

jan TITTEL
jochen BAUMANN

Kompakter
**SCHNELL
EINSTIEG**
für IT-Profis



APPS FÜR ANDROID ENTWICKELN

AM BEISPIEL EINER REALEN APP

HANSER



Bleiben Sie auf dem Laufenden!

Der Hanser Computerbuch-Newsletter informiert Sie regelmäßig über neue Bücher und Termine aus den verschiedenen Bereichen der IT. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter www.hanser-fachbuch.de/newsletter

Jan Tittel
Jochen Baumann

Apps für Android entwickeln

Am Beispiel einer realen App

HANSER

Die Autoren:

Jan Tittel, Essen

Jochen Baumann, Oberhausen

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.



Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2014 Carl Hanser Verlag München, www.hanser-fachbuch.de

Lektorat: Sieglinde Schär

Copy editing: Kathrin Powik, Lassan

Herstellung: Irene Weilhart

Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München

Umschlagrealisation: Stephan Rönigk

Gesamtherstellung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

print-ISBN: 978-3-446-43191-1

e-book-ISBN: 978-3-446-43315-1

Inhalt

1	Einführung	1
1.1	Die Android-Plattform	2
1.2	An wen richtet sich dieses Buch?	3
1.3	Buchaufbau und verwendete Technologien	3
1.4	Vorstellung des Beispielprojekts	4
1.5	Danksagung	9
2	Einrichten der Arbeitsumgebung	11
2.1	Installation von Java	11
2.2	Einrichtung und Konfiguration von Eclipse und ADT	14
2.3	Geräte für die Entwicklung einrichten	23
3	Schnelleinstieg in Eclipse und Java	27
3.1	Die erste App mit Eclipse und dem ADT erstellen	27
3.1.1	Ein neues Projekt in Eclipse anlegen	28
3.1.2	Erster Einstieg in Eclipse mit ADT	32
3.1.3	Die Oberfläche der App anpassen (XML-Layout)	35
3.1.4	Der App Funktionalität geben	41
3.1.5	Die Struktur von Android-Projekten	43
3.2	Die App im Emulator und auf einem Gerät testen	46
3.2.1	Emulator definieren	47
3.2.2	Tastaturbefehle für den Android-Emulator	49
3.2.3	Die App auf einem Emulator oder Gerät starten	49
3.3	Crash-Kurs in Java	52
3.3.1	Werte einer Variablen zuordnen	54
3.3.2	Bezeichner und Schlüsselwörter in Java	54
3.3.3	Klassen und Objekte in Java	55
3.3.4	Objekte erzeugen und initialisieren	57
3.3.5	Methoden von Objekten aufrufen	57
3.3.6	Auf Eigenschaften von Objekten zugreifen	58
3.3.7	Abstrakte Klassen und Methoden	59

3.3.8	Interfaces (Schnittstellen)	61
3.3.9	Java Packages (Pakete)	62
3.3.10	Logging	62
3.3.11	Kommentare	62
3.3.12	Ablaufsteuerung mit einfacher Verzweigung	63
3.3.13	Ablaufsteuerung mit Mehrfachverzweigungen	64
3.3.14	Wiederholungen mit Schleifen	64
3.3.15	Fehlerbehandlung	65
4	Grundlagen von Layouts, Views, Komponenten und Intents: Erste Oberflächen erstellen	67
4.1	Zielsetzung	67
4.2	Layouts, Views und Komponenten	67
4.2.1	Layouts	68
4.2.2	Views und Widgets	69
4.2.3	Bausteine einer Android-Anwendung	69
4.3	Layouts und Activities	75
4.3.1	Layouts erstellen	75
4.3.2	Activities	80
4.4	Ereignisse und Intents	81
4.4.1	Auf Klick-Ereignisse der Oberfläche reagieren	82
4.4.2	Mit Intents eine andere Activity aus der aktuellen Activity aufrufen	82
4.5	Activities im Manifest registrieren	84
5	Menüs und Ressourcen	85
5.1	Menüs	85
5.1.1	Optionsmenü erstellen	85
5.2	String-Ressourcen	88
5.2.1	Eine String-Ressource anlegen und referenzieren	88
5.3	Drawable-Ressourcen	89
5.3.1	Drawable-Ressourcen verwenden	89
5.3.2	ActionBar-Icons erstellen und verwenden	90
5.3.3	xEffekte durch Drawables	91
6	Eigene Klassen, Listen und Adapter	95
6.1	Eigene Klassen erstellen	95
6.2	Enumerationen	97
6.3	Arrays, Listen und Adapter	98
6.3.1	Einfache Strings in Spinner und Liste anzeigen	99
6.3.2	Eigene Objekte in einer ListView anzeigen	101
6.3.3	ListItem und ListAdapter für eigene Objekte	103
6.3.4	ListItem auswählen	107
6.3.5	LogCat verwenden	108

6.4	Mit Dateien arbeiten	109
6.4.1	Datei im Dateisystem speichern	110
6.4.2	Inhalte von Verzeichnissen auflisten und in ListView anzeigen	117
6.4.3	Dateien löschen	122
6.4.4	Dateien lesen	124
6.5	Zwischenstand der App (Version 0.3)	128
7	ActionBar, WebView und E-Mail	129
7.1	ActionBar erweitern und Funktionen nutzen	129
7.1.1	Eigenes Layout-Element in der ActionBar verwenden und App-Icon für die Navigation nutzen	129
7.2	E-Mail mit Anhang versenden	133
7.3	Lokale und externe Webseiten anzeigen	136
7.3.1	Erstellen und Anpassen der WebView Activity	136
7.4	Zwischenstand der App (Version 0.4)	140
8	Fragments, Touch Events und Canvas	141
8.1	Fragments	141
8.1.1	Fragments, Drawable-Ressourcen	142
8.2	Image View erweitern, Canvas und Touch Events	150
8.3	Activity for Result und Grafikbearbeitung	156
8.3.1	Mit Activity for Result den Pfad zum Bild ermitteln und sich das Ergebnis in einem Toast anzeigen lassen	156
8.3.2	Ein Bild in ein Bitmap umwandeln	157
8.4	Zwischenstand der App (Version 0.5)	161
9	Audiodaten aufnehmen, abspielen und die App mit Gesten steuern	163
9.1	Touch Events auswerten mit GestureDetector	163
9.2	Audios aufnehmen und abspielen	166
9.2.1	Audio-Notizen erstellen	166
9.2.2	Audio-Notiz abspielen	167
9.3	Threading	169
9.4	Zwischenstand der App (Version 0.6)	170
10	Dialog-Fragments und Datenbank	171
10.1	Dialog-Fragments	171
10.1.1	Klasse für Erinnerungen, Reminder	177
10.2	Datenbank in Android verwenden	179
10.2.1	Datenbank erstellen	180
10.2.2	Datensätze hinzufügen, ändern und löschen	183
10.2.3	Alle Datensätze einer Tabelle auslesen	186
10.2.4	Datenbankzugriff per Kommandozeile oder Eclipse Plug-in	187

