



Elisabeth  
Jung

# Android 4

Übungsbuch für die App-Entwicklung  
Aufgaben mit vollständigen Lösungen

# Einleitung

Dieses Buch ist aus der Erkenntnis entstanden, dass für Java-Programmierer das Entwickeln von Anwendungen mit Android im Schnelldurchlauf erlernt werden kann. Jeder, der Java programmieren kann, kann auch Applikationen für Android entwickeln. Das Buch soll zeigen, wie vorhandene Kenntnisse genutzt werden können, aber auch denjenigen, die parallel zu Android Java lernen möchten, eine Hilfestellung beim Erstellen von ersten eigenen Apps leisten.

Viele der Android-Lehrbücher arbeiten von vornherein mit Eclipse. Dazu muss das zugehörige Android-Plug-in installiert werden und Eclipse entsprechend konfiguriert werden. Das Entwickeln einer App beginnt in diesem Fall mit dem Anlegen eines Projekts in der Entwicklungsumgebung. Mit einem Android-Wizard, der mit dem Android-Plug-in installiert wird, wird der Programmierer schrittweise zum Anlegen des Projekts geführt.

Wir wollen den Weg ohne eine Entwicklungsumgebung gehen, um die einzelnen Schritte beim Anlegen eines Android-Projekts besser verfolgen zu können, und das Apache-Tool Ant zur Hilfe holen. Ungestört von dem gesamten Overhead, das eine Entwicklungsumgebung mit sich bringt, können damit die flexiblen und schnellen Kommandozeilen-Programme des Android-SDK benutzt werden, um effektiv arbeiten zu können. Falls Sie jedoch die Arbeit mit einer Entwicklungsumgebung bevorzugen, können selbstverständlich alle Beispiele aus diesem Buch darin eingebunden werden.

Meine Übungsbücher zu Java, auf die ich des Öfteren im Laufe der Kapitel verweise, sind eine Hilfestellung, aber durchaus keine Voraussetzung. Wie in diesen Büchern steht auch diesmal der praktische Teil an erster Stelle. Am Anfang eines Unterkapitels wird die erforderliche Theorie zusammengefasst, ohne auf alle Details in der Beschreibung von Eigenschaften zu Android-Komponenten einzugehen, diese kann man in der API-Dokumentation nachschlagen. Weil in Android viele der Eigenschaften über XML-Dateien definiert werden, gibt es eine ganze Reihe davon. Uns interessiert vielmehr, wie einzelne davon genau einzusetzen sind und im Java-Code zugänglich gemacht werden können. Wie immer werde ich mich auch mit diesem Buch nahe an der Programmiersprache bewegen und Parallelitäten von Abläufen einfach und kompakt darstellen.

Ich bringe nicht nur Beispiele zur Realisierung der wichtigsten Funktionalitäten von mobilen Geräten, die auch programmtechnisch erreicht werden können, wie

das Telefonieren und das Verschicken von E-Mails und SMS-Nachrichten, sondern beschäftige mich im Detail mit Applikations-Komponenten, wie Activities, Fragments, Intents und Content Provider, dem Zugriff auf die Ressourcen von Android-Applikationen und dem Datei- und Datenbanksystem von Android. Ich zeige, wie eigene Activity-, Fragment-, Dialog-, View-, Adapter-, Listener-, ContentProvider-, Animation-, Thread- und AsyncTask-Klassen erstellt werden können, aus einer Komponente in andere verzweigt werden kann und wie die Komponenten einer eigenen bzw. von Standard-Applikationen über Intents erreichbar sind. Sie erfahren, wie Activities untereinander kommunizieren, Daten mit ContentProvider über App-Grenzen hinweg ausgetauscht werden können, die verschiedenen Arten von Android-Animationen generiert werden und wie MultiMedia- und Geocoding-Eigenschaften in Android-Apps eingebunden werden können.

Weil Themen wie die rasante Ausbreitung von Android-Apps oder deren Veröffentlichung mit dem Google Play Shop in vielen Lehrbüchern zu Android beschrieben sind, kann in einem Übungsbuch darauf verzichtet werden. Dafür beschäftigen wir uns im Details bereits am Anfang von Kapitel 2 mit dem DDMS(Dalvik Debug Monitor Server)-Tool, um dieses parallel zur Entwicklung für die Fehlersuche einsetzen zu können. Eine wichtige Rolle wird auch der Benutzung des Emulators beim Programmieren und Testen von Applikationen gewidmet. Gleichzeitig zeige ich, wie Android-Apps auf einem Smartphone, das via USB an den PC angeschlossen ist, ohne Google Play installiert und mit dem DDMS überwacht werden können.

