

Sveučilište Josipa Jurja Strossmayera u Osijeku  
Odjel za matematiku  
Sveučilišni nastavnički studij matematike i informatike

*Seminarski rad iz kolegija*  
**Web programiranje i primjene**

*Naslov rada:*  
Programski jezici za mobilne aplikacije

Ivana Šimić, 875  
Katarina Vincetić, 898

Osijek, siječanj 2013.

## Sadržaj

Sažetak .....	1
1. Uvod .....	1
2. Android operativni sustav .....	2
2.1. Java .....	2
2.2. Programiranje mobilnih aplikacija za Android OS .....	3
3. Zaključak .....	7
Literatura .....	8

## Sažetak

Na početku rada dan je kratak uvod u aplikacije za mobilne uređaje. Također, dan je i pregled najpoznatijih operativnih sustava za navedene uređaje te popis najčešćih programskih jezika koji se koriste prilikom programiranja aplikacija za te uređaje. U nastavku donosimo malo više o programiranju aplikacija za uređaje s operativnim sustavom Android uz primjere u programskim jezicima Java i HTML5.

Ključne riječi: programski, jezik, mobilne, aplikacije, Android, Java, XML, HTML5

## 1. Uvod

Razvoj mobilnih aplikacija eksplodirao je Appleovim predstavljanjem iPhonea 2007. godine. U međuvremenu su se tržištu mobilnih aplikacija pridružili i Google i Microsoft, te je danas ponuda ovih aplikacija postala jedan od glavnih faktora pri kupnji mobilnog uređaja. Zbog velike popularnosti pametnih telefona i tablet uređaja, očekuje se da će vrijednost tržišta mobilnih aplikacija u sljedeće tri godine rasti 30% godišnje. Znanje razvoja mobilnih aplikacija jako je traženo i financijski isplativo, stoga ne čudi da se sve više i više programera odlučuje upravo za izradu takvih aplikacija.

Kod programiranja aplikacija za mobilne uređaje razlikujemo dvije vrste aplikacija: nativne i web aplikacije. Obje se svode na isto, i izgledaju jako slično. Ipak, postoje određene razlike. Po definiciji, nativne aplikacije su prilagođene određenom operativnom sustavu, dok se web aplikacija može izvršavati na svim operativnim sustavima u web pregledniku. Što znači da je za nativne aplikacije potrebno izraditi posebne verzije za svaki od operativnih sustava, dok se mobilna web aplikacija može koristiti na svakom operativnom sustavu. Zbog toga je izrada web aplikacije jeftinija i potrebno je manje vremena za njezin razvoj. Također, kod web aplikacije svi korisnici koriste istu verziju, dok se kod nativnih može dogoditi da neki korisnik koristi stariju verziju u kojoj ima još nekih bugova. Nadalje, web aplikaciju nije potrebno instalirati na uređaj, dok se nativna mora instalirati i time zauzima dio resursa uređaja. Ali i nativne aplikacije imaju određene prednosti nad web aplikacijama. Tako na primjer, ova vrsta mobilne aplikacije pruža mogućnost povezivanja sa funkcijama uređaja, kao što su kamera, akcelerometar, podaci iz memorije uređaja i slično; dok web aplikacija ima ograničen set mogućnosti. Također, nativna aplikacija ne ovisi o mreži, pa je njezino izvođenje brže od izvođenja web aplikacije. Vidimo da je prilikom odabira potrebno imati na umu na koje tržište ciljamo, koliko vremena imamo za izradu aplikacije, koliko funkcionalnosti uređaja nam je potrebno itd.

Donosimo kratak pregled operativnih sustava i programskih jezika koji se najčešće koriste za programiranje nativnih aplikacija za njih:

- Android: Java, C/C++, C#, Simple
- BlackBerry: Java
- Symbian: C++
- Windows Mobile: C/C++, Visual C++
- iOS: C#, Objective C

Za izradu web aplikacija se sve više koriste HTML5 i CSS3 uz JavaScript.

U nastavku ćemo obraditi programske jezike za jedan od najpopularnijih operativnih sustava za mobilne uređaje - Android.

## 2. Android operativni sustav



Slika 1. Logo operativnog sustava Android

Android je operativni sustav baziran na Linuxu. Dizajniran je primarno za mobilne uređaje poput telefona sa zaslonom na dodir i tablet uređaja. Stvorila ga je tvrtka Android, Inc. koju je 2005. kupio Google. Prvi telefon kojemu je Android bio operativni sustav prodan je u listopadu 2008. godine.