10.3 Daten aus der Datenbank anzeigen	189
10.4 Zwischenstand der App (Version 0.7)	190
11 Google Maps Api V2 und LocationService	191
11.1 Vorbereitungen zur Verwendung von Google Maps Api V2	191
11.1.1 API-Key und Rechte im Manifest anpassen	195
11.2 Ortsbestimmung mit dem LocationService	198
11.3 Eigene Activity for Result für die Map	201
11.4 Zwischenstand der App (Version 0.8)	208
12 Zeit- und ortsbasierte Erinnerungen, lokale Notifications	209
12.1 Zeitbasierte Erinnerungen mit dem AlarmManager	209
12.2 BroadcastReceiver und NotificationManager	211
12.3 Ortsbasierte Erinnerungen mit ProximityAlerts	212
12.4 Zwischenstand der App (Version 0.9)	214
13 Lokalisierung, Icons und Startbilder – Vorbereitung für die Veröffentlichung im Play Store	215
13.1 Die App lokalisieren	215
13.1.1 String-Ressourcen lokalisieren	216
13.2 Icons und Bilder	216
13.3 Der letzte Feinschliff	217
13.3.1 Mit Android Lint den Code verbessern	220
14 Veröffentlichung einer App im Play Store	223
14.1 App mit eigener Signatur exportieren	223
14.2 App in den Play Store hochladen	226
14.3 Store-Eintrag erstellen und Icons und Screenshots verwalten	228
14.4 Preisgestaltung und Vertrieb	228
Index	231

1

Einführung

Der Buchmarkt hält bereits einiges an Literatur zur App-Entwicklung für Smartphones bereit. Angefangen vom einfachen, dünnen Buch für Einsteiger bis hin zum dicken Wälzer und Nachschlagewerk für Profis. Warum also ein weiteres Buch?

Wir sind selbst seit Jahren als Entwickler tätig und haben mit dem Aufkommen des Booms selbstverständlich auch mit der App-Entwicklung für Smartphones begonnen. Dabei haben wir insbesondere zwei Schwachstellen ausgemacht.

Zum einen holen die meisten Bücher oder Tutorien sehr weit aus und beginnen mit den einfachsten Dingen. Ein Großteil der Entwickler startet mit einer App allerdings nicht sein erstes Projekt, sondern verfügt bereits über umfangreiche Kenntnisse in anderen Programmiersprachen. Für viele Entwickler ist daher ein Schnelleinstieg vollkommen ausreichend.

Zum anderen ist es für eine erfolgreiche App unerlässlich, diese für mehrere Plattformen, zumindest für die beiden derzeit erfolgreichsten (iOS und Android), zu entwickeln. Zusätzlich vermissen wir oftmals die Umsetzung von realen Beispielen, da es dabei dann doch einige Aspekte zu berücksichtigen gilt, die in einer rein technisch funktionalen Beschreibung der App-Entwicklung untergehen.

So wurde die Idee geboren, eine tatsächlich existierende Beispiel-App für mehrere Plattformen zu entwickeln und daraus gleichzeitig ein Buch – einen Schnelleinstieg für IT-Profis – zu machen. Durch dieses Konzept ist es für Sie als Leser möglich, sich an ein und demselben Beispiel einen Überblick über die verschiedenen Plattformen zu verschaffen und gleichzeitig mit möglichst wenig Aufwand die Unterschiede bei der Entwicklung herauszuarbeiten. Das vorliegende Buch konzentriert sich daher neben den notwendigen Grundlagen auf die Umsetzung eines umfangreicheren Beispiels.

Parallel erscheint dieses Buch auch für iOS (*Apps für iOS entwickeln*, Carl Hanser Verlag 2013, ISBN 978-3-446-43192-8). Weitere Plattformen sind bereits in Planung, deren konkrete Umsetzung hängt aber auch von der Entwicklung des Marktes ab, sodass wir an dieser Stelle keine falschen Versprechungen machen möchten.

■ 1.1 Die Android-Plattform

Android ist noch eine verhältnismäßig junge Plattform für Smartphones und andere mobile Geräte, wenn man bedenkt, wie lange mobile Plattformen schon eingesetzt werden. Denken Sie beispielsweise an Symbian, Java embedded, Windows CE/Windows Mobile oder BlackBerry OS.

Der Ursprung von Android liegt in dem von Andy Rubin Ende 2003 in Palo Alto gegründeten Unternehmen Android Inc., das Google im Jahr 2005 vollständig übernahm. Android Inc. entwickelte Software für Mobiltelefone und favorisierte standortbezogene Dienste.

Viel mehr war von Android Inc. nicht bekannt und auch Google hüllte sich nach der Übernahme in Schweigen. Die offizielle Lesart war: „Man habe Android Inc. wegen der talentierten Ingenieure und der Technologie übernommen.“ Letztendlich stimmte dies ja mehr oder weniger auch. Bis 2007, genauer gesagt bis zum 5. November dieses Jahres, nahm der normale Nutzer kaum Notiz von diesen Aktivitäten seitens Google. Doch an diesem Tag trat Google an die Presse und gab die Gründung der OHA (Open Handset Alliance) bekannt. Das Konsortium hatte 47 Gründungsmitglieder aus verschiedenen Branchen. Neben bekannten Geräteherstellern und Netzbetreibern wie HTC und T-Mobile gehörten auch Software- und Marketingfirmen sowie Unternehmen der Halbleiterindustrie zu den Gründungsmitgliedern. Eine Woche nach Gründung der OHA erfolgte die Veröffentlichung des ersten offiziellen Android SDK.

