

# Software Engineering 2 Tutorium Android Programmierung

Monika Brodbeck

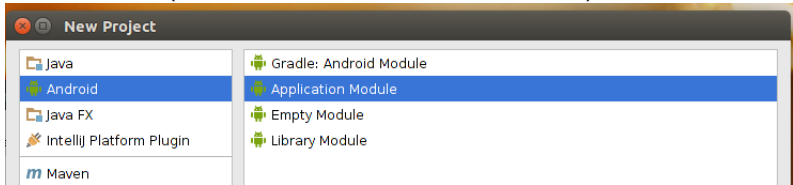
10.03.2015

# Links

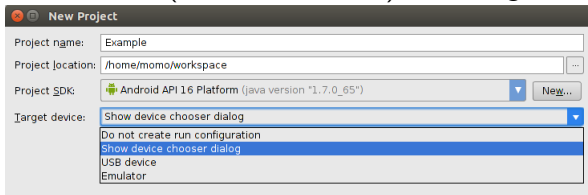
- Android Studio  
<https://developer.android.com/sdk/installing/index.html?pkg=studio>
- IntelliJ Idea + Android SDK  
<https://developer.android.com/sdk/installing/index.html?pkg=tools>
- Eclipse + Android Plugin  
<https://dl-ssl.google.com/android/eclipse/>

# Erstellen eines Android-Projekts

New Project (→ Android → Application Modul)

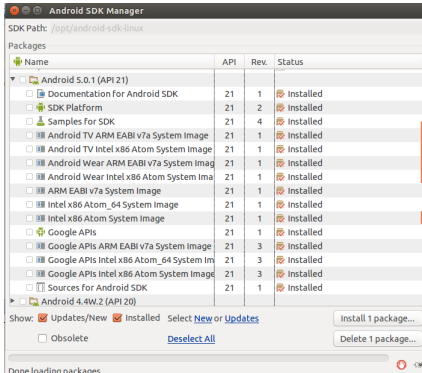


Android API (minimale Version) und Zielgerät auswählen

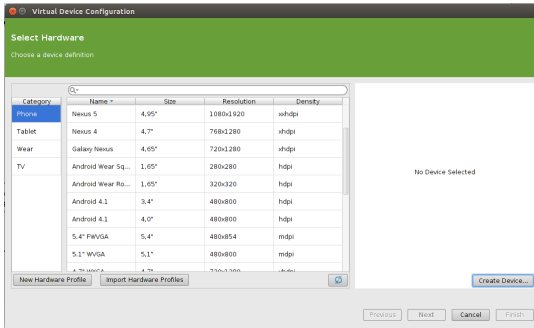


# Installation von Android-Komponenten - SDK Manager

- Installation fehlender Komponenten
- Installation von zusätzlichen API-Versionen



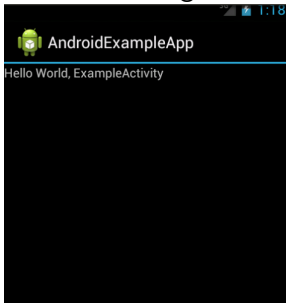
# Erstellen eines Virtual Device



- Verwendung vorhandener Konfigurationen
- Eigenes Gerät konfigurieren
- Anpassung des Geräts an Anforderungen

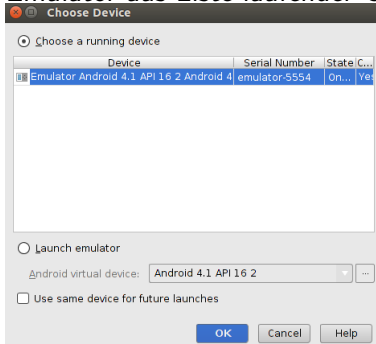
# Ausführen auf dem Emulator

- `src` → `com.example.AndroidExampleApp` → `ExampleActivity.java`
- Rechte Maustaste → Run 'AndroidExampleApp'
- Emulator und gewünschtes virtuelles Gerät auswählen



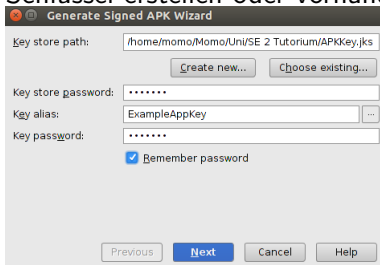
# Ausführen auf dem Emulator

- Bei erneuten Ausführen kein Neustart des Emulators nötig
- Emulator aus Liste laufender Geräte auswählen



## Ausführen auf dem Handy

- Erstellen einer APK-Datei: Build → Generate signed APK
- Schlüssel erstellen oder vorhandenen auswählen



- APK-Datei aufs Handy kopieren
- App installieren (Software aus unbekannten Quellen akzeptieren)



