

2. ZÁKLADNÍ SOUČÁSTI APLIKACE

BI-AND



Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

2. přednáška

- Android projekt
- Manifest
- Uživatelské rozhraní (základy)
- Activity

Základní stavební prvky

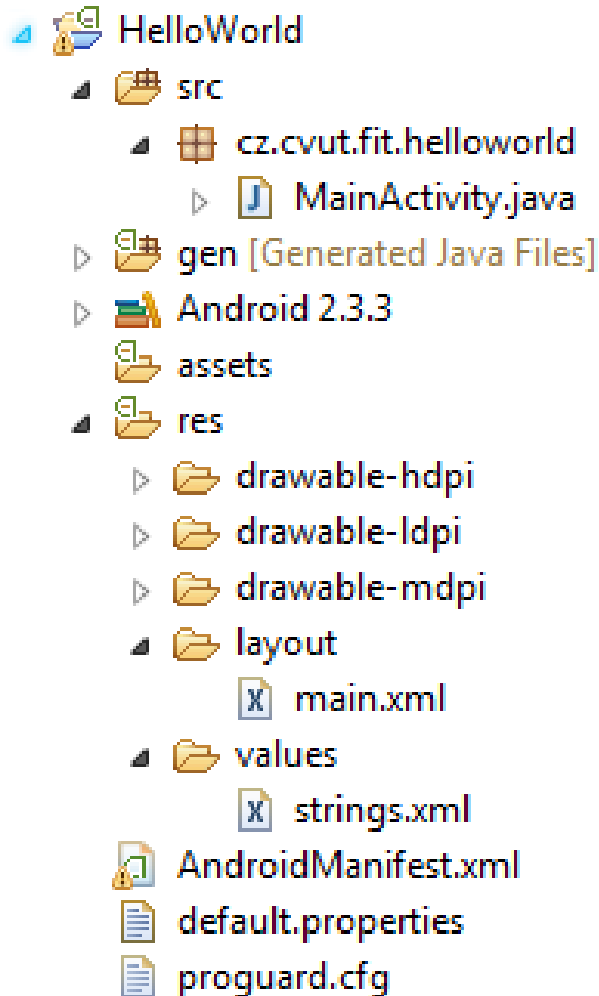
- Activity
 - Určitá obdoba AWT Frame z Javy
 - Obsahuje uživatelské rozhraní
 - XML Layout určuje rozložení prvků

- Service
 - Služba běžící na pozadí
 - Ovládána povely pocházejícími většinou z Activity
 - Použití např. pro přehrávání hudby, odesílání emailů, stahování souborů

Základní stavební prvky

- **Content Provider**
 - Rozhraní pro přístup k (strukturovaným) datům jiné aplikace
 - Např. kontakty, kalendář, zmeškané hovory apod.
- **Broadcast Receiver**
 - Zachytávání „zpráv“ (Intents), které se šíří systémem
 - Např. příchozí hovor, přijatá SMS nebo uživatelsky definovaná zpráva
- **Intent**
 - „Zpráva“ o nějaké události, která kromě druhu Intentu může obsahovat další data

Android projekt



- src
- res
 - drawable
 - layout
 - values
 - ...
- AndroidManifest.xml

Android projekt – Source (scr)

- Zdrojový kód v Javě
- Žádná „Main“ třída
 - Určeno v Android Manifestu
- Unikátní jméno balíčku
 - Konvence: [země].[autor].[jméno aplikace]
 - Více – viz. sekce rozšiřující odkazy
 - Např.: `cz.cvut.helloWorld`
- **Doporučuje se dělit na více balíčků** (Utilities, Communication, Adapters, Activities apod.)

