

# Uvod

## Mobilne aplikacije

# Agenda

- 1 Mobilni uređaji
- 2 Android platforma

# Karakteristike mobilnih uređaja

- mobilnost
- ograničeni hardverski resursi
- korisnički interfejs
- mobilne komunikacije
- senzori

# Mobilnost

- dimenzije, masa i napajanje dopuštaju prenošenje uređaja
- uređaj može koristiti servise bazirane na lokaciji

## Ograničeni hardverski resursi

- centralni procesor
- grafički procesor
- operativna memorija
- spoljna memorija
- izvor električne energije

# Korisnički interfejs

- ekran osetljiv na dodir
- prepoznavanje i sinteza govora
- senzori

# Mobilne komunikacije

- mobilna mreža
- Wi-Fi mreža
- Bluetooth
- NFC

# Senzori

- senzori pozicije (GPS, mobilna mreža, Wi-Fi mreža)
- senzori kretanja (akcelerometar, žiroskop)
- senzori okoline (magnetno polje, temperatura, pritisak, osvetljenost, blizina objekata)

# Vrste mobilnih uređaja

- pametni telefoni
- tablet računari
- wearables (satovi, naočare, itd.)
- namenski mobilni uređaji (fotoaparati, kamere, GPS, itd.)

# Google Glass



Figure 1: Google Glass

# Android Wear



Figure 2: Android Wear

# Uređaji sličnih karakteristika

- TV/STB
- uređaji u automobilima
- uređaji u kući (frižider, mikrotalasna pećnica, itd.)



Figure 3: Android TV



Figure 4: Android Auto

# Zašto Android?

- Udeo pametnih telefona i tablet računara u tržištu računara
- Udeo mobilnih uređaja na Android platformi u tržištu mobilnih uređaja
- Udeo Android aplikacija u tržištu aplikacija

# Tržište mobilnih uređaja

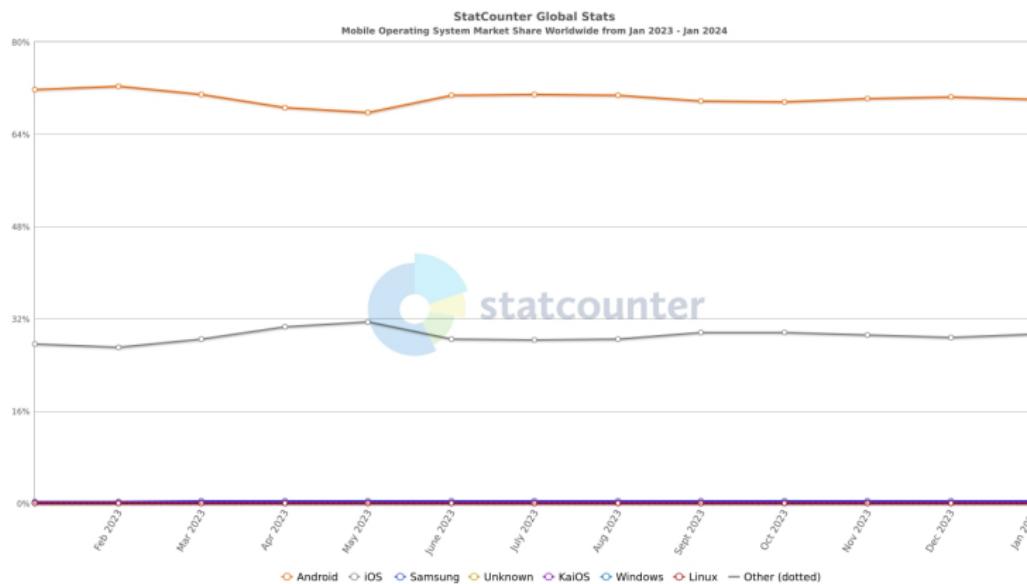
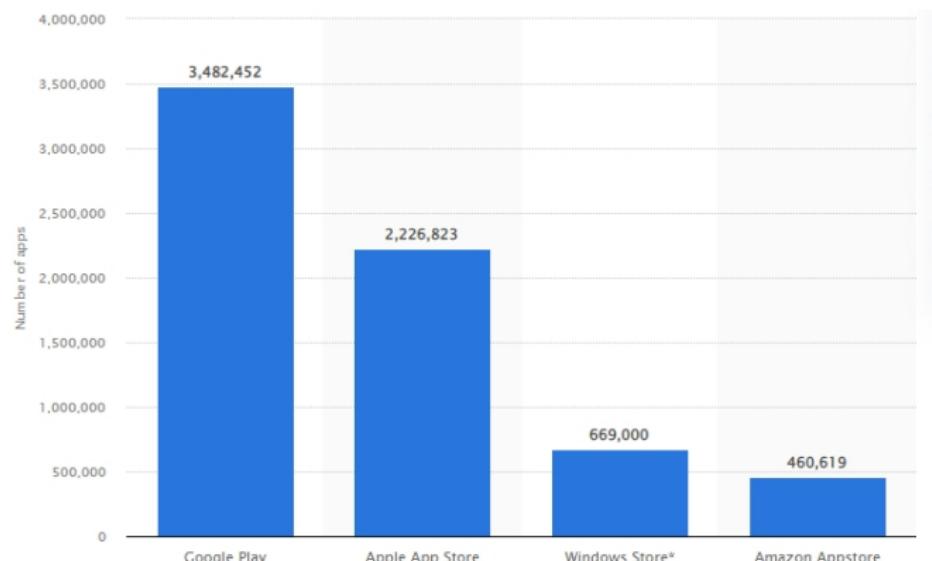


