

Aufgabe 001

[ADT Installation]

Installieren Sie das Android SDK oder das Bundle ADT. Optional installieren Sie das Intel Hardware Accelerated Execution Manager (HAXM).

Aufgabe 002

[Hello-World-App]

Erzeugen Sie Ihre erste "Hello World"-App mit einem TextView und lassen Sie sie sowohl im Simulator als auch, wenn möglich, auf einem Android-Handy laufen.

[Start: <http://developer.android.com/training/basics/firstapp/index.html>]

Aufgabe 003

[Device Management]

Erzeugen Sie ein neues Android Virtual Device im Android Virtual Device Manager und simulieren Sie Ihre App dort. Variieren Sie die Einstellungen und erzeugen zu diesem Zweck auch neue Devices.

Aufgabe 004

[Layout, Button, Textfeld]

Erzeugen Sie weitere Buttons und Textfelder und verstehen Sie die Verwendung von IDs und (String)Ressourcen.

Aufgabe 005

[Layout, Button, Click-Event]

Ergänzen Sie Ihre App aus Aufgabe 2 oder 4 um einen Button und implementieren Sie einen Click-Event-Handler. Führen Sie das unten genannte Tutorial aus und starten Sie eine neue Activity.

[<http://developer.android.com/training/basics/firstapp/starting-activity.html>]

Aufgabe 006

[LinearLayout, RelativeLayout]

Modifizieren Sie Ihre App aus Aufgabe 2, benutzen Sie LinearLayout horizontal und vertikal und füllen die Layouts mit Texten und Buttons oder Controls Ihrer Wahl. Variieren Sie insbesondere die Gewichte der Controls.

Ersetzen Sie nun das Layout durch ein RelativeLayout und passen Sie die Attribute auf das neue Layout an.

[<http://developer.android.com/training/basics/firstapp/building-ui.html>

<http://developer.android.com/guide/topics/ui/overview.html>]

Aufgabe 007

[GridLayout]

Ersetzen Sie in einer Ihrer Apps das verwendete Layout durch ein GridLayout und experimentieren Sie mit der Anordnung. Testen Sie die Konfiguration aus dem Beispiel. Passen Sie die minimal benötigte API Version (14) an.

[<http://developer.android.com/reference/android/widget/GridLayout.html>

<http://android-developers.blogspot.de/2011/11/new-layout-widgets-space-and-gridlayout.html>]

Übungsaufgaben

Aufgabe 008

[Lifecycle]

Programmieren Sie eine App und implementieren Sie alle möglichen Callbacks (onStart usw.), die beim Wechsel des Zustandes der App (Lifecycle) aufgerufen werden. Lesen Sie dazu das angegebene Kapitel "Managing the Activity Lifecycle". Geben Sie dazu Statusmeldungen im LogCat oder als Toast aus. Testen Sie diese App auch auf dem Handy, wenn möglich, und drehen Sie insbesondere das Handy. Suchen Sie hierfür die Tastenkombination für den Simulator.

[<http://developer.android.com/training/basics/activity-lifecycle/index.html>
<http://developer.android.com/tools/debugging/debugging-log.html>
<http://developer.android.com/guide/topics/ui/notifiers/toasts.html>]

Aufgabe 009

[Lifecycle]

Erweitern Sie Aufgabe 008 um Callbacks für onRestoreInstanceState und onSaveInstanceState und ergänzen Sie Ihre App um Key-Value-Paare, die Sie im Bundle in onSaveInstanceState abspeichern und in onRestoreInstanceState wieder laden.

[<http://developer.android.com/training/basics/activity-lifecycle/recreating.html>]

Aufgabe 010

[Data save]

Programmieren Sie eine App, die Daten in Files ablegt und wieder lädt.

[<http://developer.android.com/training/basics/data-storage/files.html>]

Aufgabe 011

[ListView]

Programmieren Sie eine App mit einem ListView mit Strings und verwenden Sie einen ArrayAdapter<String>. Initialisieren Sie den Adapter einmal mit einem String[] und einmal mit einer ArrayList<String>. Was passiert, wenn Sie ein Element zum Adapter hinzufügen (add)?

