Android

Android-Entwicklung

Android ist mittlerweile ein fester Bestandteil der IT. Aufgrund der großen Verbreitung dieses Betriebssystems und der kostenlosen Tools ist es relativ leicht mit geringem Aufwand Anwendungen für das eigene Smartphone oder Tablet zu schreiben. Im Folgenden wird erklärt, welche Software man benötigt, um selbst Anwendungen zu schreiben und diese im Anschluss auf das Gerät zu übertragen.

Die benötigten Software-Paket sind hier aufgelistet:

1. JDK in einer aktullen Version als Basis zur Java-Entwicklung
2. Android-SDK mit den Android-spezifischen Bibliotheken
3. Eine gute IDE[1)] zum Editieren des Programm-Codes

Nun gibt es immer mehrer Wege, die zum Ziel führen. Es gibt fix und fertige Pakete, die einem „fast“ alles abnehmen. Wenn alles gut geht, hat man die Entwicklungsumgebung im Handumdrehen fertig. Wenn allerdings etwas schief geht und das passiert leider sehr häufig, dann hat man oftmals keine Idee warum es nicht funktioniert.

Daher wird hier die Installation in einzelnen Komponenten vorgenommen, so dass man im Zweifelsfall nachvollziehen kann, woran es hapert.

Name	Zweck	Download
JDK[2)]	Android wird primär in der Programmiersprache Java entwickelt. Das JDK enthält die Basis.	Download
Android-SDK [3)]	Für Android benötigt man spezielle Programm-Bilbotheken Das Android-SDK enthält die Basis.	Download
Eclipse	Entwicklungsumgebung, die per PlugIns fast alles kann	Download

Bevor man startet: Ich empfehle eine eigene Struktur für die Entwickungsumgebung anzulegen.

- `C:\android\eclipse` für die IDE
- `C:\android\sdk` für das SDK
- `C:\android\workspace` für die Projektdateien (eigener Quellcode)
- um das JDK muss man sich keine Gedanken machen, hier ist es günstiger den Standard-Pfad zu nutzen

JDK installieren

Auf der Download-Seite wählt man die entsprechende Plattform (ich beziehe mich zunächst auf Windows).
Für ein Windows 7 64-bit wäre dies zum Beispiel die Datei `jdk-7u40-windows-x64.exe`. `7u40` gibt die Version des JDKs an und `x64` steht für die Plattform also Windows 7 64-bit.

Die Installation ist denkbar einfach. **Vorsicht:** Oracle versucht gerne eine Suchmaschinen-Bar für den Browser zu installieren. Wer dies nicht wünscht sollte also nicht gedankenlos `weiter-weiter-weiter` klicken.

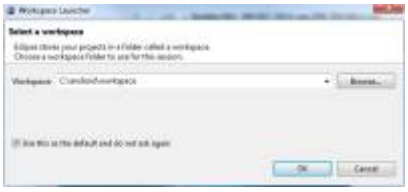
Wer mag, kann sich noch die Java-Dokumentation ansehen oder als Bookmark speichern. Diese ist nützlich, um die Klassen besser zu verstehen. Am Ende der Installation bekommt man diese unter dem Button `More` angeboten.

Eclipse installieren

Die Installation ist ebenfalls nicht weiter schwierig. Einfach die ZIP-Datei in den oben vorgeschlagenen Ordner entpacken. Der Eclipse-Starter sollte dann so liegen: `C:\android\eclipse\eclipse.exe`. Sollte eine Fehlermeldung angezeigt werden, so liegt dies unter Umständen an einer fehlenden oder nicht gefundenen JDK. Also sollte man überprüfen, ob der erste Schritt JDK installieren korrekt ausgeführt wurde. Hier wird leicht JDK mit JRE[4)] verwechselt. Letzteres ist lediglich die Laufzeitumgebung für Java. Sie reicht aber nicht aus, um Java-Programme zu entwickeln.

Wenn Eclipse zum ersten mal startet, kann dies eine Weile dauern. Also Geduld! Eclipse ist vollständig in Java geschrieben und benötigt daher relativ viel Arbeitsspeicher. Dies gilt im übrigen immer im Umgang mit Java. RAM hilft!