Figure 5: Udeo Android mobilnih uređaja

# Broj aplikacija u prodavnicama



© Statista 2022

Figure 6: Broj aplikacija u prodavnicama

# Native vs. Web Apps

- Android/iOS
- HTML5/CSS/JavaScript

# Hybrid Apps

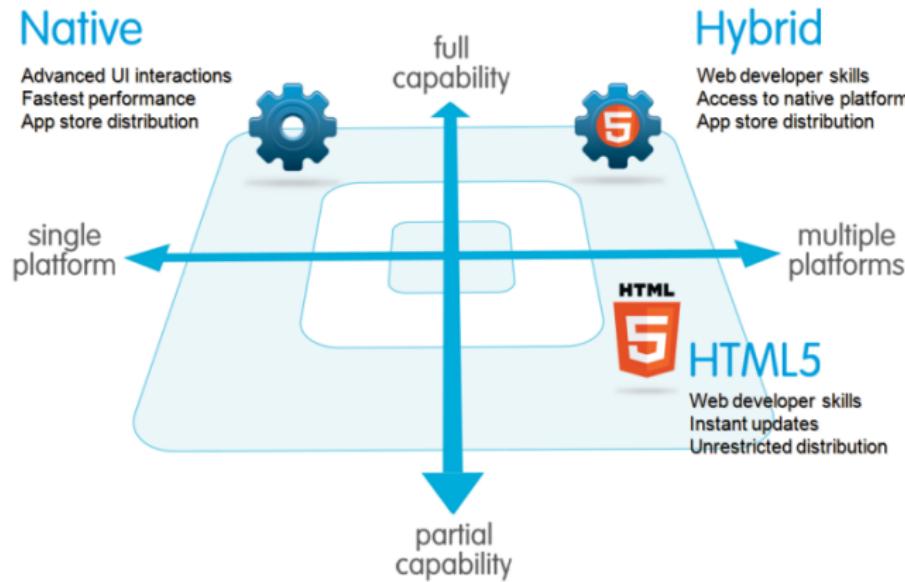


Figure 7: Native/Mobile/Hybrid Apps

# Hybrid Frameworks

- Xamarin
- React Native
- Flutter
- NativeScript
- Cordova (Ionic)