# Ressourcen Ordner

- res/layout: Layouts der Benutzeroberfläche (XML)
- res/drawable: Abbildungen (PNG, JPG)
- res/values: Konstanten (z.B. strings.xml)
- res/raw: Videos
- res/menu: Layouts für Menüs

# AndroidManifest.xml

Einstiegspunkt beim Ausführen des Projekts

- Referenz zum Java-Code
- Komponenten (z.B. Android Icon)
- Minimale SDK-Version

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.AndroidExampleApp"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="16"/>
    <application android:label="AndroidExampleApp" android:icon="@drawable/ic_launcher">
        <activity android:name="ExampleActivity"
            android:label="AndroidExampleApp">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

# Layout Dokument

## res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Hello World, ExampleActivity"
        android:id="@+id/helloText"
    />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:text="World"
        android:ems="10"
        android:id="@+id/nameInput" android:layout_gravity="center_horizontal"/>
</LinearLayout>
```

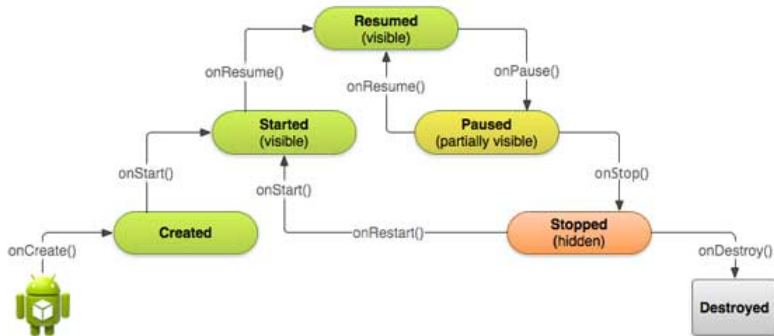
# Layout Dokument - Komponenten

- LinearLayout - orientation: Anordnung der Komponenten (vertical = untereinander)
- TextView/EditText - text: Angezeigter Text
- TextView/EditText - id: Id, welche Zugriff auf das Objekt im Java-Code ermöglicht
- weitere Mögliche Objekte wie Buttons, Checkboxes, Layout-Typen, Kalender,...

## <ActivityName>.java

- src/<packageName>/<ActivityName>.java
- Klasse, die im XML-Manifest referenziert wird und welche als Einstiegsklasse in den Code dient
- Implementierung der Methoden des Activity Lifecycles

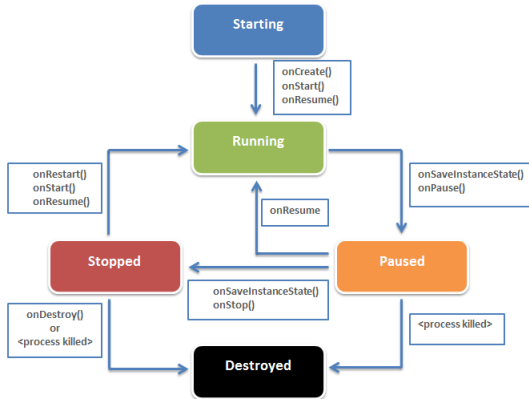
# Activity Lifecycle Übersicht



# Callback Methoden

- onCreate(Bundle savedInstanceState): App wird erstellt
- onStart(): App wird sichtbar
- onResume(): App kommt in den Vordergrund und Benutzer interagiert mit der App
- (onSaveInstanceState(Bundle savedInstanceState))
- onPause(): App verliert den Fokus (ist aber z.T. noch sichtbar)
- (onSaveInstanceState(Bundle savedInstanceState)): Speichern von eingegeben Daten zur Weiterverwendung bei einem erneuten Start
- onStop(): App unsichtbar im Hintergrund
- onDestroy(): App wird beendet

# Aufruf der Callback Methoden





## (Einfache) Beispiele für Callback Methoden

```
super.onCreate(savedInstanceState);  
setContentView(R.layout.main);  
if(savedInstanceState != null){  
    name = (String) savedInstanceState.get("name");  
    TextView hello = (TextView) findViewById(R.id.helloText);  
    hello.setText("Hello "+name+", ExampleActivity");  
}  
  
Toast.makeText(this, "onCreate", Toast.LENGTH_SHORT).show();
```

```
protected void onSaveInstanceState(Bundle savedInstanceState){  
    super.onSaveInstanceState(savedInstanceState);  
    EditText usernameEt = (EditText) findViewById(R.id.nameInput);  
    String username = usernameEt.getText().toString();  
  
    savedInstanceState.putString("name", username);  
}
```

# Online Tutorials

- Android Developer Tutorial von Google
- Basteldroid - Android Einführung
- Video Tutorial auf Youtube