

MVHS Währungsrechner

*Unterlagen zum Kurs „Einblick in die
Programmierentwicklung für Android Smartphones“ an
der Münchner Volkshochschule*

Lizenz: CC-BY-NC 3.0 (<http://creativecommons.org/licenses/by-nc/3.0/deed.de>)
Ersteller: Dipl.-Ing. (FH) Eugen Richter (WebDucer – IT & Internet Service)

Inhaltsverzeichnis

Geschichte von Android.....	3
Versionsgeschichte.....	3
Versionen.....	4
Versionsunterschiede (Features).....	4
Versionsverbreitung.....	5
Verlauf der Verbreitung.....	6
Hardware.....	7
Sensoren.....	7
Auflösungen.....	7
Auflösungsverbreitung.....	7
Prozessoren.....	7
Verbreitungswege.....	8
Programmierung.....	9
Lebenszyklus der Activity.....	10
MVHS Währungsrechner.....	11
Layout.....	11
Layout-Struktur.....	11
Einstellungen für Views.....	11
Ressourcen.....	12
Events.....	12
onCalculateButtonClick (Button).....	12
onCreateOptionsMenu.....	12
onOptionsItemSelected.....	12
Programmlogik.....	12
Menüs.....	12
Weitere Ausbaumöglichkeiten.....	14
Anhang.....	15
Installation von Eclipse IDE und ADT.....	15
Installation von Eclipse.....	15
Installation von ADT.....	15

Geschichte von Android

Quelle: [http://de.wikipedia.org/wiki/Android_\(Betriebssystem\)](http://de.wikipedia.org/wiki/Android_(Betriebssystem))

Stand: 13.04.2011

Die Geschichte von Android beginnt im Jahr 2005 mit der Übernahme der Firma Android von Andy Rubin durch Google. Diese Firma war auf mobile Anwendungen mit Schwerpunkt auf standort-bezogene Dienste. Im Jahr 2007 gründete Google mit weiteren 33 Partnern die Open Handset Alliance und kündigte die Entwicklung eines neuen mobilen Betriebssystems an. Im Oktober 2008 erschien das erste Android Mobiltelefon (HTC Dream / T-Mobile G-1).

Seit dem wurde das Android Betriebssystem weiter verbessert und neue Funktionen erweitert.

Versionsgeschichte

Version (Name)	Freigabedatum
1.1	10. Februar 2009
1.5 (Cupcake)	30. April 2009
1.6 (Donat)	15. September 2009
2.0 (Eclair)	26. Oktober 2009
2.1 (Eclair)	12. Januar 2010
2.2 (Froyo)	20. Mai 2010
2.3 (Gingerbread)	6. Dezember 2010
2.3.3 (Gingerbread)	23. Februar 2010
3.0 (Honeycomb)	23. Februar 2010 (nur Tablets)
3.1 (Ice Cream)	2011 (Zusammenführung des Codes von 2.3.3 und 3.0)

Versionen

Versionsunterschiede (Features)

