

Praktikum Mobile und Verteilte Systeme (MSP)

Sommersemester 2019

Übungsblatt 1

Ziel der heutigen Praxisveranstaltung ist die Einrichtung einer Android-IDE (Android Studio) sowie das Programmieren einer ersten Anwendung für Android-basierte Endgeräte. Detaillierte Informationen zur Einrichtung von Android finden sich unter:

<https://developer.android.com/studio/index.html>

Aufgabe 1: Activities, Services und Nebenläufigkeit

Nachdem nun ein erster Einstieg geschafft ist, werden Sie in dieser Aufgabe die Verwendung von *Activities* und *Services* sowie die komponentenübergreifende Kommunikation erlernen. Zu diesem Zweck sollen Sie eine App programmieren, die im Hintergrund Bilder aus dem Internet herunterladen kann. Über einen Button kann der Nutzer den Download-Vorgang starten. Den Download-Fortschritt kann der Nutzer mit Hilfe eines *ProgressBar*-Objekts für jeden Download einzeln verfolgen. Sobald ein Download fertig ist, wird der Fortschrittsbalken durch eine Miniaturansicht des heruntergeladenen Bildes ersetzt. Achten Sie beim Programmieren auf einen guten Stil: Beispielsweise sollten String-Literale bei der Programmierung mit Android immer als *resource* verwaltet und nicht statisch im Code definiert werden.

- a. Legen Sie ein neues *Android Application Project* an. Vergeben Sie sinnvolle Namen und beginnen Sie mit einer leeren *Activity*. Versuchen Sie als erstes, das Layout für die neue *Activity* in XML festzulegen. Am Ende sollte es in etwa so aussehen wie der unten angefügte Screenshot. Sie können zu diesem Zweck den grafischen UI-Editor von Android Studio verwenden. Legen Sie zusätzlich zu jeder der fünf *ProgressBar*-Instanzen auch noch je ein *ImageView*-Objekt an. Setzen Sie die Sichtbarkeit dieser Bildanzeigen zunächst jedoch auf GONE.
- b. Versuchen Sie nun, im Programmcode der *Activity* auf die einzelnen *View*-Objekte zuzugreifen. Nachdem wir diese Elemente später alle verwenden werden, legen sie für alle Objekte geeignete Instanzvariablen an. In der `onCreate()`-Methode der *Activity* können Sie die Methode `findViewById()` verwenden, um die Verbindung zwischen XML-Layout und Programmcode herzustellen. Dazu müssen Sie Ihrer *Activity* noch das entsprechende Layout zuweisen.
- c. Bei einem Klick auf den GO!-Button soll ein *Service* gestartet werden, der das Herunterladen der Bilddateien übernimmt. Legen Sie hierfür zunächst eine neue Klasse an, die von *Service* erbt. Mit Hilfe eines `onClickListeners` können Sie auf Nutzereingaben horchen. In der entsprechenden Event-Behandlung können Sie so den *Service* mit Hilfe eines geeigneten *Intent*-Objekts starten. Bei einem Klick auf den Kill-Button soll die App vollständig beendet werden.

- d. Kümern wir uns nun um den Service. Dieser soll im Hintergrund Bilddateien aus dem Internet herunterladen. Achten Sie bei Ihrer Implementierung darauf, dass diese potentiell langwierige Aufgabe tatsächlich im Hintergrund abläuft und nicht den UI-Thread blockiert. Das Herunterladen können Sie beispielsweise mit Hilfe eines *URLConnection*-Objekts umsetzen. Sollten Probleme auftreten, behalten sie die Logausgabe Ihrer IDE im Auge - hier werden Sie nützliche Hinweise finden. Ob Sie den Download der Bilder parallel oder sequentiell durchführen, ist Ihnen überlassen.
- e. Nun geht es darum, die komponentenübergreifende Kommunikation zwischen der *Activity* und dem soeben erstellen *Service* umzusetzen. Im Rahmen der heutigen Übung soll dies mit Hilfe des *BroadcastReceiver*-Mechanismus und entsprechenden *Intent*-Nachrichten realisiert werden. In der *Activity* muss hierzu ein geeigneter *BroadcastReceiver* angelegt und registriert werden. Der *Service* verschickt *Intents* mit entsprechendem Inhalt. Der *BroadcastReceiver* in der *Activity* empfängt diese Nachrichten und aktualisiert den jeweiligen Fortschrittsbalken.
- f. Sobald ein Bild vollständig heruntergeladen wurde, soll der Fortschrittsbalken durch eine Miniaturansicht des Bilds ersetzt werden. Die Bilddaten können Sie erneut via *Intent* vom Service an die Activity weiterreichen. Weisen Sie dem *ImageView*-Objekt die Bilddaten zu. Um den Austausch zu realisieren, genügt es nun, die Sichtbarkeit des Fortschrittsbalkens auf *GONE*, die der Bildanzeige auf *VISIBLE* zu setzen.
- g. Testen Sie, ob Sie Ihre App verlassen können (z.B. auf den Homescreen zurückkehren), ohne dass der Download der Bilddaten davon unter- bzw. abgebrochen wird.