Android projekt – Generated (gen)

- Obsahuje soubory vygenerované ADT
- Ručně nemodifikovat
- R.java – **jedinečné odkazy** na různé Resources (viz. dále) a další prvky s vlastním ID (tlačítka, editovatelné boxy, apod.) **přístupné „odkudkoliv“ z kódu**
 - Přístup ke zdroji pomocí `Context.getResources()`
`String s = getResources().getString(R.string.app_name);`
- Interfaces vytvořené pomocí AIDL (Android Interface Definition Language)
 - Forma interprocess communication (IPC)

Android projekt - Assets

- Nezávislé zdroje (raw asset files)
- Přibaleny do .apk beze změn
- Android negeneruje ID obsahu této složky
- Většinou se používají pro textury
- Přístup přes `Context.getAssets()` ;

- Např.:

```
InputStream is =  
getAssets().open("folder/sample.txt");
```


Android projekt – Resources (res)

- **Drawable**
 - Bitmap File (.png, .jpg, .gif)
 - Nine-Patch File (.9.png)
 - Drawable shapes/object (.xml)
- **Layout**
 - Rozložení UI (.xml)
 - Např. Activity nebo jiné komponenty (řádek v seznamu)
- **Values**
 - Mnoho nejrůznějších Resources - strings, themes, styles, attrs (.xml)
 - Na rozdíl od ostatních nejsou XML soubory referované v R.java pomocí jména, ale podle XML element type

Android projekt – Resources (res)

- Menu
 - Definice Menu v aplikace
 - Kvůli oddělení kódu a vlastního vzhledu aplikace
- Raw
 - Podobné jako assets, liší se pouze v přístupu k nim
 - Vytvořený odkaz v R.java
 - Většinou umístěny MP3 soubory, gestures
- Další složky v Resources (res) – color, xml, anim
- Resources přístupné i jiným aplikacím (např. Launchers)



Parametry displeje

- Velikost displeje (Screen size)
 - Úhlopříčka displeje v palcích (inches – např. 3.7“)
- Hustota displeje (Screen density)
 - Počet pixelů na danou plochu displeje (dpi - dots per inch)
- Orientace
 - Na šířku (landscape) nebo na výšku (portrait)
 - **Žádná orientace není defaultní**

Parametry displeje

- Rozlišení (Resolution)
 - Počet fyzických pixelů na displeji (px - pixels)
 - Nepoužívá se při návrhu pro podporu více různých displejů
- Density independent pixel
 - Virtuální jednotka používaná pro pozicování a rozměry prvků nezávislé na dpi (dip, dp nebo dps)
 - 1 dip je 1 fyzický pixel na 160 dpi displeji
 - Převod zajišťuje OS

$$px = dp \times (dpi / 160)$$

Android projekt – Resources (res)

Konfigurace	Přípony	Upřesnění
Jazyk	-en -cs ...	
Rozměry displeje	-small -normal -large -xlarge	Min. 426dp x 320dp Min. 470dp x 320dp (baseline size) Min. 640dp x 480dp Min. 960dp x 720dp
Hustota displeje	-ldpi -mdpi -hdpi -xhdpi -nodpi	~120dpi ~160dpi (baseline density) ~240dpi ~320dpi OS nebude modifikovat tyto resources
Orientace displeje	-land -port	Poloha na šířku Poloha na výšku
API Level	-v11 -v10 -v9 ...	Pro Android 3.0 a vyšší Pro Android 2.3.3 a vyšší Pro Android 2.3 a vyšší

Struktura APK

- Application Package File (APK)
- ZIP balíček postavený na **formátu JAR**
- META-INF:
 - MANIFEST.MF (manifest file) – údaje o APK a jeho obsahu
 - CERT.SF (signature file) – verifikace APK
 - CERT.RSA – uložený podpis SF, certifikát
- RES: resources nezkompilevané do resources.arsc



SHA-1 – hashovací funkce

Struktura APK

- AndroidManifest.xml – viz. dále
- classes.dex – zkompilevané třídy určené pro DVM
 - Obdoba .class souborů
 - Více tříd v jednom souboru
- resources.arsc – předkompilované resources (např. XML)



Android Manifest

- „Nadstavba“ Java manifestu
- Specifikuje parametry aplikace
 - Název package
 - Komponenty (Activity, Intent Filters, Broadcast Receivers, Services)
 - Oprávnění (internetové připojení, zjišťování polohy apod.)
 - Importované knihovny
 - Minimální verze API
 - Mnohé další informace o aplikaci – viz. dále

