

Programski Sistemi

Android programiranje



Šta je Android?

- Android je kompletna open source platforma projektovana za mobilne uređaje
- Nastala u kompaniji Google, sada u vlasništvu Open Handset Alliance (preko 80 kompanija, među kojima su Samsung, Motorola, HTC, Vodafone, ARM)
- Razdvaja hardver od softvera koji se na njemu izvršava, što omogućuje velikom broju različitih uređaja da izvršavaju istu aplikaciju
- Android programeri imaju na raspolaganju kompletno okruženje za brz i lak razvoj; potrebni su samo IDE i Android SDK

Istorijat Androida

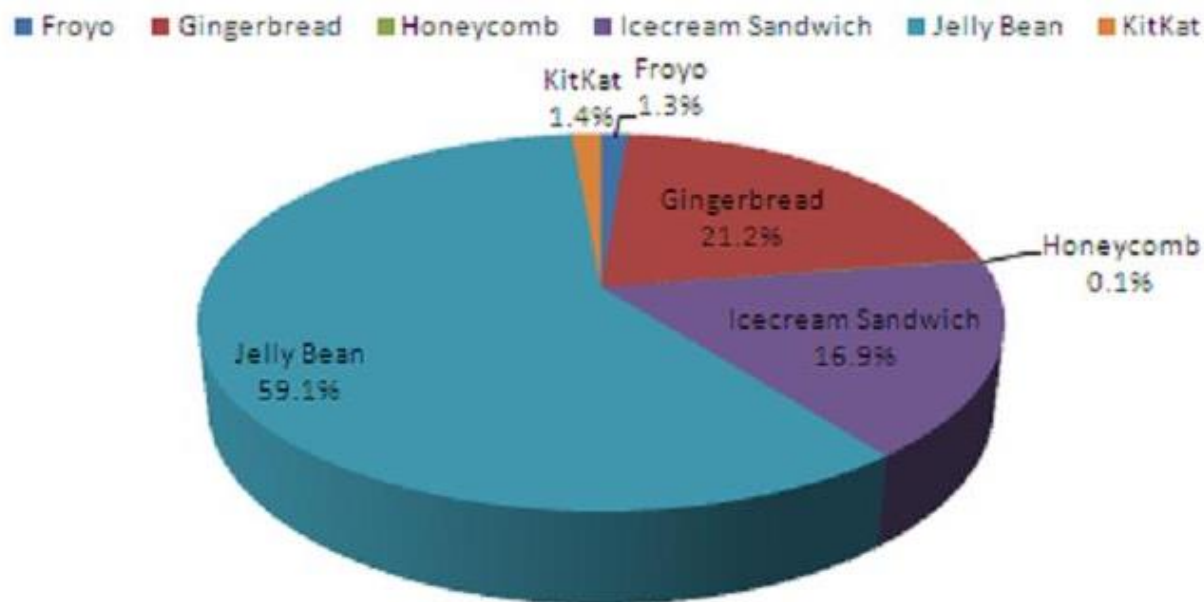
- 2005. Google kupuje Android, Inc.
- 2007 pojavljuje se Open Handset Alliance, a Android je zvanično open source
- 2008 pojavio se Android SDK 1.0. Uskoro sledi i HTC-ov telefon G1
- 2009 pojavljuju se novi uređaji i verzije operativnog sistema: Cupcake (1.5), Donut (1.6), and Eclair (2.0 and 2.1)
- 2010 objavljen je Froyo (Android 2.2) koji radi na preko 60 uređaja

Verzije Androida

Verzija Androida	API nivo	Kodno ime
1.0	1	
1.1	2	
1.5	3	Cupcake
1.6	4	Donut
2.0	5	Eclair
2.01	6	Eclair
2.1	7	Eclair
2.2	8	Froyo
2.3	9	Gingerbread
2.3.3	10	Gingerbread
3.0	11	Honeycomb
3.1	12	Honeycomb
3.2	13	Honeycomb
4.0, 4.0.1, 4.0.2	14	Icecream Sandwich
4.0.3, 4.0.4	15	Icecream Sandwich
4.1	16	Jellybean
4.2	17	Jellybean
4.3	18	JellyBean
4.4	19	KitKat

