

Uvod u Android programiranje

Doc. dr Vladimir Milićević



UVOD

Android operativni sistem predstavlja najrasprostranjeniju distribuciju Linux operativnog sistema.

Cilj jeste sticanje znanja i veština neophodnih za razumevanje koncepata i principa razvoja softvera za mobilne uređaje.

U fokusu biće Android operativni sistem i aplikacije koje se izvršavaju upravo na njemu. Android predstavlja najrašireniju distribuciju Linux operativnog sistema. Istorijat operativnog sistema počinje od 2005. godine kada je kompanija *Android INC.* razvila prvu verziju operativnog sistema.

Pored vlastitih razvojnih timova, kompanija *Google*, za unapređenje i razvoj operativnog sistema, brzo stiče saveznike u kompanijama koje su bile direktno pogođene ekspanzijom operativnog sistema IOS za *Apple* mobilne uređaje. U početku se radilo o kompanijama: *Motorola* i *SonyEricsson*, a danas su glavni nosioci primene *Android operativnog sistema kompanije*: Samsung, LG, Sony, kao i kineski giganti *Huawei* i *Xiaomi*.

Veoma brzo Android stiče veoma moćnu i široko rasprostranjenu programersku zajednicu okupljenu oko developer.android.com u okviru koje se:

- ❖ veoma brzo dobijaju odgovori na sva pitanja od interesa za korisnike i programere;
- ❖ obezbeđuju Android treninzi za aktuelne i buduće programere;
- ❖ Iznose iskustva i zapažanja u vezi sa prednostima i nedostacima aktuelnih verzija operativnog sistema i sl.

VERZIJE ANDROID OS

Sve verzije Android operativnog sistema nose zvaničan naziv dobijen po nekoj poslastici.

Sledećom tabelom prikazan je istorijat Google Android operativnog sistema sa funkcionalnostima koje su nove verzije uvele (izvor: <http://www.blic.rs/IT/514074/Sve-verzije-Android-OS-Od-G1-do-Lollipopa>).

Verzija	Datum objave	Karakteristike
1.0 G1	Februar 2008.	GPS, Bluetooth, multitasking, Integracija sa Google servisima, YouTube, Android Market (oko 35 aplikacija).
1.5 Cupcake	April 2009.	Uvedene kategorije u Android Marketu, univerzalna pretraga za sve, integrisana galerija.
1.6 Donut	Septembar 2009.	Virtualna tastatura, kamkorder opcija za kameru, stereo Bluetooth, vidžeti i folderi na početnom ekranu, copy/paste za pretragu, direktan upload na YouTube.
2.0 Eclair	Oktobar 2009.	Više korisničkih naloga, univerzalni email inbox, pretraga SMS poruka, podrška za digitalni zum, Bluetooth 2.1.
2.2 Froyo	Maj 2010.	Pet početnih ekrana, podrška za HotSpot, prebacivanje aplikacija na SD karticu, Adobe Flash 10.1.
2.3 Gingerbread	Decembar 2010.	Redizajniran copy/paste, podrška za NFC, podrška za prednju kameru.
3.0 Honeycomb	Februar 2011.	Podrška za 3D grafiku, privatni mod za internet pretragu, video četovanje za Google Talk, full screen za galeriju.

3.1 - 3.2.6 Honeycomb	May 2011. - Februar 2012.	Podrška za periferije poput miševa i tastatura, više opcija sa vidžetima, ispravljani brojni bagovi.
4.0 Ice Cream Sandwich	Oktobar 2011.	Lakše pravljenje foldera, redizajnirani Google servisi, podrška za 16 otvorenih tabova u Chromeu, mogućnost promene Launchera, Roboto font, gašenje aplikacija u pozadini, Android Beam, podrška za Wi-Fi Direkt.
4.1 Jelly Bean	Jul 2012.	Lakša interakcija sa notifikacijama, promena veličine vidžeta iz aplikacija, podrška za više jezika, pregled fotografija u filmstrip modu, Chrome kao glavni pregledač, Google Now servis.
4.2 - 4.3 Jelly Bean	Novembar 2012 - Oktobar 2013.	Vidžeti na zaključanom ekranu, brza podešavanja u notifikacijama, novi skrinsejver, isti interfejs za sve uređaje, podrška za 4K rezoluciju.
4.4 KitKat	Oktobar 2013.	Velika promena u interfejsu, providni status bar, mogućnost da se aplikaciju vide preko celog ekran, veličina operativnog sistema je značajno smanjena da mogu da ga koriste i slabiji telefoni.
5.0 Lollipop	Oktobar 2014.	Velika promena u dizajnu interfejsa, notifikacije se prikazuju na početnom ekranu, novi pregled pokrenutih aplikacija, mod za "gostujućeg korisnika", poboljšana štednja baterije.

Slika-1 Verzije Android operativnog sistema

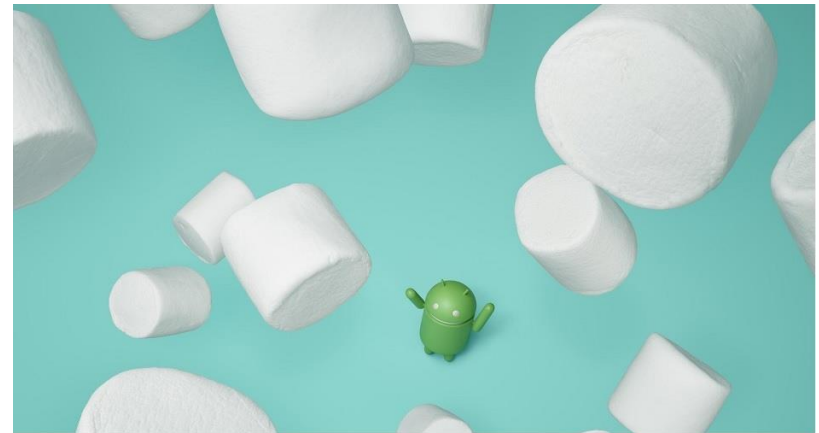
ANDROID 6.0

U 2015. predstavljen je Android 6.0 sa brojnim novinama.

U 2015. godini kompanija *Google* počela je sa distribucijom najnovije generacije operativnog sistema Android pod nazivom Android 6 – Marshmallow. Ovaj operativni sistem uvodi brojne novine:

- **Android NOW ON TAP:** mogućnost otvaranja dodatne opcije bilo koje aplikacije ili ekrana zadržavajući Android Home taster.
- **Dozvole za aplikacije:** Mogućnost odlučivanja o tome koja aplikacija će pristupati kojim opcijama telefona (kontakti, mikrofon, poruke, memorija, veb, kartica i sve drugo).
- **Podrška za senzore otiska prsta:** Sad gotovo svaki Android može da ima skener otiska prsta, ako to proizvođač poželi da stavi na svoj uređaj.
- **Doze Mode:** Nova opcija za uštedu potrošnje baterije.
- **Novi meni aplikacija:** Ikonice aplikacija u glavnom meniju se prikazuju sa poboljšanjima koje uključuju aplikacije koje se najčešće koriste, predlog Androida koje aplikacije da koristite i pretragu aplikacija.

U toku je ažuriranje premium uređaja ovom verzijom Android operativnog sistema.



Slika-2 Android 6.0

KARAKTERISTIKE ANDROID OS

Unificirani pristup razvoju aplikacija je jedna od glavnih prednosti Android operativnog sistema.

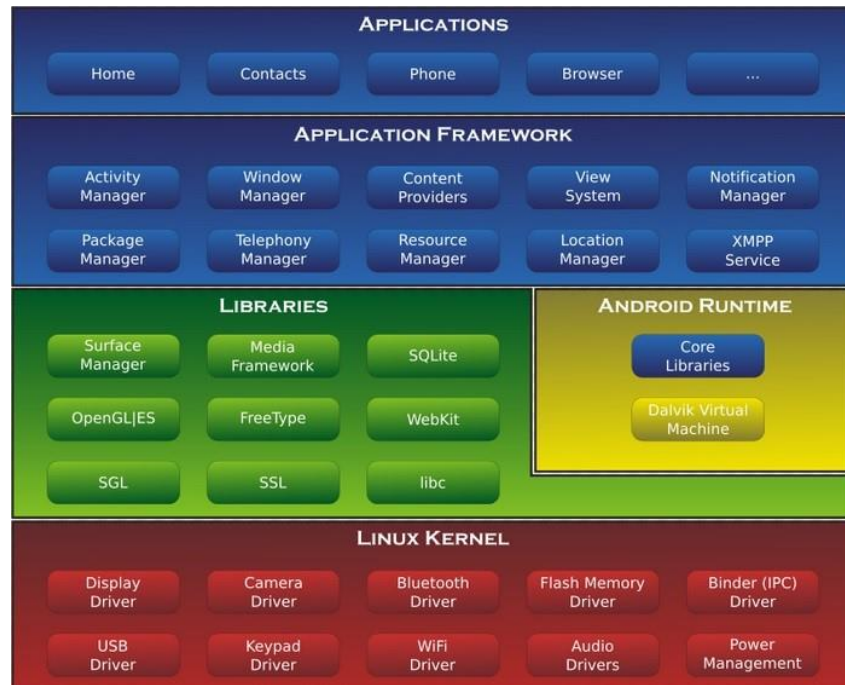
Osnovna prednost Android operativnog sistema jeste postojanje unificiranog pristupa razvoju aplikacija. Sva softverska rešenja se razvijaju pod Android platformom i izvršavaju na uređajima koje pokreće neka od verzija Android operativnog sistema. Ovaj operativni sistem pripada tzv. **open-source** konceptu, a to znači da su alati i tehnologije za prilagođavanje i unapređenje operativnog sistema, kao i aplikacija koje se izvršavaju pod Androidom, dostupni potpuno besplatno. Otuda, ne postoje posebno određene hardversko – softverske konfiguracije za Android ali moraju biti ispoštovani sledeći zahtevi:

- skladištenje podataka jednostavnom SQLite relacionom bazom podataka;
- pristupanje mobilnim mrežama putem svih poznatih mobilnih standarda (GSM, GPRS, EDGE, UMTS, LTE) kao i umrežavanje putem Bluetooth, NFC, WiFi, WiMax i drugih poznatih standarda;
- slanje tekstualnih i multimedijalnih poruka;
- web čitač;
- hardverska podrška za akcelerometar, GPS, kameru i sl.
- podrška za ekrane osetljive na dodir;
- multi-tasking;
- deljenje Internet konekcije itd.