# Native vs. Hybrid Apps

Native Apps	Hybrid Apps
UI/UX	portabilnost
performanse	brži razvoj

Table 1: Native vs. Hybrid Apps

# Agenda

- 1 Mobilni uređaji
- 2 Android platforma

# Šta je Android?

- OS
- Application framework
- Applications

# Istoriјa Androida

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

# Istorija Androida

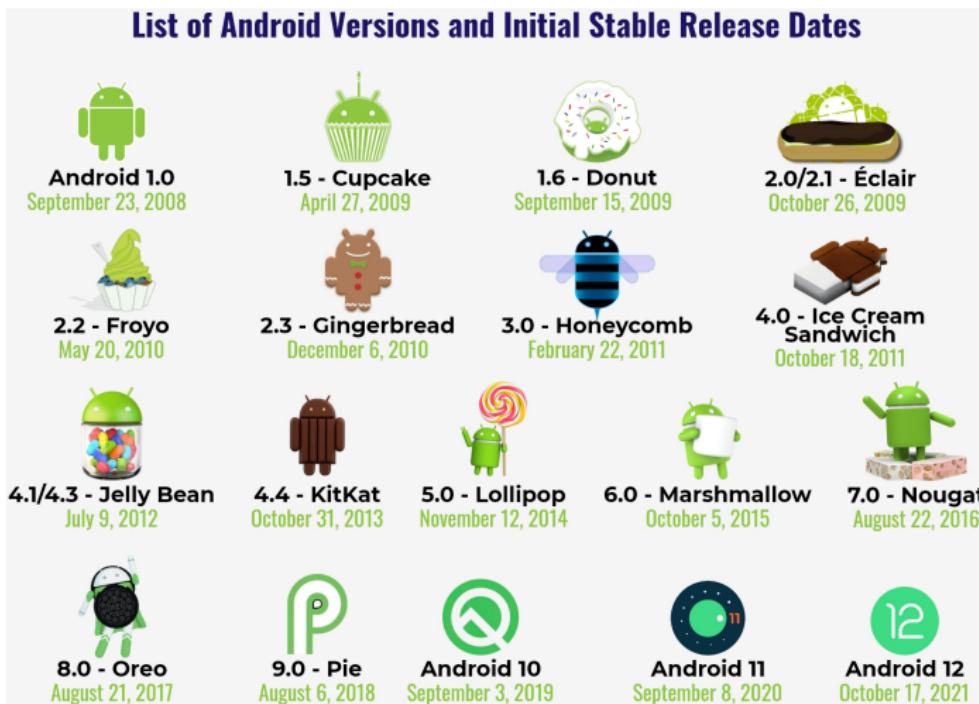


Figure 8: Istorija Androida

# Karakteristike Androida

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

# Arhitektura Androida



Figure 9: Arhitektura Androida

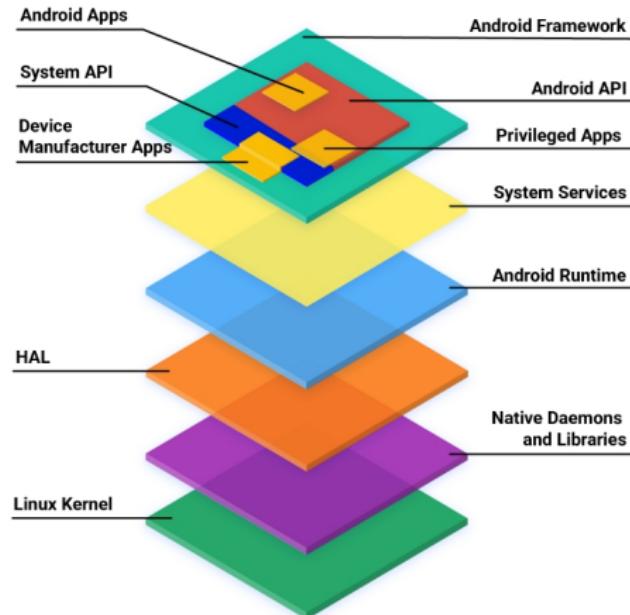


Figure 10: Android stack

# Android platforma

- GNU/Linux
  - Bionic C biblioteka
- JVM
  - Dalvik do Androida 4.4
  - Android Run Time (ART) od Androida 5.0
- standardna Java biblioteka
  - Apache Harmony do Androida 6.0
  - OpenJDK od Androida 7.0