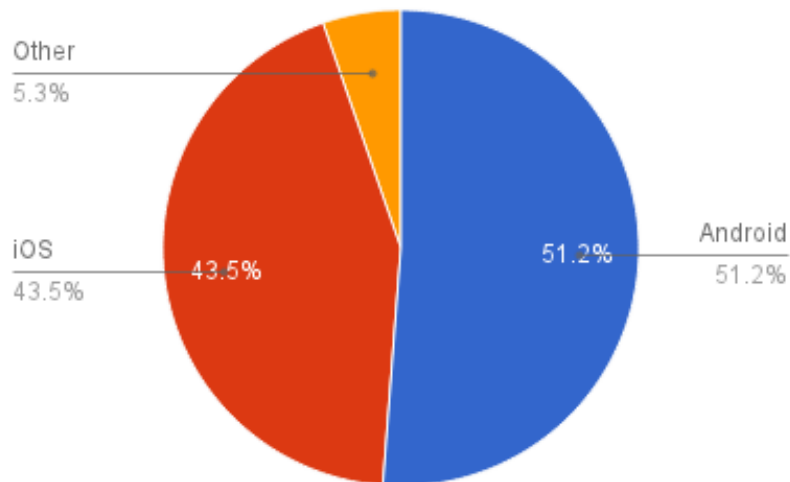
Verzije Androida

- Većina korisnika Androida u 2014. koristi Android 4.0 (Jelly Bean)
- To znači da je poželjno da aplikacija podržava minimalan API 16