Der Launch des ersten Smartphones mit Android wurde allerdings erst für das Jahr 2008 angekündigt. Seit dieser ersten Version wurde Android stetig verbessert. Während anfangs der Fokus noch auf der Funktionalität und weniger auf dem Design lag, so ist die Version 4.0 (*Ice Cream Sandwich*) auch in Sachen Design ein echter Meilenstein. Google reagierte auf die zunehmende Fragmentierung der Plattform, auf die Erfahrungen mit der Version 3.0 (*Honeycomb*), die Tablets vorbehalten war, und erstellte eine verbesserte Version für alle Plattformen (Smartphones, Tablets und TV). TV findet in diesem Buch jedoch keine Beachtung.

Seit *Ice Cream Sandwich* ist es nun viel einfacher, eine App für Smartphones und Tablets zu erstellen, die auf allen Geräten so aussieht und funktioniert, wie man es erwartet, wenn man diese im Standard erstellt hat.

Ein Problem war unter anderem, dass viele Hardware-Hersteller die Android-Oberfläche anpassten und mit eigenen UIs, wie HTC Sense, Samsung TouchWiz, Motorola Motoblur oder Sony, Codename: „Rachel“, auslieferten. Es konnte nun passieren, dass Apps oder Widgets, die auf Standardelementen von Android basieren, auf diesen Geräten anders aussahen oder sogar nicht funktionierten. Dieses Problem besteht seit Version 4.0 nun nicht mehr. Um dieses Problem in den Griff zu bekommen, führte Google das *Holo Theme* und die *Action-Bar* ein, die jetzt auf jedem Gerät verfügbar sind.

Die Version 4.2 (*Jelly Bean*) brachte wieder viele Verbesserungen, und Google legte hier den Fokus der Verbesserungen auf die User Experience und die Stabilität.

Dazu zählen unter anderem folgende Punkte:

- UIs sind noch flexibler zu erstellen und somit ansprechender.
- Verbesserte Browser und App-Widgets
- USB-Audio ist neu hinzugekommen.

- Die DalvikVM wurde stabiler und performanter.
- Ein neuer Bluetooth Stack wurde integriert.
- NFC wurde um Dateiaustausch erweitert.
- Multi-User-Unterstützung

Weitere Informationen zur aktuellen Version erhalten Sie unter <http://www.android.com>.

■ 1.2 An wen richtet sich dieses Buch?

Dieses Buch richtet sich an Entwickler, die bereits grundlegend mit der modernen objekt-orientierten Programmierung vertraut sind. Vorkenntnisse in Java sind nicht notwendig. Hierfür haben wir einen kleinen Crash-Kurs in Java mit in das Buch aufgenommen. Der Schwerpunkt liegt dabei allerdings auf der Syntax und den Grundlagen von Java, die für das weitere Verständnis hilfreich sind. Es wird nicht auf Fragen für Einsteiger eingegangen, worum es sich beispielsweise bei der objektorientierten Programmierung überhaupt handelt. Dieses allgemeine Wissen setzen wir voraus. Außerdem sollten Sie über gute Computerkenntnisse verfügen, da Eclipse und das ADT manchmal Ihre Unterstützung benötigen.

Intention dieses Buches ist es, Ihr vorhandenes allgemeines Programmierwissen auf Android zu übertragen und Ihnen somit einen schnellen und kompakten Einstieg in die App-Entwicklung zu bieten. Die Entwicklung unter Android hat eine geringe Einstiegshürde und sorgt gleich am Anfang für eine steile Lernkurve.



HINWEIS: Für absolute Anfänger und Einsteiger in die Anwendungsentwicklung, ohne Vorkenntnisse in irgendeiner objektorientierten Programmiersprache, ist dieses Buch wenig geeignet, da allgemeine Kenntnisse vorausgesetzt werden.

■ 1.3 Buchaufbau und verwendete Technologien

Nach dieser Einführung wird zunächst die Installation und Konfiguration der Entwicklungsumgebung Eclipse und des ADT-Plug-ins beschrieben. Wir beschränken uns auf die Erläuterung für ein Windows 7-System. Anschließend erhalten Sie einen kurzen Überblick zu Eclipse und den Android Development Tools.

In Kapitel 3 erhalten Sie einen Schnelleinstieg in Java und eine kurze Einführung in Eclipse. Wir haben uns für Eclipse als IDE entschieden, da diese derzeit noch, als favorisiertes Entwicklungssystem von Google angesehen, als Download angeboten wird, und kostenlos sowie für alle Plattformen verfügbar ist. Eine gute Alternative wäre IntelliJ IDEA von JetBrains.

Zum Zeitpunkt der Drucklegung dieses Buches arbeitete Google noch an der neuen Entwicklungsplattform *Android Studio*, die auf IntelliJ IDEA basiert.

Die ersten Kapitel haben wir bewusst mit zahlreichen Screenshots versehen, damit sich auch die Leser unter Ihnen problemlos zurechtfinden, die bislang kaum Erfahrungen mit Eclipse und Java besitzen, und die sich erstmalig mit der Entwicklung unter Eclipse auseinandersetzen.

Die weiteren Kapitel zeigen anhand eines umfangreichen und realen Beispielprojekts die einzelnen Schritte der App-Entwicklung für Android. Das Beispielprojekt umfasst unter anderem folgende Technologien und Features:

- Benutzeroberflächen erstellen
- Mit der ActionBar arbeiten
- Eigene Navigationselemente erstellen
- Eigene Klassen erstellen
- Daten zwischen Activities übergeben
- Texte erstellen und bearbeiten
- Listen darstellen und bearbeiten
- Toasts (Bildschirmnachrichten) einblenden
- Dateien lesen und schreiben sowie mit dem Dateisystem arbeiten
- Datenbanken erstellen und nutzen
- Audio aufnehmen und abspielen
- Standortdienste und Karten verwenden, inklusive Reverse Geocoding
- Notifications nutzen
- Lokalisierung der App für mehrere Sprachen
- Fotos aus der Galerie nutzen

Um zu verstehen, was das Ziel unserer Aufgabe ist, wird die App im folgenden Abschnitt kurz vorgestellt, sodass Sie nicht ins kalte Wasser springen müssen und erst am Ende den Sinn und Zweck der beschriebenen Schritte verstehen.

■ 1.4 Vorstellung des Beispielprojekts

Das Beispielprojekt, welches wir im Verlauf dieses Buches erstellen werden, ist eine App für die Erstellung von Notizen in Form von Text, Bild oder Audio.