Příklad AndroidManifest.xml

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <manifest
3.     xmlns:android="http://schemas.android.com/apk/res/android"
4.     package="cz.cvut.hello.world"
5.     android:versionCode="5"
6.     android:versionName="1.5">
7.     <application>
8.         <activity
9.             android:name=".MainActivity"
10.            android:label="Hello world">
11.            <intent-filter>
12.                <action
13.                    android:name="android.intent.action.MAIN" />
14.                <category
15.                    android:name="android.intent.category.LAUNCHER" />
16.            </intent-filter>
17.        </activity>
18.    </application>
19.    <uses-sdk
20.        android:minSdkVersion="7"
21.        android:targetSdkVersion="10" />
22.    <supports-screens
23.        android:smallScreens="true"
24.        android:xlargeScreens="false"
25.        android:anyDensity="true" />
26.    <uses-permission
27.        android:name="android.permission.INTERNET" />
28.</manifest>
```

Důležité elementy a atributy

- `<uses-sdk>`
 - `minSdkVersion` – minimální API level pro spuštění
 - `targetSdkVersion` – API level, pro který je aplikace „určena“
 - OS určuje vypnutí/zapnutí určitých funkcí kvůli forward compatibility
 - Např. pro API 11 a vyšší - Screen compatibility mode, Holo Theme apod.
 - **Více informací v `Build.VERSION_CODES`**
- `<supports-screens>`
 - Seznam podporovaných rozlišení
 - Pro větší nepodporované displeje zapnut screen compatibility režim (API Level 13)
- `<uses-library>` - seznam využívaných knihoven
 - Např. `com.google.android.maps`

Pozor na rozdíl mezi
`minSdkVersion` a `targetSdkVersion`

Důležité elementy a atributy

- `<uses-configuration>` - SW nebo HW nároky aplikace
 - Týká se především ovládání
 - Např. `reqFiveWayNav`, `reqKeyboardType`, `reqTouchScreen`
- `<uses-feature>` - SW nebo HW nároky aplikace
 - Týká se především tech. vlastností zařízení
 - Např. Bluetooth, fotoaparát, GPS
- Rozdíl?
 - Configuration – typicky bude potřebovat jiné resources (Google TV)
 - Feature – typicky nebude potřebovat odlišné resources (GPS)
- `<uses-permission>` - přístup ke zdrojům OS
 - Např. čtení kontaktů, přístup na internet či k fotoaparátu

Android Market filtruje aplikace např. podle `<uses-feature>` a `minSdkVersion`

UI komponenty (Widgets)

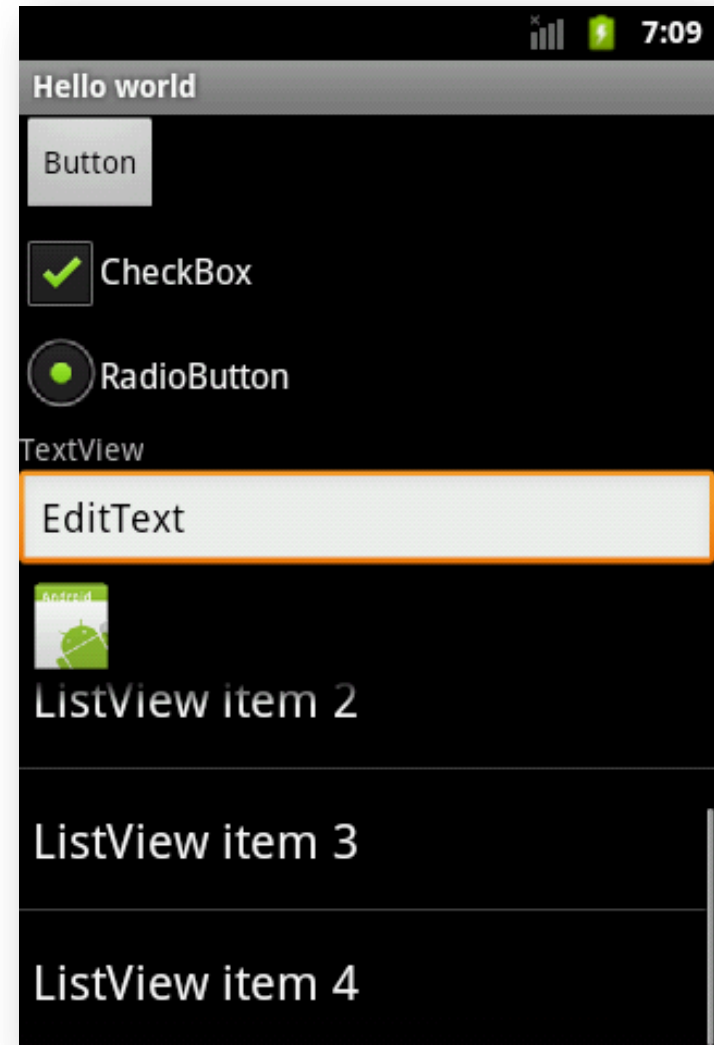
- Dědí od třídy View
- Možné vytvářet programově nebo **pomocí XML**
- Výška a šířka se nastavuje pomocí
 - `match_parent`
 - Roztáhne se přes celou velikost rodiče
 - Dříve `fill_parent` - do API Level 8 (Android 2.2)
 - `wrap_content`
 - Přizpůsobí svou velikost podle obsahu
 - Přímo číselnou hodnotou nebo pomocí `wrap` (`layout_weight`)

Nejpoužívanější widgety

- Nejpoužívanější widgety:
 - Button
 - CheckBox
 - RadioButton
 - TextView
 - EditText
 - ImageView
 - ListView
 - **WebView**

Neplést si s widgety
umístěnými na pohotovostní
obrazovce (tzv. App Widgets)

