doch auf einer höheren Abstraktionsstufe angesiedelt als die reine Erstellung von Klassen und Packages und erfordert bereits beim Entwurf eines Softwaresystems die entsprechende Berücksichtigung. Wer generell eine Einführung in das Thema Modularisierung sucht, wird hier ebenfalls fündig werden. Ebenso derjenige, der sich bereits mit Modularisierung auskennt, aber mehr über die Migration erfahren will oder wie sie im Kontext von z. B. Microservices und Containern Verwendung findet.

Entwickler

Architekten

Übersicht über die Kapitel

Der erste Teil des Buches ist den Grundlagen zur Modularisierung gewidmet und den Zielen, die mit dem Java-Modularisierungskonzept verfolgt werden. Es wird ausführlich erläutert, was unter Modularisierung verstanden wird, was ein Modul ist und wie Module entworfen werden. Abgeschlossen wird der erste Teil mit der Betrachtung dessen, wie Modularisierung vor Java 9 gelöst wurde und welche Probleme mit dem Modularisierungskonzept gelöst werden.

Teil I

Teil II stellt ausführlich das Java-Modulkonzept vor. Es wird gezeigt, wie mit Java Module erstellt werden und wie eine modularisier-

Teil II

te Anwendung gebaut wird. Zudem werden das modularisierte JDK und die sich daraus ergebenden neuen Möglichkeiten für Entwickler vorgestellt. Der Migration von Anwendungen hin zu Modulen ist ein weiteres Kapitel gewidmet. Zur Abrundung der Darstellung des Java-Modulsystems wird ein Vergleich zum OSGi-Modularisierungsframework gezogen. Abgeschlossen wird der zweite Teil mit konkreten Anleitungen, wie die gängigsten Entwicklungswerkzeuge bei der modularen Softwareentwicklung eingesetzt werden können, und ein »Real World«-Projekts zeigt, wie eine modulare Java-Anwendung gebaut wird.

Der letzte Teil des Buches wird den neuen Entwicklungen in der Welt der Softwareentwicklung gerecht. Nach dem Lesen dieses Teiles wird klar sein, was Microservices und Container-Technologien wie Docker mit der Modularisierung gemeinsam haben und wie Java-Module in Kombination mit diesen Technologien verwendet werden können.

Pfade durch das Buch

Um den Erwartungshaltungen der verschiedenen Lesern gerecht zu werden, gibt es auch verschiedene Möglichkeiten, sich den Inhalt des Buches zu erschließen. Während der eine möglichst schnell ohne allzu viel Theorie in die konkrete Anwendung von Modulen mit Java eingeführt werden will, möchte ein anderer vielleicht etwas mehr über die Konzepte dahinter erfahren. Ein weiterer Leser ist eventuell in OSGi sehr gut bewandert und interessiert sich für die Unterschiede und Gemeinsamkeiten zur Java-Modularisierung und was für Vorzüge ein modularisiertes JDK im Embedded-Bereich mit sich bringt. Für diese unterschiedlichen Anforderungen sind im Folgenden Leseempfehlungen bzgl. der Reihenfolge der Kapitel gegeben. Oft bietet es sich an, erst einmal das zu lesen, was am meisten interessiert, und sich danach erst mit dem Rest zu beschäftigen. Die einzelnen Kapitel sind weitestgehend autark und können leicht separat gelesen werden.

Fundierter Einstieg

Einstieg in die Modularisierung

Bei dieser Lesereihenfolge wird zunächst geklärt, was Modularisierung ist, wofür diese gut ist und was es mit dem modularisierten JDK auf sich hat. Danach werden die Modularisierung mit Java vorgestellt und die konkrete Verwendung.

Kapitel 1 → 2.1 → 2.2 → 3 → 4

Schnelleinstieg

■ Schnelleinstieg Module

Wer schon weiß, was Module sind, und sofort wissen möchte, wie Modularisierung mit Java funktioniert, ist hier gut aufgehoben.

Kapitel 3 → 4 → 10

Für den OSGi-Kenner

OSGi-Kenner

Wer OSGi kennt, wird sich die Java-Modularisierung schnell erarbeiten können. Hilfreich dabei wird die Erklärung zu den Unterschieden und Gemeinsamkeiten beider sein.

Kapitel 8 → 3 → 4

Danksagung

Mein Dank geht zunächst an meinen Lektor René Schönfeldt, der mich stets mit guten Tipps und Verbesserungsvorschlägen versorgt hat und an das gesamte dpunkt.team. Bei den mir unbekanntem Gutachtern bedanke ich mich für viele Anregungen, die letztlich für die Zusammenstellung der in diesem Buch behandelten Themen mit gesorgt haben. Und zu guter Letzt bedanke ich mich bei meiner Frau Rosanne für ihre Geduld und Unterstützung, auch wenn es mal wieder sehr spät geworden ist.