Version (Name)	Datum	Neue Features	Status
1.1	10.02.2009	<ul style="list-style-type: none">• Speichern von MMS-Anhängen	unsupported
1.5 (Cupcake)	30.04.2009	<ul style="list-style-type: none">• Automatischer Wechsel zwischen Hoch- und Querformat• Bildschirm-Tastatur• Aufnahme und Wiedergabe von Videos• Automatisches Verbinden und Stereo für Bluetooth• Weitere Sprachen neben Englisch und Deutsch	Supported
1.6 (Donut)	15.09.2009	<ul style="list-style-type: none">• Virtual Private Networks konfigurierbar• Differenzierte Energieverbrauchssteuerung• Suchfunktion quellenübergreifend und selbstoptimierend• Text-to-Speech, Gestenerkennung und mehr als eine Bildschirmauflösung	Supported
2.0 (Eclair)	26.10.2009	<ul style="list-style-type: none">• Digitalzoom und Unterstützung von Blitzlicht• Unterstützung von Microsoft Exchange• Bluetooth 2.1	Supported
2.1 (Eclair)	12.01.2010	<ul style="list-style-type: none">• Animierte Hintergrundbilder• Informationen zur Signalstärke• Erweiterungen von Webkit (HTML5-Unterstützungen, WebStorage, Geolocation, Video ...)	Supported
2.2 (Froyo)	20.05.2010	<ul style="list-style-type: none">• Linux-Kernel 2.6.32, der weniger Arbeitsspeicher benötigt• Verwendung von Arbeitsspeicher, der größer als die bisher nutzbaren 256 MByte ist.• JIT-Compiler, Erweiterungen für OpenGL ES 2.0, Unterstützung von Flash 10.1• Tethering• Speicherbarkeit von Apps auf der SD-Karte (App2SD)• Android Cloud to Device Messaging Framework: Möglichkeit PUSH in die eigenen Anwendungen zu implementieren.• Bluetooth-Sprachwahl	Supported
2.3 (Gingerbread)	06.12.2010	<ul style="list-style-type: none">• Linux-Kernel 2.6.35.7• Unterstützung von WebM• Unterstützung von HTML5 Audio• Unterstützung von Google TV• Unterstützung von Near Field Communication	Supported

Version (Name)	Datum	Neue Features	Status
		<ul style="list-style-type: none"> • Parallele Garbage Collection für ruckelfreie Animationen • Verbesserte Integration von sozialen Netzwerken • Unterstützung von Gyroskopen (nicht zu verwechseln mit Bewegungssensoren) und anderen Sensoren (u.a. Barometer, Schwerekraftsensor) • Integrierter SIP-Client für VoIP • Integrierter Downloadmanager • Unterstützung des Ext4-Dateisystems 	
2.3.3 (Gingerbread)	23.02.2011	<ul style="list-style-type: none"> • Dual-Core-Unterstützung • Unterstützung von Dual-Core-Apps auf Single-Core-Geräten • Verbesserte Unterstützung der NFC-Technik • Verbesserte Bluetooth-Unterstützung • Kleinere Verbesserungen 	Current
3.0 (Honeycomb)	23.02.2011	<ul style="list-style-type: none"> • Benutzerfreundlichere Oberfläche • Verbesserte Unterstützung für Tablet-Computer • Google Talk mit Videotelefonie • Neue Browser-Funktionen: Synchronisierung der Lesezeichen mit Google Chrome, Tabs, automatisches Ausfüllen von Formularen und Inkognito-Modus beim Surfen 	Current
3.1 (Ice Cream)	2011	<ul style="list-style-type: none"> • Zusammenführung der Entwicklungslinien 2.x und 3.x und Google TV 	Future

Versionsverbreitung

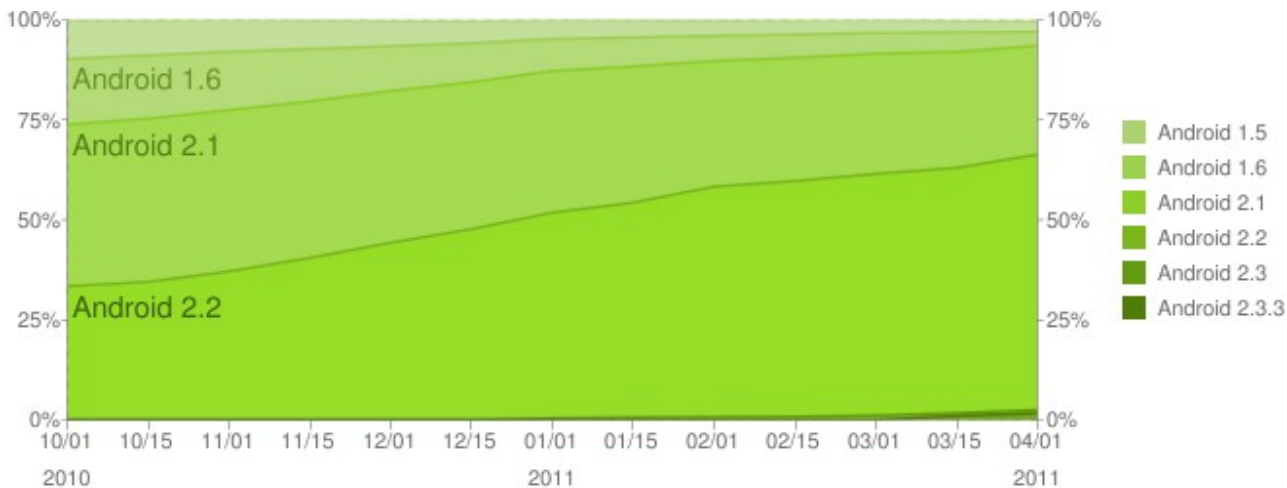
Quelle: <http://developer.android.com/resources/dashboard/platform-versions.html>