```
1. <Button
2.     android:text="Button"
3.     android:layout_width="wrap_content"
4.     android:layout_height="wrap_content" />
5. <CheckBox
6.     android:text="CheckBox"
7.     android:layout_width="wrap_content"
8.     android:layout_height="wrap_content" />
9. <RadioButton
10.    android:text="RadioButton"
11.    android:layout_width="wrap_content"
12.    android:layout_height="wrap_content" />
13. <TextView
14.    android:text="TextView"
15.    android:layout_width="wrap_content"
16.    android:layout_height="wrap_content" />
17. <EditText
18.    android:text="EditText"
19.    android:layout_height="wrap_content"
20.    android:layout_width="match_parent" />
21. <ImageView
22.    android:src="@drawable/icon"
23.    android:layout_width="wrap_content"
24.    android:layout_height="wrap_content" />
25. <ListView
26.    android:layout_width="wrap_content"
27.    android:layout_height="fill_parent"
28.    android:entries="@array/listView" />
```

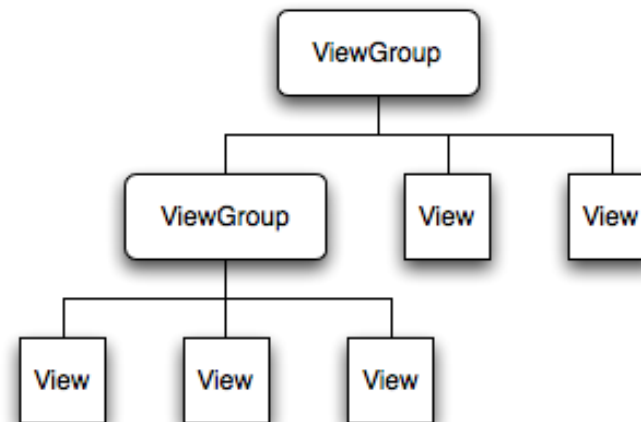


Layouty (rozvržení)

- Dědí od ViewGroup
- Kontejner pro widgety (View) a jiné layouty (ViewGroup)
- Vychází z `java.awt` layoutů
- Možné vytvářet programově nebo **pomocí XML**
- Umístěny ve složce `res/layout`
- Nastavení šířky a výšky stejné jako u widgetů

- Typy:

- `FrameLayout`
- **`LinearLayout`**
- **`RelativeLayout`**
- `TableLayout`
- `GridLayout`



LinearLayout

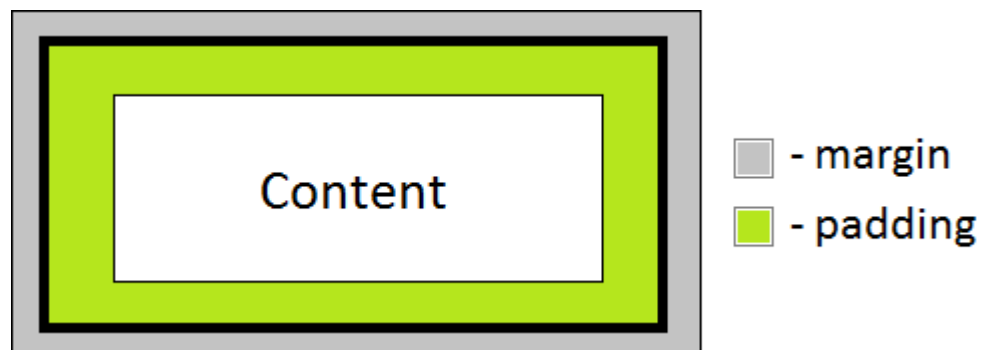
- Srovná všechny widgety vertikálně nebo horizontálně
- Lze vkládat nejruzněji do sebe
 - Problémy s rychlostí vykreslování při velké úrovni vnořování

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout
3.     xmlns:android="http://schemas.android.com/apk/
4.         res/android"
5.         android:orientation="vertical"
6.         android:layout_width="fill_parent"
7.         android:layout_height="fill_parent">
8.     <TextView
9.         android:layout_width="fill_parent"
10.        android:layout_height="wrap_content"
11.        android:text="Hello World, HelloWorld"
12.        android:textSize="50sp" />
12.</LinearLayout>
```



