

# Uvod u Android

## Android App Development 5

20. januar 2018.

# Agenda

- 1 Kurs "Android App Development"
- 2 Android platforma

# Platforma



Figure 1: Samsung Galaxy Tab S2



Figure 2: Android 7.0 Nougat

# Programski alati

- JDK
- Android SDK
- Android Studio

# Literatura

- Charlie Collins, Michael Galpin, Matthias Käppler, "Android in Practice", Manning Publications, 2012
- Jonathan Simon, "Head First Android Development", O'Reilly Media, Inc., 2015
- Android Developers, <http://developer.android.com/>
- Stack Overflow, <http://stackoverflow.com/>
- Google, <http://www.google.com/>

# Website

Website:

- <https://enastava.ftninformatika.com>
- <https://gitlab.com/groups/AndroidAppDevelopment>

# Agenda

- 1 Kurs "Android App Development"
- 2 Android platforma

# Šta je Android?

- OS
- Middleware libraries
- Application framework
- Applications
- SDK



# Istorija Androida

- Android, Inc.
- Google
- Open Handset Alliance (OHA)

# Istorija Androida



Figure 3: Istorija Androida

# Karakteristike Androida

- FLOSS (OS, VM, API, Apps, SDK)
- multitasking (više procesa, više niti)
- sandboxing (izolovanje aplikacija)

# Arhitektura Androida

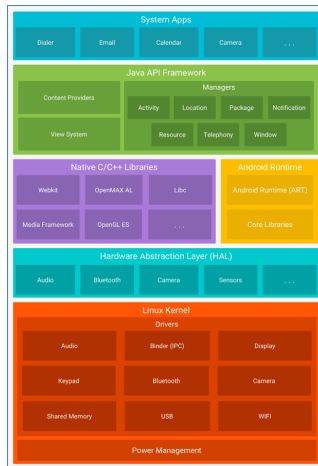


Figure 4: Arhitektura Androida

# Android platforma

- GNU/Linux
  - Bionic C
- Dalvik (just-in-time)
  - Apache Harmony
- Android Run Time (ART) (ahead-of-time)
  - Apache Harmony
  - OpenJDK

# Inicijalizacija aplikacija

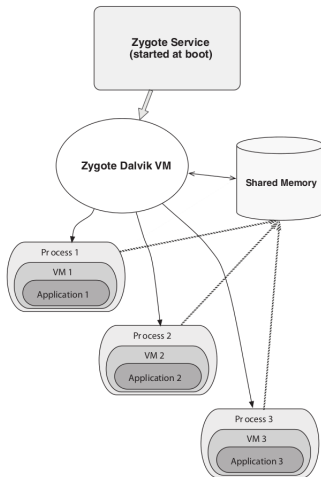


Figure 5: Inicijalizacija aplikacija

# Verzije Androida

- Android 7.0 Nougat
- API level 24

# Komponente Android aplikacije

- Šta je softverska komponenta?
- Softverska komponenta je jedinica kompozicije softvera sa specificiranim interfejsom i eksplicitnom kontekstnom zavisnošću.



# Komponente Android aplikacije

- Aktivnost (activity)
- Servis (service)
- Prijemnik poruka (broadcast receiver)
- Dobavljač sadržaja (content provider)

# Aktivnosti

- Aktivnost predstavlja pojedinačan ekran Android aplikacije
- Aplikacija se sastoji iz više slabo povezanih aktivnosti
- Glavna aktivnost je aktivnost koja će se prikazati korisniku kada pokrene aplikaciju

# Activity

```
package com.example.project;  
import android.app.Activity;  
  
public class ExampleActivity extends Activity {  
  
    @Override  
    public void onCreate (...) {  
        ...  
    }  
  
    ...  
  
    @Override  
    public void onDestroy () {  
        ...  
    }  
}
```

# Activity

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
  <application ... >
    <activity android:name=".ExampleActivity" ... >
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

# Servisi

- Servis se izvršava u pozadini
- Obično ne zahteva interakciju sa korisnikom
- Omogućava izvršavanje više paralelnih zadataka (multitasking) i implementaciju klijent-server softverske arhitekture