ARHITEKTURA ANDROID OS

Arhitektura Adroida organizovana je po slojevima.

Android operativni sistem podeljen je na slojeve kao što je prikazano sledećom slikom (izvor: <http://www.plagosus.net>).



Slika-4 Slojevi Android OS

- Linux jezgro predstavlja osnovu operativnog sistema i u njemu su sadržani pokretački programi svih hardverskih komponenta integriranih u mobilni uređaj;
- Biblioteke čuvaju programske kodove svih modula neophodnih za funkcionisanje operativnog sistema;
- Android Runtime je istog nivoa kao i biblioteke i obezbeđuje arhive pomoću kojih JAVA programeri pišu Android aplikacije;
- Radni okvir aplikacija obezbeđuje različite funkcionalnosti Android operativnog sistema bez kojih kreirane aplikacije ne bi mogle da se izvršavaju;
- Nivo aplikacija obuhvata sve aplikacije koje su pre-instalirane na uređaju, preuzete sa Android Marketa ili drugih izvora ili samostalno kreirane i implementirane.

PREGLED NEOPHODNIH ALATA I TEHNOLOGIJA

Programski jezik JAVA je osnovna tehnologija koju koriste alati za razvoj Android aplikacija

Biće prikazano preuzimanje i podešavanje sledećih tehnologija i alata neophodnih za razvoj aplikacija za Android operativni sistem:

- JAVA JDK SE;
- Android SDK;
- Eclipse IDE;
- Android Development Tools (ADT).

Nakon instalacije navedenih tehnologija i alata biće prikazano kreiranje **Android virtuelnog uređaja** (AVD) kojim će biti omogućeno testiranje kreiranih Android aplikacija.

Sledećom slikom ilustrovane su tehnologije i alati koji će biti preuzeti i instalirani za razvoj Android aplikacija.



Slika-1 Tehnologije i alati za razvoj Android aplikacija

JAVA DEVELOPMENT KIT (JDK)

Pre bilo kakvog rada neophodno je prvo instalirati JDK paket.

JAVA Development Kit (JDK) predstavlja implementaciju JAVA platforme predstavljene od strane kompanije *Oracle* u formi paketa binarnih datoteka namenjenih programerima za razvoj JAVA softverskih rešenja na različitim hardversko – softverskim platformama. JDK obuhvata i JAVA virtuelnu mašinu (JVM) i sve prateće resurse kojima je omogućeno razvijanje i izvršavanje JAVA softvera.

Od 2007. godine JAVA se distribuira kroz GNU General Public Licence (GPL), a to znači da je omogućeno potpuno besplatno nabavljanje JAVA tehnologije i odgovarajućih razvojnih alata.

Od 1995. godine, kada je predstavljena prva verzija programskog jezika JAVA, objavljeno je osam generacija ovog programskog jezika zajedno sa tekućim ispravkama.

Sledećom tabelom prikazane su JAVA generacije sa vremenom objavljivanja.

JDK verzija	Datum objavljivanja
JDK Alfa i Beta	1995.
JDK 1.0	23. 01. 1996.
JDK 1.1	19. 02. 1997.
J2SE 1.2	08. 12. 1998.
J2SE 1.3	08. 05. 2000.
J2SE 1.4	06. 02. 2002.
J2SE 5.0	30. 09. 2004.
JAVA SE 6	11. 12. 2006.
JAVA SE 7	28. 07. 2011.
JAVA SE 8	18. 03. 2014.

Slika-2 Generacije JDK paketa

PREUZIMANJE JDK 7

JKD se potpuno besplatno preuzima sa Oracle sajta.

Budući da se JAVA 8 i dalje razvija i dorađuje, JAVA 7 i dalje predstavlja aktuelni standard za razvoj širokog spektra softverskih rešenja. U daljem radu, a sa ciljem formiranja unificiranog alata za razvoj mobilnih aplikacija, prvo će biti demonstrirana instalacija sedme generacije JDK paketa. Prvi korak jeste odlazak na stranicu www.oracle.com i izbor JAVA platforme koja će biti instalirana.



Slika-3 Izbor JAVA platforme za instalaciju

INSTALACIJA JDK 7

Neophodno je izabrati i instalirati verziju JDK koja odgovara verziji OS instaliranog na računaru.

Nakon izbora platforme, neophodno je da se prihvate uslovi licenciranja softvera i da se izabere verzija JDK u zavisnosti od operativnog sistema na kojem se vrši instalacija, a to je prikazano sledećom slikom.

Java SE Development Kit 7u25		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	80.38 MB	jdk-7u25-linux-i586.rpm
Linux x86	93.12 MB	jdk-7u25-linux-i586.tar.gz
Linux x64	81.46 MB	jdk-7u25-linux-x64.rpm
Linux x64	91.85 MB	jdk-7u25-linux-x64.tar.gz
Mac OS X x64	144.43 MB	jdk-7u25-macosx-x64.dmg
Solaris x86 (SVR4 package)	136.02 MB	jdk-7u25-solaris-i586.tar.Z
Solaris x86	92.22 MB	jdk-7u25-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.77 MB	jdk-7u25-solaris-x64.tar.Z
Solaris x64	15.09 MB	jdk-7u25-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	136.16 MB	jdk-7u25-solaris-sparc.tar.Z
Solaris SPARC	95.5 MB	jdk-7u25-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.05 MB	jdk-7u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.67 MB	jdk-7u25-solaris-sparcv9.tar.gz
Windows x86	89.09 MB	jdk-7u25-windows-i586.exe
Windows x64	90.66 MB	jdk-7u25-windows-x64.exe

Slika-4 Izbor JDK za preuzimanje

Izabrana je instalacija za 64 bitnu verziju operativnog sistema Windows i pokrenuta je instalacija kao što je prikazano sledećom slikom.

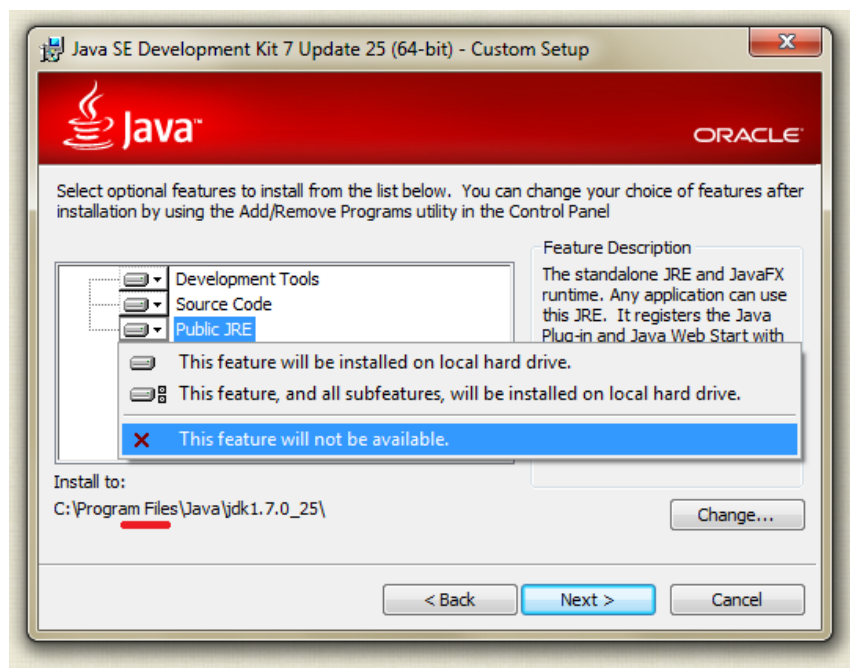


Slika-5 Instalacija JDK

INSTALACIJA JDK 7 - NASTAVAK

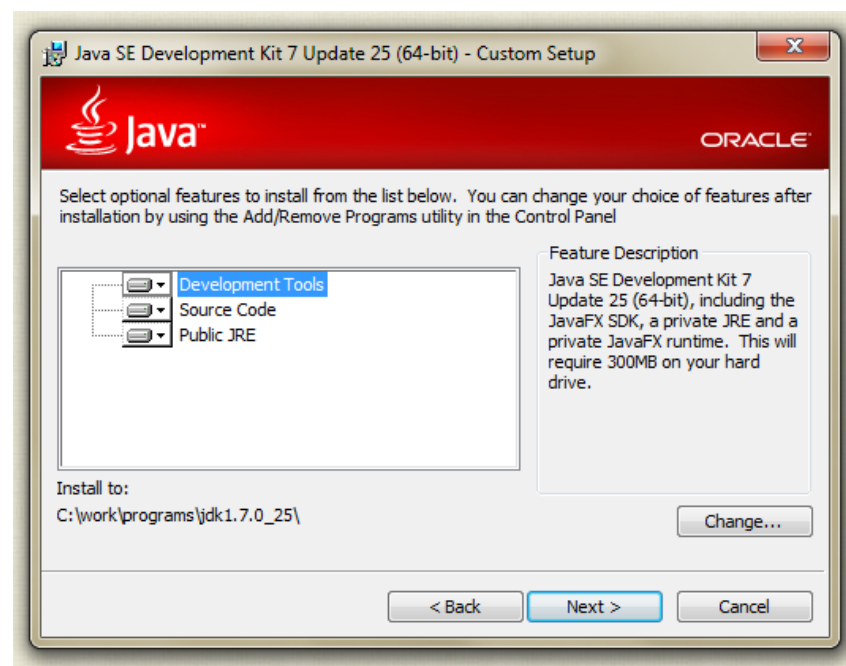
Neophodno je izabrati lokaciju i opcione alate za instalaciju.