# Inicijalizacija Android aplikacija

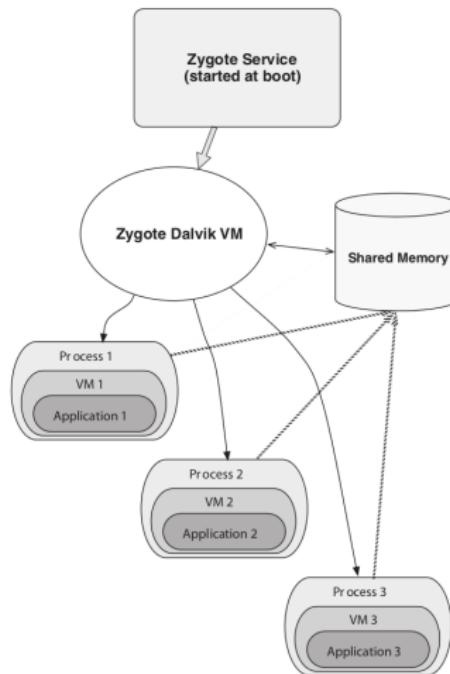


Figure 11: Inicijalizacija aplikacija

# Verzije Androida

- Android 11 Red Velvet Cake
- Android API level 30

# Komponente Android aplikacije

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

# Komponente Android aplikacije

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

# Aktivnosti

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

# Activity

```
1 package com.example.project;
2 import androidx.appcompat.app.AppCompatActivity;
3
4 public class ExampleActivity extends AppCompatActivity {
5
6     @Override
7     public void onCreate(...) {
8         ...
9     }
10
11     ...
12
13     @Override
14     public void onDestroy() {
15         ...
16     }
17
18 }
```

# Activity

```
<?xml version="1.0" encoding="utf-8"?>
1 <manifest ... >
2   <application ... >
3     <activity android:name=".ExampleActivity" ... >
4       <intent-filter>
5         <action android:name="android.intent.action.MAIN" />
6         <category android:name="android.intent.category.LAUNCHER" />
7       </intent-filter>
8     </activity>
9   </application>
10 </manifest>
```

12

# Servisi

- Servis se izvršava u pozadini
- Ne zahteva interakciju sa korisnikom

# Service

```
1 package com.example.project;
2 import android.app.Service;
3
4 public class ExampleService extends Service {
5
6     @Override
7     public void onCreate() {
8         ...
9     }
10
11     @Override
12     public int onStartCommand(...) {
13         ...
14     }
15
16     @Override
17     public void onDestroy() {
18         ...
19     }
20 }
```

# Service

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4     <service android:name=".ExampleService" ... />
5   </application>
6 </manifest>
```

# Prijemnici poruka

- Prijemnici poruka obrađuju asinhrone događaje
- Događaje mogu da izazovu operativni sistem ili druga komponenta

# Broadcast Receiver

```
1 package com.example.project;
2 import android.content.BroadcastReceiver;
3
4 public class ExampleReceiver extends BroadcastReceiver {
5
6     @Override
7     public void onReceive(...) {
8         ...
9     }
10 }
```

# Broadcast Receiver

```
<?xml version="1.0" encoding="utf-8"?>
1 <manifest ... >
2   <application ... >
3     <receiver android:name=".ExampleReceiver" ... >
4       <intent-filter>
5         <action name="android.provider.Telephony.SMS_RECEIVED" />
6       </intent-filter>
7     </receiver>
8   </application>
9 </manifest>
```

# Dobavljači sadržaja

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

# Content Provider

