


# 4 Übung Web- und Multimedia-Engineering

Aufgabenstellung Ü4 *Mobile App Development*  
und thematische Einführung

- *Überblick (Terminplan)*
  1. Aufgabenstellung Ü4: Mobile App Development (Android)
  2. Einführung: Entwickeln auf Android
  3. Hilfreiches, Tipps und Links

# Terminplan / Ablauf

Woche	Datum	Übungsthema /-inhalt
<b>KW 42</b>	13./14.10.	keine Übung
<b>KW 43</b>	20./21.10.	Einführung Aufgabenstellung Ü1: HTML5 und CSS3  → Materialien für Übungsaufgabe 1: HTML5 und CSS3
<b>KW 44</b>	27./28.10.	Details zur Abgabe/Einreichung von Ergebnissen, Konsultation
<b>KW 45</b>	03.11. 9.00 Uhr	Abgabe Ü1 HTML5 und CSS3
<b>KW 45</b>	03./04.11.	Einführung Aufgabenstellung Ü2: JavaScript, Ajax, JSON und jQuery → Materialien für Übungsaufgabe 2
<b>KW 46</b>	10./11.11.	Konsultation
<b>KW 47</b>	17./18.11.	keine Übung
<b>KW 48</b>	24.11. 9.00 Uhr	Abgabe Ü2 JavaScript, Ajax, JSON und jQuery
<b>KW 48</b>	24./25.11.	Einführung Aufgabenstellung Ü3: XML und PHP → Materialien für Übungsaufgabe 3
<b>KW 49</b>	01./02.12.	Konsultation
<b>KW 50</b>	08./09.12.	Konsultation
<b>KW 51</b>	15.12. 9:00 Uhr	Abgabe Ü3 PHP und XML
<b>KW 51</b>	15./16.12.	Einführung Aufgabenstellung Ü4: Mobile App Development → Materialien für Übungsaufgabe 4 Version 2 → Schnittstellenbeschreibung (API)
<b>KW 52</b>	22./23.12.	keine Übung
<b>KW 1</b>	29./30.12.	keine Übung
<b>KW 2</b>	06./07.01.	Android Netzwerkkommunikation und Einführung inf_box_lib
<b>KW 3</b>	12./13.01.	Konsultation
<b>KW 4</b>	19.01. 9:00 Uhr	Abgabe Ü4 Mobile App Development
<b>KW 4</b>	19./20.01.	keine Übung
<b>KW 5</b>	26./27.01.	Anschluss, Feedback und Fragen
<b>KW 6</b>	02./03.02.	keine Übung
Woche	Datum	Übungsthema /-inhalt

- 4 Aufgabenstellungen



- Je ein Themenkomplex
- Je ca. 3 Wochen Bearbeitungszeit
- Gemeinsames Thema: inf\_box
- **Ü1:** Grundlagen client-seitige Technologien: HTML5 und CSS3
  - Grundgerüst für eine Webseite als Interface für inf\_box
- **Ü2:** Dynamische Webseiten: JavaScript, Ajax, JSON und jQuery
  - Manipulation, Animationen und dynamische Inhalte für die Webseite
- **Ü3:** Grundlagen server-seitige Technologien: XML und PHP
  - XML-Schema, XSLT, Grundlagen PHP einer inf\_box Serverkomponente
- **Ü4:** Mobile App Development: Android SDK
  - Einfache inf\_box Android-Client-App



Teil 1

# **Aufgabenstellung Ü4:**

## **Mobile App Development (Android)**

- Entwicklung einer Client-App für den Zugriff auf den Webservice
- Aufgabe umfasst 3 Arbeitspakete
  1. Anlegen von fünf Activities für Anmeldung, Registrierung, App-Infos, Listenansicht und Detailansicht ★★
  2. Umsetzung des vorgegebenen Layouts für alle Activities ★★
  3. Abfrage und Anzeige der Daten des Webservice ★★★

## 1. Anlegen Activities ★★

- ☐ Definition der fünf Activities entsprechend der folgenden logischen Struktur

- ☐ für Anmeldung (Login)
- ☐ Registrierung
- ☐ App-Infos
- ☐ Listenansicht
- ☐ Detailansicht