Tokom procesa instalacije biće neophodno izvršiti određena podešavanja, poput izbora lokacije na kojoj će biti JDK snimljen, kao i izbora opcionih JDK alata.



Slika-6 Izbor lokacije na kojoj će JDK biti instaliran

Opcioni alati biraju se iz ponuđenih instalacionih menija, instalacija se nastavlja (sledeća slika) i nakon malo vremena dobija se poruka da je instalacija JAVA paketa uspešno obavljena.

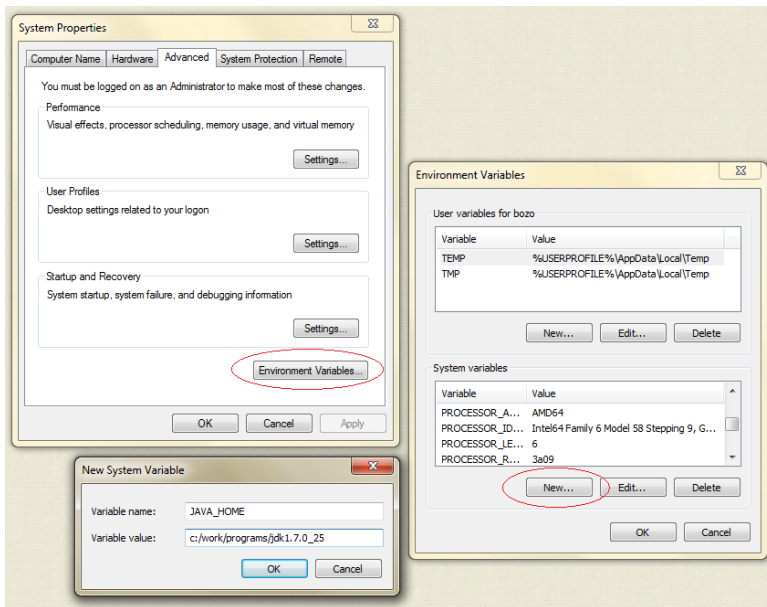


Slika-7 Izbor opcionih JAVA alata

PODEŠAVANJE JDK NA RAČUNARU

Neophodno je JAVU obezbediti dostupnom iz bilo kojeg direktorijuma na računaru.

Sledeći korak je veoma važan, **neophodno je JAVU obezbediti dostupnom iz bilo kojeg direktorijuma na računaru**. To se postiže podešavanjem sistemskih varijabli. Klikom na My Computer, zatim Properties, pa Advanced System Settings i Enviroment Variables, otvara se prozor u kojem će biti izvršena navedena podešavanja. Sada se definiše jedna sistemska promenljiva po imenu JAVA_HOME čija će vrednost biti putanja do JDK foldera. Nakon toga iz liste postojećih varijabli, bira se varijabla PATH u okviru koje je neophodno dodati sledeći string ;%JAVA_HOME%/bin kojim je određena putanja do JAVA prevodioca. Navedeno je prezentovano sledećom slikom.

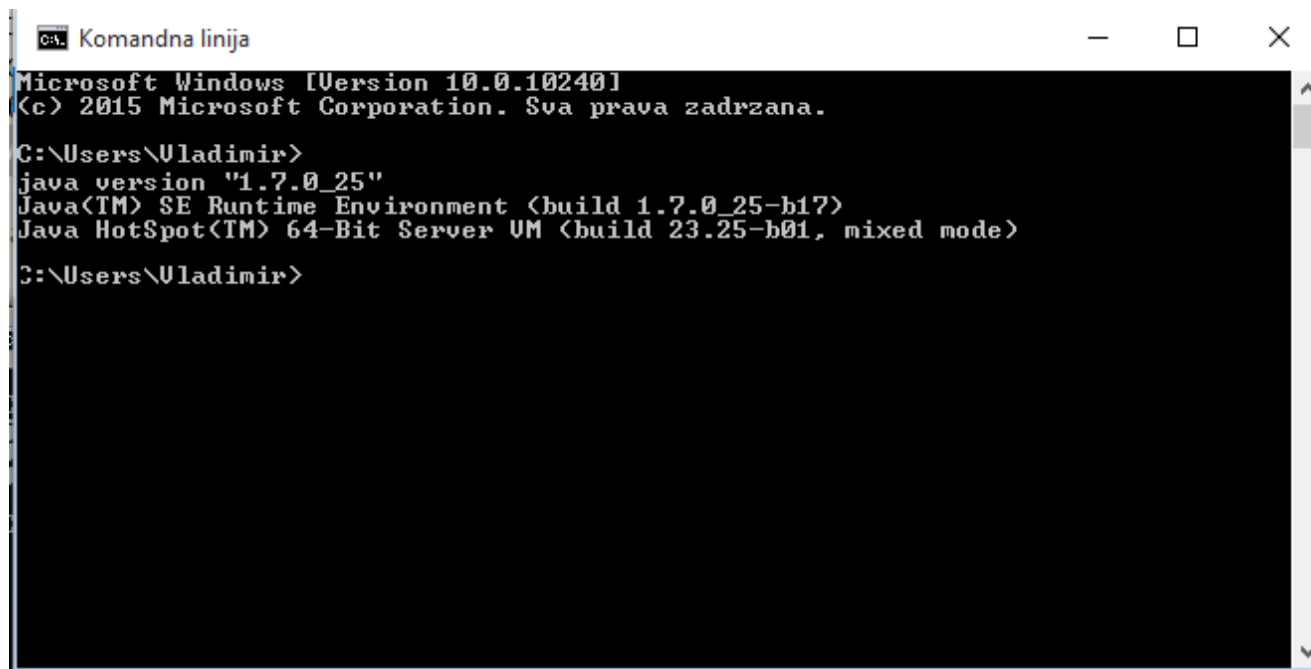


Slika-8 Podešavanje JDK sistemskih varijabli

PROVERA KONKRETNOSTI JDK INSTALACIJE.

U MS DOS-u se proverava konkretnost instalacije JAVA paketa.

Na samom kraju neophodno je proveriti da li je JAVA stvarno dostupna iz bilo kojeg foldera. Otvaranjem MS DOS Command Prompt i kucanjem instrukcije `java -version` vrši se provera. Ukoliko se na ekranu ispiše instalirana verzija, zadatak je uspešno obavljen (sledeća slika), u suprotnom biće ispisana komentar 'java' is not recognized as an internal or external command, operable program or batch file.



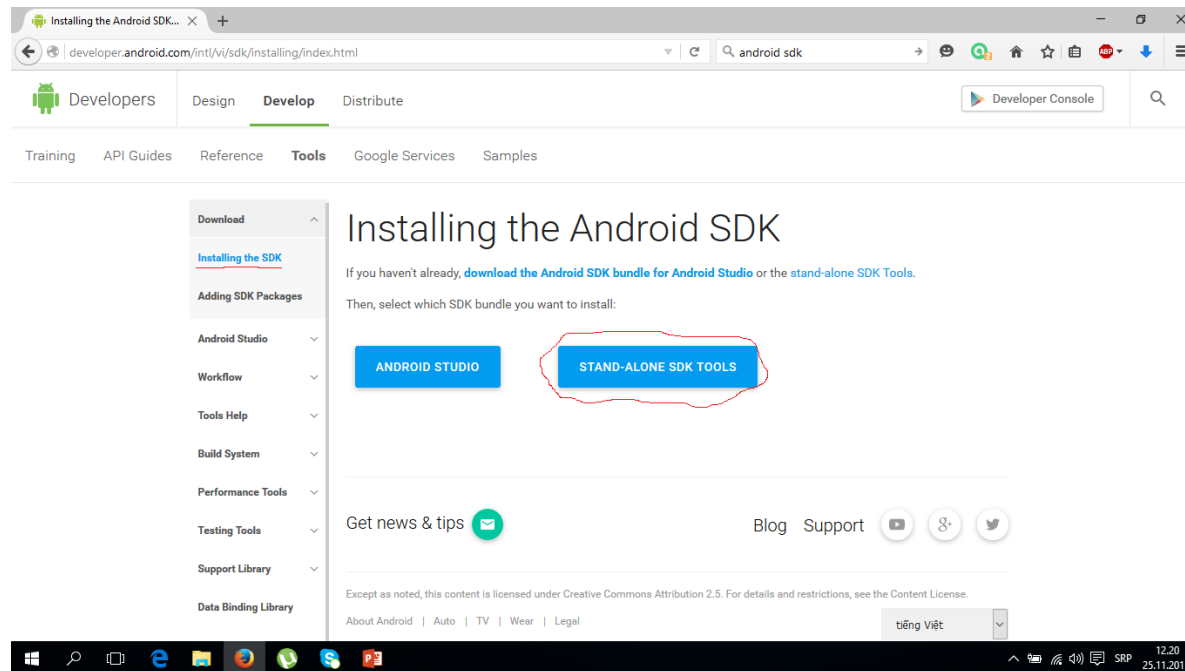
```
C:\Users\Uladimir> java version "1.7.0_25"
Java(TM) SE Runtime Environment (build 1.7.0_25-b17)
Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)
C:\Users\Uladimir>
```

Slika-9 Provera uspešnosti JDK instalacije

ANDROID SDK

Android SDK sadrži debager, emulator, dokumentaciju, primere koda i uputstva.

Pored JDK paketa, najvažniji softver koji je neophodno preuzeti je **Android SDK**. U okviru ovog paketa nalaze se debager, emulator, dokumentacija, primeri koda i uputstva. Odlaskom na web stranicu <http://developer.android.com/intl/vi/sdk/installing/index.html> i izborom opcije *STAND-ALONE SDK TOOLS*, iz menija, *Instaling The SDK*, vrši se preuzimanje instalacionog paketa za Android SDK.