Atributy View a ViewGroup

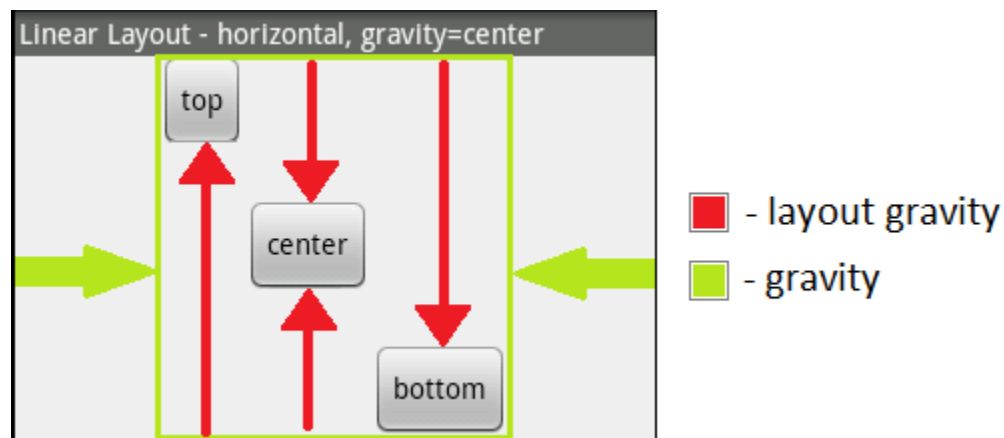
- ID – jedinečné ID View
- Padding – vnitřní okraj
- Margin – vnější okraj



