

<b>Vorwort</b> .....	<b>XI</b>
<b>Teil I: Werkzeuge und Grundlagen</b> .....	<b>1</b>
<b>1 Das Android-SDK</b> .....	<b>3</b>
Das Android-SDK und die Voraussetzungen installieren .....	3
Testlauf: Prüfen, ob die Installation funktioniert .....	15
Komponenten des SDKs .....	26
Aktualisierungen .....	34
Beispielcode .....	37
Über das Lesen von Code .....	38
<b>2 Java für Android</b> .....	<b>41</b>
Android verändert das clientseitige Java .....	41
Das Java-Typsistem .....	42
Geltungsbereiche .....	66
Idiome der Java-Programmierung .....	70
<b>3 Die Zutaten einer Android-Anwendung</b> .....	<b>87</b>
Traditionelle Programmiermodelle im Vergleich zur Android-Programmierung .....	87
Activities, Intents und Tasks .....	89
Weitere Android-Komponenten .....	91
Der Lebenszyklus von Komponenten .....	96
Statische Anwendungsressourcen und Kontext .....	100
Die Android-Laufzeitumgebung für Anwendungen .....	109
Android erweitern .....	113

Nebenläufigkeit in Android	120
Serialisierung	136
<b>4 Ihre Anwendung in die Hände des Benutzers bringen</b>	<b>145</b>
Anwendungssignierung	145
Eine App auf Google Play bringen	156
Alternativen für den Vertrieb	159
Google Maps-API-Schlüssel	161
Die API-Level-Kompatibilität angeben	162
Kompatibilität mit vielen Displays	163
<b>5 Eclipse bei der Android-Softwareentwicklung</b>	<b>165</b>
Eclipse-Konzepte und Terminologie	166
Eclipse-Views und Perspektiven	172
Java-Programmierung in Eclipse	175
Eclipse und Android	177
Bugs verhindern und den Code sauber halten	178
Eclipse-Eigenarten und Alternativen	187
<b>Teil II: Das Android-Framework</b>	<b>189</b>
<b>6 Einen View aufbauen</b>	<b>191</b>
Die Android-GUI-Architektur	191
Eine grafische Benutzerschnittstelle zusammenbauen	196
Den Controller einbinden	202
Das Menü und die Aktionsleiste	224
View-Debugging und -Optimierung	227
<b>7 Fragmente und die Unterstützung unterschiedlicher Plattformen</b>	<b>231</b>
Ein Fragment erstellen	232
Fragment-Lebenszyklus	235
Der Fragment-Manager	237
Fragmenttransaktionen	238
Die Support Library	243
Fragmente und Layout	245
<b>8 2D- und 3D-Grafiken zeichnen</b>	<b>253</b>
Eigene Widgets gestalten	253
Augenschmaus	280

<b>9</b>	<b>Datenverarbeitung und -speicherung</b>	<b>295</b>
	Einführung in relationale Datenbanken	295
	SQLite	296
	Die Sprache SQL	297
	SQL und das datenbankzentrierte Datenmodell für Android-Anwendungen	308
	Die Android-Datenbankklassen	310
	Datenbankdesign für Android-Anwendungen	311
	Die Datenbank-API nutzen: MJAndroid	315
<b>Teil III:</b>	<b>Ein Anwendungsgerüst für Android</b>	<b>329</b>
<b>10</b>	<b>Ein Framework für ordentliche Anwendungen</b>	<b>331</b>
	Den Lebenszyklus visualisieren	332
	Der Fragment-Lebenszyklus	346
	Die Activity-Klasse und ordentliche Anwendungen	349
	Lebenszyklusmethoden der Klasse Application	350
<b>11</b>	<b>Eine Benutzerschnittstelle aufbauen</b>	<b>355</b>
	Übergeordnetes Design	356
	Grafische Bearbeitung der Benutzerschnittstelle	359
	Mit einem leeren Blatt beginnen	360
	Das Layout der Fragmente	364
	Ein skalierbares UI in Komponenten zerlegen	367
	Activity, Fragment, Aktionsleiste und mehrere Layouts zur Zusammenarbeit bringen	374
	Die andere Activity	378
<b>12</b>	<b>Content-Provider nutzen</b>	<b>383</b>
	Wie Content-Provider funktionieren	385
	Die öffentliche Provider-API implementieren	389
	Einen Content-Provider schreiben und integrieren	394
	Dateimanagement und Binärdaten	396
	Android-MVC und Content-Observation	398
	Ein vollständiger Content-Provider:	
	Der SimpleFinchVideoContentProvider-Code	400
	Ihren Content-Provider deklarieren	409

<b>13 Ein Content-Provider als Fassade für einen REST-basierten Webservice</b>	<b>411</b>
REST-basierte Android-Anwendungen entwickeln	412
Ein »Netzwerk-MVC«	413
Zusammenfassung der Vorteile	415
Codebeispiel: YouTube-Videoinhalte dynamisch anzeigen und cachen	417
Die Struktur des Quellcodes für das Finch-YouTube-Videobeispiel	419
Die Videoanwendung Schritt für Schritt	420
Schritt 1: Das UI nimmt Benutzereingaben entgegen	421
Schritt 2: Unser Controller überwacht Events	421
Schritt 3: Der Controller führt mit einer ManagedQuery einen Request auf Content-Provider bzw. Modell aus	421
Schritt 4: Den REST-Request implementieren	422
<b>Teil IV: Fortgeschrittenere Themen</b>	<b>443</b>
<b>14 Suche</b>	<b>445</b>
Die Suchschnittstelle	445
Suchvorschläge	454
<b>15 Location und Mapping</b>	<b>461</b>
Positionsbasierte Dienste	462
Mapping	463
Die Google Maps-Activity	463
MapView und MapActivity	464
Mit MapViews arbeiten	465
MapView- und MyLocationOverlay-Initialisierung	466
Eine MapActivity aussetzen und wiederaufnehmen	469
Die Karte mit Menübuttons steuern	471
Die Karte mit dem Keypad steuern	473
Positionsdaten ohne Karten	474
StreetView	482
<b>16 Multimedia</b>	<b>485</b>
Audio und Video	485
Audio- und Videowiedergabe	486
Audio- und Videoaufzeichnung	490
Gespeicherte Medieninhalte	496

<b>17 Sensoren, NFC, Sprache, Gesten und Barrierefreiheit</b>	<b>499</b>
Sensoren	499
Near Field Communication (NFC)	505
Gesteneingabe	517
Barrierefreiheit	519
<b>18 Kommunikation, Identität, Synchronisierung und Social Media</b>	<b>523</b>
Kontaktkonten	523
Authentifizierung und Synchronisierung	526
Bluetooth	542
<b>19 Das Android Native Development Kit (NDK)</b>	<b>559</b>
Native Methoden und JNI-Aufrufe	560
Das Android-NDK	562
Native Bibliotheken und Header aus dem NDK	566
Eigene Bibliotheksmodule erstellen	567
Native Activitys	571
<b>Index</b>	<b>579</b>