Den Quellcode der App finden Sie unter www.downloads.hanser.de. Außerdem können Sie sich das fertige Projekt aus dem Google Play Store herunterladen und installieren. Suchen Sie dazu einfach nach *scyte notes*. Weitere Informationen zum Projekt finden Sie auf der dazugehörigen Webseite www.scyte.eu.

Bild 1.1 zeigt die Startseite der App mit der Auflistung aller Notizen.

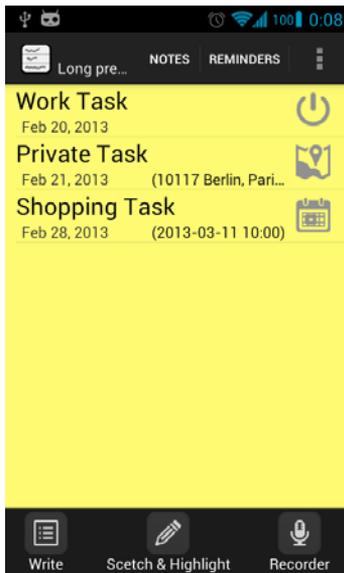


Bild 1.1

Listenansicht aller Notizen

Durch Auswahl des Notiz-Typs und Hinzufügen beziehungsweise Auswählen einer vorhandenen Notiz, gelangt der Benutzer in die spezifische Bearbeitungsansicht der Notiz. Das Beispiel der Bearbeitungsansicht einer Text-Notiz ist in Bild 1.2 zu sehen. Die Bearbeitungsmöglichkeiten sind bewusst einfach gehalten, alle Notizen werden in gängigen Formaten, wie *.txt* für Text-Notizen, *.jpg* für Bild-Dateien und *.3gp* für Audio-Notizen, gespeichert. Somit steht dem Austausch über Plattformgrenzen nichts im Wege, und die Notizen können einfach als E-Mail versendet werden.

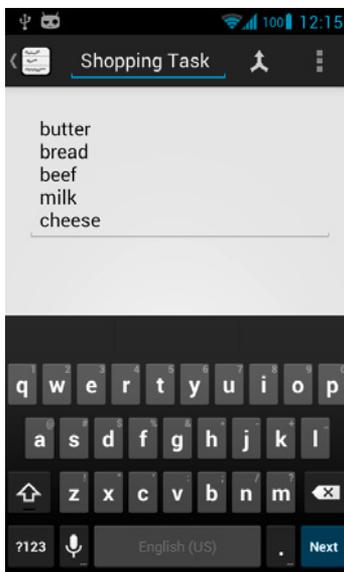


Bild 1.2

Bearbeitungsansicht für Text-Notizen

Nach dem erstmaligen Speichern einer neuen Notiz über das Options-Menü **SAVE**, erscheint ein Auswahldialog, ob eine zeit- oder ortsbasierte Erinnerung oder keine Erinnerung eingerichtet werden soll, wie in Bild 1.3 dargestellt. Außerdem erscheint dieser Auswahldialog, wenn in der Startansicht auf das **ICON** einer Notiz ohne Erinnerung geklickt wird, wie in Bild 1.4 zu sehen. Pro Notiz kann jeweils nur eine Erinnerung eingerichtet werden. Im Gegensatz zu den Notizen selbst, werden die Erinnerungen nicht in Form von Dateien, sondern in einer Datenbank gespeichert.

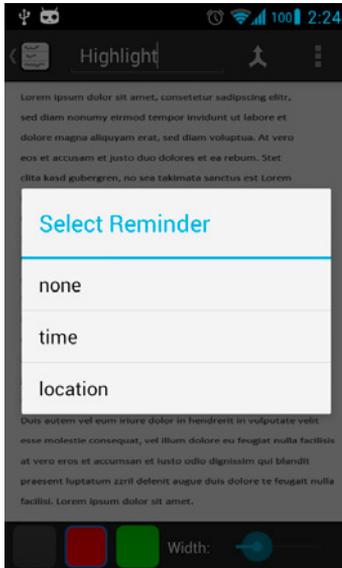


Bild 1.3 Auswahldialog für Erinnerungsart beim Speichern

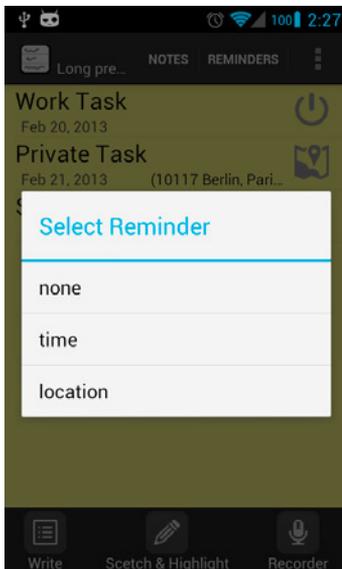


Bild 1.4 Auswahldialog für Notiz ohne Erinnerung in der Startansicht

Nach der Auswahl der Erinnerungsart öffnet sich der eigentliche Dialog für die Erinnerungen. In Bild 1.5 sehen Sie den Dialog der zeitbasierten und in Bild 1.6 der ortsbasierten Erinnerung, mit eigenem Marker. Bei der ortsbasierten Erinnerung gibt es die Möglichkeit, die Karte zu zoomen und auch einen eigenen Marker für die Erinnerung zu setzen. Wird kein eigener Marker hinzugefügt, wird der aktuelle Standort als Erinnerung genutzt. Der eigene Marker gibt dem Benutzer jedoch die Möglichkeit, abweichend von der eigenen Position, den Ort in der Umgebung für die Erinnerung selbst zu bestimmen. Gründe dafür können sein: Die Standortbestimmung ist zu ungenau, oder die Erinnerung wird an einer anderen Position eingerichtet. Die Erinnerungen können durch Auswahl des Icons in der Liste der Startansicht auch geändert oder gelöscht werden.

