

Android-Grundlagen

e-Portfolio von Heiko Faller



Abb. 1: Android-Roboter

Agenda

1. Android
2. Aufbau einer Android-App
3. Material Design
4. Android-Studio
5. Erste Schritte
6. Übung



Android

- mobiles Betriebssystem
- Marktanteil in Deutschland: ca. 75% (Aug. 2015)
- basiert auf Linux
- Daher Open Source



Aufbau einer Android-App

Android-App

Source-Code Java

- Programmlogik
- Bibliotheken

Manifest XML

- Verknüpfung
- Beinhaltet Informationen für das System
 - Name/Icon
 - Permissions
 - Activities
 - Services

Ressourcen XML, Grafiken, ...

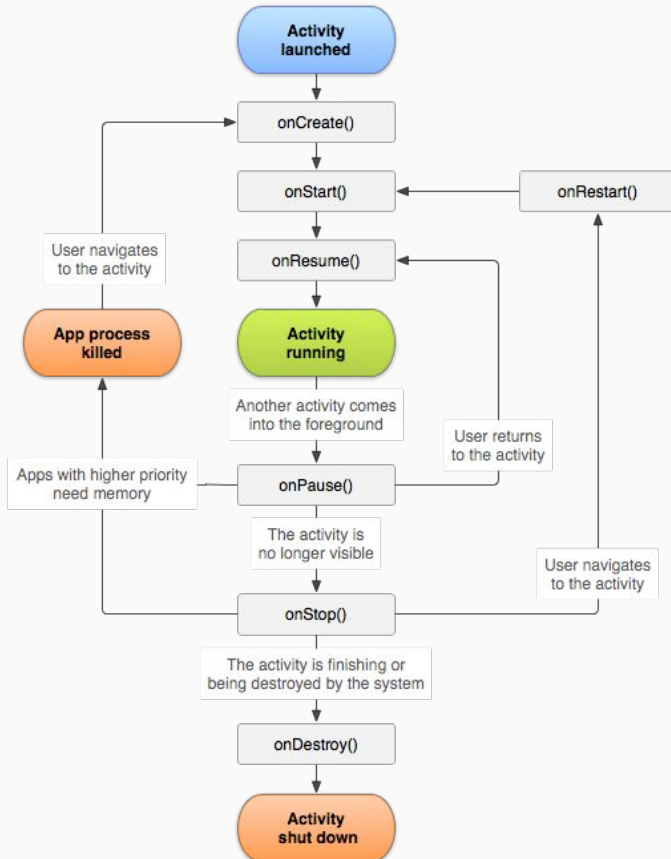
- UI
- Werte
 - Strings
 - Arrays
 - Farben
 - ...
- Grafiken



- Etwas, kleines fokussiertes, was der Nutzer tun kann
- Erbt von “Activity”
- Zuständig für den Zugriff auf Ressourcen
- Also auch für die Anzeige der Layouts