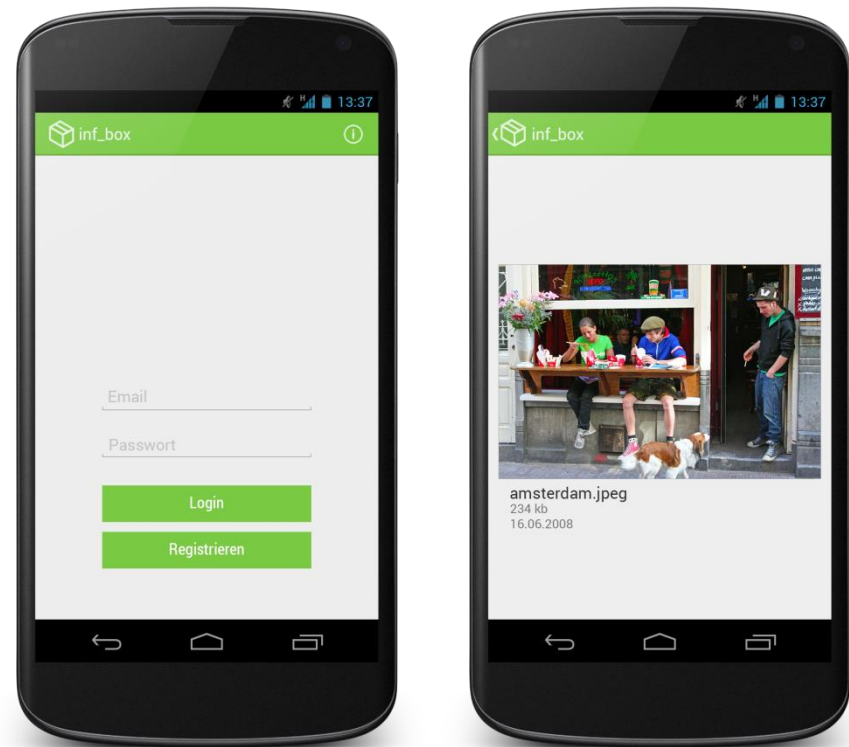


- ☐ Verknüpfung der Activities mittels Buttons
- ☐ Eingabevalidierung bei Registrierung: E-Mail-Adresse validieren

## 2. Umsetzung des vorgegebenen Layouts für alle einzelnen Activities



- ☐ Umsetzung der Layoutvorgaben in den jeweils zugehörigen Layout-XML-Dateien → siehe Material





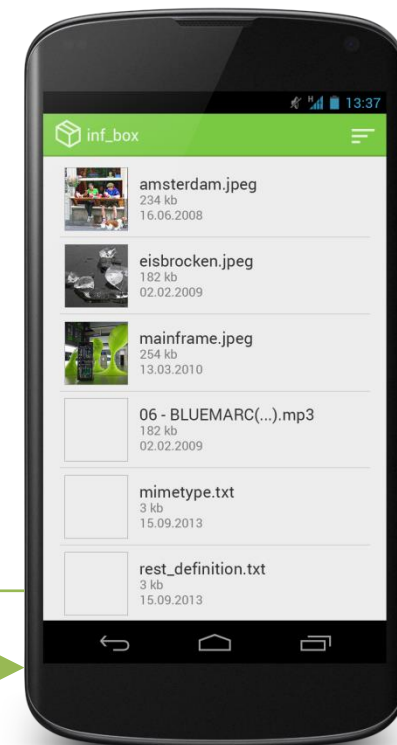
## 3. Abfrage und Anzeige der Daten des Webservice




- ☐ Abfrage von allen Items des Nutzers „Thomas Müntzer“ (User ID = 1)
- ☐ Anzeige der Daten in einer Liste, zugehörigen ListAdapter erstellen (!)
  - ☐ Anzeige von quadratischem Thumbnail, Dateiname, Dateigröße sowie Erstelldatum
- ☐ Anzeige von Items in Detailansicht
  - ☐ Versch. Dateitypen der Reihe nach berücksichtigen:  
*Bilder/Fotos (Pflicht),*  
*Textdateien (Bonus),*  
*Videos (Bonus), Musik (Bonus)*



..1010110001111..



- Allgemeine Kriterien
  - ☐ Projekt als UTF-8 codiert
  - ☐ Integrated Development Environment (Alternativen)
    - ☐ Eclipse + ADT Plugin
    - ☐ Android Studio
  - ☐ API Version
    - ☐ Minimum  $\geq 14$  (entspricht Android 4.0)
    - ☐ Target = 19 ( entspricht Android 4.4)
  - ☐ Projekt korrekt benannt: „infbox\_android\_t<XX>“   
→ „<XX>“ durch Teamnummer ersetzen
  - ☐ Gut dokumentierter, sauberer Code

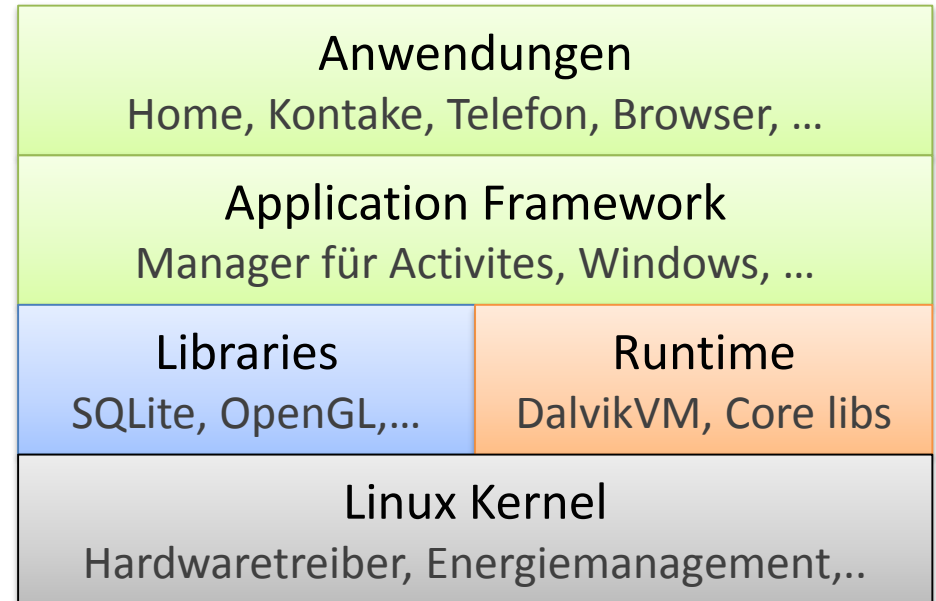
- Erweiterung des vorgegebenen Basisprojekts um die einzelnen Aufgabenteile
- Erlaubte Hilfsmittel:
  - Eclipse-Projekt (Prototyp und Design) als Basis
  - Keine zusätzlichen Frameworks oder Toolkits erlaubt
- Testen: Android Emulator / eigenes Android-Gerät
- Abgabe: Montag, **19.01.2015 bis 9.00 Uhr** 
  - Readme in Abgabe muss u.a. Name und Version der verwendeten IDE (Eclipse oder Android Studio) enthalten
  - Abgabe muss u.a. exportiertes Projekt ohne Binaries (bspw. \*.class Dateien) beinhalten