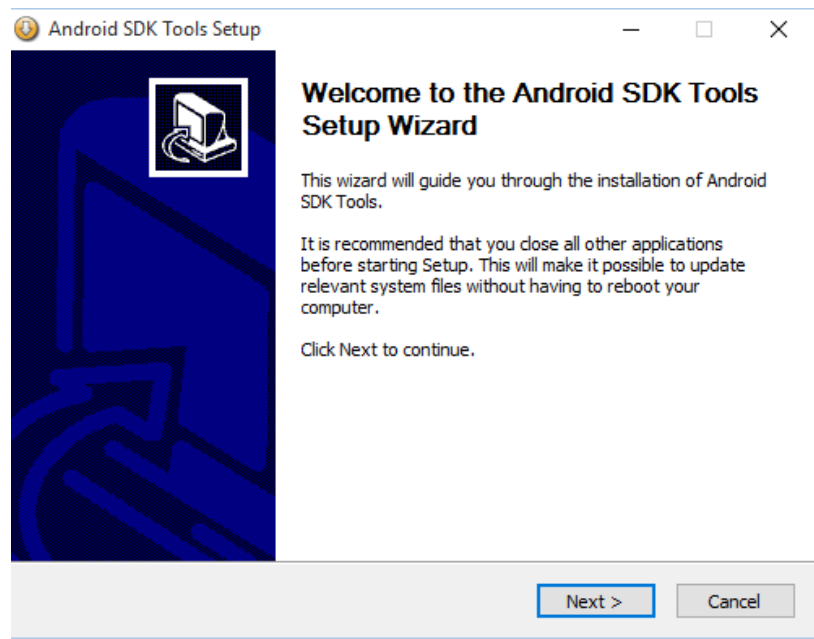


Slika-10 Preuzimanje Android SDK

ANDROID SDK - INSTALACIJA

Android SDK može biti preuzet na dva načina: kao .zip paket ili spakovan u Windows Installer.

Sledeće aktivnosti podrazumevaju pokretanje instalacije, izbor lokacije na kojoj će paket Android SDK biti instaliran, kao i obavljanje konačnih podešavanja da bi sve funkcionisalo kako je očekivano. Android SDK može biti preuzet na dva načina: kao .zip paket ili spakovan u Windows Installer. U prvom slučaju neophodno je raspakovati arhivu na željenoj lokaciji i izvršiti podešavanja sistemskih varijabli, kao u slučaju sa JAVA JDK. Ako se preuzme Android SDK sa Windows Installerom ova podešavanja će biti izvršena automatski (sledeća slika).

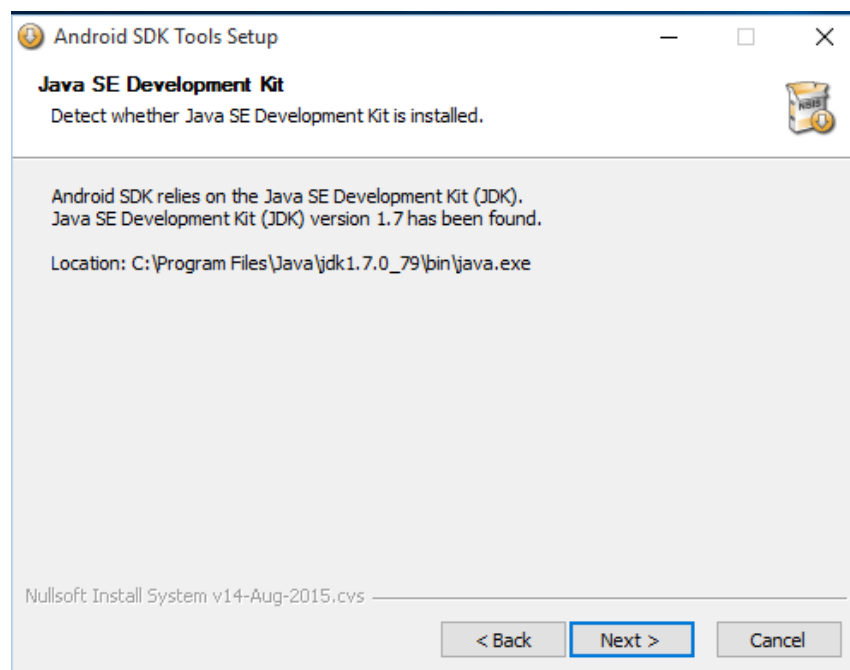


Slika-11 Instalacija Android SDK

ANDROID SDK – IZBOR JDK I INSTALACIONOG FOLDERA

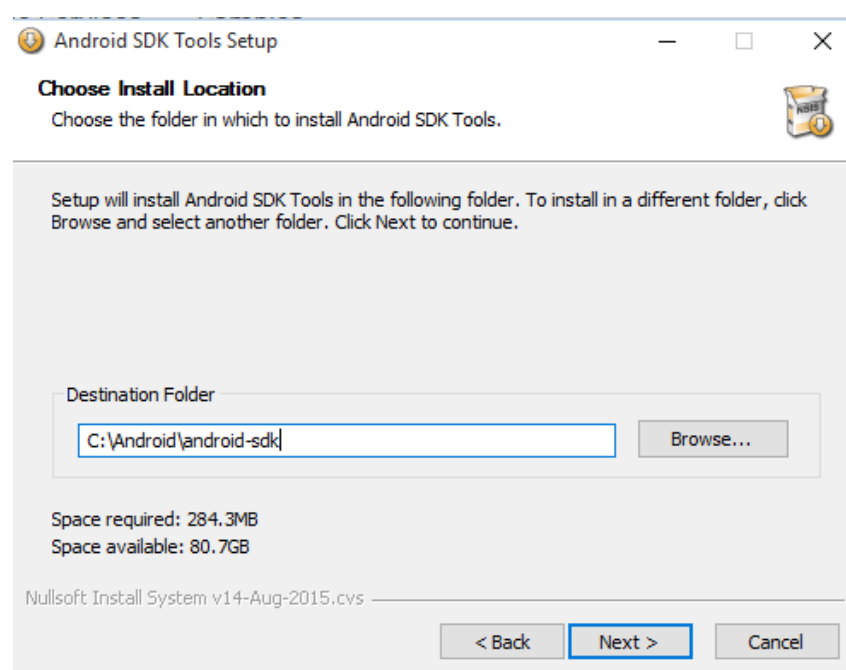
Android SDK se vezuje za instalirani JAVA JDK paket.

U sledećem koraku instalacija prepoznaje instaliranu verziju JAVA JDK za koju vezuje instalaciju Android SDK. Klikom na Next instalacija se nastavlja.



Slika-12 Povezivanje JAVA JDK i Android SDK

Dalje, neophodno je izabrati lokaciju na hard disku gde će biti instaliran Android SDK paket. Izborom lokacije i SDK opcionih alata kompletira se instalacija.



Slika-13 Izbor lokacije za instalaciju Android SDK

KONFIGURISANJE ANDROID SDK MENADŽERA

Android SDK menadžer upravlja različitim verzijama Android SDK-a instaliranim na računaru.

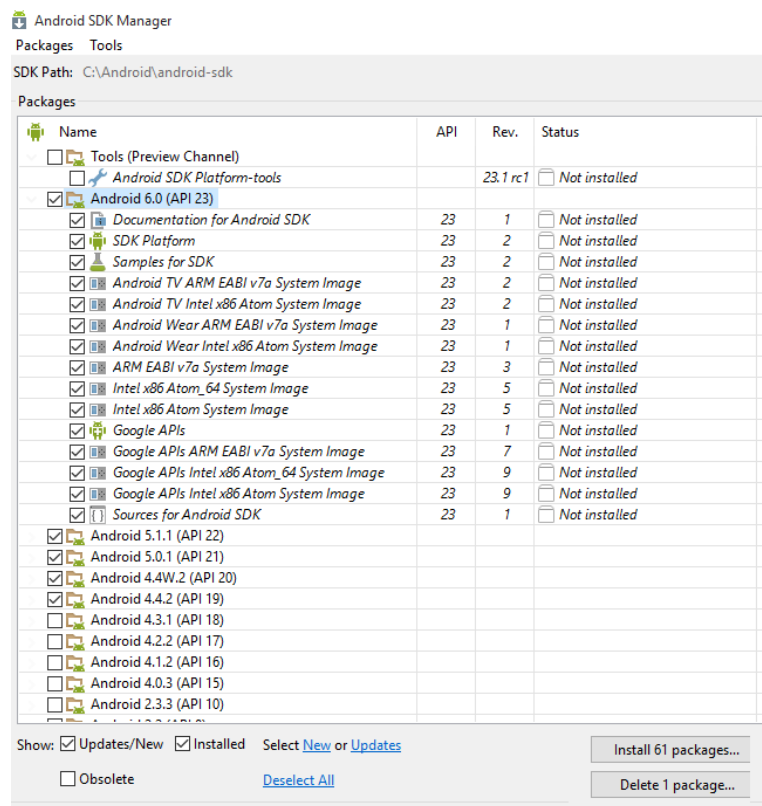
Pokretanjem instaliranog Android SDK menadžera prikazuje se lista stavki koje su trenutno instalirane na računaru, kao i onih koje su dostupne za naknadnu instalaciju. Izborom iz liste moguće je instalirati dodatne alate, dokumentaciju i platforme koji će biti korišćeni u konkretnim softverskim projektima. Ovde je bitno napomenuti da je svaka verzija Android operativnog sistema određena vlastitim API (Application Programming Interface) nivoom: Android 6 je API nivoa 23, Android 5.1.1 je API nivoa 22, Android 4.4.2 je API nivoa 19 itd. Za svaki API nivo postoje dve platforme:

- SDK platforma;
- Google API interfejs.

Google API interfejs proširuje SDK funkcionalnostima koji omogućavaju rad sad sa Google servisima, poput Google Maps biblioteka i sl.

Sledećom slikom prikazan je izbor za instalaciju iz Android SDK menadžera. Pribiranjem uslova licenciranja pokreće se instalacija. Po završetku instalacije biće neophodno restartovati ADB (Android Debug Bridge) što će biti i traženo

Pribiranjem završava se instalacija izabranih alata.