Man wird nach dem Ordner für den sogenannten Workspace gefragt:



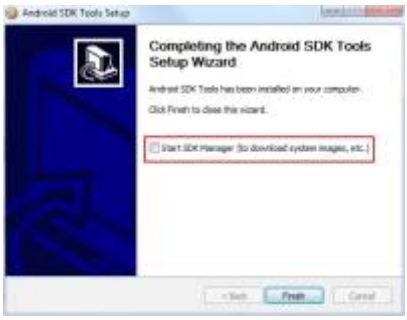
Im Workspace werden alle Projekte abgelegt. Es ist sinnvoll diesen Ordner nicht unterhalb eines Programms zu wählen, sondern parallel dazu, damit nicht aus Versehen seine wertvolle Arbeit löscht, wenn man eine neue Version eines Programms einspielt.

Android-SDK installieren

Bei der Installation des SDKs ist nicht viel zu beachten. Die Datei lautet zum Beispiel `installer_r22.2.1-windows.exe`. Falls es eine neure Version gibt, so kann man auf der Download-Seite im Bereich `USE AN EXISTING IDE` nur das SDK herunterladen.

Wichtig: Den Pfad, in dem das SDK installiert wird, sollte man sich notieren. Während der Installation findet die Detektion des JDKs statt. Wenn dies korrekt installiert ist, dann wird es automatisch angezeigt. Weiter wird man gefragt, ob das SDK für alle User oder nur für den aktuellen installiert werden soll. Als Speicherort kann man selbstverständlich den Standardpfad nutzen. Ich bevorzuge, aber alles in einem Unterordner zu haben (s.o.).

ACHTUNG: Der Installer fragt am Ende nach, ob der SDK-Manager gestartet werden soll. Da wir Eclispe bereits installiert haben, darf der SDK-Manager NICHT direkt im Anschluss gestartet werden. Das wird später erledigt.



Das SDK in Eclipse einbinden

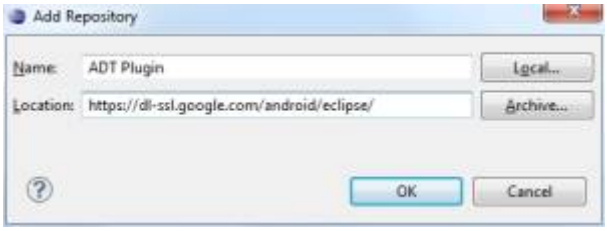
Die folgenden Schritt kommen von google selbst.

Was passiert nun? Wir haben zwar die Bibliotheken auf dem PC installiert, aber uns fehlen noch die sogenannte `ADT-Plugins` mit deren Hilfe aus Eclipse eine Android-IDE wird. Die `ADT-Plugins` sind wenn man so will Scripte oder Mini-Programme, die uns viel Arbeit abnehmen. So kann Eclipse mit Hilfe dieser Plugins einen Android-Projekt-Rumpf anlegen und für die verschiedenen Android Versionen (1.0-4.3) entsprechende Bibliotheken einbinden. Alles sehr praktisch und vor allem notwendig.

Die Schritt aus der Original-Anleitung sehen wie folgt aus:

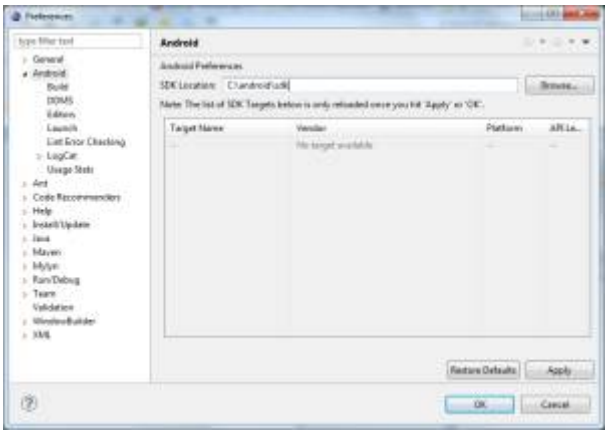
1. Eclipse starten, dann `Help → Install New Software` auswählen.
2. Oben rechts auf `Add` klicken
3. Der Dialog `Add Repository` erscheint. Als *Name* wird `ADT Plugin` und \\als *Location* der folgende Pfad angegeben: `https://dl-ssl.google.com/android/eclipse/`
4. Auf `OK` klicken
5. Im Dialog `Available Software` wählt man `Developer Tools` aus und klickt auf `Next`.
6. Der nächste Dialog zeigt nochmal alle Tools an, die nun runtergeladen werden sollen. Auf `Next` klicken.
7. Die Lizenzen angezeigt und sollten gelesen werden. Anschließend bestätigen und auf `Finish` klicken.
8. Falls eine `security warning` erscheint, die sinngemäß auf nicht validierte bzw. authentifizierte Software hinweist, kann ignoriert werden. Auf `OK` klicken.
9. Nachdem die Installation beendet ist muss Eclipse neu gestartet werden.