Aufgabe 012

[ListView]

Ergänzen Sie Ihr Programm aus Aufgabe 11 durch Buttons zum Hinzufügen und Löschen von Einträgen im ListView. Was macht in diesem Zusammenhang die Funktion notifyDataSetChanged()?

[<http://developer.android.com/guide/topics/ui/layout/listview.html>
<http://developer.android.com/guide/topics/ui/declaring-layout.html#Adapter-Views>]

Übungsaufgaben

Aufgabe 013 [Layout]

Programmieren Sie eine App, die unterschiedliche Layouts für Portrait und Landscape anzeigt.

[<http://developer.android.com/training/basics/supporting-devices/screens.html>]

Aufgabe 014 [Fragment]

Programmieren Sie eine App mit einem Fragment 1 für die Anzeige im Portrait-Modus und mit zwei Fragmenten 1 und 2 für die Anzeige im Landscape-Modus (Nutzen Sie das Fragment 1 in beiden Modi).

[<http://developer.android.com/training/basics/fragments/index.html>
<http://developer.android.com/guide/components/fragments.html>]

Aufgabe 015 [Fragment]

Programmieren Sie eine App, die auf einer Activity zwischen mehreren (mind. drei) Fragmenten dynamisch umschalten kann.

Aufgabe 016 [Intent]

Programmieren Sie eine App, die unterschiedliche andere Apps via Intent aufruft und verwendet. Nutzen Sie (soweit vorhanden) die Apps: Phone, Mail, Kalender, Kontakte, Web.

[<http://developer.android.com/training/basics/intents/index.html>
<http://developer.android.com/training/basics/intents/sending.html>]

Aufgabe 017 [Intent, Camera, ImageView]

Programmieren Sie eine App, die ein Foto machen kann und zeigen Sie dieses in einem ImageView an.

[<http://developer.android.com/training/camera/index.html>
<http://developer.android.com/guide/topics/media/camera.html>]

Aufgabe 018 [Intent]

Programmieren Sie zwei Apps. Eine davon wandelt, quasi als Dienst, einen per Intent übergebenen String in Großbuchstaben um und die andere nutzt diesen "Uppercase"-Service.

[<http://developer.android.com/training/sharing/index.html>]

Aufgabe 019 [ActionBar]

Ergänzen Sie in einer Ihrer Apps einen Button in der ActionBar, der nach Drücken die ActionBar (für immer?) verschwinden lässt.

[<http://developer.android.com/training/basics/actionbar/index.html>
<http://developer.android.com/guide/topics/ui/actionbar.html#Tabs>]

Übungsaufgaben

Aufgabe 020 [Sensor]

Ermitteln Sie, welche (Motion)Sensoren in Ihrem Handy zur Verfügung stehen, lesen Sie die Werte aus und zeigen Sie sie an.

[http://developer.android.com/guide/topics/sensors/sensors_motion.html]

Aufgabe 021 [Touch, Gestures]

Verfolgen Sie die Koordinaten eines Fingers auf dem Handy. Identifizieren Sie darüber hinaus Gesten wie SingleTap, Scroll, Fling oder LongPress.

[<http://developer.android.com/training/gestures/index.html>]

Aufgabe 022 [Touch, Gestures]

Bewegen/Schieben Sie eine Grafik oder ein Icon mit dem Finger über den Bildschirm.

Aufgabe 023 [Views]

Implementieren Sie ein eigenes View und lassen Sie es sich im Layout anzeigen. Orientieren Sie sich an dem Beispiel PieChart.

[<http://developer.android.com/training/custom-views/index.html>]

Aufgabe 024 [Datenbanken]

Sofern Sie Kenntnis in Datenbanken bzw. SQL-Statements haben, implementieren Sie eine einfache Datenbank für Namen und Telefonnummern. Sie finden erste Informationen dazu hier:

[<http://developer.android.com/training/basics/data-storage/databases.html>
<http://developer.android.com/guide/topics/data/data-storage.html#db>]

Beachten Sie auch App 007.