Teil 2

## **Einführung: Entwickeln auf Android**

# Einführung Android: Grundlagen

- Basiert auf einem Linux-Kernel (Speicher-/ Prozessverwaltung, Schnittstelle zwischen Hard- und Software)
- Implementierungssprache Java, Bytecode jedoch in spezieller Form: *Dalvik Bytecode*  
→ nicht zu Java  
Bytecode kompatibel
- Es gibt keine main-Methode!

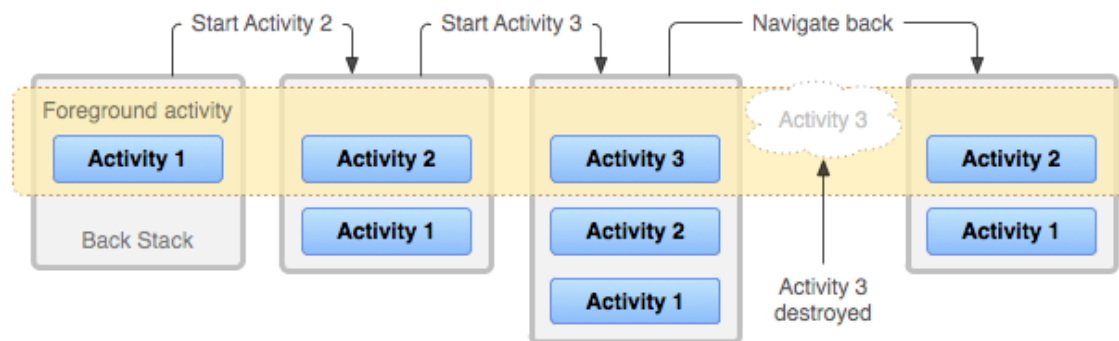


- Hauptbestandteile einer Android-Anwendung
  - Activities
  - Services
  - Broadcast Receiver
  - Content Provider