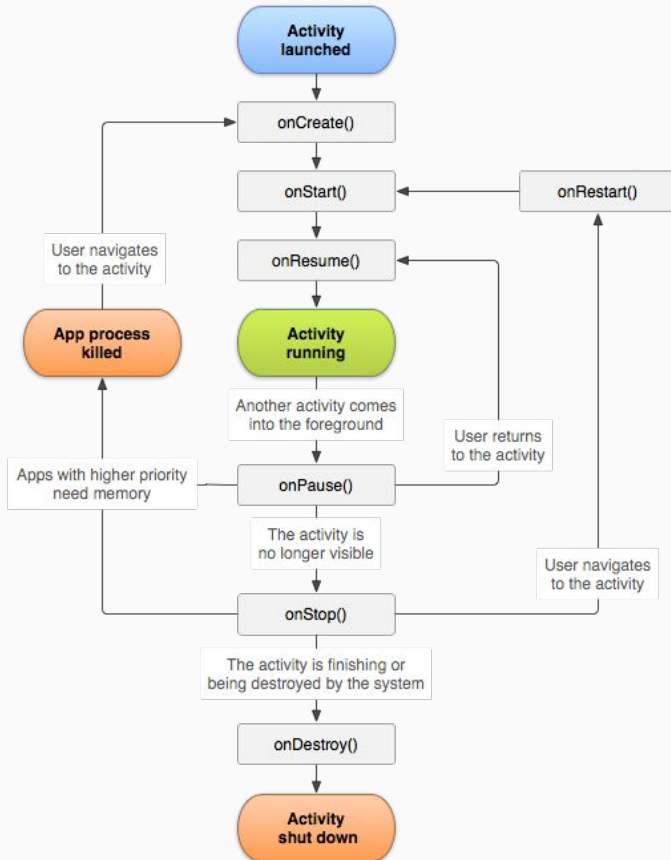
Activities



- `onCreate()` beim ersten Start der Activity
- `onDestroy()` wenn das System oder die Activity sich selbst beendet
- ...



Abb. 2: Lebenszyklus einer Activity:



- Auf verschiedene Ereignisse reagieren
- Zum Beispiel Daten oder Zustände speichern und wieder laden
- ohne aktives Eingreifen des Nutzers



Abb. 2: Lebenszyklus einer Activity

Material Design

- Design-Richtlinie für Android-Apps
- Sorgt für Einheitlichkeit
- Ist intuitiver
- Soll Dreidimensionalität durch Schichten visualisieren
- Enthält Vorgaben zu Animationen, Farben, Icons, Typographie, ...