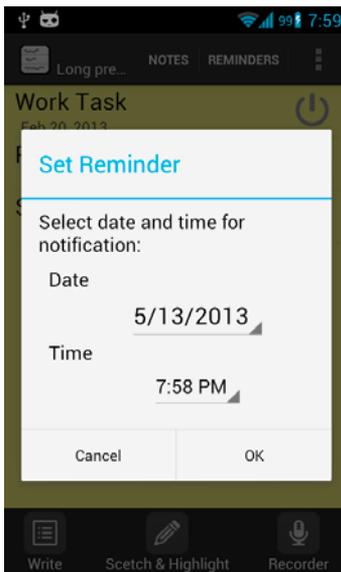


Bild 1.5
Ansicht der zeitbasierten Erinnerung

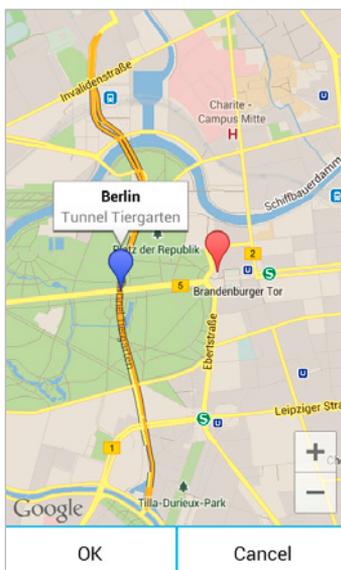


Bild 1.6
Ansicht der ortsbasierten Erinnerung mit eigenem Marker

Der Benutzer hat nicht nur die Möglichkeit, einfache Skizzen als Bild-Notiz anzufertigen, sondern kann auch Fotos aus der Galerie wählen, diese mit Markierungen versehen und abspeichern. Mit den beiden Farben Rot und Grün wird deshalb ausschließlich transparent gezeichnet, wie in Bild 1.7 zu sehen. Außerdem kann die Breite der Markierung über den Schieberegler eingestellt werden.

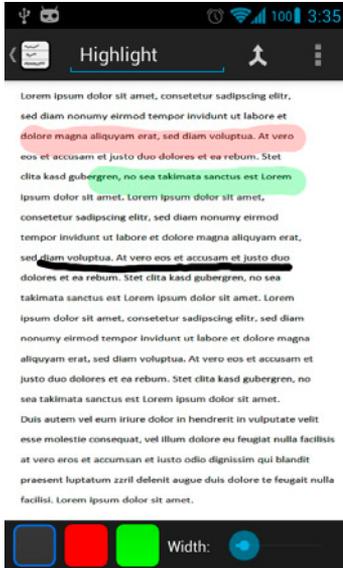


Bild 1.7
Bearbeitungsansicht für Skizzen und Markierungen

Über das OPTIONS-Menü in der oberen ActionBar stehen dem Benutzer weitere Funktionen, je nach Notiz-Typ, zur Verfügung. Im Fall einer Bild-Notiz zum Beispiel, kann mit einem leeren Bild neu gestartet werden, ein Bild aus der Galerie gewählt oder die Notiz per E-Mail versendet werden, wie in Bild 1.8 zu sehen ist.

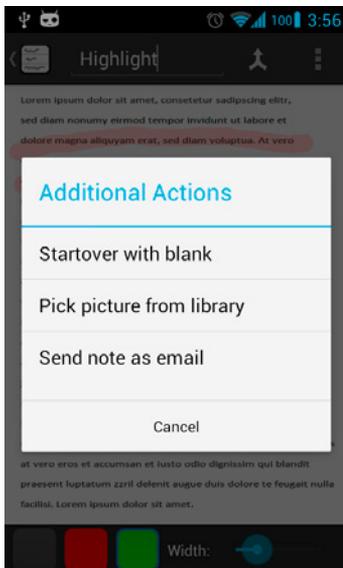


Bild 1.8
Optionen für Skizzen und Markierungen

Da es sich bei der vorgestellten App nicht nur um ein reales Projekt, sondern auch um das Beispiel für dieses Buch handelt, werden wir den Funktionsumfang der App nicht erweitern oder ändern, sodass Sie im Rahmen dieses Buches möglichst eins zu eins genau die App entwickeln, die sich auch für jedermann kostenlos im Google Play Store befindet.



HINWEIS: Wenn wir Feedback oder Bugs von Usern mitgeteilt bekommen, werden wir dies, sofern notwendig, selbstverständlich in eine umgehende Aktualisierung der App einfließen lassen. Als kostenlosen Zusatz zu diesem Buch werden Sie dann eine ausführliche Beschreibung der vorgenommenen Änderungen erhalten. Die PDF wird unter www.downloads.hanser.de zum Download zur Verfügung stehen. Auf dem gleichen Weg werden wir Sie bei Bedarf mit Zusatzinformationen oder Bonuskapiteln versorgen.

■ 1.5 Danksagung

Ein Buch ist keine Einzelleistung. Als Autoren tragen wir viele Bausteine zusammen. Dieses Buch basiert einerseits auf Erfahrungen und Überlegungen aus eigenen Projekten, andererseits aber auch aus öffentlich zugänglichen Quellen des World Wide Web, deren Autoren und Organisatoren wir unseren Dank aussprechen. Ebenso danken wir Frau Sieglinde Schärl und Frau Julia Stepp vom Hanser Verlag für die Anregung zu diesem Buch und die sehr gute Zusammenarbeit.

Besonderen Dank möchte ich, Jochen Baumann, meiner Lebensgefährtin Anke und meinem Sohn Fabian aussprechen, die mit unendlicher Geduld und Unterstützung dazu beitrugen, dass dieses Buch entstehen konnte.

Oberhausen/Essen, im August 2013

Jan Tittel und Jochen Baumann

2

Einrichten der Arbeitsumgebung

In diesem Kapitel wird die Installation und Einrichtung der Entwicklungsumgebung Eclipse und des ADT-Plug-ins unter Windows beschrieben. Weiterhin werden die Android Developer Tools kurz vorgestellt. Bevor wir uns um die Einrichtung von Eclipse kümmern, starten wir mit der Installation von Java.

■ 2.1 Installation von Java

Als Erstes sollten Sie eine aktuelle Version des Java Development Kit (JDK) installieren. Dies kann parallel zur Java Runtime Environment (JRE) erfolgen. Den Download zum JDK (Bild 2.1) finden Sie unter:

- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- <http://java.oracle.com>



Bild 2.1 Downloadlink zum JDK

Downloaden Sie das passende JDK für Ihr Betriebssystem (Bild 2.2). Sie sollten immer die aktuellste Version auswählen.