- Layout weight – určuje jakou část volného prostoru View získá

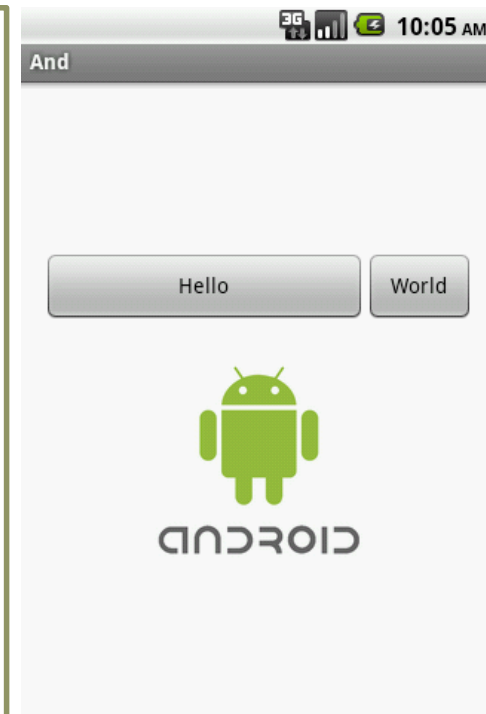
- Gravity
 - Pozicování obsahu ViewGroup

- Layout gravity
 - Pozicování View uvnitř rodiče
 - Způsob, jakým se má „dotýkat“ rodiče



Složitější LinearLayout

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:layout_width="fill_parent"
4.     android:layout_height="fill_parent"
5.     android:orientation="vertical"
6.     android:gravity="center">
7.     <LinearLayout android:orientation="horizontal"
8.         android:layout_width="fill_parent"
9.         android:layout_height="wrap_content"
10.        android:layout_margin="15dp">
11.         <Button android:text="Hello"
12.             android:layout_width="0dp"
13.             android:layout_height="wrap_content"
14.             android:layout_weight="3" />
15.         <Button android:text="World"
16.             android:layout_width="0dp"
17.             android:layout_height="wrap_content"
18.             android:layout_weight="1" />
19.     </LinearLayout>
20.     <ImageView android:src="@drawable/android"
21.         android:layout_width="wrap_content"
22.         android:layout_height="wrap_content"
23.         android:id="@+id/imageView1" />
24.</LinearLayout>
```



Activity

- Prezentační vrstva aplikace
- Každá musí být definována v Android Manifestu
- **Všechny activity běží v rámci jednoho procesu**
 - Pokud není v Android Manifestu řečeno jinak
- Mohou se navzájem spouštět a ovlivňovat
 - Při spuštění jedné je druhá pozastavena a **umístěna na zásobník**
 - Při stisknutí Back Button je opět vyvolána
- Řídí se životním cyklem (3. přednáška)

Hello World

- Vytvořit ve složce `res/layout` soubor `main.xml` s layoutem, textem a tlačítkem

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <LinearLayout
3.      xmlns:android="http://schemas.android.com/apk/res/android"
4.      android:orientation="vertical"
5.      android:layout_width="fill_parent"
6.      android:layout_height="fill_parent">
7.
8.      <TextView
9.          android:id="@+id/myTextView"
10.         android:layout_width="wrap_content"
11.         android:layout_height="wrap_content"
12.         android:textSize="50sp" />
13.     <Button
14.         android:text="Click me"
15.         android:id="@+id/myButton"
16.         android:layout_width="wrap_content"
17.         android:layout_height="wrap_content" />
18.
19. </LinearLayout>
```

Vytvoření activity

MainActivity.java:

```
1. public class MainActivity extends Activity {  
2.     @Override  
3.     public void onCreate(Bundle savedInstanceState) {  
4.         super.onCreate(savedInstanceState);  
5.         setContentView(R.layout.main);  
6.     }  
7. }
```

Nezapomenout přidat do `AndroidManifest.xml`:

```
1. <activity  
2.     android:name=".MainActivity"  
3.     android:label="@string/app_name">  
4. </activity>
```