Aufgabe 025 [Async]

Programmieren Sie eine App, wo nach einem Button-Click im Hintergrund ein Zähler von 1 bis 10 im Sekundentakt hochgezählt und angezeigt wird. Solange gezählt wird, ist der Button disabled. Ihre App soll weiterhin bedienbar sein.

[<http://developer.android.com/reference/android/os/AsyncTask.html>]

Aufgabe 026 [Async, Notification]

Erweitern Sie Aufgabe 25 um eine Notification, dass der Zähler bei 10 angekommen ist.

[<http://developer.android.com/design/patterns/notifications.html>
<http://developer.android.com/training/notify-user/index.html>]

Übungsaufgaben

Aufgabe 027

[TimePicker, DialogFragment, Notification]

Zeigen Sie nach einem Button-Click einen Dialog zur Auswahl einer Zeit (TimePicker) an und zeigen Sie das Ergebnis in Ihrer App an, wenn der Dialog nicht abgebrochen (cancel) wurde.

[<http://developer.android.com/guide/topics/ui/controls/pickers.html>]

Analog zu Aufgabe 26 geben Sie eine Notification aus, wenn der Dialog abgebrochen wurde.

Aufgabe 028

[Service, IntentService, Notification]

Angenommen Sie haben eine Passwortverwaltungs-App, die sich, nachdem Sie sich eingeloggt haben, immer nach einer gewissen Zeit wieder ausloggt bzw. ausloggen möchte.

Schreiben Sie einen Service (lokal), den Sie einmal aktivieren und der nach 10s eine Notification ausgibt, die vorschlägt, sich wieder auszuloggen.

Erweitern Sie den Service so, dass Sie ihn abschalten können.

Legen Sie ein (Passwort)Eingabefeld auf das Layout Ihrer Activity und lassen Sie den Service das Eingabefeld immer nach 10s löschen (nerv...).

Dokumente zum Thema *Service* finden Sie hier:

[<https://developer.android.com/guide/components/services.html>

<https://developer.android.com/training/run-background-service/index.html>

<https://developer.android.com/guide/components/intents-filters.html>

<https://developer.android.com/guide/components/bound-services.html>]

Aufgabe 029

[Service, IntentService]

Schreiben Sie einen Service (lokal), den Sie aktivieren und deaktivieren können und der alle Datenbankoperationen übernehmen soll (alternativ File-Operationen), z.B. als Erweiterung von Aufgabe 24.

Sie benötigen dafür nicht nur einen Service, sondern auch eine Art "Sprache", wie Sie mit dem Service kommunizieren und in welcher Form dieser seine Ergebnisse zurück liefert. Stellen Sie die Daten, z.B. das Ergebnis eines SQL-Queries, auf der Activity geeignet dar.

Aufgabe 030

[Service]

Erweitern Sie Aufgabe 29 so, dass Ihr Service als eigenständiger Service läuft, der von mehreren Apps genutzt werden kann.

Übungsaufgaben

App 001 [Buttons, Textfelder]

Programmieren Sie einen Taschenrechner.

- Dieser beherrscht die Grundrechenarten.
- Er arbeitet mit UPN.
- Man sieht die letzten zwei Eingaben bzw. Ergebnisse auf dem Stack.

App 002 [Activities, InstanceStates]

Programmieren Sie eine Master-Passwort-App.

- Auf dem "Login"-Bildschirm (MainActivity) kann man durch Eingabe eines korrekten Passwortes den Zugang zu einer zweiten Activity freischalten.
- Speichern Sie das Master-Passwort im privaten Datenbereich der App.

[<http://developer.android.com/training/basics/data-storage/index.html>]

App 003 [ListView]

Erweitern Sie Aufgabe 12 zu einer Einkaufsliste.

App 004 [Camera]

Kontrollieren Sie die Camera.

[<http://developer.android.com/training/camera/cameradirect.html>]

App 005 [Intent]

Programmieren Sie eine Notruf-App, die aus der Menge der vorhandenen Kontakte einen bestimmten Kontakte heraussucht und direkt anruft (oder alternativ eine Mail schreibt bzw. SMS sendet).

App 006 [Sensor]

Programmieren Sie eine Wasserwaage.