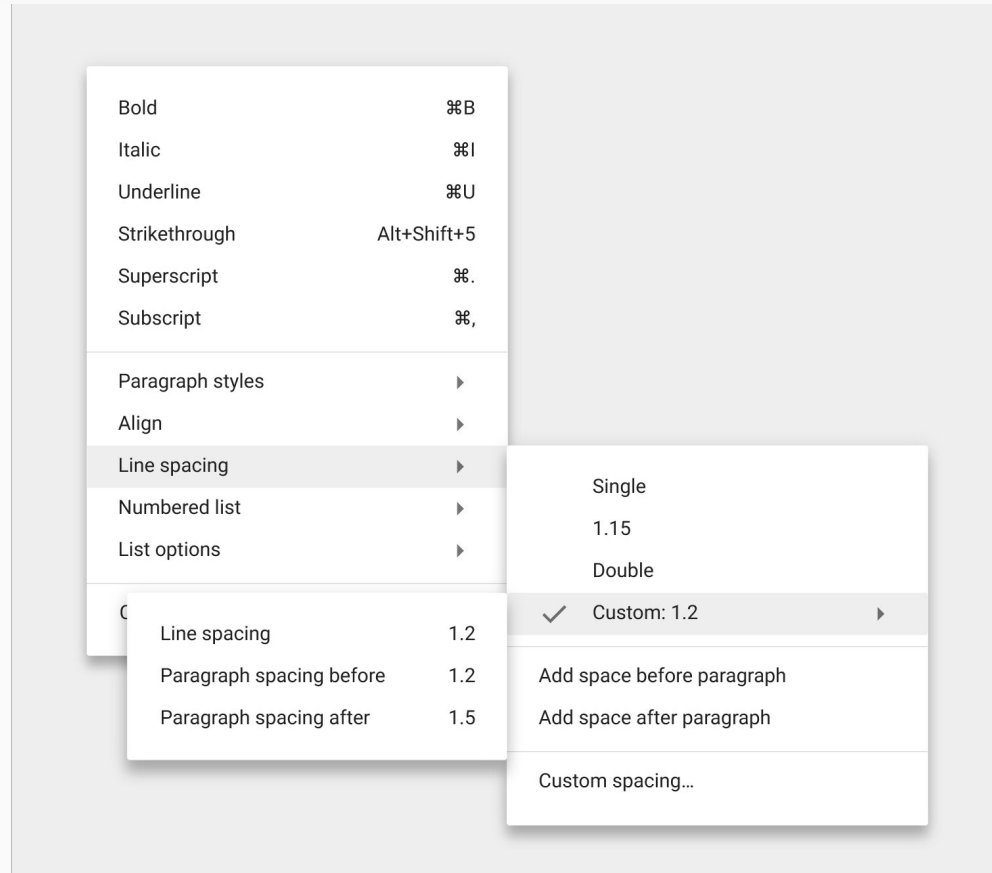


Abb. 3: Material Design



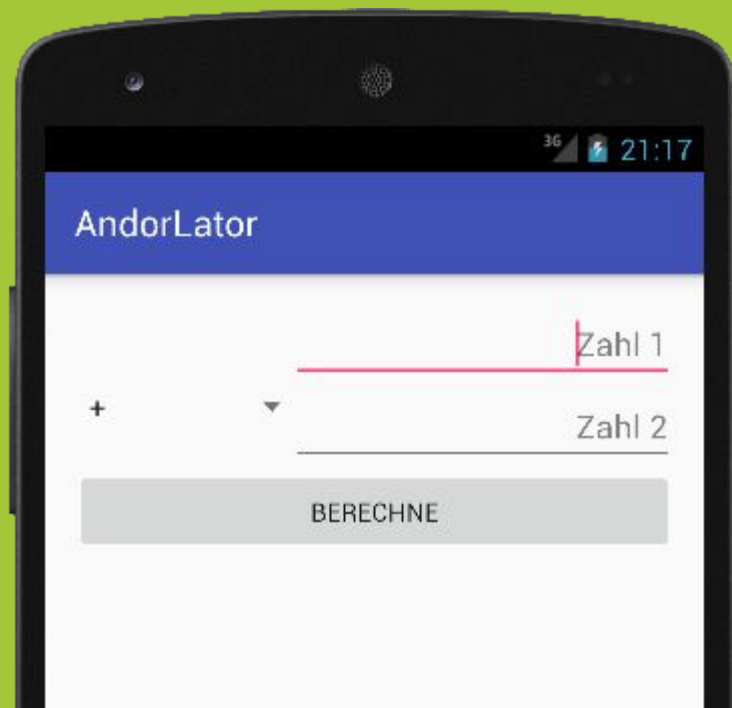
Android-Studio

- IDE basierend auf IntelliJ IDEA
- Speziell für Android-Entwicklung angepasst
 - angepasste Datei-Hierarchie
- Beinhaltet unter anderem
 - Android Virtual Device Manager
 - viele Tools zur Analyse und zum Monitoring



Erste Schritte

Übung



- Einfacher Taschenrechner
- Zwei Zahlenfelder und ein Spinner für die Operatoren
- Button zum Start der Berechnung
- Label zur Ausgabe des Ergebnisses
- Benachrichtigung des Nutzers bei Falscheingaben per Toast
- Texte und Beschriftungen sollen in den Ressourcen abgelegt werden
- Es sollen die Sprachen Deutsch und Englisch unterstützt werden.



Erweiterung:

- Ergebnisse werden mit Rechnung untereinander in einer Liste angezeigt.
- Diese Liste soll Scrollbar sein



Anzeigen eines Layouts:

```
setContentView(Layout-ID oder View);
```

Einbinden eines Views aus einer Ressource in den Java-Code:

```
EditText textfeld = (EditText) findViewById( ID des Textfeldes );
```

Kurze Nachricht an den Nutzer

```
Toast.makeText(getApplicationContext(), "Text", Toast.LENGTH_SHORT).show();
```

Praktische Links:

developer.android.com

stackoverflow.com



Android:

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

<http://www.kantarworldpanel.com/global/smartphone-os-market-share/>

Aufbau einer Android-App:

<http://developer.android.com/reference/android/app/Activity.html>

Material Design:

<http://www.google.com/design/spec/material-design/>

Android-Studio:

<http://developer.android.com/tools/studio/index.html>

Abb. 1: Android-Roboter:

<http://developer.android.com/distribute/tools/promote/brand.html>

The Android robot is reproduced or modified from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License.

Abb. 2: Lebenszyklus einer Activity:

<http://developer.android.com/reference/android/app/Activity.html>

Abb. 3: Material Design:

<http://www.google.com/design/spec/what-is-material/elevation-shadows.html>

Vielen Dank