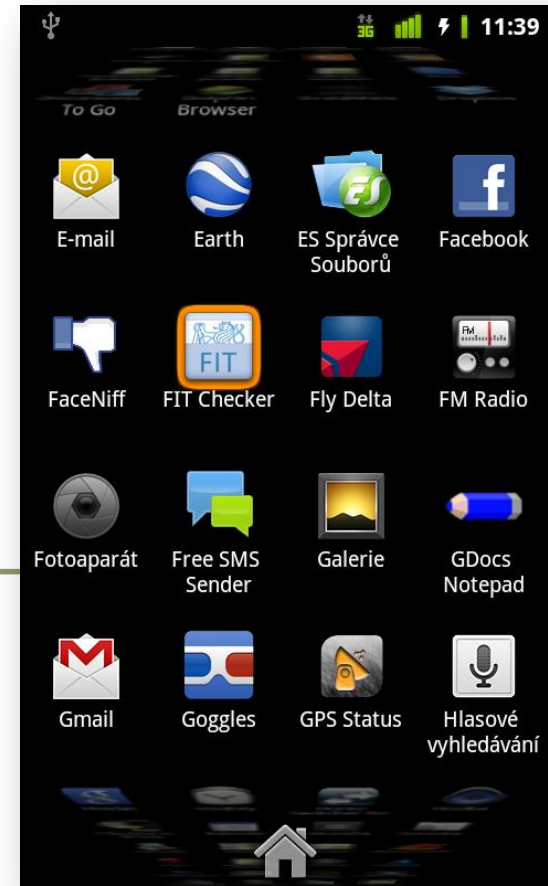
cz.cvut.hello.world.MainActivity.java



Ikonka v app launcheru

- Pro přidání ikonky do app launcheru je zapotřebí přidat příslušný `intent-filter`

```
1. <activity
2.     android:name=".MainActivity"
3.     android:label="FIT Checker"
4.     <intent-filter>
5.         <action
6.             android:name="android.intent.action.MAIN" />
7.         <category
8.             android:name="android.intent.category.LAUNCHER"
9.         />
10.</intent-filter>
11.</activity>
```



MainActivity.java

- Nastavíme aktivitě layout

res/layout/main.xml



```
setContentView(R.layout.main) ;
```

- Získáme reference na widgety, se kterými chceme pracovat

```
TextView myTextView = (TextView) findViewById(R.id.myTextView) ;  
Button myButton = (Button) findViewById(R.id.myButton) ;
```


Nastavení události

- Změna textu v *TextView*:

```
myTextView.setText("Hello World");
```

- Nastavíme tlačítku událost (*View.OnClickListener*)

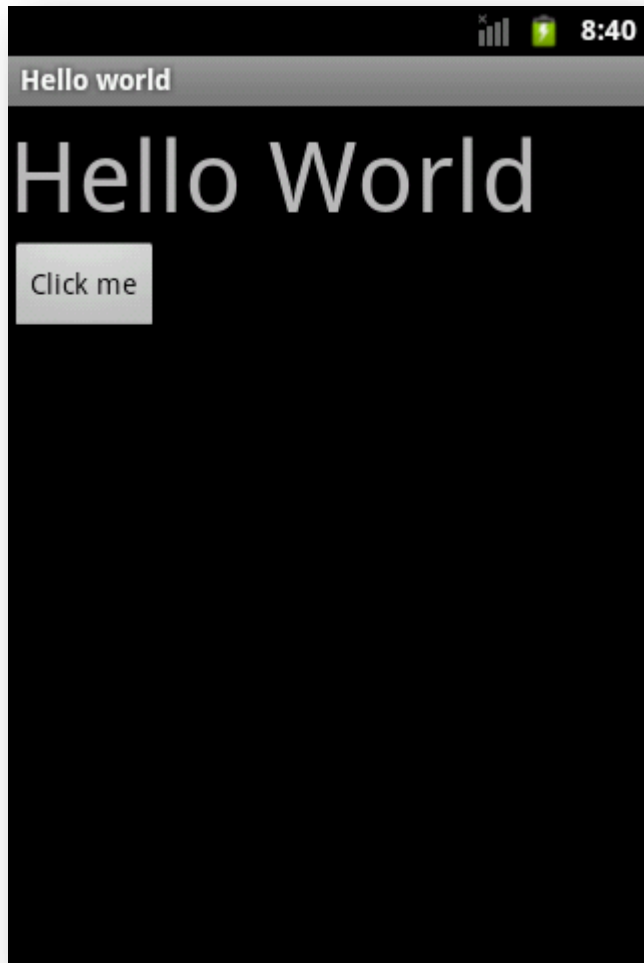
```
1. myButton.setOnClickListener(new View.OnClickListener() {  
2.     @Override  
3.     public void onClick(View v) {  
4.         myTextView.setText("Android FTW!!!");  
5.         myButton.setVisibility(View.INVISIBLE);  
6.     }  
7. });
```