Java SE Development Kit 7u15

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement
 Decline License Agreement

Product / File Description	File Size	Download
Linux x86	106.64 MB	jdk-7u15-linux-i586.rpm
Linux x86	92.97 MB	jdk-7u15-linux-i586.tar.gz
Linux x64	104.77 MB	jdk-7u15-linux-x64.rpm
Linux x64	91.68 MB	jdk-7u15-linux-x64.tar.gz
Mac OS X x64	143.75 MB	jdk-7u15-macosx-x64.dmg
Solaris x86 (SVR4 package)	135.52 MB	jdk-7u15-solaris-i586.tar.Z
Solaris x86	91.94 MB	jdk-7u15-solaris-i586.tar.gz
Solaris SPARC (SVR4 package)	135.92 MB	jdk-7u15-solaris-sparc.tar.Z
Solaris SPARC	95.26 MB	jdk-7u15-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	22.92 MB	jdk-7u15-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.59 MB	jdk-7u15-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	22.53 MB	jdk-7u15-solaris-x64.tar.Z
Solaris x64	14.96 MB	jdk-7u15-solaris-x64.tar.gz
Windows x86	88.75 MB	jdk-7u15-windows-i586.exe
Windows x64	90.4 MB	jdk-7u15-windows-x64.exe

Java SE Development Kit 7u10

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement
 Decline License Agreement

Bild 2.2 Versionsauswahl von Java

Installieren Sie nun das JDK und setzen Sie gegebenenfalls die Windows-Path-Variablen zum JDK-bin-Ordner wie folgt:

1. Öffnen Sie in der Systemsteuerung den Punkt SYSTEM.
2. Klicken Sie auf ERWEITERTE SYSTEMEINSTELLUNGEN und dann auf UMGEBUNGSVARIABLEN.
3. Wählen Sie aus den *Systemvariablen* den Eintrag *Path* aus, und rufen Sie anschließend den Befehl BEARBEITEN auf, worauf das Fenster wie in Bild 2.3 angezeigt wird.
4. Tragen Sie in das Feld *Wert der Variablen* den Pfad zum JDK ein (Bild 2.3).

Zum Abschluss überprüfen Sie die Java-Umgebung, öffnen die Eingabeaufforderung (cmd) und geben `java -showversion` ein. Sie sollten jetzt die Ausgabe aus Bild 2.4 erhalten.

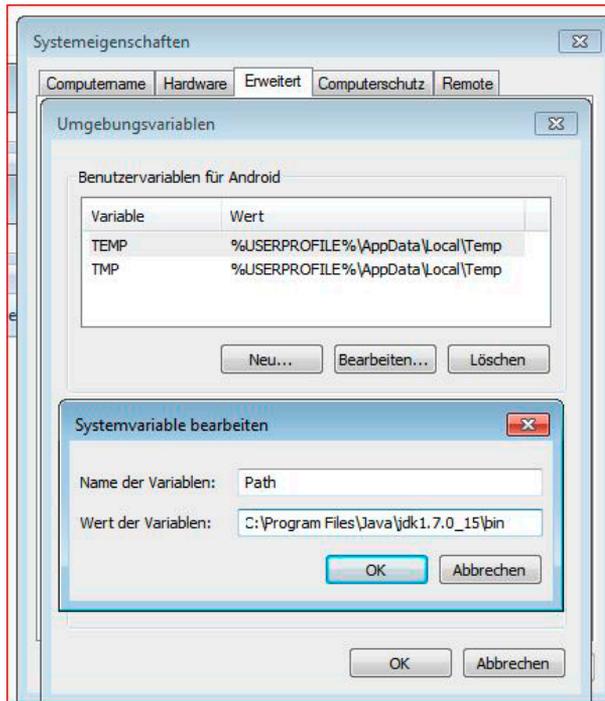


Bild 2.3 Windows-Systemvariablen: Path-Variable JDK

```

java version "1.7.0_15"
Java(TM) SE Runtime Environment (build 1.7.0_15-b03)
Java HotSpot(TM) 64-Bit Server VM (build 23.7-b01, mixed mode)

Verwendung: java [-options] class [args...]
             (zur Ausführung einer Klasse)
             oder java [-options] -jar jarfile [args...]
             (zur Ausführung einer JAR-Datei)
wobei options Folgendes umfasst:
-d32         Verwendet ein 32-Bit-Datenmodell, sofern verf"ngbar
-d64         Verwendet ein 64-Bit-Datenmodell, sofern verf"ngbar
-server      zur Auswahl der "server" UM
-hotspot     ist ein Synonym f"r die "server" UM [verworfen]
             Die Standard-UM ist server.

-cp <Class-Suchpfad von Verzeichnissen und .zip-/.jar-Dateien>
-classpath <Class-Suchpfad von Verzeichnissen und .zip-/.jar-Dateien>

```

Bild 2.4 Ausgabe des Befehls Java-showversion



PRAXISTIPP: Die Ausgabe aus Bild 2.4 ist die Ausgabe des JDK auf einem Windows 7 64-bit-System. Unter einem 32-bit-System erscheint natürlich eine entsprechend andere Ausgabe.

■ 2.2 Einrichtung und Konfiguration von Eclipse und ADT

Als Nächstes benötigen Sie eine aktuelle Version von Eclipse und dem ADT-Plug-in. Google stellt hier mittlerweile ein komplettes Paket zum Download zur Verfügung. Sie finden das Paket unter <http://developer.android.com>.

Sie können Eclipse und das ADT-Plug-in aber auch einzeln installieren. Eine Anleitung dazu finden Sie unter <http://developer.android.com/sdk/installing/index.html>.

1. Downloaden Sie das Bundle von der Android-Developer-Seite (<http://developer.android.com>).
2. Entpacken Sie die *zip*-Datei auf Ihrer Festplatte.
3. Verschieben oder kopieren Sie den Ordner *Eclipse* zur besseren Übersicht in ein separates Verzeichnis mit einfacher Namensgebung Ihrer Wahl (Bild 2.5).