Slika-14 Izbor alata iz Android SDK menadžera

ECLIPSE IDE

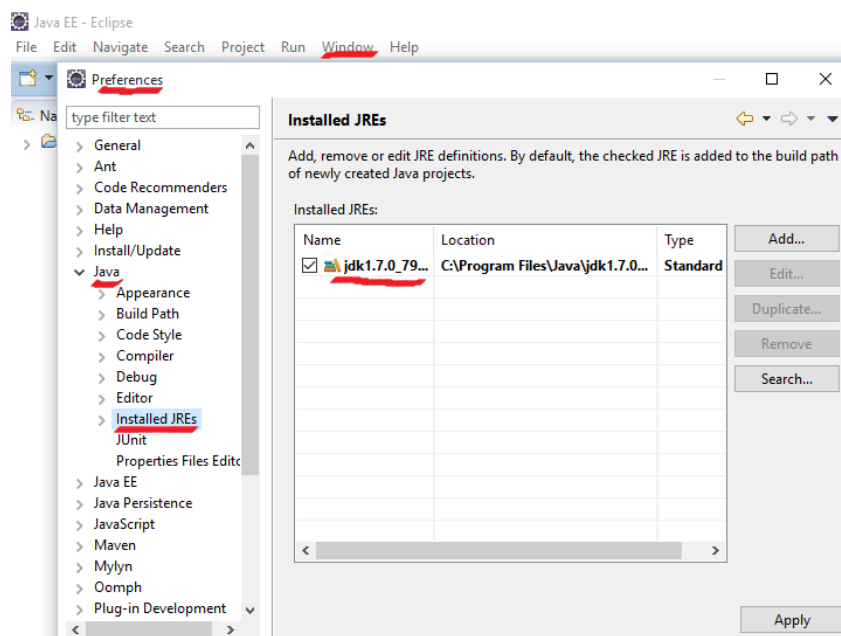
Eclipse IDE je nazastupljenije razvojno okruženje u industriji softvera.

U daljem radu neophodno je obezbediti razvojno okruženje koje će omogućiti primenu svih navedenih tehnologija i alata. Za razvoj Android aplikacija najčešće se koristi Eclipse IDE (eng. Integrated Development Environment). Eclipse je mnogo više od JAVA IDE – radi se o platformi za razvoj i pokretanje radnih okruženja za različite namene koje nisu ograničene samo na programiranje (mada takvih ima najviše). Najkvalitetnija osobina Eclipse-a jeste proširivost koja se ogleda u preuzimanju različitih dodataka (eng. plug-in) kojima kvalitet razvoja softvera dobija novu dimenziju. Pored podrške JAVA programskom jeziku, Eclipse može da se koristi kao razvojna podrška za druge programske jezike poput: C, C++, Ada, Python itd.

Za razvoj Android aplikacija neophodno je da se sa lokacije eclipse.org/downloads/ preuzme arhiva koja odgovara operativnom sistemu instaliranom na računaru. Jednostavnim raspakivanjem i pozivanjem datoteke eclipse.exe, Eclipse IDE razvojno okruženje se pokreće. Nakon pokretanja biće još neophodno odabrati lokaciju radnog prostora (workspace) gde će biti čuvani svi projekti na kojima će se raditi.

Eclipse je takođe prošao kroz brojne evolucije i ovde će biti govora o najnovijoj verziji Eclipse IDE pod nazivom MARS.

Neophodno je, na kraju, povezati Eclipse sa instaliranim JAVA JDK paketom, navigacijom kroz menije Window – Preferences – Java – Installed JREs (sledeća slika):



Slika-15 Povezivanje Eclipse IDE sa JDK

ANDROID DEVELOPMENT TOOLS (ADT)

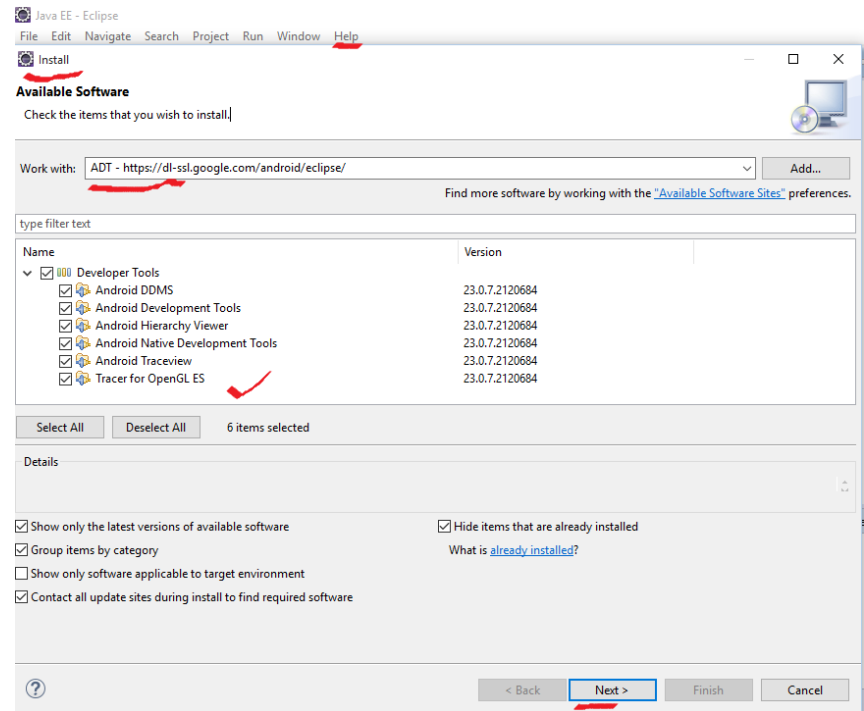
ADT je dodatak za Eclipse IDE za razvoj mobilnih aplikacija.

Da bi bilo omogućeno razvijanje Android mobilnih aplikacija neophodno je instaliranom Eclipse IDE razvojnom okruženju priključiti dodatak pod imenom **Android Development Tools**. Ovim proširenjem omogućen je razvoj mobilnih aplikacija ali i kvalitetnije identifikovanje i otklanjanje grešaka nastalih tokom procesa programiranja. Dodatak ADT omogućava da se kroz Eclipse IDE uradi neki od sledećih zadataka:

- Kreiranje novog Android projekta;
- Upotreba Android emulatora;
- Upravljanje greškama;
- Preslikavanje Android aplikacija u Android pakete (APK);
- Kreiranje digitalnih sertifikata APK paketa.

Instalacija ADT je jednostavna. U Eclipse IDE bira se meni Help, a zatim se bira opcija Install New Software. U ponuđenom okviru za dijalog neophodno je uneti sledeći string **<https://dl-ssl.google.com/android/eclipse/>** koji odgovara URL-u sa kojeg se poziva instalacija ADT. Klikom na dugme *Next* pokreće se instalacija i ADT se integriše u Eclipse IDE.

Navedeno je ilustrovano sledećom slikom.



Slika-16 Instalacija ADT za Eclipse IDE

KREIRANJE ANDROID VIRTUELNOG UREĐAJA

AVD je instanca emulatora koja omogućava modeliranje realnog Android uređaja.

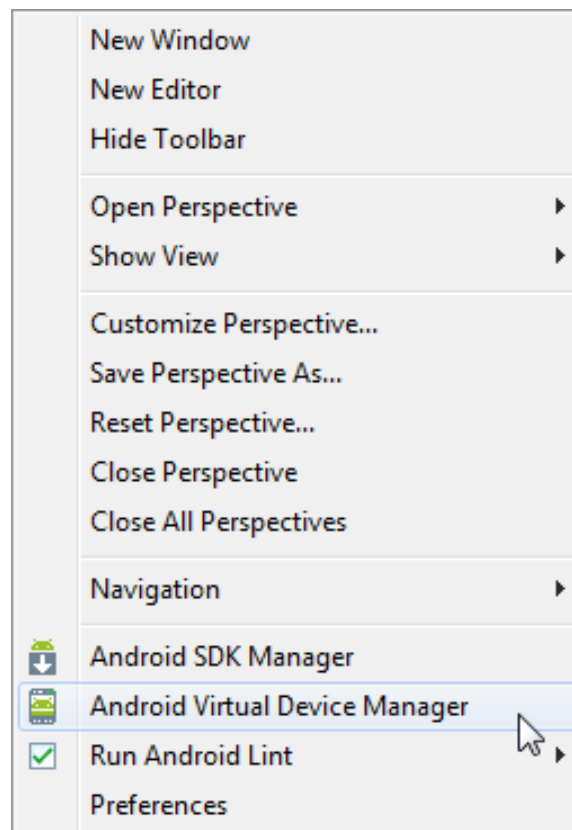
Pod AVD-om podrazumeva se instancu emulatora, koja modeluje neki stvarni Android uređaj. Svaki AVD sadrži parametre koji ga opisuju – hardverski profil uređaja, verzija Androida koja je instalirana u emulatoru, kao i veličina SD kartice koju emulator poseduje.

Moguće je napraviti veći broj AVD-ova za testiranje aplikacija na više različitih konfiguracija. Android aplikacije se mogu testirati direktno na android telefonu (ili tabletu), ali osnovna prednost korišćenja AVD je što je moguće raditi testove da bi se proverilo kako se aplikacija ponaša na uređajima sa različitim sposobnostima i na raznim verzijama Androida.

Android virtuelni uređaj se kreira u nekoliko koraka:

- U Eclipse IDE razvojnom okruženju bira se opcija **AVD Manager** iz menija Window.

Navedeno je prikazano sledećom slikom.