# Einführung Android: Activity

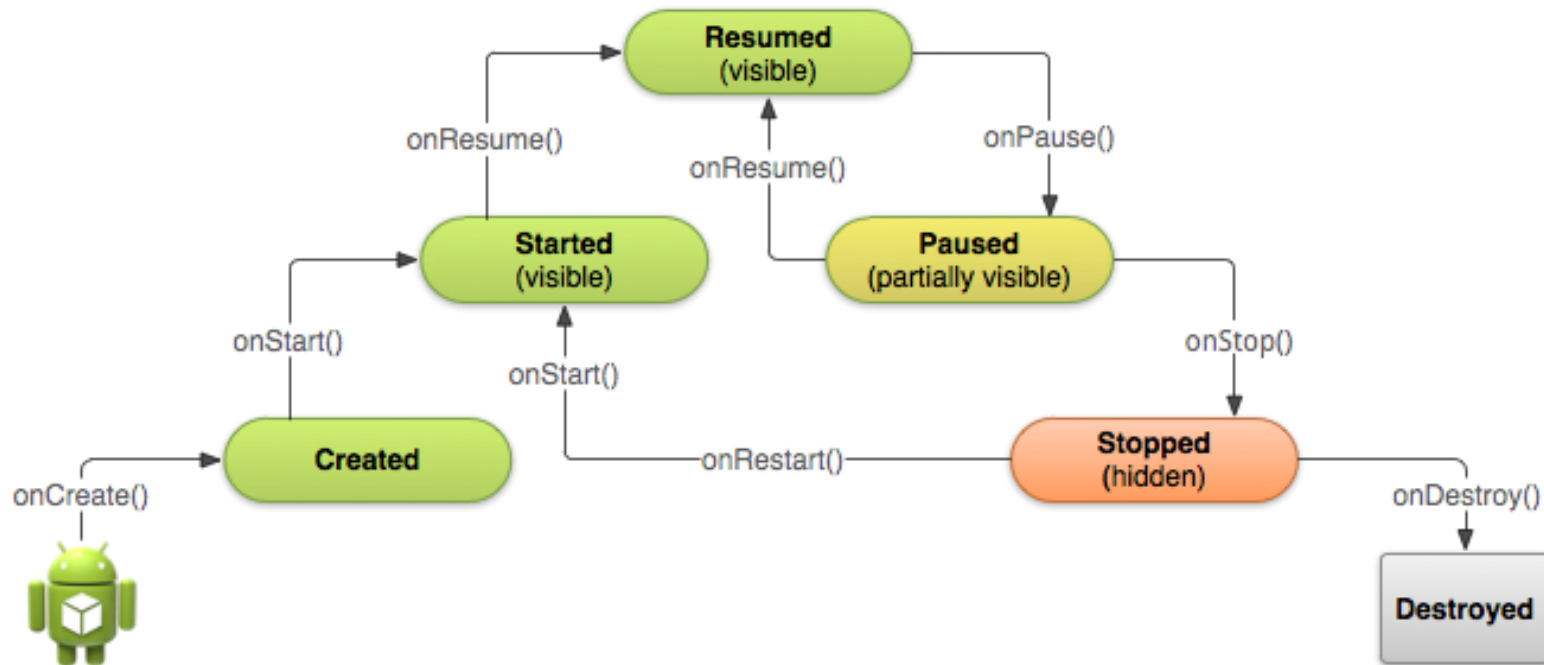
## ■ Activity

- Definiert eine einzelne View (in XML beschrieben) zur Anzeige auf dem Bildschirm (eine App hat meist mehrere Activities) und deren Funktionalität
- Behandelt Events von Elementen der Benutzerschnittstelle (z. B. Klick auf einen Button)
- Immer eine Activity aktiv, Vorgängeraktivität wird deaktiviert und für spätere Wiederverwendung auf dem "History Stack" abgelegt → Activity Lifecycle



# Einführung Android: Activity Lifecycle

- Activity Lifecycle





# Einführung Android: View

- View
  - Ein *View* ist der sichtbare Teil der *Activity*
  - Auf zwei verschiedene Arten definierbar:
    - In einer XML-Layout-Datei
    - UI-Elemente direkt im Code beschrieben

```
<RelativeLayout xmlns:android= ...  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
    <TextView android:id="@+id/mytext"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/hello_world" />  
  
</RelativeLayout>
```



# Einführung Android: Beispiel

- Beispiel: Definition eines Views und Starten einer 2. Activity

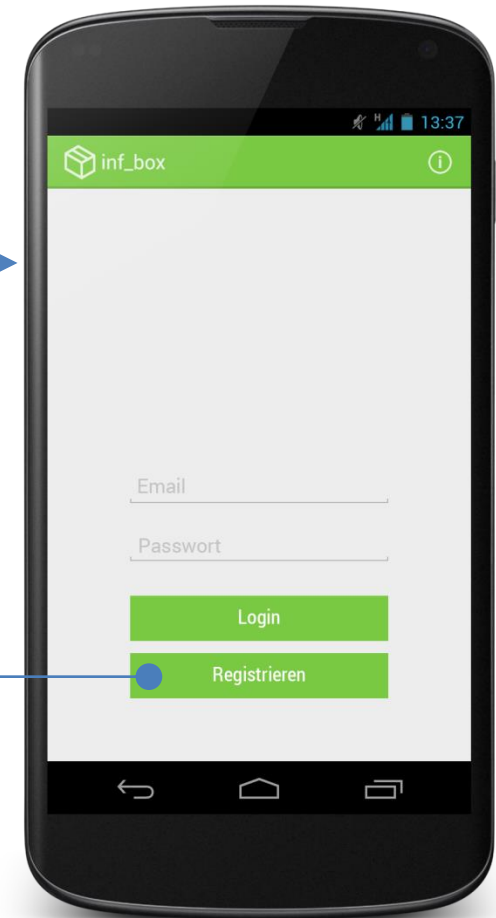
```
public class MyActivity extends Activity
{
    onCreate(..)
    {
        setContentView(R.layout.myLayout);
    }

    onClick(..)
    {
        ..
        Intent intent = ..
        startActivity(intent);
    }
}
```

definiert View

Event

Neue Activity



# Einführung Android: Intent

## ■ Intent

- Asynchrone Nachrichtenobjekte, die abstrakte Operation beschreiben
- Startet z. B. eine andere *Activity* („öffne ein neues Fenster“)
- Kann Daten an die zu startende *Activity* übergeben
- Unterscheidung zwischen expliziten und impliziten *Intents*:
  - Explizite Intents definieren ihr Ziel genau (z. B. durch Angabe der Klasse)
  - Implizite Intents definieren nur die Aktion (Auswahl der Anwendung durch System oder den Nutzer, beispielsweise Webbrowser)

```
..  
Intent i = new Intent(this, MySecondActivity.class);  
i.putExtra("Key1", "Value1");  
startActivity(i);  
..
```

Übertragung von Daten zw. Activities

## ■ Broadcast Receiver

- Ermöglicht das Empfangen von *Intents* anderer Anwendungen (z. B. Reaktion auf Timer, geänderte Verfügbarkeit eine WLANs, usw.)
- IntentBroadcasts werden über Anwendungsgrenzen vermittelt
- Ist eine Applikation, die für bestimmte IntentBroadcasts registriert und wurde nicht aktiv, wird diese aktiviert