Stand: 1. April 2011

Unter <http://developer.android.com/resources/dashboard/platform-versions.html> liefert Google Statistiken zur Verbreitung der unterschiedlichen Versionen. Die Daten werden durch die Zugriffe auf Android Market zusammengetragen. Somit beinhaltet diese Statistik keine Geräte, die keinen Zugriff auf Android Market haben (z.B.: 1&1 Tablet, der Zugriff nur auf 1&1 AppStore hat und weitere günstige Geräte).

Version	API Level	Verbreitung
Android 1.5	3	2,70%
Android 1.6	4	3,50%
Android 2.1	7	27,20%
Android 2.2	8	63,90%
Android 2.3	9	0,80%
Android 2.3.3	10	1,70%
Android 3.0	11	0,20%

Verlauf der Verbreitung



Hardware

Sensoren

- GPS
- Kompass
- Gyroskop
- Beschleunigungssensor
- Lagesensor
- NFC
- Annäherungssensor
- Kamera
- Mikrofone

Auflösungen

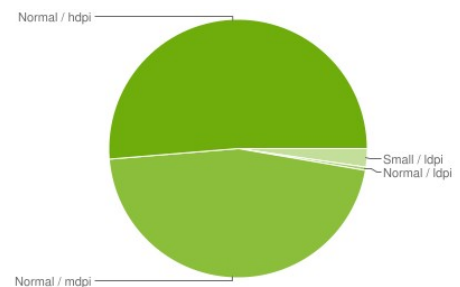
- QVGA (320 x 240) [Low Density]
- HVGA (320 x 480) [Midle Density]
- WVGA (800 x 480) [Hight Density]
- WXGA (1024 x 600 / 1280 x 800) [High Density] Tablets

Auflösungsverbreitung

Quelle: <http://developer.android.com/resources/dashboard/screens.html>

Stand: 13.04.2011

	Low Density	Medium Density	High Density
Small	2.3 %		
Normal	0.4 %	45.9 %	51.2 %
Large			



Prozessoren

- ARM Prozessoren mit 400 – 1GHz (Single Core)
- ARM Prozessoren mit 600 – 1,6 GHz (Dual Core)

Verbreitungswege

Die Android Apps können im Gegensatz zu Apple Apps von jeder Quelle bezogen werden. Es gibt keine Pflicht das Google Market zu benutzen. Aus diesem Grund etablieren sich langsam auch alternative Markets, die unter anderem andere Vergütungsmodelle für Programmierer und andere Bezahlmethoden anbieten.

Weiterhin bieten einige Smartphone Hersteller (entweder nur diesen oder parallel zu Google Market) herstellersistenspezifische Markets (z.B.: für Samsung Galaxy Tab).

Folgende Bezugsquellen für Apps sind möglich (kein Anspruch auf Vollständigkeit):

- Google Market
- Amazon AppStore (nur USA momentan)
- Androidpit
- Entwickler-Homepage
- Eigener PC/Mac

Bei allen Quellen, außer natürlich Google Market, muss auf dem Android Gerät der Bezug von Apps aus unbekannten Quellen aktiviert sein (Einstellungen → Anwendungen).