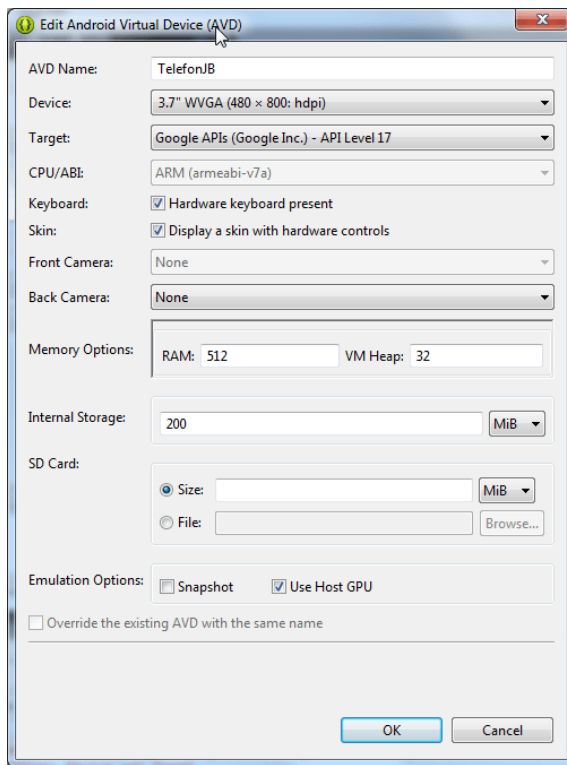


Slika-1 Otvaranje AVD menadžera

DEFINISANJE KONKRETNOG AVD

Klikom na New počinje kreiranje nove AVD instance.

U okviru Menadžera neophodno je kliknuti na New da bi bila kreirana nova AVD instancu (sledeća slika).



Slika-2 Kreiranje nove AVD instance

DEFINISANJE KONKRETNOG AVD

Za svaki AVD neophodno je podesiti odgovarajuće parametre.

Podešavaju se sledeći parametri:

AVD Name: Ime po kome se raspoznaje ova instancu (telefonJB)

Device: Tip uređaja koji se emulira.

Target: Ovde se bira verziju Androida koja će biti emulirana. Moguće je izabrati samo one verzije za koje je instaliran API u okviru SDK Menadžera. Ako pored verzije stoji Google API, znači da će u emulatoru biti dostupni i neki guglovi servisi (kao na primer Google Maps), a u suprotnom biće emuliran 'čist' Android.

Keyboard: Daje mogućnost korišćenja fizičke tastature u okviru emulatora. U suprotnom će za svako tekst polje iskakati virtuelna tastatura koja se koristi pomoću miša.

Skin: Obezbeđuje okvir sa standardnim android tasterima (Back, Home, Menu) kada se pokrene AVD.

Front Camera i Back Camera: Ako se izabere veb kamera oponaša kameru telefona.

Memory Options: Preporučljivo je da bude bar 512Mb RAM (deo ovoga će zauzeti sam Android sistem), a VM Heap će biti zadržan na vrednosti 32.

Internal Storage: Definiše vrednost za kapacitet internog skladišta memorije (za početak 200 MB će biti dovoljno).

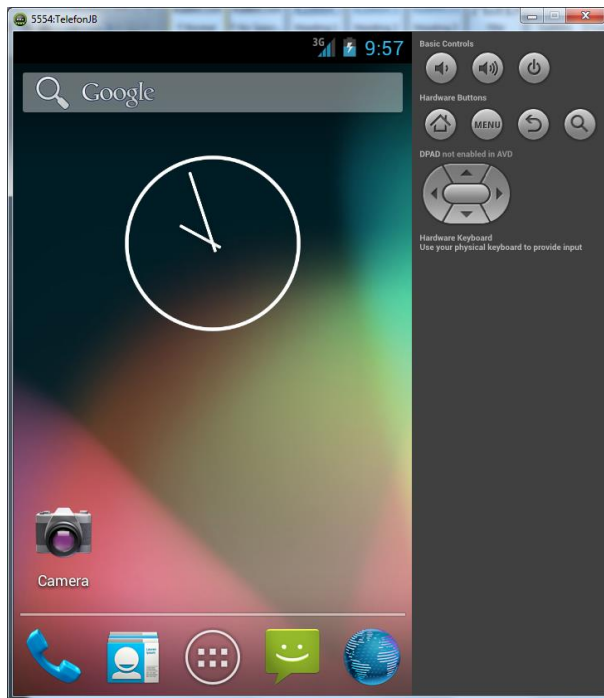
SD Card: Definiše vrednost za kapacitet eksternog skladišta memorije.

Emulation Options: Izбором Use Host GPU emulator će raditi brže.

AVD POKRETANJE

Kreiran Android emulator koristi se za testiranje aplikacija.

Klikom na OK biće kreiran Android Emulator, i u okviru AVD Menadžera, klikom na Start pa na Launch biće pokrenut (sledeća slika). Sada je sve spremno za kreiranje prve Android aplikacije.

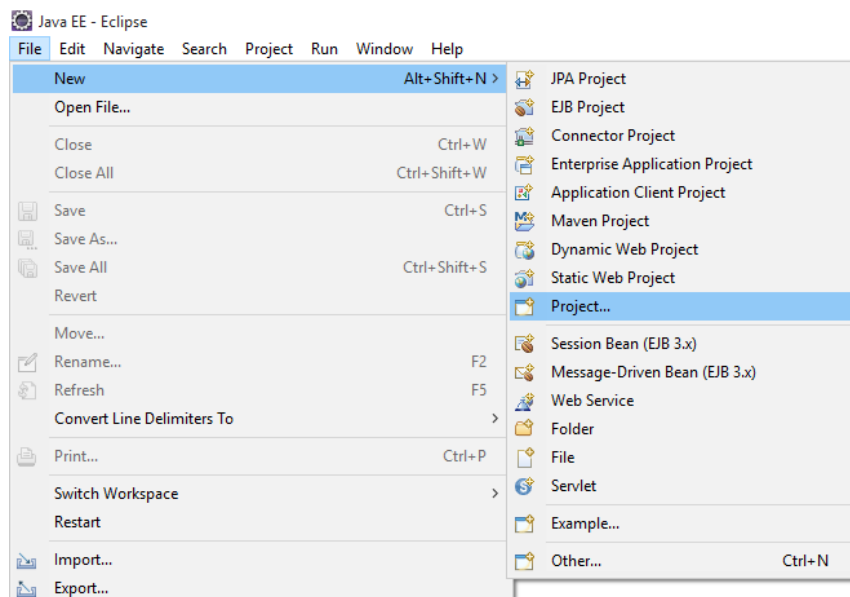


Slika-3 Kreirana AVD instanca

PRVA ANDROID APLIKACIJA

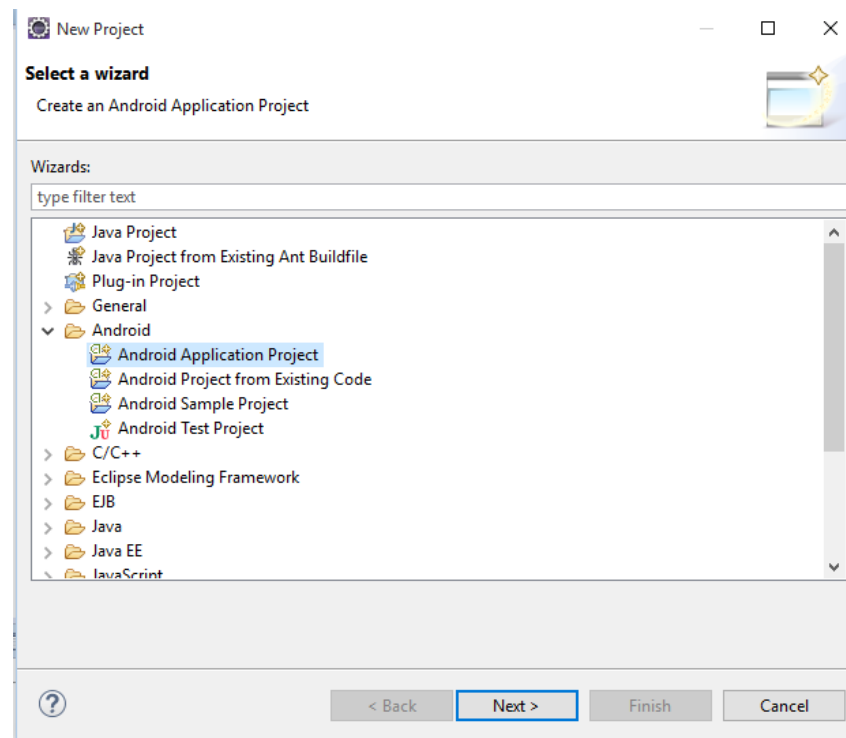
Android aplikacije imaju formu projekata.

Razvoj svake Android aplikacije započinje pokretanjem Eclipse IDE razvojnog pkrženja. Nakon toga iz menija *File*, bira se opcija *New*, za zatim *Project*.



Slika-4 Novi Android projekat

Nakon klika na opciju *Project* otvara se prozor u kojem se, u meniju *Android*, bira opcija *Android Project*.



Slika-5 Izbor opcije Android projekat

KREIRANJE ANDROID PROJEKTA

Svi fajlovi čuvaju se u folderu projekta.

U sledećem koraku neophodno je dodeliti naziv projektu, aplikaciji i paketu koji će čuvati dokumentaciju projekta. Takođe, biće izabran i API koji odgovara verziji Android operativnog sistema za koju se aplikacija razvija. U konkretnom slučaju ciljani API je verzija Android 4.4 KitKat, kompajliranje će biti obavljeno najnovijom verzijom za Android 6.0 (sledeća slika). Ključne informacije o aplikaciji su sledeće:

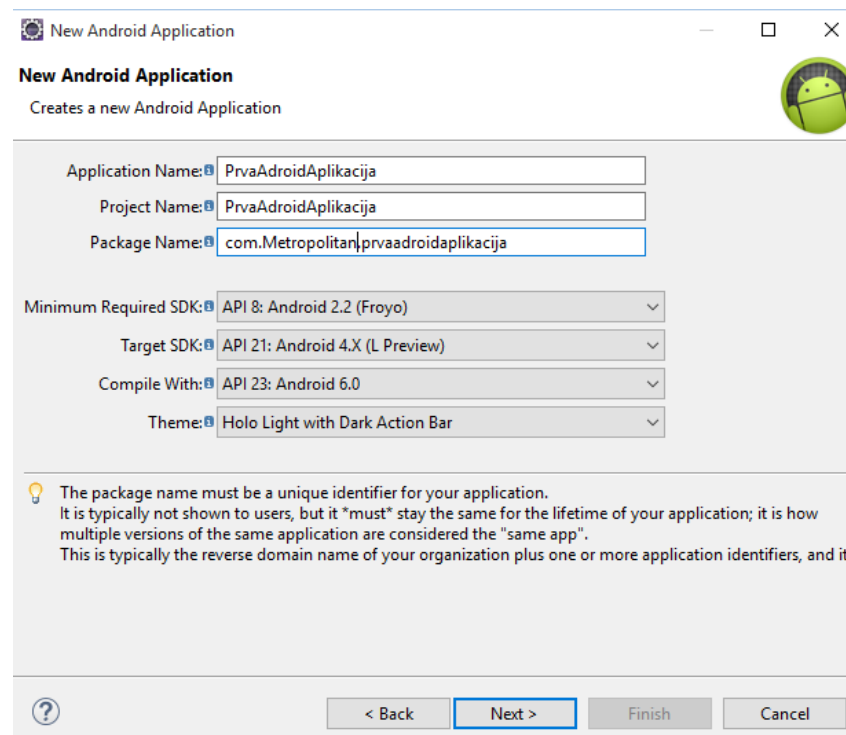
Naziv projekta: PrvaAdroidAplikacija;

Naziv aplikacije: PrvaAdroidAplikacija;

Naziv paketa: com.Metropolitan.prvaadroidaplikacija.