Mit Apps, die an die wichtigsten Themen gebunden sind, gehe ich auf die Java-Klassen von Android ein und beschreibe jedes Mal parallel das Einbinden von Ressourcen über XML-Dateien, die eine wichtige Rolle in der Gestaltung von Oberflächen spielen. Es wird auf viele View-Elemente und Ressourcentypen Bezug genommen. Ich habe nicht jedes einzelne Thema in eine eigene App umgesetzt, sondern ganz bewusst, um Wiederholungen in den Formulierungen von Aufgaben zu vermeiden, verwandte Funktionalitäten in die gleiche App zusammengefasst, die dann einzeln über Auswahlmöglichkeiten am Bildschirm der App erreicht werden können.

Die mit allen zugehörigen Projektdateien präsentierten Apps aus diesem Buch können als Basis für die Entwicklung von eigenen Apps dienen.

Ich mache auf die Unterschiede zwischen nativen Apps und Web-Apps aufmerksam und zeige dies auch mit Beispielen, indem Webapplikationen aus dem Band III des Übungsbuchs »Servlets und JavaServer Pages« in diesem Buch mit Android entwickelt werden.

## Aufbau des Buchs

Das erste Kapitel des Übungsbuchs bringt eine Beschreibung der Installation des Android-SDK und einer Android-Plattform für die Entwicklung und beschäftigt sich mit der Architektur und Installation von Android-Applikationen. Des Weiteren wird in diesem Kapitel die Art und Weise, wie das Ant-Tool beim Erstellen von Android-Projekten eingesetzt werden kann, präsentiert.

Die Ressourcen eines Projekts bilden zusammen mit den Layoutdateien und Java-Klassen die drei wichtigsten Bausteine einer App. Anhand der von Android standardmäßig erstellten App beim Anlegen eines Projekts beschreibe ich in diesem Kapitel alle dazugehörigen Projektdateien und Verzeichnisse.

Um ein besseres Verständnis für die Programmierung von Android-Apps allgemein und für das Zusammenspiel von XML- und Java-Klassen bzgl. der Definition von Oberflächen zu erlangen, soll mit der ersten Aufgabe die erste eigene App sowohl mithilfe einer Layoutdatei als auch wie in Java üblich ohne diese programmiert werden.

Im zweiten Kapitel, das mit einer Beschreibung der Funktionalitäten des DDMS-Tools beginnt, widmen wir uns erstmals dem Lebenszyklus von Applikationen und der Protokollierung von deren Abläufen in der LogCat und am Bildschirm. Danach werden die Dialog- und Menü-Klassen von Android vorgestellt. Die in diesem Kapitel enthaltenen Aufgaben präsentieren neue Ressourcentypen, Layoutdefinitionen und Widget-Klassen, zeigen, wie eigene Menü-, Dialog-, View- und Adapter-Klassen erstellt werden können, explizite und implizite Intents eingesetzt werden und wie Standard- sowie eigene App-Komponenten über implizite Intents nutzbar gemacht werden können. Sie stellen die wichtigsten Zugriffe im Android-Dateisystem beispielhaft dar und zeigen, wie das Telefonieren und das Verschieken von E-Mails und SMS-Nachrichten programmtechnisch erfolgen können. Das Kapitel beschäftigt sich des Weiteren mit der Beschreibung und Benutzung von Fragment-Komponenten und wird mit dem Realisieren der Animationsarten, die von Android unterstützt werden, beendet.

Das dritte Kapitel befasst sich mit der erweiterten App-Programmierung. Das in Android integrierte SQLite-Datenbanksystem wird vorgestellt und ich zeige anhand von Übungsaufgaben, wie Daten in einer Datenbank gespeichert, geändert, gelöscht und abgefragt werden können. Diese Daten gehören der App, in der die DB erstellt wurde. Sollten sie auch anderen Applikationen zur Verfügung gestellt werden, kommen Content Provider zum Einsatz.

Ich zeige, wie eigene Content Provider erzeugt werden, aber auch, wie die Standard Content Provider von Android, wie z.B. MediaStore und CallLog, erreichbar sind. In Android besteht die Möglichkeit, mithilfe der Kamera eines mobilen Geräts Fotos zu erstellen. Mit einer App, die das Fotografieren mit der Kamera

ermöglicht, werden die aufgenommenen Fotos in der MediaStore-Datenbank gespeichert, wieder aufgefunden und per E-Mail verschickt.