App 007 [Fragments, ListView, Datenbanken, File-IO, Async]

Programmieren Sie eine Kontakte-App, deren Daten in einer Datenbank oder einer Datei abgelegt sind. Ihre App sollte mindestens enthalten:

- Eine Ansicht, um Kontakte auszuwählen bzw. zu suchen.
- Eine Ansicht, um Kontakte zu pflegen, d.h. Sie anzulegen, zu ändern und zu löschen.
- Von einem Kontakt aus direkt das Telefon aufzurufen. Alternativ einen Messenger.
- Nutzen Sie AsyncTask für längere Datenbankoperationen.

App 008 [Stoppuhr,Async]

Programmieren Sie eine Stoppuhr, bei der die Zeit im Hintergrund läuft und zu sehen ist.

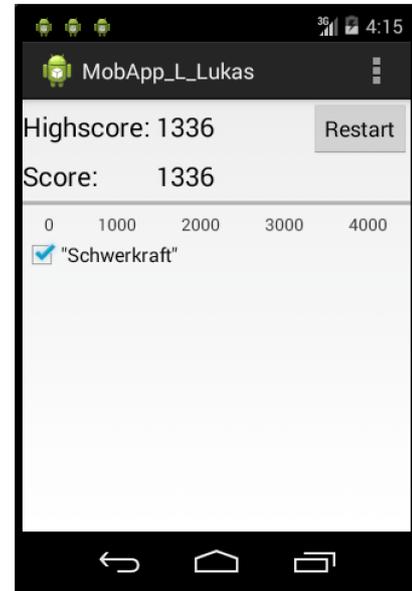
Übungsaufgaben

App 009

[Async, Fling, Notification]

Ermitteln Sie die maximale Geschwindigkeit (Highscore) eines waagerechten Flings und zeigen diesen Highscore in einem Fortschrittsbalken an.

- Wie dargestellt gibt es Textfelder für den Highscore und den letzten Versuch. Dort wird die jeweilige Geschwindigkeit angezeigt.
- Der Fortschrittsbalken zeigt grafisch, den Highscore an. Max. Wert sei 5000.
- Ein Button "Restart" setzt den Highscore auf 0 zurück.
- Ist die (waagerechte) "Schwerkraft" aktiv, so wird der Fortschrittsbalken jede Sekunde um 100 verringert. Eine Checkbox aktiviert oder deaktiviert dieses Feature.
- Wird der Highscore geknackt, so spendieren Sie eine Notification über diesen Erfolg.



App 010

Programmieren Sie eine ToDo-Liste bzw. eine Aufgabenverwaltung. Folgende Anforderungen könnten interessant sein:

- Ein ToDo hat mindestens eine Beschreibung, eine Priorität, ein Erledigt-Flag und optional ein Datum, bis zu dem es erledigt sein soll.
- Todos können angelegt und gelöscht werden.
- Eigenschaften können geändert werden.
- Ist ein ToDo erledigt, gibt es einen (Belohnungs) Jingle.
- Die Todos werden in einer Liste angezeigt, die nach Eigenschaften wie etwa der Priorität oder dem Datum geordnet werden kann.
- Die Menge der Todos wird abgespeichert und steht beim Neustart wieder zur Verfügung.

Überlegen Sie sich für jede Anforderung eine geeignete technische Umsetzung und ein ansprechendes GUI.

Übungsaufgaben

Projekt MobApp_...	Stichpunkte
A_FirstDemo	<ul style="list-style-type: none"> • Basics • Intents • Intent Data • Activity starten • Views via findViewById finden
B_GettingStarted	<ul style="list-style-type: none"> • ...
C_NoFragments D_OneFragment E_TwoFragments F_FragmentUsage	<ul style="list-style-type: none"> • ...
G_StartIntents	
H_RequestUpper H_UpperService	
I_Gui1 I_Gui2 I_Gui3 I_Gui4 I_Gui5 I_Gui6	
J_Sensors	
K_Mouse1 K_Mouse2 K_Skribble	
L_AsyncTask	
L_Database	
L_Dialogs	
L_Lukas	
M_Notifications	
N_Media1	
O_AIDLService O_UseService1 O_UseService2	
P_GPS	