## ■ Service

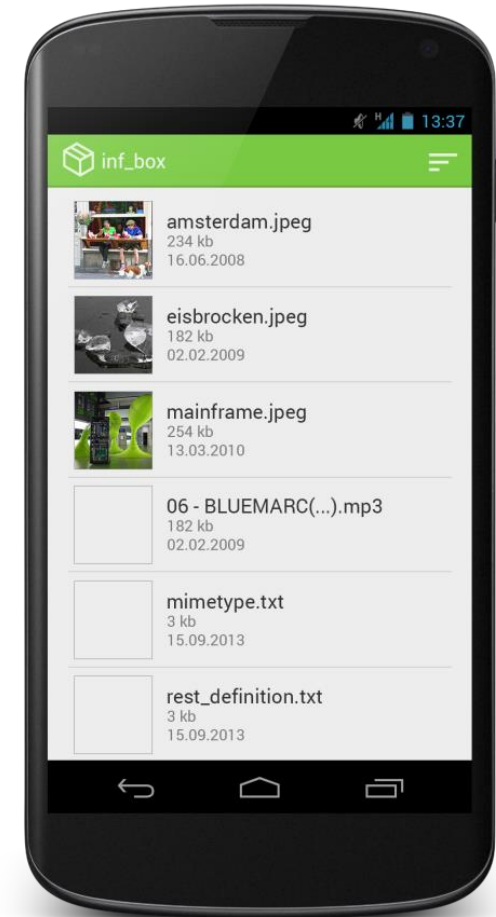
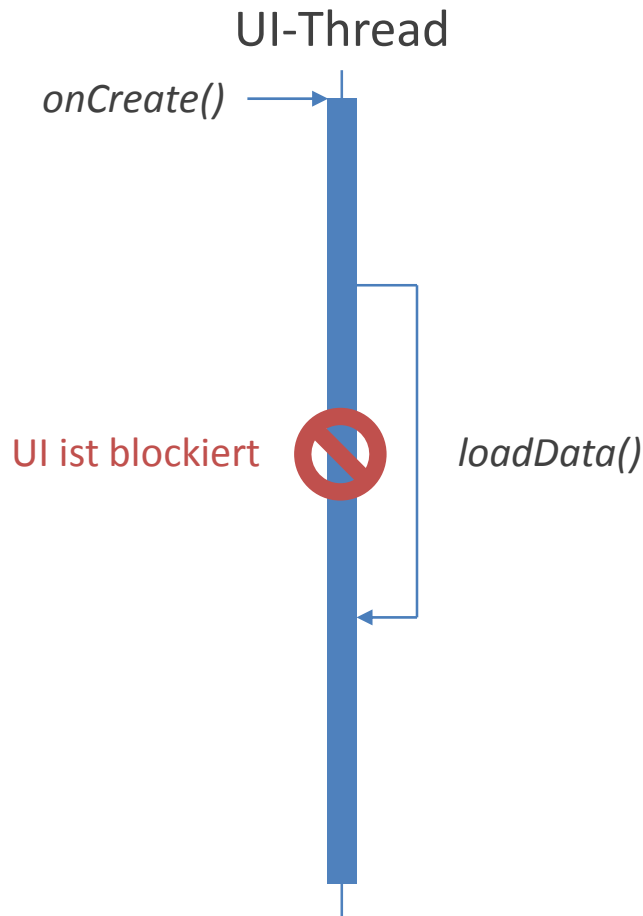
- Hintergrundthreads, die unabhängig von der Benutzerschnittstelle abgearbeitet werden
- *Activities* können jederzeit auf diese Services zugreifen
- Zugriff auf Servicefunktionalität über spezifizierte Schnittstellen

## ■ Content Provider

- Stellen Anwendungsdaten bereit
- Implementieren Standardschnittstellen zum Speichern und Abrufen von Daten

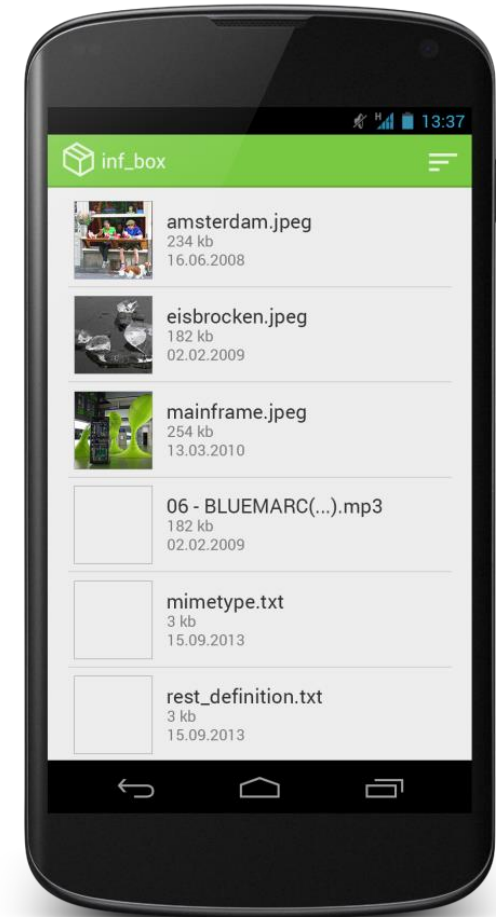
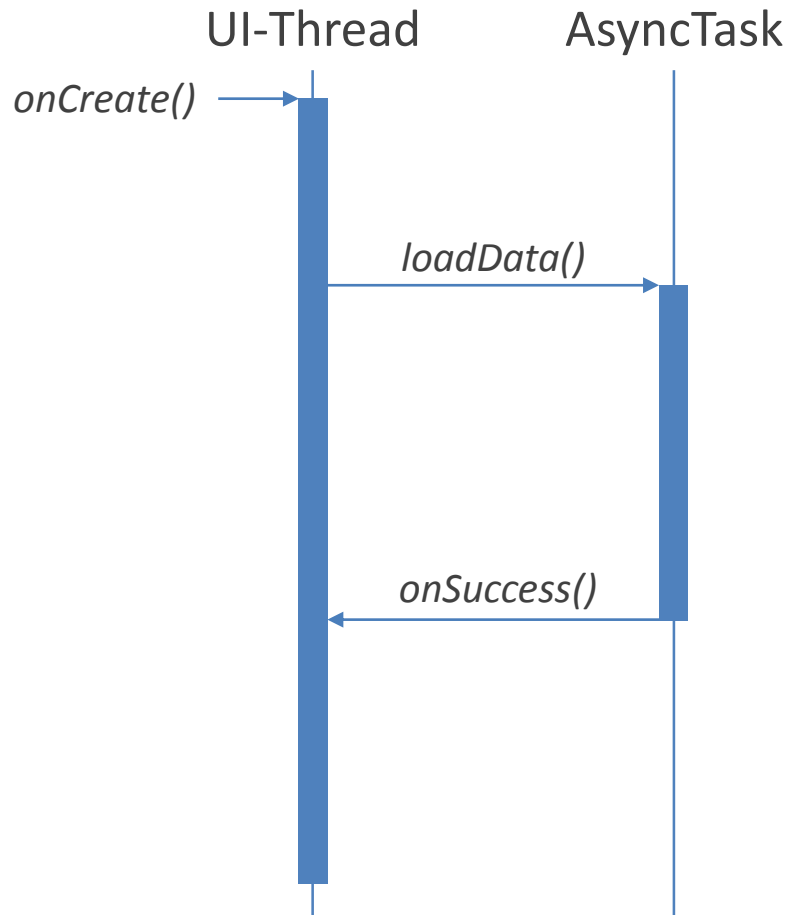
# Einführung Android: Netzwurkkommunikation