Android je projekt otvorenog koda što znači da je sam njegov kod dostupan na njegovoj službenoj web stranici. Općenito, svatko može skinuti kod i optimizirati ga za svoje potrebe i pridonijeti razvijanju ovog operativnog sustava. Google paralelno uvijek radi na novoj inačici, u koju može uvrstiti i dijelove koda koji razvijaju sustav u (njima) dobrom smjeru.

### 2.1. Java

Programski jezik Java trenutno je najčešći za programiranje aplikacija za uređaje s operativnim sustavom Android. Isto tako, Java je još uvijek jedan od najraširenijih programskih jezika općenito. Postoje određene razlike između programiranja u ovom programskom jeziku za desktop aplikacije u odnosu na programiranje aplikacija za mobilne uređaje, ali pogledajmo prvo malo općenito.

Java je programski jezik kojeg je stvorio James Gosling iz Sun Microsystemsa 1995. godine. Programi pisani ovim programskim jezikom kompajliraju se u bytecode koji se može izvršiti na bilo kojem uređaju ako ima JVM (Java Virtual Machine). To znači da možemo napisati kod na jednom računalu i izvršavati na bilo kojem drugom bez obzira na arhitekturu računala. Potrebno je samo instalirati JVM. Sintaksa ovog jezika jako je slična sintaksi programskih jezika C i C++. Kao primjer pogledajmo „Hello world“ program napisan u C++-u i u Javi:



Slika 2. Logo programskog jezika Java

Java	C++
<pre>class HelloWorldApp {     public static void main(String[] args) {         System.out.println("Hello World!");     } }</pre>	<pre>#include&lt;iostream&gt; using namespace std;  int main() {     cout &lt;&lt; "Hello world!" &lt;&lt; endl;     return 0; }</pre>

Tablica 1. Usporedba sintakse Jave i C++-a na jednostavnom primjeru

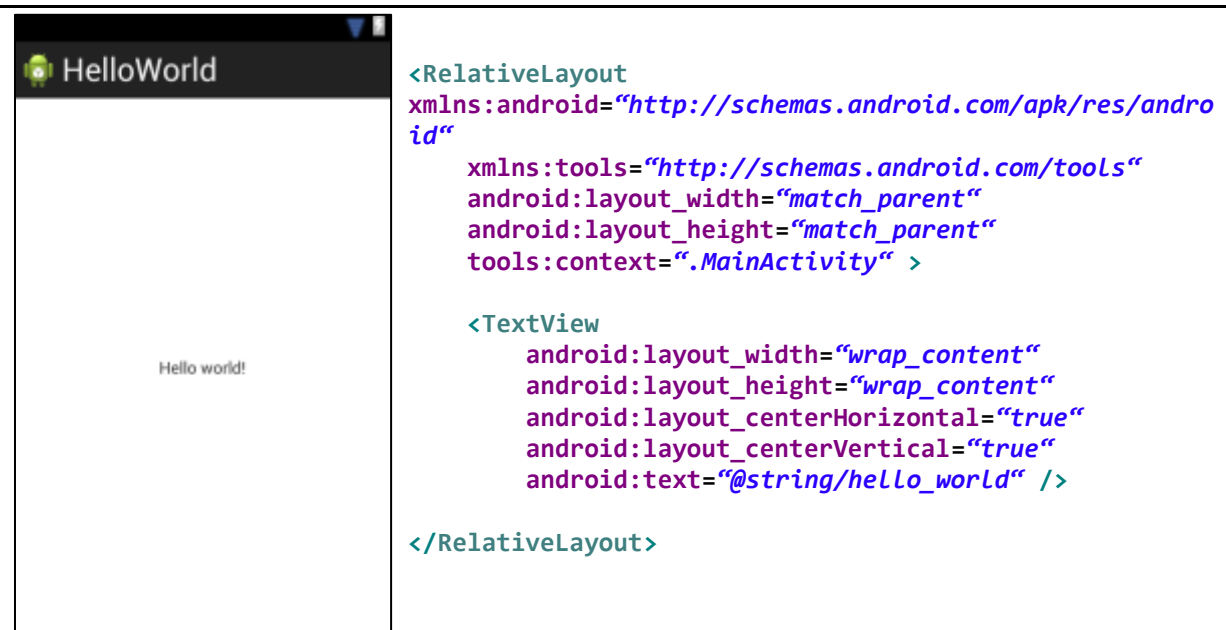
Iako se razlika sintakse u jednostavnijim programima može činiti velika, u složenijima se vidi dosta sličnosti. Logika programiranja u ovim jezicima je ista, pa prijelazom s jednog programskog jezika na drugi nije potrebno ništa drugo osim možda upoznavanja sa sintaksom jezika na koji prelazimo.