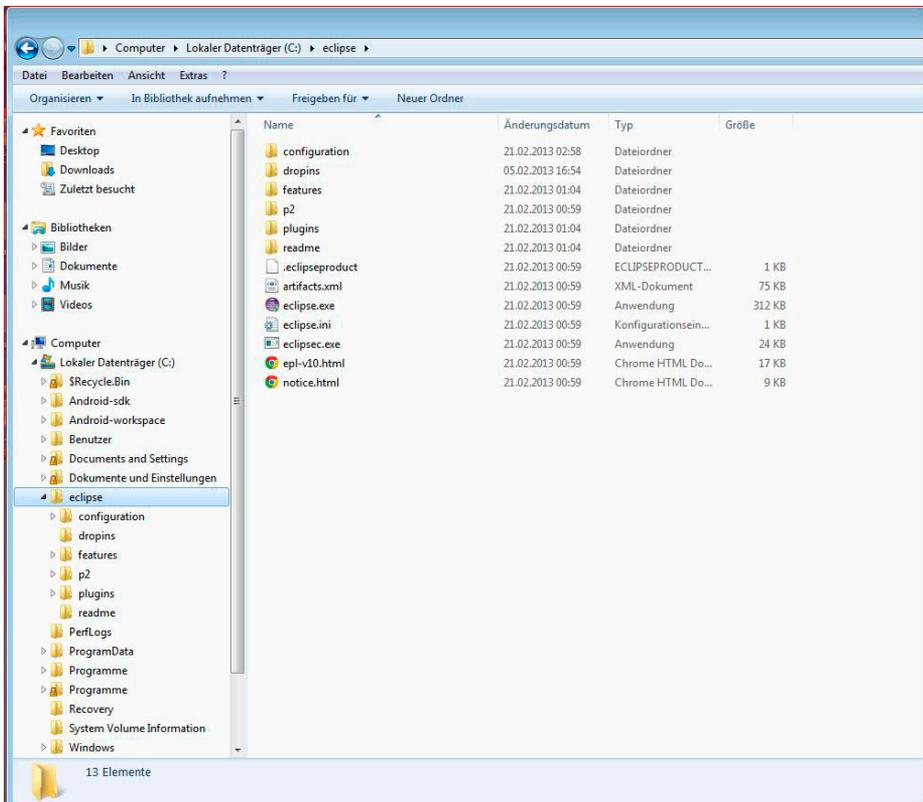


Bild 2.5 Ordner Eclipse

4. Als Nächstes legen Sie ein Verzeichnis für das Android-SDK an und kopieren den Ordner *sdk* und die Datei *SDK Manager.exe* in diesen Ordner (Bild 2.6).

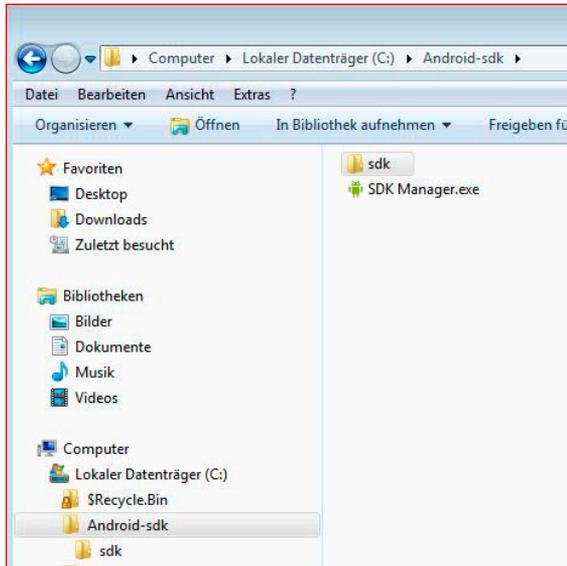


Bild 2.6 Ordner Android-SDK

5. Jetzt tragen Sie, wie bei der Java-Installation, den Pfad zu den Android Plattform-Tools als Systempfad ein, wie in Bild 2.7 abgebildet.

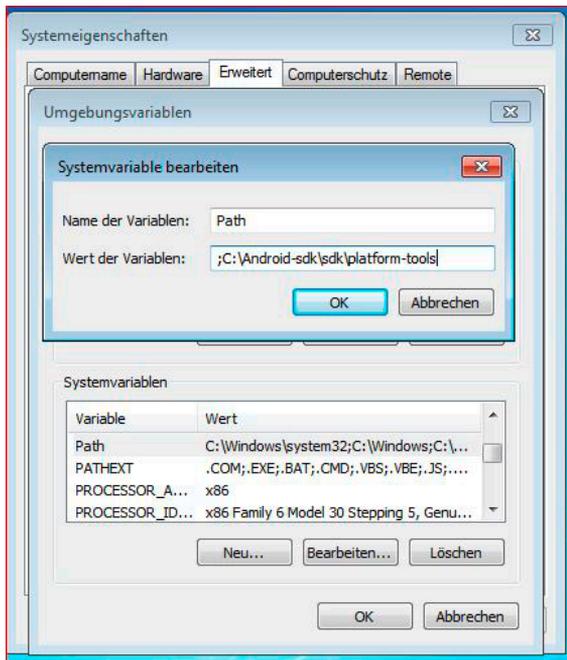


Bild 2.7 Windows-Systemvariablen: Path-Eintrag für die Android-Plattform-Tools

6. Nun aktualisieren Sie das Android-SDK. Dazu starten Sie den SDK-Manager in Ihrem Android-SDK-Ordner durch Doppelklick auf SDK MANAGER.EXE. Auf einigen Systemen erhalten Sie sofort die Fehlermeldung (Bild 2.8), dass der *SDK-Manager* keine Verbindung zum Update-Server herstellen konnte. Dies liegt meist an Ihrer Antivirus-/Security-Lösung oder an der Windows-Firewall.

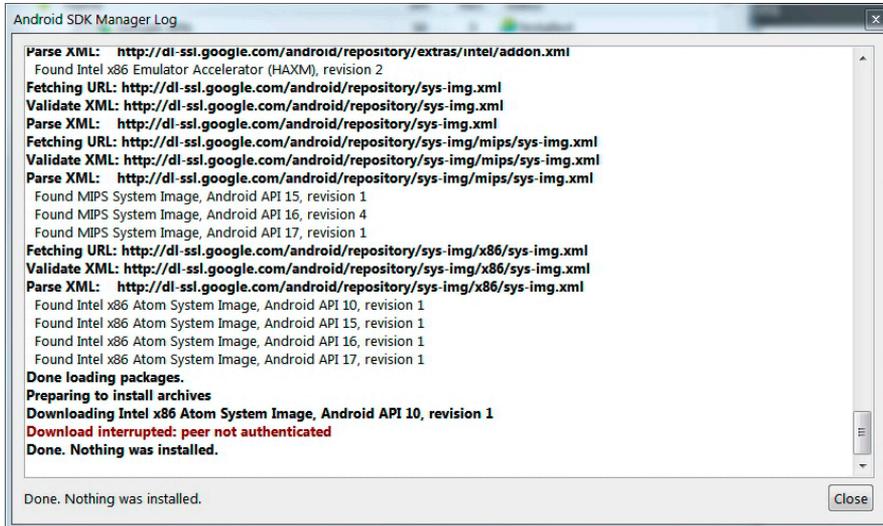


Bild 2.8 Fehlermeldung beim Start von SDK Manager.exe

Sie beheben das Problem, indem Sie auf **TOOLS/OPTIONS** klicken und im Bereich *Others* (Bild 2.9) die Option *Force https://... sources to be fetched using http://...* aktivieren.