- Beispiel: Laden von Netzwerkdaten



# Einführung Android: Netzwerkkommunikation

## ■ Beispiel: Laden von Netzwerkdaten



# Einführung Android: Netzwerkkommunikation

## ■ Notwendige Voraussetzungen

### – AndroidManifest.xml enthält:

- `<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>`
- `<uses-permission android:name="android.permission.INTERNET"/>`

### – Erlauben grundsätzlichen Zugang zum Internet und sind wichtig für die Fehlerbehandlung (kein Netz o.ä.)

## ■ Typischer Ablauf der Netzwerkkommunikation



- Komponenten der Bibliothek: AsyncTasks

## InfboxTask

```
#doInBackground(String..urls)
-getResult()
-getStringFromInputStream()
+onPostExecute(String result)
+setResultHandler
  (IInfboxResultHandler handler)
...
+execute(String url)
```

## DownloadImageTask

```
#doInBackground(String..urls)
+onPostExecute(String result)
...
+execute(String url)
```



- Weitere Komponenten der Bibliothek

## ConnectionChecker

+isDeviceConnected(Context c)

## IInboxResultHandler

+handleResult(String result)

## InboxDataConverter

+getInboxItem(String input)

+getInboxItemList(String input)

+getInboxUser(String input)

+getInboxUserList(String input)

+getInboxMetadata

## ■ Beispiel: neuer InfboxTask

```
InfboxTask itemsTask = new InfboxTask();

itemsTask.setResultHandler(new IInfboxResultHandler() {

    @Override
    public void handleResult(String result) {

        //display result at UI
        ...

    }

});

itemsTask.execute(StaticValues.BASE_URL + "/users/1/items");
```

- Beispiel: neuer DownloadImageTask

```
imageView.setTag(StaticValues.BASE_URL + "items/" + item.getId() +  
    "/thumbnail");  
  
new DownloadImageTask(imageView)  
    .execute(StaticValues.BASE_URL + "items/" + item.getId() +  
        "/thumbnail");
```

- Beispiel: sicheres Ausführen eines Netzwerktasks

```
Context c = getApplicationContext();  
  
if(ConnectionChecker.isDeviceConnected(c)) {  
    itemsTask.execute(StaticValues.BASE_URL + "/users/1/items");  
}  
else {  
    Toast.makeText(c, "Kein Internet.",  
    Toast.LENGTH_SHORT).show();  
}
```

↑  
Nicht vergessen!