Bei den kostenpflichtigen Apps bekommt der Entwickler (bei Google und Amazon) 70% der Einnahmen und 30% der Betreiber des Markets.

Weitere Verdienstmöglichkeiten ergeben sich durch die Werbeeinblendungen in den Apps (z.B. durch AddMobile – gehört mittlerweile auch zu Google).

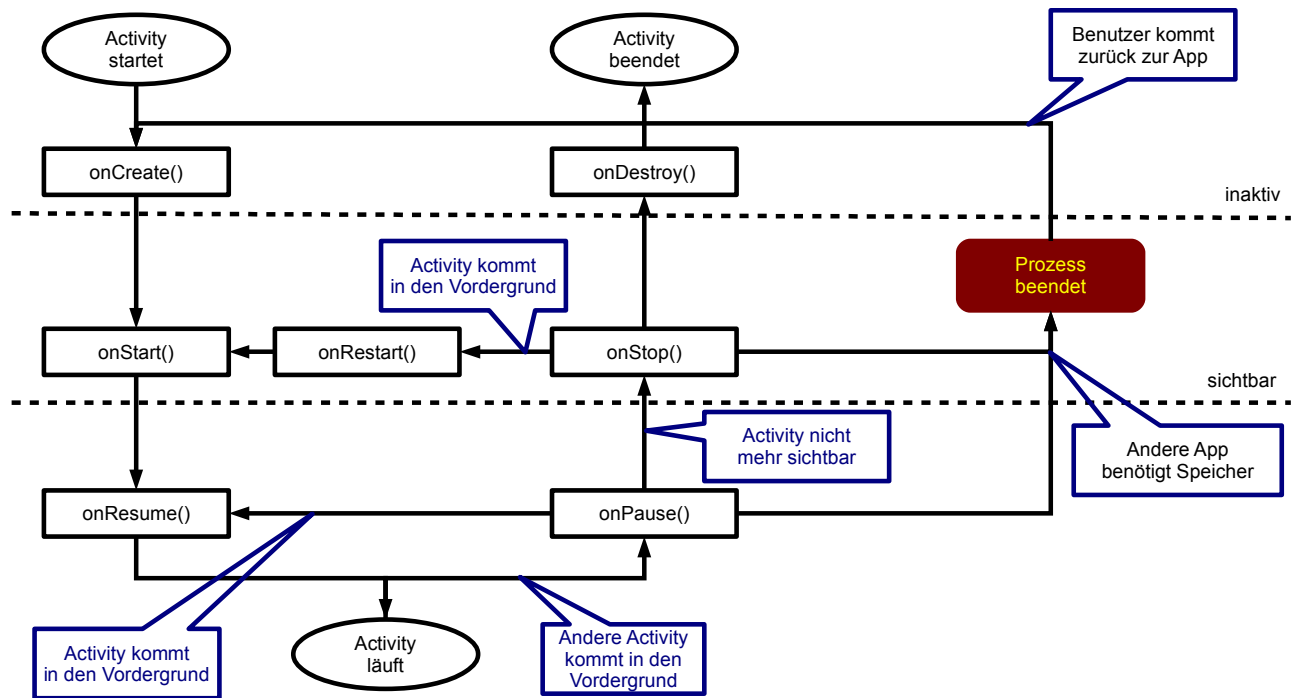
Programmierung

Die Entwicklung von Android Apps erfolgt grundsätzlich in Java. Android unterstützt dabei Java 1.5 komplett und das Android SDK liefert zusätzlich noch spezielle Bibliotheken, die an das mobile System angepasst wurden (Bluetooth, JSON, XML.Sax, JUnit, W3C.DOM). Die Apps laufen auf dem Gerät (und im Emulator) in einer von anderen Apps abgeschotteten Sandbox. Die eingesetzte JVM wurde von Google neu entwickelt und an die Architektur eines mobilen Systems mit wenig Ressourcen angepasst. Die JVM heißt unter Android Dalvik VM diese kann keinen nativen Java-Code ausführen, genauso wenig wie die Oracle JVM den Dalvik Byte-Code nicht ausführen kann. Beim Kompilieren eines Android Projektes wird aus dem Java-Byte-Code (Kellerautomat) ein Dalvik-Byte-Code (Registerautomat) erstellt.

