

Praktikum 1 – Grundlagen und erste Schritte

Präambel

Im Rahmen dieses Praktikums sollen Sie sich zunächst mit der generellen Verwendung der Werkzeuge zur Entwicklung von Android-Anwendungen vertraut machen

Auf den Rechnern im Pool SI 0024/25 ist unter Windows für die App-Entwicklung **Android Studio** als IDE installiert. Sie sollten sich auf den Rechnern mit Ihrem Standard-Account anmelden können.

Nach Möglichkeit sollten Sie immer am gleichen Rechner arbeiten. Der Grund dafür ist, dass sich Android Studio eigentlich gar nicht für den Betrieb in Pool-Umgebungen eignet. Die lokal erzeugten Projekte können recht groß werden, vor allem im Hinblick auf die Anzahl von Dateien. Insofern ist es auch ratsam, die Projekte *lokal* und *nicht im Profil* zu speichern, da dies spätere Anmelde-Vorgänge massiv verzögern kann.¹ Erfahrungsgemäß führt außerdem die Ablage der Projekte im zentralen Home-Verzeichnis (aus den gleichen Gründen) zu erheblichen Wartezeiten aufgrund der Latenzen hinsichtlich der Dateitransfers über das Netzwerk.

Natürlich können Sie die lokalen Daten nach Praktika der Arbeiten an einem Projekt mit dem Datenserver synchronisieren und so dauerhaft erhalten.

Im Betrieb werden *kontinuierlich Updates* von IDE-Komponenten angeboten, was Administrationsrechte auf dem Rechner bedingt. Auf diese Updates sollten Sie verzichten.

Wenden Sie sich an die Betreuung im Praktikum, falls es Schwierigkeiten beim Zugang gibt.

Einarbeitung Kotlin

Überprüfen Sie zunächst die **Installation** von Android Studio und melden Sie sich, wenn sich die Anwendung nicht starten lässt (hier ist beim ersten Start etwas Geduld vonnöten).

- Bevor Sie mit der Bearbeitung der Aufgaben beginnen, sollten Sie einmal <https://kotlinlang.org/docs/basic-syntax.html> durchgegangen sein. Die Code-Beispiele in Kotlin lassen sich in einer web-basierten Playground-Applikation ausführen und ggf. auch abändern.
- Auf der o.g. Web-Seite befinden sich weitere Informationen zur Programmierung mit Kotlin. Wichtig sind hier die Kapitel zu *Lambdas* (https://play.kotlinlang.org/byExample/04_functional/01_Higher-Order%20Functions) und *Scope Funktionen* (https://play.kotlinlang.org/byExample/06_scope_functions/01_let).
- Im Lernraum finden Sie ein Archiv eines Kotlin-Kurses (**Kotlin_Starter.zip**). Entpacken Sie das Archiv und öffnen Sie das Projekt in Android Studio. Sie finden dann ein paar kleinere Aufgaben vor, mit denen Sie Ihren Wissensstand überprüfen können. Zumeist sind die vorgesehen Platzhalter zu editieren. Zu den Aufgaben werden auch Hinweise zur Lösung

¹ Wenn man trotzdem Projekte im Profil anlegen möchte, dann sollte vor dem Ausloggen ein ‚Clean‘ in Android Studio durchgeführt werden.

gegeben. Versuchen Sie jedoch zunächst eine Lösung anhand der Dokumentation unter <https://kotlinlang.org/docs/home.html> zu entwickeln.

- Eine Übersicht der Kotlin-Syntax findet Sie unter der URL https://koenig-media.raywenderlich.com/uploads/2018/08/RW_Kotlin_Cheatsheet_1_0.pdf.

Einfache Android App

Arbeiten Sie sich dann anhand des **Android-Tutorials HelloWorld** auf der Seite <https://developer.android.com/training/basics/firstapp/creating-project.html> in die Benutzung der IDE und der Android-Programmierung ein.

- Bei der Konfiguration des Projektes können Sie unter *Language* die zu verwendende Programmiersprache (stellen Sie *Kotlin* ein) auswählen.
- Unter *Running Your App* sollten Sie nur den Punkt *Run on the Emulator* behandeln. Das Debugging der Applikation über die USB-Schnittstelle auf einem realen Gerät werden wir uns zu einem späteren Zeitpunkt ansehen.

Hinweise

Beachten Sie dabei die folgenden **Hinweise**:

- Android Projekte können recht groß werden. Vor allen Dingen werden hier sehr viele Dateien erzeugt. Durch den Wizard wird Ihnen vorgeschlagen, diese im Windows-Profil zu speichern, was den Anmelde-Prozess signifikant verlängert. Speichern Sie das Projekt stattdessen auf dem **Arbeitsrechner lokal** auf der **D:** Partition.
- Images von neuen zu emulierenden virtuellen Geräten (AVDs) sollten ebenfalls **D:** gespeichert werden. Sie sollten sich also beim Wechsel des Arbeitsplatzes nicht wundern, wenn die zuvor erstellten AVDs am neuen Arbeitsplatz nicht mehr vorhanden sind.
- Achten Sie darauf, den **Emulator** nach einem Test nicht direkt wieder zu schließen, damit die längliche Start-Phase (in der ein virtuelles Gerät gebootet wird) nur einmal durchlaufen werden muss.
- Sie sollten **keine Updates** oder neue SDKs mit dem SDK Manager herunterladen!
- Beim ersten Start von Android Studio wird der **Wizard** aufgerufen.
 - Dort zunächst immer auf *next* drücken.
 - Fehler und Warnungen können ignoriert werden.
 - Ab dem zweiten Start kann der Wizard mit *close* beendet werden. Sie sollten den Haken, um den Wizard beim nächsten Mal wieder zu starten, nicht setzen.
- Prüfen Sie im Android Studio anhand der **Statuszeile** unten stets, ob noch Build-Prozesse aktiv sind. Sie sollten jeweils erst weiterarbeiten, wenn diese Prozesse abgeschlossen sind, da sonst unter Umständen im Editor irrtümlich Fehler angezeigt werden können.

Im Folgenden finden Sie einige sinnvolle Einstellungen für **Android Studio**:

- Zeilennummern einblenden:

- *File -> Settings... -> Editor -> General -> Appearance -> Show line numbers*
- Auto Import bei Copy & Paste:
 - *File -> Settings -> Editor -> General -> Auto Import*
 - *Insert import on paste* auf *All* setzen
 - *Optimize imports on the fly* anhängen
- Automatische Formatierung des Codes (ggf. bzgl. Auswahl):
 - *<STRG>-<ALT>-L*
- Code Completion (Autovervollständigung)
 - *<STRG>-<SPACE>*
- Block Kommentare
 - *<STRG>-<SHIFT>-/* (nur auf dem Numblock möglich)
 - Wer den Schrägstrich über der "7" verwenden will muss die Keymap ändern:
 - *Settings -> Keymap*
 - *Toggle Bookmark 7* entfernen
 - Unter *Comment Block*, neuen mit *<STRG>-<ALT>-7* hinzufügen