Des Weiteren beschäftigen wir uns in diesem Kapitel mit der Art und Weise, wie MultiMedia- und Geocoding-Eigenschaften in Android-Apps eingebunden werden können. Anhand zahlreicher Tests wird das Laden von Bild-, Sound- und Videodateien aus dem lokalen Verzeichnis von Applikationen, von der SD-Karte und aus dem Internet geprüft.

Das parallele Laden von Dateien aus dem Internet im Hintergrund wird mit einer App demonstriert, die die Benutzung von Java-Threads und der AsyncTask von Android gegenüberstellt. Wenn langläufige Threads gebraucht werden, um ein Problem zu lösen, sollte auf die Klassen und Interfaces des Concurrent Utilities von Java zurückgegriffen werden. Anhand der Berechnung von Primzahlen zeige ich, wie Runnables, Callables und Executors für das Verteilen von Aufgaben an mehrere Arbeitsthreads eingesetzt werden können, und beschreibe den Arbeitsverlauf mithilfe von ProgressBar-Widgets und Progress-Dialogen.

Mit einer App »für den Urlaub«, die uns ermöglicht, an dem Ort, in dem wir uns gerade befinden, Fotos aufzunehmen, diese zusammen mit dem Namen des Orts und den Location-Koordinaten anzuzeigen und per E-Mail an Freunde zu senden, und einer App, die mithilfe von Fragments in ihrer Ausrichtung zwischen Portrait- und Landscape-Modus wechseln kann, wird der Aufgabenteil aus dem 3. Kapitel ergänzt. Als Abschluss präsentiere ich eine Aufgabe, die Tween- und Property-Animationen realisiert.

Es folgen Hinweise für das Laden und Testen der Apps auf einem Smartphone, Remote Debugging, Erstellen von Screenshots auf dem Handy, die Veröffentlichung von Applikationen und das Android-SDK 4.2.2. Dadurch, dass für Android das Laden und Installieren von Apps auf dem Smartphone nicht zeitaufwendig ist, ist beim Erstellen von Apps gleich von Anfang an ein paralleler Test auf dem Emulator und Smartphone, was auch durch das DDMS-Tool von Android unterstützt wird, ebenfalls denkbar.

## Benötigte Software

Für die Durchführung der Übungen können Sie das Java Development Kit (JDK) der Java Standard Edition (SE) Version 5.x, 6.x oder 7.x verwenden. Das aktuelle JDK der Java SE können Sie sich kostenlos von der Java-Homepage von Oracle unter der Adresse <http://www.oracle.com/technetwork/java/javase/downloads/index.html> herunterladen. Das JDK umfasst sowohl die Software zur Programmerstellung als auch das Java Runtime Environment (JRE) für die Programmausführung.

Ich habe mich in der Entwicklung für die zum Zeitpunkt der Redaktion dieses Buchs gerade aktuelle Version 4.0.3 API 15 von Android entschieden. Verfolgen Sie für die Installation die in Kapitel 1 im Detail beschriebene Vorgehensweise. In der Zwischenzeit wurden die Versionen 4.2.1 und 4.2.2 von Android freigegeben. Diese können ebenfalls von der Webseite <http://developer.android.com/sdk/index.html> heruntergeladen werden. Alle Applikationen aus diesem Buch wurden auch mit der Android-Plattform 4.2.2 API 17 von Android getestet.

Für die Installation der Android Updates auf dem Galaxy-S3-Smartphone, auf dem alle Apps aus diesem Buch geladen und getestet wurden, verweise ich auf das Internet.

## Webseite zum Buch

Unter [www.mitp.de/9501](http://www.mitp.de/9501) finden Sie einen Link mit Download-Dateien zum Buch. Dieser beinhaltet den Quellcode der Lösungsvorschläge und die für Windows 7 kompilierte ausführbare Version als Download-Archiv. Dort finden Sie alle Java-Quellcodes, übersetzten Klassen sowie Layout- und Ressourcendateien in einer Verzeichnisstruktur, die mit der im Buch beschriebenen übereinstimmt.

Ich wünsche Ihnen weiterhin viel Spaß und Erfolg beim Programmieren mit Java und Android.

Elisabeth Jung