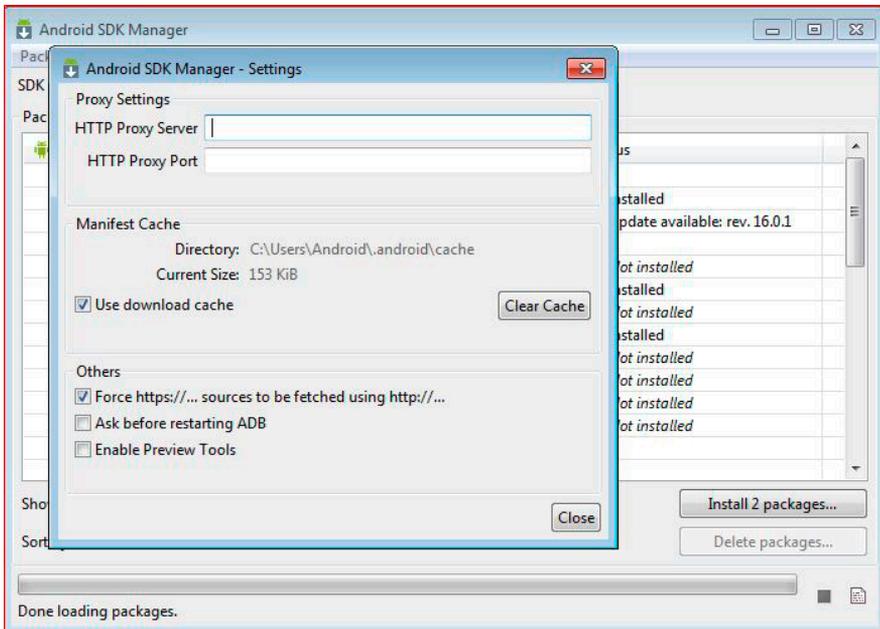


Bild 2.9 Tools/Options im Android-SDK-Manager

7. *Ask before restarting ADB* fragt nach, bevor die Android Debug Bridge (ADB) neu gestartet wird.
8. Aktivieren Sie die Option *Enable Preview Tools*, werden Ihnen auch die Vorab-Versionen des SDK angezeigt.



HINWEIS: Diese Versionen enthalten meist noch Fehler. So hatte die Preview Version 21.1r massive Probleme mit Java-64-bit. Sie sollten diese Versionen nicht in einer produktiven Entwicklungsumgebung einsetzen, sondern nur zu Testzwecken.

9. Nun kann das Android-SDK aktualisiert werden. Wenn nötig, installieren Sie auf jeden Fall die Pakete *Google USB-Driver*, *Google APIs* sowie *Samples for SDK* (Bild 2.10).

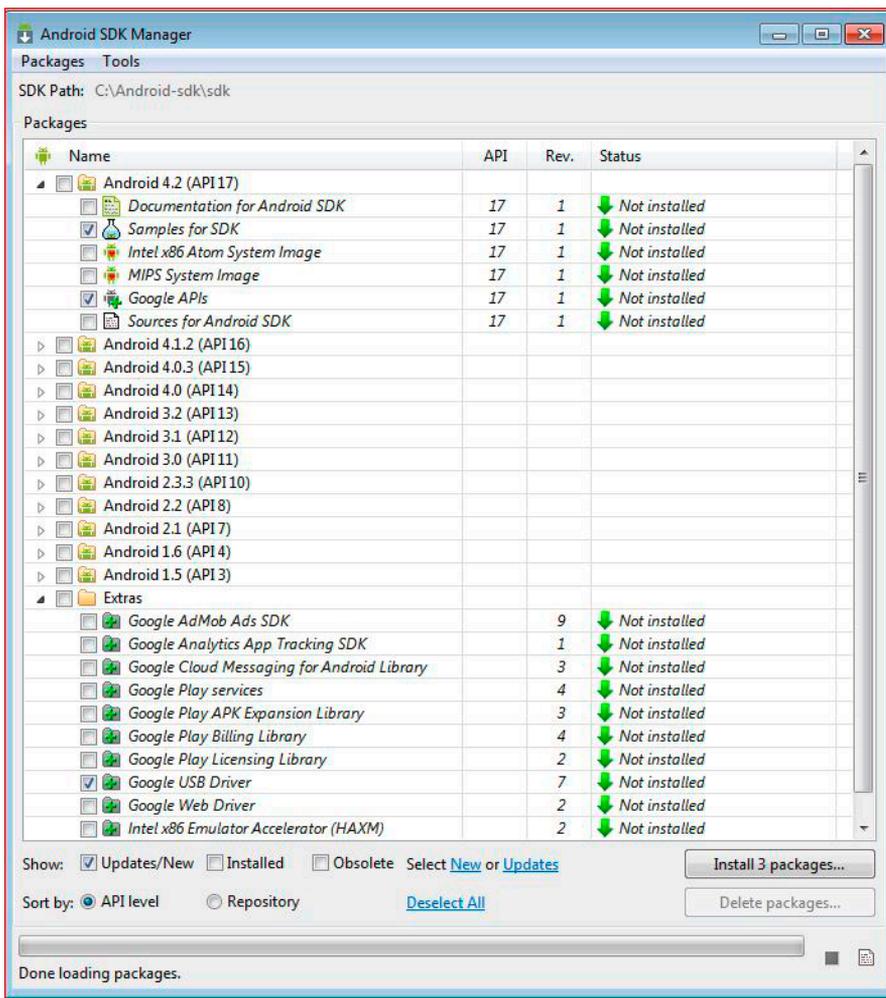


Bild 2.10 Android-SDK-Installation