Objasnimo sada malo kod Java programa: kao što vidimo, prvo je potrebno otvoriti novu klasu, ovdje smo ju nazvali HelloWorldApp. Klasa u kojoj se nalazi main funkcija treba imati isto ime kao i sama java datoteka. Izvršavanje programa počinje izvršavanjem funkcije main (kao i u C++-u). Ispis ovdje izvršavamo funkcijom System.out.println koju (najčešće) koristimo za ispis jednog reda na ekran. Kako je ovdje funkcija main tipa void, nije potrebno na kraju pisati naredbu return kao što smo je pisali u kodu napisanom u C++-u. Ali tamo je funkcija main bila tipa int, pa smo na kraju morali vratiti neku vrijednost.

## 2.2. Programiranje mobilnih aplikacija za Android OS

Nativne aplikacije za Android OS najčešće se programiraju u Java programskom jeziku. Ipak, dijelovi koda mogu biti napisani i u programskim jezicima C i C++. Dio koda pisan u jednom od tri navedena jezika u aplikaciji zapravo određuje događaje. Na primjer, što se dogodi ako korisnik pritisne određeni gumb i slično. Određivanje izgleda aplikacije pišemo u XML-u. Tako kod za jednostavnu mobilnu aplikaciju postaje pravi projekt koji se sastoji od nekoliko dateteka.

Za izradu aplikacija za seminar koristili smo Eclipse. To je jedan od alata koji uvelike olakšava programiranje aplikacija za Android. U nastavku ćemo vidjeti koliko se zapravo sve zakomplicira kada program „Hello world“ želimo kao aplikaciju izvršiti na uređaju s Android operativnim sustavom. Pogledajmo najprije XML kod i izgled aplikacije:



Tablica 2. Izgled aplikacije „Hello world“

Vidimo da prvo pravimo layout koji nam odgovara. Najčešće je početni layout upravo RelativeLayout, pa onda kasnije u njega možemo stavljati i više novih layouta. Odredili smo mu veličinu: rekli smo da se i po dužini i po širini rastegne koliko može tako da popuni cijeli ekran. U sljedećem redu (tools:context) pozivamo Java datoteku koja je u našem slučaju, iako se čini da ima puno koda, trivijalna. Proučimo ostatak XML koda prije nego pređemo na Java datoteku. Sljedeća naredba ovdje „piše po ekranu“. Njome smo rekli programu da tekst centrira i horizontalno i vertikalno, te da sadržaj teksta uzme iz datoteke strings.xml. Navedena datoteka ima sljedeći sadržaj:

---

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">HelloWorld</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>

</resources>
```

---

*Tablica 3. strings.xml*

Pogledajmo sada sadržaj Java datoteke:

---

```
package com.test.helloworld;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

---

*Tablica 4. MainActivity.java*

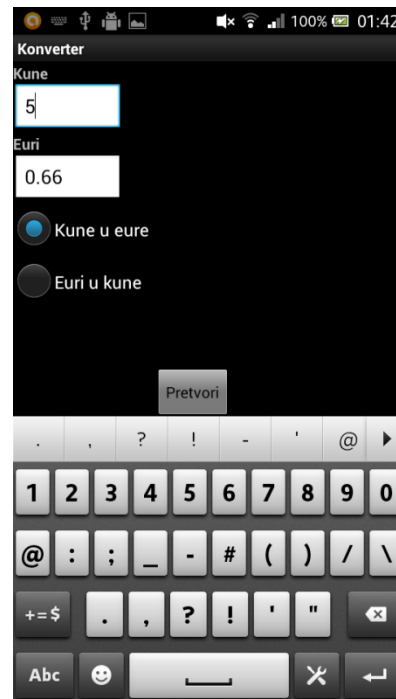
Kod ove datoteke je tzv. „Blank activity“, što zapravo znači da ne radi ništa. U našem primjeru, nema potrebe da nešto radimo u Java kodu, jer smo u XML datoteci rekli našoj aplikaciji da ispiše tekst na ekran. Ovaj dio koda alat Eclipse generira sam. Ako želimo raditi nešto drugo, trebamo to sami napisati. Ovdje kao predložak imamo nekoliko libraryja koji nam mogu trebati i početak klase.