```
1 package com.example.project;
2 import android.content.ContentProvider;
3
4 public class ExampleProvider extends ContentProvider {
5
6     @Override
7     public void onCreate() {
8         ...
9     }
10
11     @Override
12     public ... query(...) {
13         ...
14     }
15
16     @Override
17     public ... insert(...) {
18         ...
19     }
20
21     @Override
22     public ... update(...) {
23         ...
24     }
25
26     @Override
27     public ... delete(...) {
28         ...
29     }
30 }
```

# Content Provider

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

# Namera

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

# Resursi

- Android aplikacije pored izvornog koda (najčešće pisano u Javi) sadrže i resurse
- Resursi mogu bili tekst, rasterska i vektorska grafika, audio i video klipovi, itd.

# Organizacija Android projekta

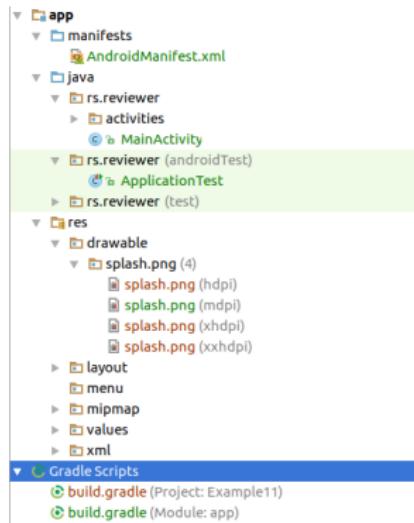


Figure 12: Organizacija izvornog koda

# Koraci u razvoju Android aplikacija

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

# Alati

- Java Development Kit (JDK)
- Android SDK (Software Development Kit)
- Android Studio (Android 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)
- sistemske 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 IntelliJ)
- interno koristi SDK (platform) tools
- za izgradnju projekta koristi Gradle
- za upravljanje izvornim kodom koristi Git

# Android Studio

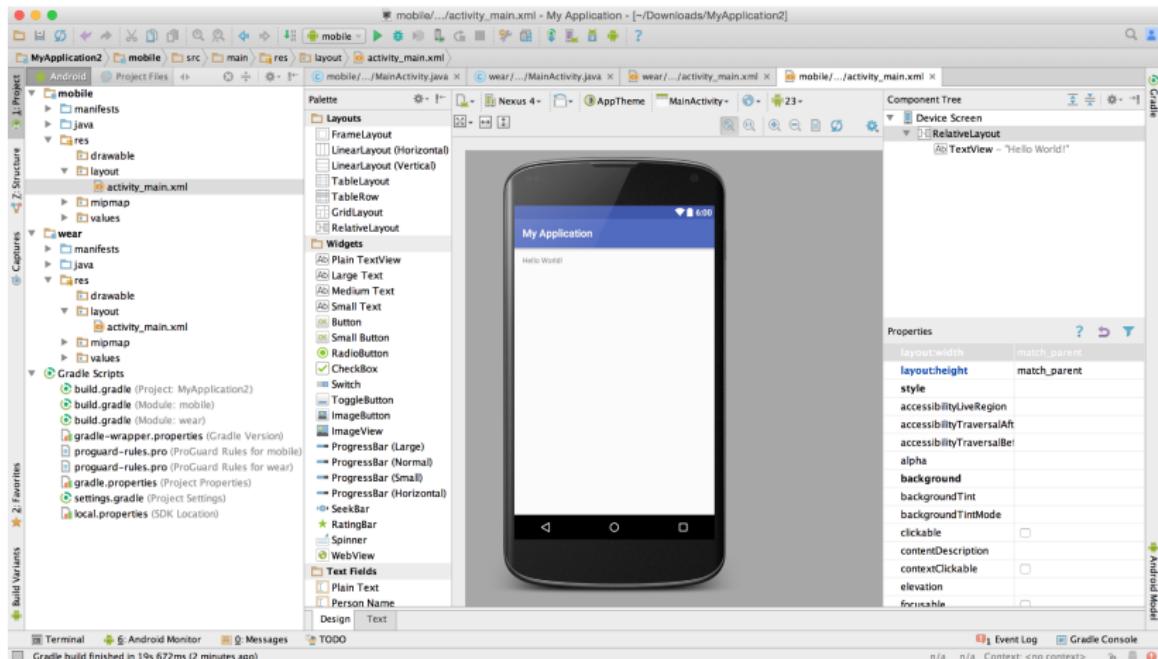


Figure 13: Android Studio

# Proces izgradnje Android aplikacija

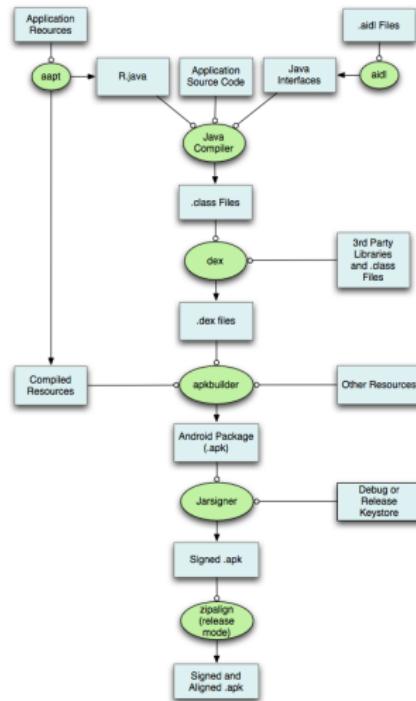


Figure 14: Izgradnja Android aplikacije