# Service

```
package com.example.project ;
import android.app.Service ;

public class ExampleService extends Service {

    @Override
    public void onCreate () {
        ...
    }

    @Override
    public int onStartCommand (...) {
        ...
    }

    @Override
    public void onDestroy () {
        ...
    }
}
```

# Service

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
  <application ... >
    <service android:name=".ExampleService" ... />
  </application>
</manifest>
```

# Prijemnici poruka

- Prijemnici poruka obrađuju asinhronne događaje
- Događaje može da izazove operativni sistem ili druga komponenta



# Broadcast Receiver

```
package com.example.project ;  
import android.content.BroadcastReceiver ;  
  
public class ExampleReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive (... ) {  
        ...  
    }  
}
```

# Broadcast Receiver

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
  <application ... >
    <receiver android:name=".ExampleReceiver" ...>
      <intent-filter>
        <action name="android.provider.Telephony.SMS_RECEIVED" />
      </intent-filter>
    </receiver>
  </application>
</manifest>
```

# Dobavljači sadržaja

- Dobavljači sadržaja upravljaju podacima
- Omogućavaju razmenu podataka između aplikacija (tj. komunikaciju između procesa)

# Content Provider

```
package com.example.project;  
import android.content.ContentProvider;  
  
public class ExampleProvider extends ContentProvider {  
  
    @Override  
    public void onCreate() {  
        ...  
    }  
  
    @Override  
    public ... query(...) {  
        ...  
    }  
  
    @Override  
    public ... insert(...) {  
        ...  
    }  
  
    @Override  
    public ... update(...) {  
        ...  
    }  
  
    @Override  
    public ... delete(...) {  
        ...  
    }  
}
```