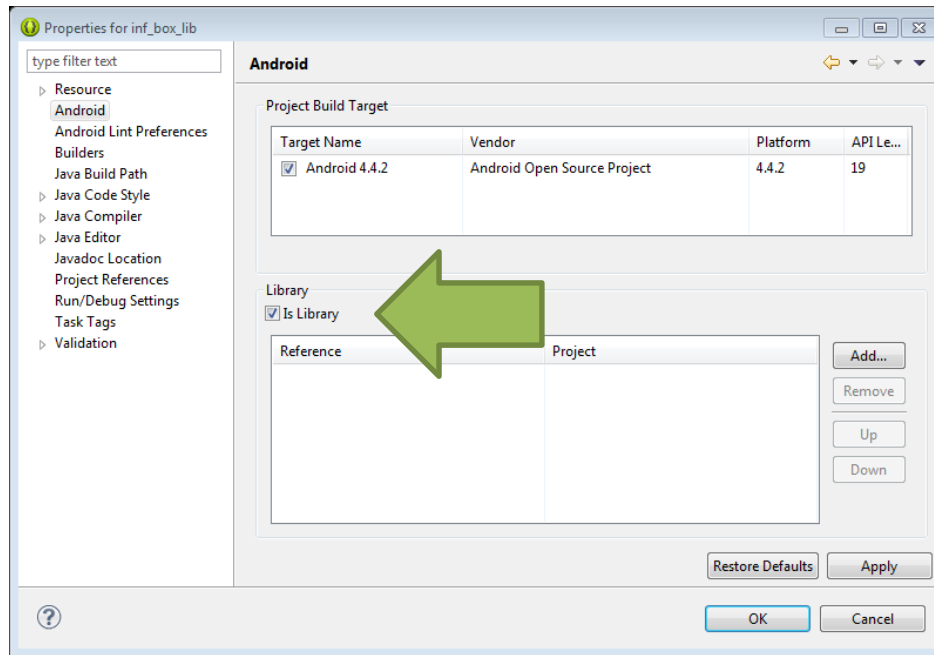


- Beispiel: Umwandlung der Netzwerkdaten in Objekte

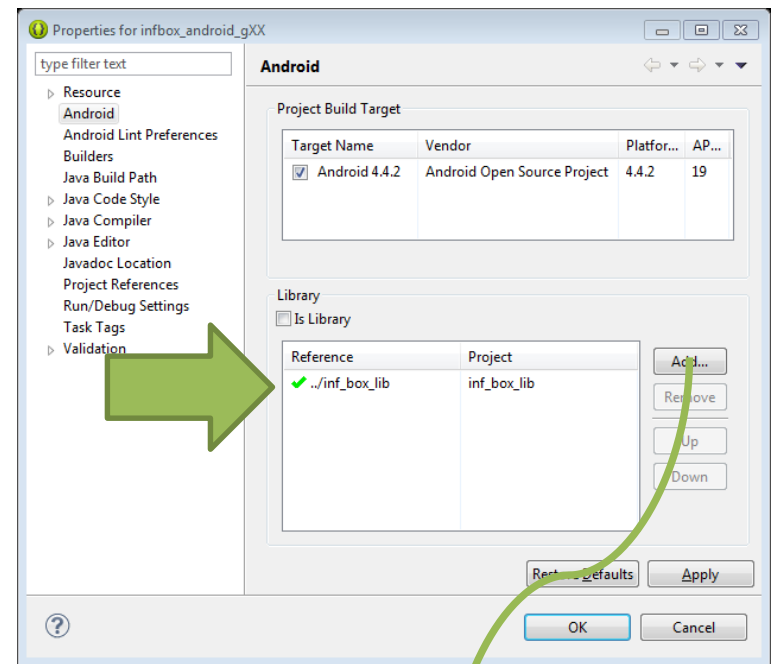
```
List<Item> items = new LinkedList<Item>();

try {
    items = InfboxDataConverter.getInfboxItemList(result);
} catch (Exception e) {
    e.printStackTrace();
}
```