Nur Performance lastige Anteile wurden in Android in C/C++ erstellt (Treiber, Mediacodecs, Webbrowser Engine, SQLite und OpenGL). Diese Möglichkeit ist auch für die Entwicklung eigener Apps gegeben, und sollte nur dann verwendet werden, wenn es Performance-Engpässe gibt. Da für C/C++ Code kein Garbage Collector vorhanden ist, muss der Programmierer sich selbst um die Freigabe der Ressourcen kümmern.

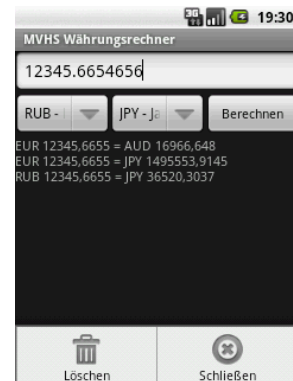
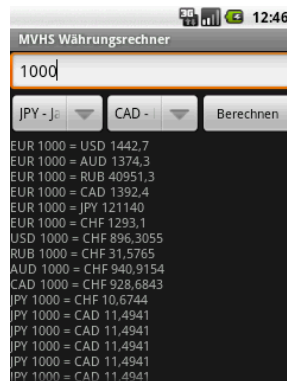
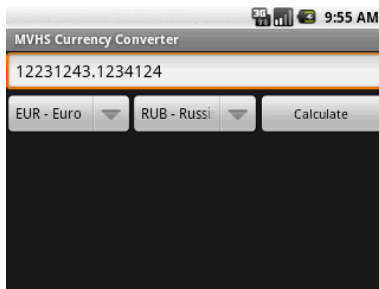
Neben den eigentlichen Apps sind auch die sogenannten Widgets möglich. Diese können einen schnellen Zugriff auf die Daten Ihrer App gewähren, ohne dass der Benutzer die komplette App starten muss. Eine gut durchdachte Erweiterung einer App um ein oder mehrere Widgets kann den Ausschlag zur Konkurrenz geben.

Lebenszyklus der Activity



MVHS Währungsrechner

Layout



Layout-Struktur

- LinearLayout_1 (vertikal) [Hauptrahmen]
 - LinearLayout_2 (horizontal) [Eingabefeld Platzierung]
 - EditText [Eingabefeld für Ausgangswährung]
 - LinearLayout_3 (horizontal) [Platzierung von 3 Elementen]
 - Spinner_1 [Ausgangswährung]
 - Spinner_2 [Zielwährung]
 - Button [Berechnung]
 - ScrollView [View erlaubt das Scrollen bei den Inhalten, die mehr Platz benötigen]
 - TextView [Ergebnisliste]

Einstellungen für Views

TextEdit

- Breite: FillParent
- Eingabemethode: numberDecimal
- max. Länge: 16

LinearLayout_3

- Gewichtungssumme: 9

Spinner

- Breite: 0dip
- Layout-Gewichtung: 3
- Einträge: CurrencyName-Array

Button

- Text: res_calculate
- Breite: 0dip

- Layout-Gewichtung: 3

TextView

- Breite: FillParent
- Höhe: FillParent

Ressourcen

- values
 - strings (Text-Übersetzungen für die Rückfall-Sprache, hier Englisch)
 - arrays (Text-Übersetzungen für Listen in der Rückfall-Sprache)
- values-de
 - strings (deutsche Übersetzungen für Texte)
 - arrays (deutsche Übersetzungen für Listen)

Events

onCalculateButtonClick (Button)

Event für den Klick auf den Button. Dieser führt direkt im Event die Umrechnung der Währung. Bei Berechnungen, die längere Zeit in Anspruch nehmen (> 3 Sekunden) muss an dieser Stelle asynchron (mit Threads) programmiert werden, da sonst Android OS eine Fehlermeldung zeigt.

onCreateOptionsMenu

Event zur Erstellung eines Optionsmenüs. Dabei werden die Einträge des Menüs über Menü-XML definiert.