Falls man Schwierigkeiten beim Download hat, dann kann man es mit `http` statt `https` versuchen. Hier nochmal der Dialog dazu.



Eventuell kommt es zu einer Fehlermeldung beim Start von Eclipse, da wir den Standardpfad für das SDK ja verändert haben. Dies kann man in den `Windows → Preferences` anpassen. Der Dialog

sollte dann so aussehen:

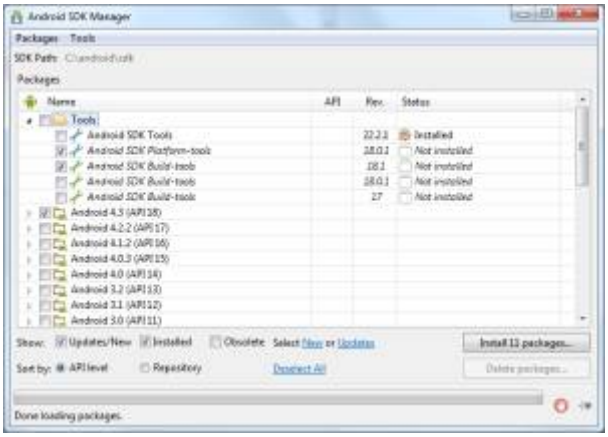


Leider fehlt noch immer etwas. Wir haben Eclipse zwar den SDK-Ordner angegeben, doch dort liegen noch keine versionsspezifischen Bibliotheken. Diese müssen nun noch heruntergeladen werden. Dazu wird der SDK-Manager gestartet:

```
c:\android\sdk\SDK Manager.exe
```

[5)]

Für die Version 4.3 von Android würde der SDK-Manager Dialog so aussehen:



Wenn man weiss, dass „nur“ für ARM-basierte Systeme (das ist die Mehrzahl) entwickeln wird, dann kann man das `Intel Atom Image` abwählen und spart sich ein paar MB. Ähnlich ist dies mit der `Google-API` diese stellt Klassen für den Umgang mit Google-Dienste bereit. Man wird wieder nach Zustimmung der Lizenzen gefragt. Danach kann man erstmal einen Kaffee trinken. Der Download und die Installation dauert eine Weile.

Im Wesentlich sind wir mit der Einrichtung fertig. Nun kann eigentlich mit dem ersten Projekt begonnen werden.

Alternative All-in-One Pakete

Wer es schnell mag dem sei ein Blick auf die beiden folgenden Pakete empfohlen. Wie gut diese allerdings funktionieren kann ich nicht sagen, da ich sie noch nicht getestet habe.

Name	Zweck	Download
Eclipse+SDK=ADT-Bundle[6] for Windows	Eine mögliche Entwicklungsumgebung [7], die alles enthält	Download
Android-Studio	Eine weitere mögliche Entwicklungsumgebung (IDE) im Aufbau befindlich!	Download

Android IDE einrichten

Wenn Eclipse bereits installiert wurde ist der Installationsprozess anders.
Der SDK-Manager sollte dann NICHT am Ende der SDK-Installation gestartet werden!

SDK mit vorhandenem Eclipse einrichten Fortsetzung

Projekt einrichten

Falls libgdx nicht starten will, dann kann das hier helfen: Libgdx in eclipse einbinden

Info Sammlung zu Spieleentwicklung mit Java

Freie Texturen

Libgdx

TexturePacker für Libgdx

Tiled der Map-Editor

1)
IDE: Integrated Development Enviroment; Integrierte Entwicklungsumgebung
2)
JDK: Java SE Development Kit; Sammlung von Java-Programm-Bibliotheken
3)
SDK: Software Development Kit; Sammlung von Android-Programm-Bibliotheken
4)
Java-Runtime Envirorment
5)

da man hier ab und zu mit arbeiten muss, liegt er nah bei der IDE. Entwickler sind nun mal faul!
6)
ADT: Android Development Tools
7)
IDE

From:
http://www.kopfload.de/ - **kopfload - Lad Dein Hirn auf!**

Permanent link:
http://www.kopfload.de/doku.php?id=allgemein:android&rev=1380378636

Last update: **2025/11/19 16:12**