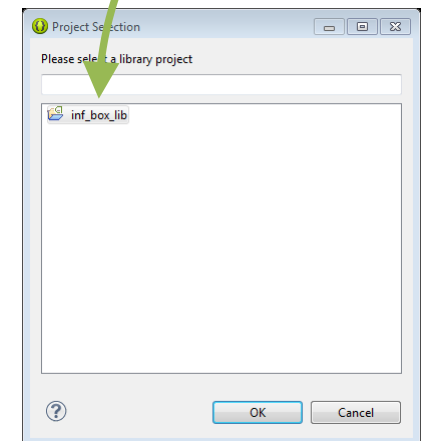
# inf\_box\_lib - Korrekte Verknüpfung



Library



Sampleproject



Teil 3

## **Hilfreiches, Tipps und Links**

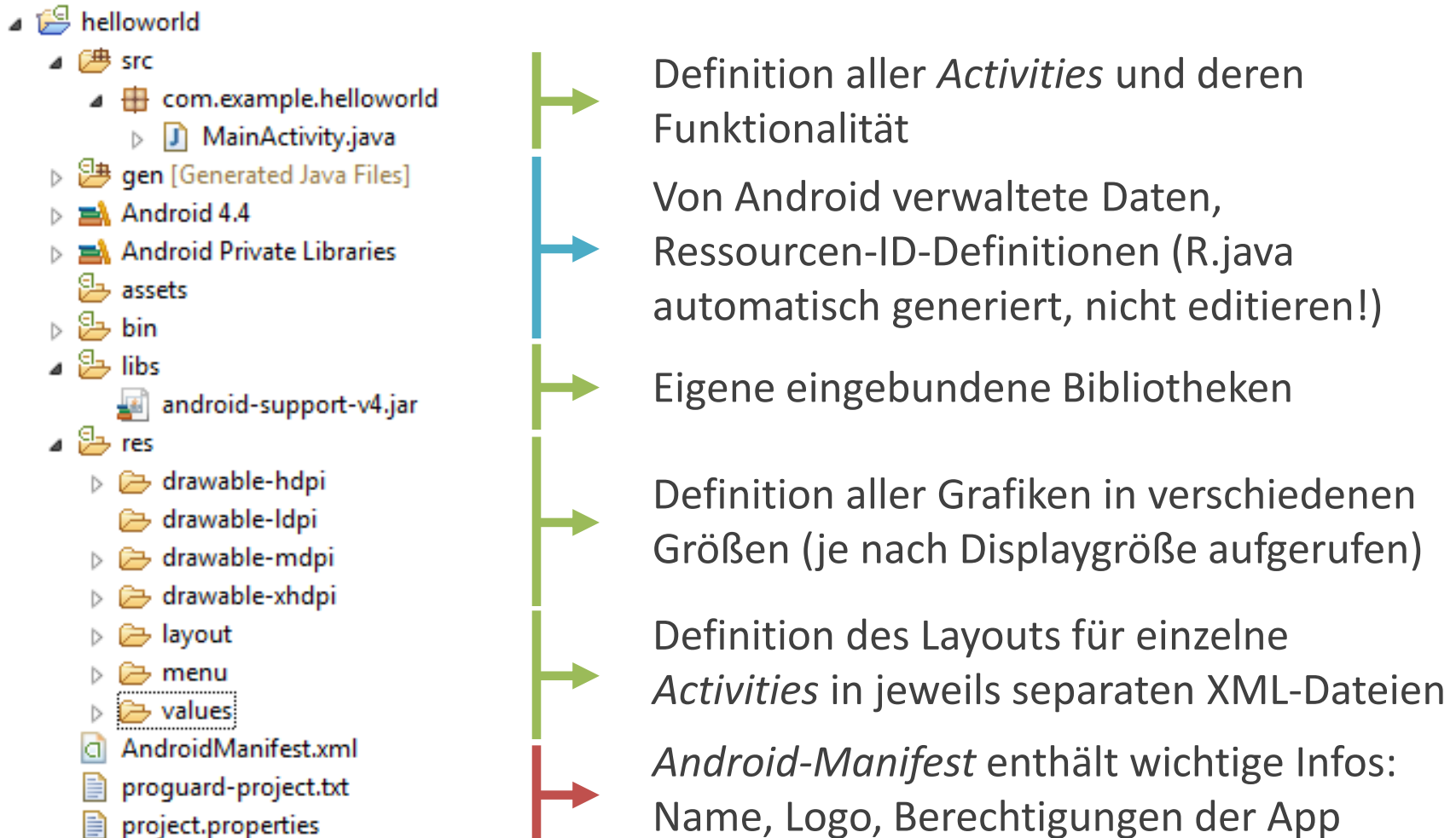
# Entwicklungsumgebung

- Komfortable Entwicklung von Anwendungen für Android mittels Eclipse (Java-IDE) und zugehörigem Plug-In
- Sämtliche weitere Hilfen zur Entwicklung bringt das Android-SDK (Software Development Kit) mit sich
- → **Anleitung** zur Installation:  
[\[http://developer.android.com/sdk/installing/installing-adt.html\]](http://developer.android.com/sdk/installing/installing-adt.html)
- Im Anschluss Download der noch fehlenden SDK-Komponenten über den SDK-Manager → Android 4.4 („KitKat“, API Version 19)
- Import des Basisprojekts über: *Import* → *Archive File* oder *Import* → *Existing Android Project*, falls Projekt schon entpackt



# Android Projektstruktur

## ■ Struktur eines Android-Projekts



- Testen von Anwendungen
    - auf dem eigenen Smartphone (per USB anschließen, Treiber installieren nicht vergessen)
    - im Android-Emulator (beim SDK enthalten, teilweise langsam)
      - Z. B. vordefiniertes Nexus 4
    - Emulator-Alternativen:
      - BlueStacks: [\[http://www.bluestacks.com\]](http://www.bluestacks.com)
      - Genymotion: [\[http://www.genymotion.com\]](http://www.genymotion.com)
      - Manymo: [\[https://www.manymo.com\]](https://www.manymo.com)
- ← **Empfehlung:**  
Sehr schnell und  
kostenlos

- www** Android Developers: *Getting Started*  
[<http://developer.android.com/training/index.html>]
- www** Lars Vogel: Solutions for common Android development problems  
[<http://www.vogella.com/articles/AndroidDevelopmentProblems/article.html>]
- www** Lars Vogel: Android Development Tutorials  
[<http://www.vogella.com/android.html>]
- www** Video2brain: *Meine erste Android-App* Lehrvideo **Kostenlos im Uninetz unter**  
[[http://slubdd.de/katalog?TN\\_vid2bra\\_video2brain\\_1251](http://slubdd.de/katalog?TN_vid2bra_video2brain_1251)]
- Thomas Künneth: *Android 4 / Apps entwickeln mit dem Android SDK* 2. Auflage.  
Galileo Press, Bonn 2012, ISBN 3836219484
- www** Heise.de: Android-Emulatoren für Windows, Mac und Linux  
[<http://www.heise.de/download/special-android-emulatoren-fuer-windows-mac-linux-151041.html>]
- www** AndroidPit, Anfängerworkshop  
[[http://www.androidpit.de/de/android/wiki/view/Android\\_Anf%C3%A4nger\\_Workshop](http://www.androidpit.de/de/android/wiki/view/Android_Anf%C3%A4nger_Workshop)]

# Fragen?



**Interactive Media Lab Dresden**  
Professur für Multimedia-Technologie

*Kontakt:*

Ricardo Langner ([ricardo.langner@tu-dresden.de](mailto:ricardo.langner@tu-dresden.de))

Tom Horak ([tom.horak@tu-dresden.de](mailto:tom.horak@tu-dresden.de))

# Changelog

**Datum / Zeit**

**Beschreibung**

2014-12-15 09:20

- Initiale Downloadversion