Klikom na Finish završavaju se inicijalna podešavanja i Eclipse IDE je spreman za razvoj prve Android aplikacije,

Navedeno je prikazano sledećom slikom.

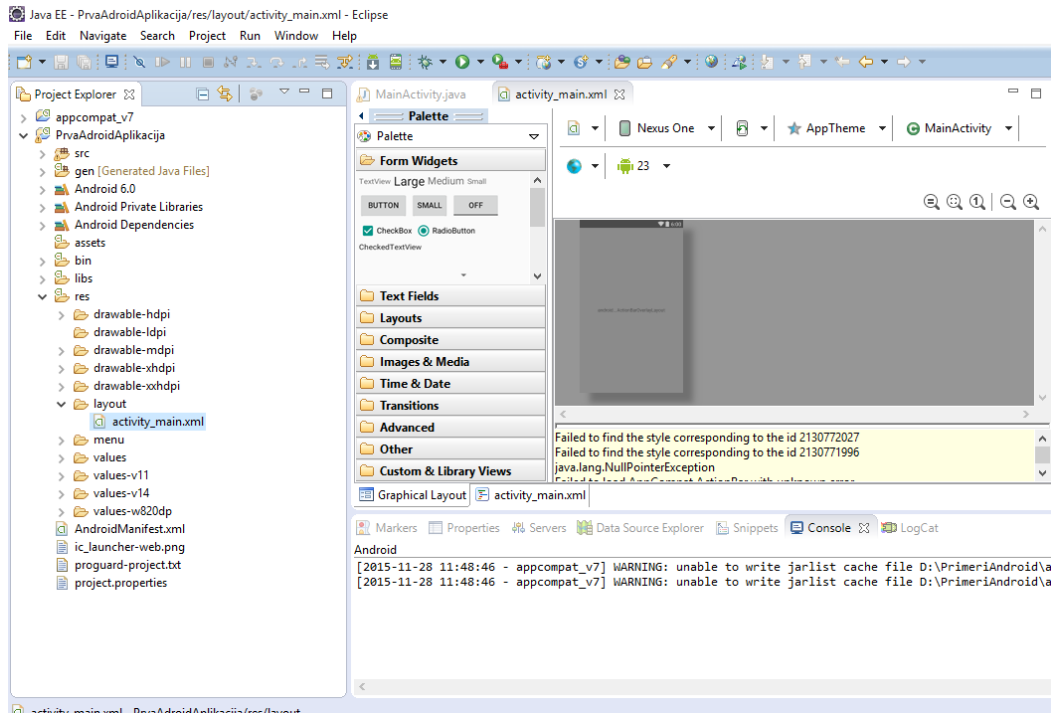


Slika-6 Application Info

ACTIVITY_MAIN.XML DATOTEKA

Korisnički interfejs aplikacije definisan je activity_main.xml datotekom.

Sada je pažnju neophodno usmeriti na panel pod nazivom Project Explorer U Eclipse integrisanom razvojnom okruženju. U **res/layout** folderu dvostrukim klikom bira se datoteka pod imenom **activity_main.xml**. Ova datoteka definiše korisnički interfejs aplikacije.



Slika-7 izbor datoteke activity_main.xml

ACTIVITY_MAIN.XML DATOTEKA - PRIMER

Inicijalni kod datoteka moguće je korigovati dodavanjem novih komponenata korisničkog interfejsa.

Inicijalni kod datoteke activity_main.xml sledi ispod:



```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:paddingBottom="@dimen/activity_vertical_margin"
7   android:paddingLeft="@dimen/activity_horizontal_margin"
8   android:paddingRight="@dimen/activity_horizontal_margin"
9   android:paddingTop="@dimen/activity_vertical_margin"
10  tools:context=".MainActivity" >
11
12   <TextView
13     android:layout_width="wrap_content"
14     android:layout_height="wrap_content"
15     android:text="@string/app_name" />
16
17 </LinearLayout>
```

Slika-8 activity_main.xml inicijalni kod

Sada je moguće ubaciti i neke vlastite korekcije. Neka to bude još malo teksta i jedno dugme. Ispod prvog xml taga <TextView.../>, a pre završnog taga </LinearLayout>, može se ubaciti sledeći kod:

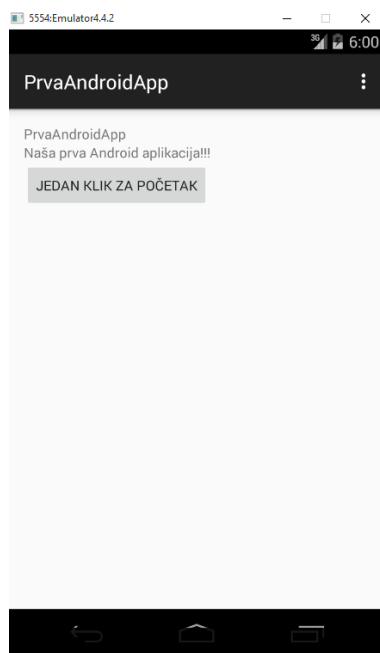
```
<TextView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Naša prva Android aplikacija!!!" />
<Button
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Jedan klik za početak" />
```

Slika-9 Dodavanje koda u xml datoteku

PREVOĐENJE I DEMONSTRACIJA PRIMERA

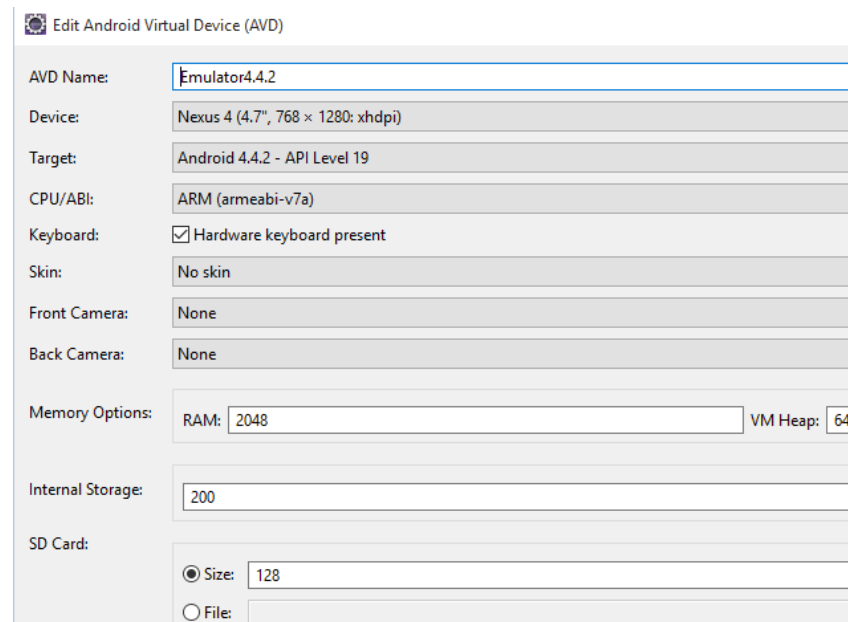
Nakon snimanja projekta, izborom Run As (Android Application) emulatorom se startuje kreirana aplikacija.

Sada je moguće pristupiti prevođenju i testiranju kreirane aplikacije. Prvo je neophodno snimiti projekat, npr sa ctrl+s, za zatim desnim klikom na naziv projekta selektovati Run As, pa Android Application.



Slika-10 Pokretanje prve aplikacije na emulatoru

Emulator koji je kreiran simulira rad uređaja na KitKat Androidu i definisan je na način prikazan sledećom slikom.



Slika-11 Izabrani emulator za testiranje aplikacije

ARHITEKTURA ANDROID APLIKACIJE

Datoteke koje čine hijerarhiju Android projekta prikazane su u Package Exploreru u Eclipse IDE.

Kreirana Android aplikacija će čitaocima ovog materijala, koji se prvi put susreću sa ovom problematikom, biti možda nejasna. Njen osnovni zadatak je, u ovom trenutku, da pokaže početne korake u kreiranju aplikacije – definisanje i podešavanje projekta, kao i hijerarhiju koju je moguće uočiti u Package Explorer-u integrisanog razvojnog okruženja. Budući da postoji više foldera i datoteka koji čine stablo Android projekta, u sledećem izlaganju će biti govora upravo o njihovom sadržaju i nameni.