Primjer u kojemu Java datoteka odrađuje velik dio posla možete naći na sljedećem linku: <http://www.mathos.unios.hr/~isimic1/wp-isimic/Seminar/Konverter.zip>. Taj primjer je primjer jednostavne aplikacije koja pretvara kune u eure i obrnuto. Za pretvorbu je uzeta fiksna vrijednost, jer bi skidanje vrijednosti s interneta dodatno zakompliciralo kod.

Donosimo dvije slike aplikacije (pokrenuta je na uređaju Sony Xperia P). U prvoj slici vidimo što imamo ako napišemo samo XML datoteku, tj. vidimo izgled aplikacije, a na drugoj primjer korištenja.



*Slika 3. Aplikacija Konverter prije korištenja*



*Slika 4. Aplikacija Konverter za vrijeme korištenja*

Kad je riječ o programiranju web aplikacija za mobilne uređaje, u posljednje vrijeme sve se više koristi HTML5 sa CSS3 i JavaScriptom. HTML5 je novija inačica opisnog jezika HTML. On donosi neke novosti i nije kompatibilan sa starijim verzijama web preglednika. Jednostavni HTML5 dokument se ne razlikuje puno od HTML dokumenta. Pogledajmo kako izgleda kod minimalnog HTML5 dokumenta:

---

```
<!DOCTYPE html>
<html>
<head>
<title>Naslov dokumenta</title>
</head>

<body>
Tijelo dokumenta...
</body>

</html>
```

---

*Tablica 5. Minimalni HTML5 dokument*

Kao jednu od novosti koju donosi HTML5, pogledajmo canvas. Koristimo ga za crtanje, najčešće korištenjem JavaScripta. Pogledajmo primjer dokumenta koji sadrži element canvas:

---

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px
solid #c3c3c3;">
Vaš preglednik ne podržava HTML5 tagove.
</canvas>

<script>

var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.fillStyle="#FF0000";
ctx.fillRect(0,0,150,75);

</script>

</body>
</html>
```



*Slika 5. Rezultat navedenog koda*

---

*Tablica 6. Primjer dokumenta s elementom canvas*

Kada programiramo web aplikaciju za Android uređaj pomoću HTML5, postupak je sličan kao i kod programiranja native aplikacije u Javi. Potreban nam je XML dokument, koji, zajedno s HTML5 dokumentom i još nekima tvori projekt koji na uređaju daje prikaz aplikacije.



### **3. Zaključak**

U današnje vrijeme sve više i više programera se odlučuje za karijeru programera mobilnih aplikacija. Znanje izrade ovih aplikacija vrlo je traženo i financijski zahvalno.

Postoji više operativnih sustava za mobilne uređaje i svaki od njih ima programski jezik u kojemu je izrada aplikacija za njega najjednostavnija. Kako je Android OS danas najpopularniji operativni sustav za mobilne uređaje, programski jezik Java je najrašireniji. Ali polako se, umjesto nativnih aplikacija koje rade samo na jednom operativnom sustavu, pišu web aplikacije koje je dovoljno napisati samo jednom, i rade na svim operativnim sustavima. Sve popularniji u tom području postaje HTML5.

## Literatura

1. Kim W. Tracy, Mobile application development experiences on Apple's iOS and Android OS, 2011
2. Android - Discover Android, <http://www.android.com/about/> (4.1.2013.)
3. Android Technical Information | Android Open Source, <http://source.android.com/about/philosophy.html> (4.1.2013.)
4. Android Tutorial - How to create a GUI application in Android, <http://www.skill-guru.com/blog/2011/01/18/currencyconverter-tutorial/> (3.1.2013.)
5. David Meyer, Mobile Application Development: Must Know Key Point Aspects, <http://www.webgranth.com/mobile-application-development-must-know-key-point-aspects> (3.1.2013.)
6. HTML5 introduction, [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp) (4.1.2013.)