Die ersten 5 Einträge werden direkt nach dem Klick auf das Menü-Button incl. Piktogramme angezeigt. Die restlichen Menüeinträge werden durch den Klick auf den „Mehr ...“ Eintrag angezeigt.

onOptionsItemSelected

Event zur Steuerung des Klicks auf einen Menüeintrag.

Programmlogik

In dem zum Spinner gehörenden Array werden auch die Umrechnungskurse in Bezug auf Euro abgelegt. Diese können dann in Abhängigkeit des ausgewählten Eintrages ermittelt und zur Berechnung herangezogen werden.

Berechnungsformel:

$$\text{Ergebnis} = \frac{\text{Eingegebener Wert}}{\text{Ausgangswährungskurs}} \cdot \text{Zielwährungskurs} \quad .$$

Menüs

- Schließen (Beenden der App)

- Löschen (Löschen der Liste mit den Währungsumrechnungen)

Weitere Ausbaumöglichkeiten

- Resistenz der Eingaben bei Rotation (Spinner)
- Mehrsprachigkeit
- Auswahl der verfügbaren Währungen
 - Auswahlbegrenzung pro Seite
- Aktualisierung der Kurse über Webservice (DB)
 - Auswahl des Webservices (DB)
- Anlegen selbst definierter Währungen (Spielwährungen) (DB)
- Angepasste Layouts für unterschiedliche Auflösungen und Ausrichtungen
- Widgets für Home Screen
 - Aktueller Kurs für Währung X
 - Währungsrechner als Widget
 - Ergebnis der letzten Umrechnung
- Export
 - Ergebnisliste
 - Währungskurse

Anhang

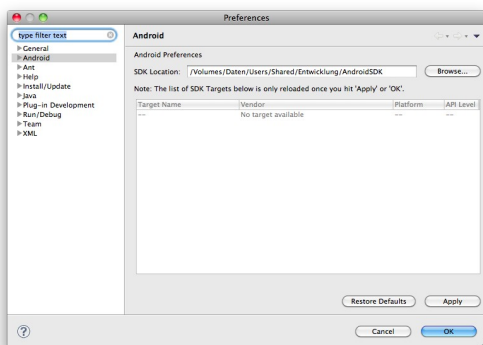
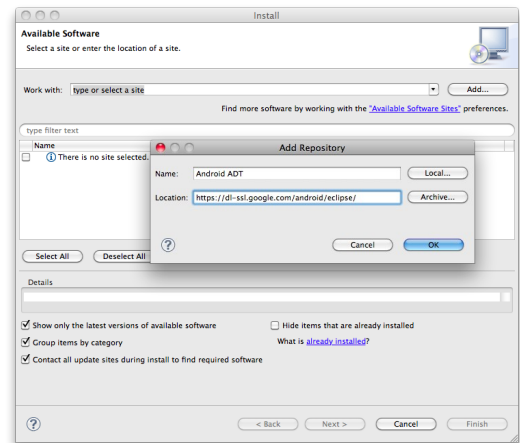
Installation von Eclipse IDE und ADT

Installation von Eclipse

- Eclipse Classic ab Version 3.5 herunterladen (<http://www.eclipse.org/downloads/>).
- Die heruntergeladene Datei in einen Ordner entpacken (z.B.: C:\Entwicklung\Eclipse unter Windows, oder /Users/Shared/Entwicklung/Eclipse unter Mac OS X).
- Eclipse starten und das Standard-Workplace (z.B.: C:\Entwicklung\Android unter Windows oder /Users/Shared/Entwicklung/Android unter Mac OS X) vergeben.