Activity – interakce s widgety

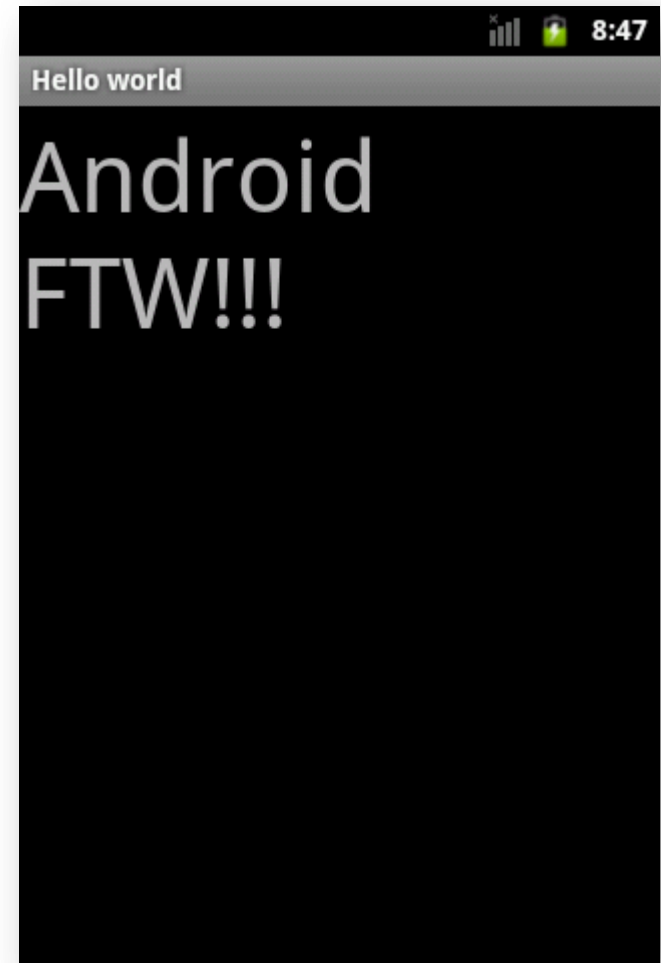
- MainActivity.java

```
1.  public class MainActivity extends Activity {
2.  private TextView myTextView;
3.  private Button myButton;
4.
5.  @Override
6.  public void onCreate(Bundle savedInstanceState) {
7.      super.onCreate(savedInstanceState);
8.      setContentView(R.layout.main);
9.      myTextView = (TextView) findViewById(R.id.myTextView);
10.     myButton = (Button) findViewById(R.id.myButton);
11.     myTextView.setText("Hello World");
12.     myButton.setOnClickListener(new OnClickListener() {
13.         @Override
14.         public void onClick(View v) {
15.             myTextView.setText("Android FTW!!!");
16.             myButton.setVisibility(View.INVISIBLE);
17.         }
18.     });
19. }
20. }
```

Activity – interakce s widgety



Click
→



Další zdroje

- <http://java.sun.com/developer/Books/javaprogramming/JAR/basics/manifest.html>
- <http://java.sun.com/developer/Books/javaprogramming/JAR/sign/intro.html#digest>
- <https://plus.google.com/u/0/108967384991768947849/posts/DcsFABkyuYM>
- <http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources>
- <http://docs.oracle.com/javase/tutorial/java/package/namingpkgs.html>