# Content Provider

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
  <application ... >
    <provider android:name=".ExampleProvider" ... />
  </application>
</manifest>
```

# Namera

- Poruka koje se šalje između komponenti
- Apstraktni opis operacije koja treba da se izvrši
- Služi za povezivanje komponenti Android aplikacije

# Koraci u razvoju Android aplikacija

- design
- develop (Android Studio, Android SDK, JDK, itd.)
- distribute (Google Play)

# Organizacija izvornog koda

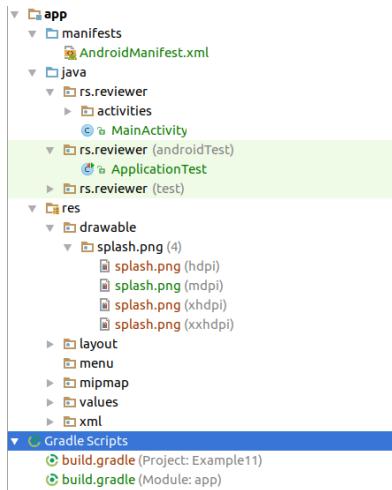


Figure 6: Organizacija izvornog koda



# Alati

- JDK (Java Development Kit)
- Android SDK (Software Development Kit)
- Android Studio (IDE)
  - Gradle
  - Git

# Android SDK

- SDK tools (alati koji ne zavise od platforme)
- SDK platform tools (backward compatible alati koji zavise od platforme)
- SDK platform (Android API)
- systemske slike
- dokumentacija
- primeri
- izvorni kod

# SDK (Platform) Tools

- Android SDK Manager (android sdk)
- AVD Manager (android avd)
- Emulator (emulator)
- Dalvik VM Bytecode Compiler (dx)
- Android Asset Packaging Tool (aapt)
- Android Debug Bridge (adb)
- Dalvik Debug Monitor Service (ddms)
- SQLite (sqlite3)

# Android Studio

- Android Studio je razvojno okruženje za Android aplikacije (zasnovano na IntelliJ)
- interno koristi SDK (platform) tools
- za izgradnju projekta koristi Gradle
- za upravljanje izvornim kodom koristi Git

# Android Studio

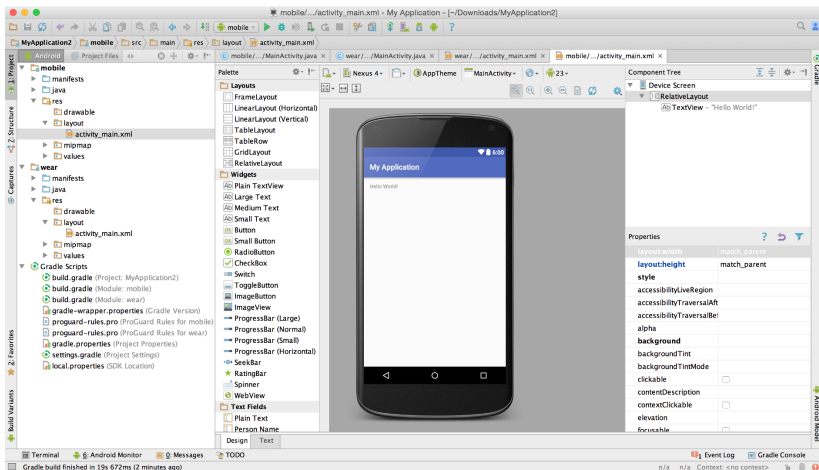


Figure 7: Android Studio

# Proces izgradnje Android aplikacija

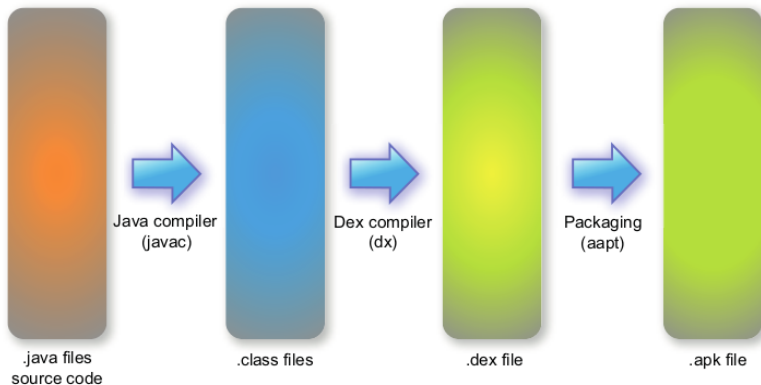


Figure 8: Izgradnja Android aplikacije

# Google Play

- Google Play je platforma za distribuciju Android aplikacija (kao i filmova, muzike, knjiga, itd.)
- Da bi se aplikacija objavila na Google Play, potrebno je da prođe kroz kontrolu kvaliteta

# Google Play

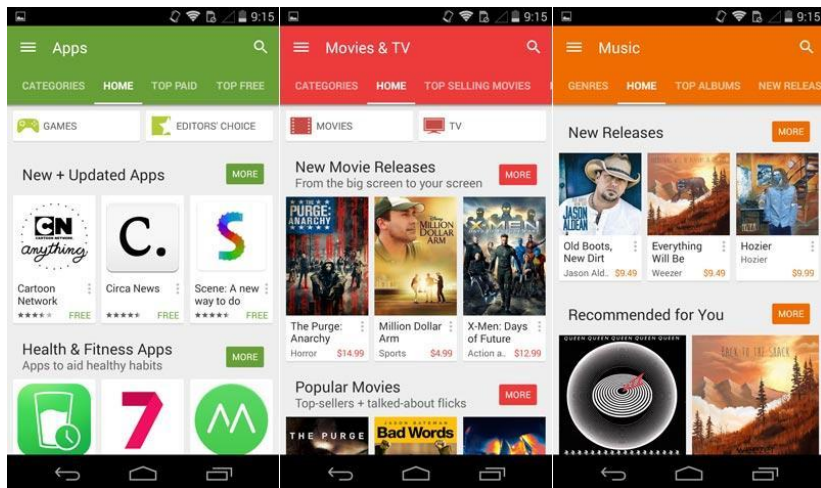


Figure 9: Google Play



# Pitanja?



Figure 10: Pitanja?