Za Android projekat od posebnog značaja su sledeći folderi:

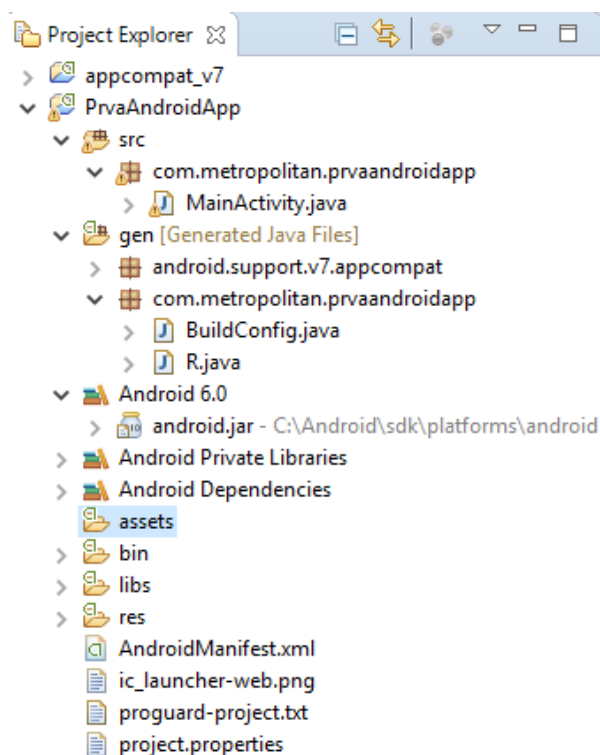
- **src** - sadrži paket sa izvornim datotekama ekstenzije .java. U navedenim datotekama sadržan je kod kojim je omogućeno funkcionisanje posmatrane aplikacije.;
- **gen** – sadrži **R.java** datoteku. Ovu datoteku je automatski kreirao JAVA prevodilac i ona upućuje na sve resurse u okviru projekta;
- **Android 6.0** - folder zavisi od generacije Android SDK kojim se vrši prevođenje, a u ovom slučaju je uključena trenutno najnovija verzija. U svakom slučaju, folder sadrži datoteku android.jar koja predstavlja JAVA arhivu sa svim klasama neophodnim za razvoj neke Android aplikacije;

- **assets** – folder sadrži sve neohodne alate koje će aplikacija koristiti npr: HTML, tekstualne datoteke, bazu podataka itd;
- **bin** – folder sadrži sve datoteke koje su nastale angažovanjem ADT u procesu prevođenja aplikacije. Među njima se nalazi i datoteka sa ekstenzijom **.apk (Android Package)** koje predstavlja binarni kod aplikacije i koja se instalira na mobilnom uređaju.
- **res** – folder u kojem su sadržani svi resursi koji se koriste u aplikaciji;
- **AndroidManifest.xml** – datoteka o kojoj će biti više govora u kasnijem izlaganju. Osnovna namena joj je definisanje privilegija neophodnih za aplikaciju, kao i filtera sadržaja, primaoca itd.

ORGANIZACIJA PACKAGE EXPLORERA

Svi folderi i datoteke aplikacije dostupni su u Package Exploreru,

Sledećom slikom je prikazan sadržaj projekta u okviru Package Explorera.



Slika-12 Hijerarhija Android projekta

U daljem izlaganju biće govora o nekim datotekama projekta koje su od posebnog značaja za kreiranje i funkcionisanje aplikacije:

1. activity_main.xml (u starijim verzijama main.xml) datoteka ima za namenu definisanje korisničkog interfejsa datoteke. U prikazanom primeru pokazano je dodavanje tekstualnih komponenata na radnu površinu mobilnog uređaja, kao i kontrole dugme (Button). Posebno bi trebalo napomenuti da je instrukcijom `@string/app_name` preuzet string definisan u datoteci strings.xml, a koji odgovara nazivu aplikacije. Upravo je i preporuka da se stringovi, koji se koriste u aplikaciji, čuvaju u navedenoj datoteci i da se na njih vrši referenciranje primenom identifikatora `@string/*`.
2. AndroidManifest.xml je veoma važna datoteka koja sadrži detaljne informacije o aplikaciji kao što su:
 - naziv paketa – u našem slučaju paket ima naziv `com.metropolitan.prvaandroidapp`;
 - identifikator verzije aplikacije;
 - minimalnu i ciljanu verziju Android OS kojima je aplikacija namenjena;

ORGANIZACIJA PACKAGE EXPLORERA - NASTAVAK

AndroidManifest.xml je datoteka u kojoj su definisane aktivnosti aplikacije.

- aplikacija koristi sliku ic_launcher.png iz drawable foldera;
- android:name=".MainActivity" instrukcijom ukazuje se na aktivnost u aplikaciji;
- Posebnu pažnju, u okviru aktivnosti, trebalo bi obratiti na xml tag <intent-filter> ... </intent-filter> u okviru kojeg se ukazuje na početnu tačku aplikacije (android.intent.action.MAIN), kao i na mogućnost pokretanja aplikacije pomoću launcher ikone(android.intent.category.LAUNCHER).

Sledećim xml kodom prikazan je sadržaj datoteke AndroidManifest.xml sa navedenim informacijama.



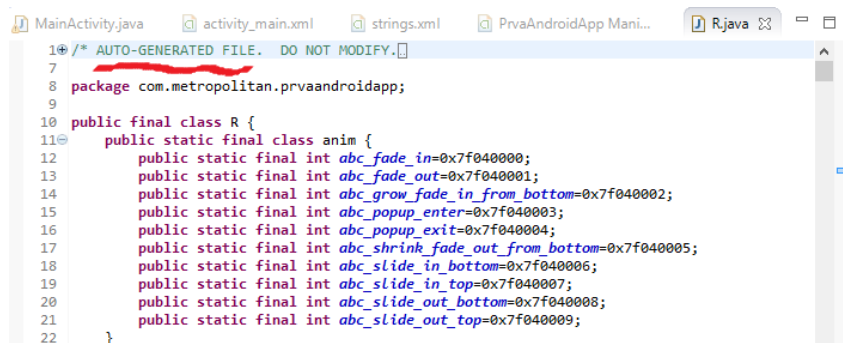
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.metropolitan.prvaandroidapp"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk
8         android:minSdkVersion="8"
9         android:targetSdkVersion="21" />
10
11     <application
12         android:allowBackup="true"
13         android:icon="@drawable/ic_launcher"
14         android:label="@string/app_name"
15         android:theme="@style/AppTheme" >
16         <activity
17             android:name=".MainActivity"
18             android:label="@string/app_name" >
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />
21
22                 <category android:name="android.intent.category.LAUNCHER" />
23             </intent-filter>
24         </activity>
25     </application>
26
27 </manifest>
```

Slika-13 Pogled u AndroidManifest.xml datoteku

R.JAVA DATOTEKA

R.java je automatski ažurirana od strane Eclipse IDE.

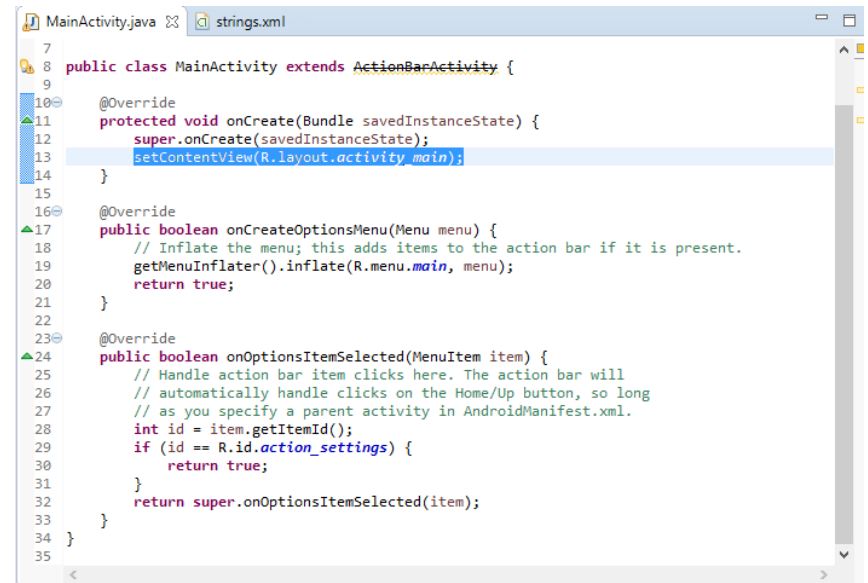
Tokom procesa dodavanja datoteka i foldera u Android projekat, datoteka R.java će automatski biti ažurirana od strane Eclipse IDE razvojnog okruženja i nije predviđeno da programer na bilo koji način modifikuje navedenu datoteku.



```
1+ /* AUTO-GENERATED FILE. DO NOT MODIFY.
7
8 package com.metropolitan.prvaandroidapp;
9
10 public final class R {
11     public static final class anim {
12         public static final int abc_fade_in=0x7f040000;
13         public static final int abc_fade_out=0x7f040001;
14         public static final int abc_grow_fade_in_from_bottom=0x7f040002;
15         public static final int abc_popup_enter=0x7f040003;
16         public static final int abc_popup_exit=0x7f040004;
17         public static final int abc_shrink_fade_out_from_bottom=0x7f040005;
18         public static final int abc_slide_in_bottom=0x7f040006;
19         public static final int abc_slide_in_top=0x7f040007;
20         public static final int abc_slide_out_bottom=0x7f040008;
21         public static final int abc_slide_out_top=0x7f040009;
22     }
```

Slika-14 Pogled u R.java datoteku

Konačno, datoteka MainActivity.java metodom setContentView() povezuje korisnički interfejs sa aktivnošću.



```
MainActivity.java strings.xml
7
8 public class MainActivity extends ActionBarActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14     }
15
16     @Override
17     public boolean onCreateOptionsMenu(Menu menu) {
18         // Inflate the menu; this adds items to the action bar if it is present.
19         getMenuInflater().inflate(R.menu.main, menu);
20         return true;
21     }
22
23     @Override
24     public boolean onOptionsItemSelected(MenuItem item) {
25         // Handle action bar item clicks here. The action bar will
26         // automatically handle clicks on the Home/Up button, so long
27         // as you specify a parent activity in AndroidManifest.xml.
28         int id = item.getItemId();
29         if (id == R.id.action_settings) {
30             return true;
31         }
32         return super.onOptionsItemSelected(item);
33     }
34 }
35
```

Slika-15 Pogled u MainActivity.java datoteku