Installation von ADT

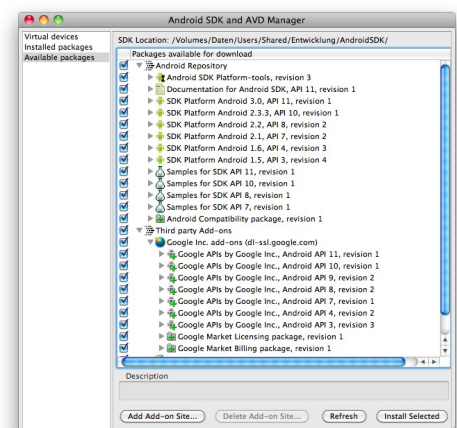
- Android SDK für das gewünschte Betriebssystem herunterladen (<http://developer.android.com/sdk/index.html>).
- Die heruntergeladene Datei in einen Ordner entpacken (z.B.: C:\Entwicklung\AndroidSDK unter Windows oder /Users/Shared/Entwicklung/AndroidSDK unter Mac OS X).
- Eclipse starten und zu **Help** → **Install New Software** gehen.



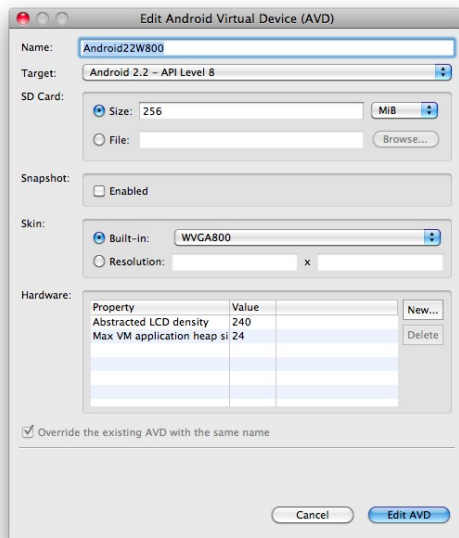
- Bei „**Work with:**“ auf „**Add...**“ klicken, um neues Software-Repository hinzuzufügen (siehe Bild).
- Namen frei vergeben und als „**Location**“ <https://dl-ssl.google.com/android/eclipse/> eingeben
- **OK** klicken.
- Developer Tools als zu installierende

Komponenten auswählen (es dauert einige Zeit, bis die Repository eingelesen wird).

- Die Installation mit „**Next**“ → „**Next**“ → „**Accept**“ und „**Finish**“ abschließen.
- Eclipse neu starten.
- Gehen Sie zu Preferences (unter Windows über **Window** → **Preferences**, unter Mac OS X über **Eclipse** → **Preferences** zu erreichen).
- Gehen Sie zum Punkt „**Android**“ und tragen Sie da



den Speicherort von Android-SDK ein (nach dem Vorschlag wäre es unter Windows der Ordner `C:\Entwicklung\AndroidSDK` und unter Mac OS X `/Users/Shared/Entwicklung/AndroidSDK`).



- Nun gehen Sie zu „**Window**“ → „**Android-SDK and AVD Manager**“ → „**Available packages**“ und installieren Sie die benötigten (z.B.: Android 1.6 und Android 2.2) oder alle SDK-Versionen.
- Nach dem Neustart von Eclipse gehen Sie wieder zu „**Android-SDK and AVD Manager**“.
- Legen Sie unter „**Virtual Devices**“ die virtuellen Android-Geräte, für die Sie entwickeln möchten (z.B.: ein Android 2.2 Gerät mit WXGA Auflösung [800x480] und 2GB SD-Karte). Sie können auch mehrere virtuelle Geräte anlegen und beim Start eines Android-Projektes dann eins davon auswählen.
- Damit sind alle Vorbereitungen für die Android-Programmierung getroffen.