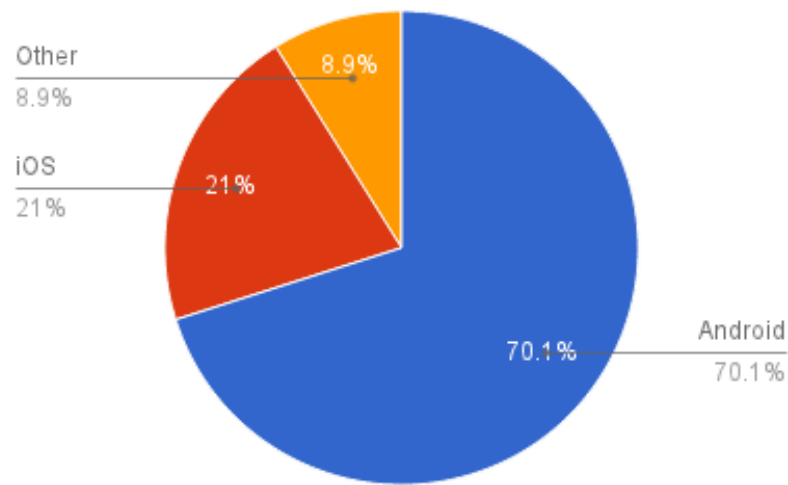


Android vs drugi

U.S. smartphone sales, 11/12-2/13

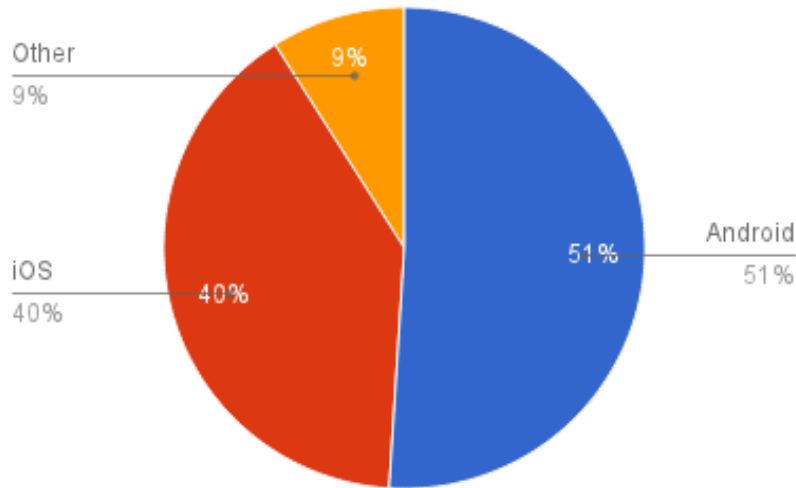


worldwide smartphone shipments, Q4 2012

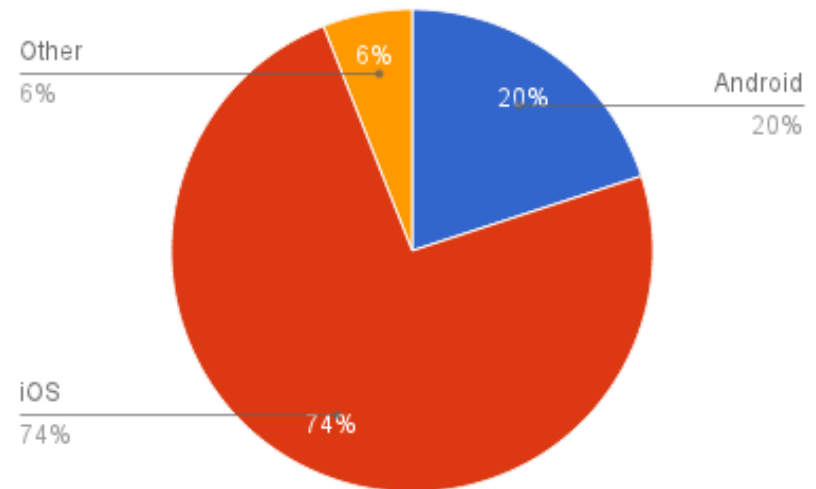


Android vs drugi

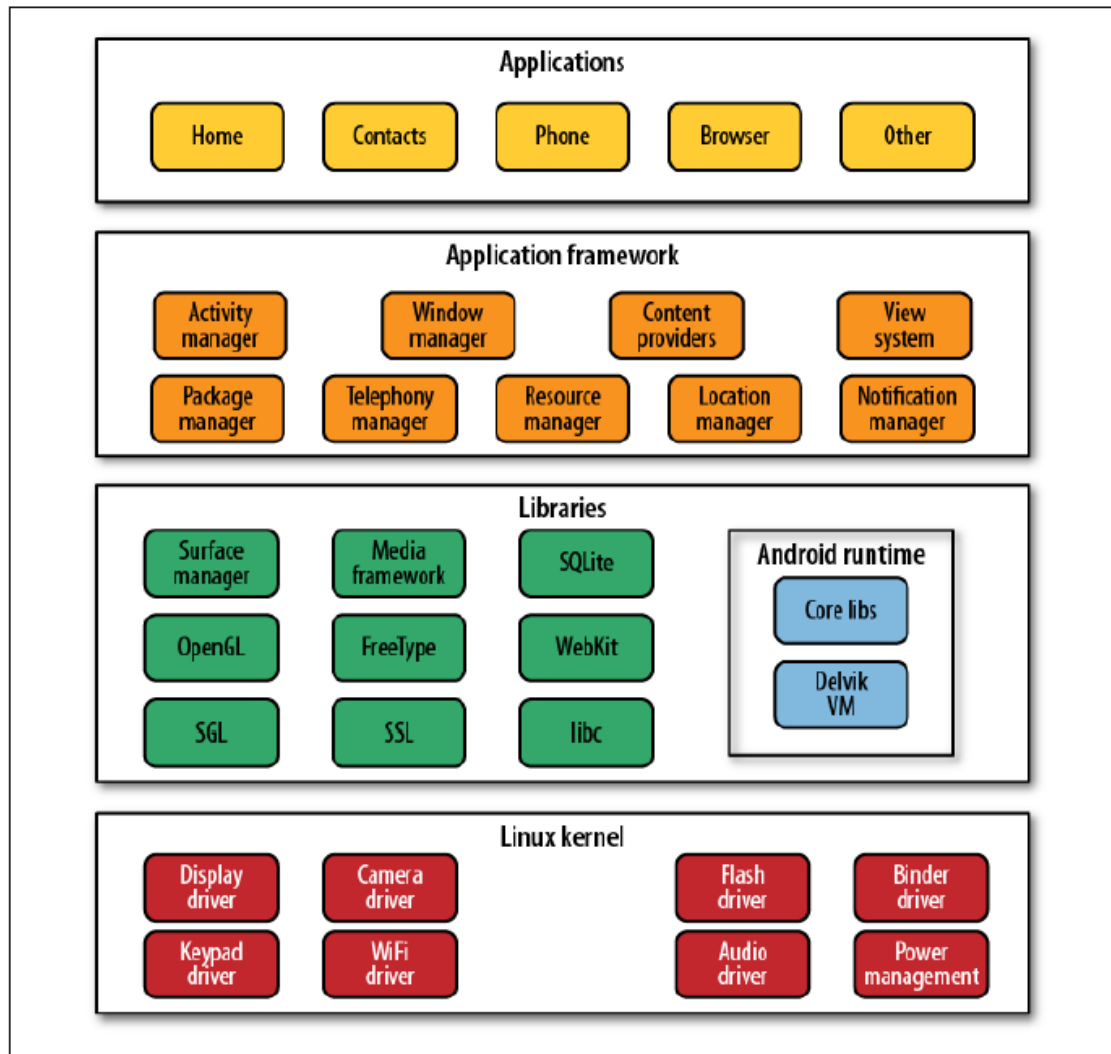
App Downloads, Q1 2013



Download Revenue, Q1 2013



Android stek



Programski jezik

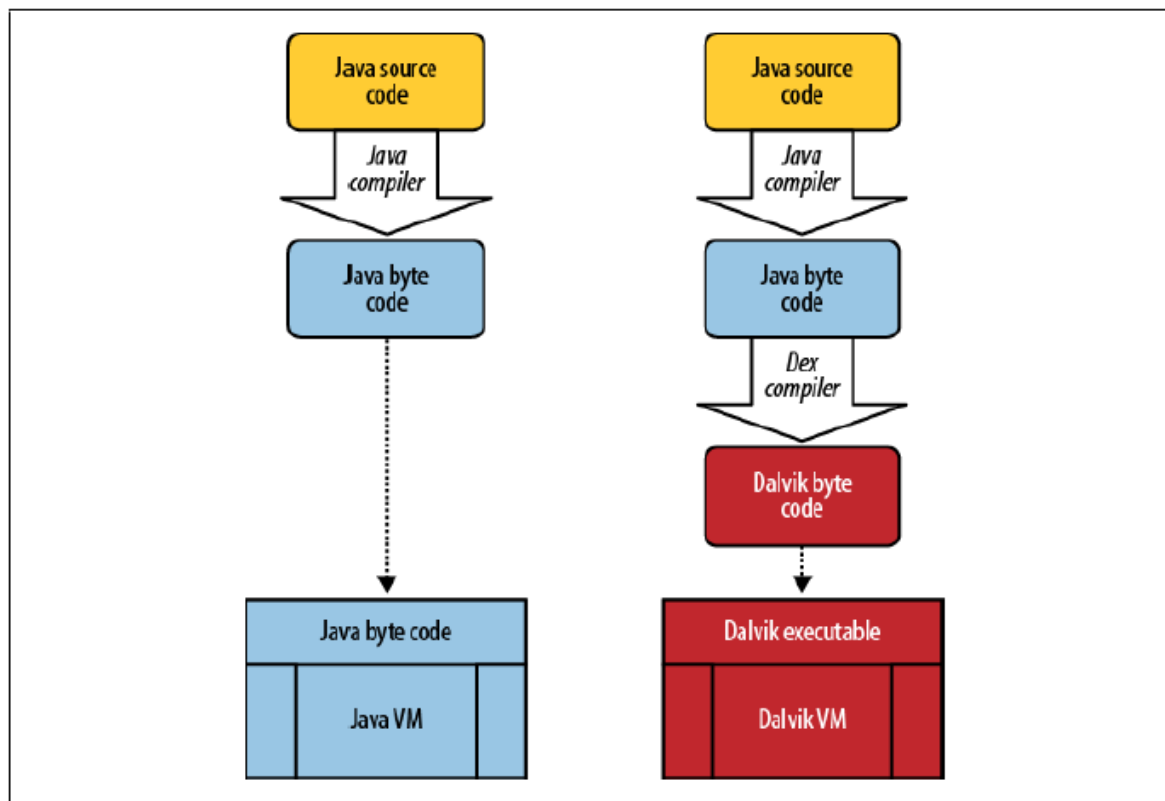
- Najveći deo Android aplikacija napisan je u Javi, i mi ćemo se isključivo baviti time
- Postoje i druge opcije:
 - Delovi koda mogu se pisati u jezicima C/C++ da bi se poboljšale performanse, ili ako se koristi OpenGL za 3D animacije
 - Delovi aplikacije mogu se pisati kao HTML, CSS i JavaScript, pa se taj materijal pakuje u Android aplikaciju koja se može distribuirati preko Play Store i sl.

Java i Android

- Skup Javinih biblioteka za Android najbliži je verziji Java SE
- Najveća razlika je u tome što su Javine biblioteke za korisnički interfejs (AWT i Swing) izostavljene i zamenjene bibliotekama specifičnim za Android
- Android standardnoj Javi dodaje i brojne nove funkcionalnosti (lokacije, senzori, WiFi, telefonija itd.).
 - To su manageri u trećem sloju Android steka

Virtuelna mašina Dalvik

- Java virtuelna mašina projektovana specijalno za Android



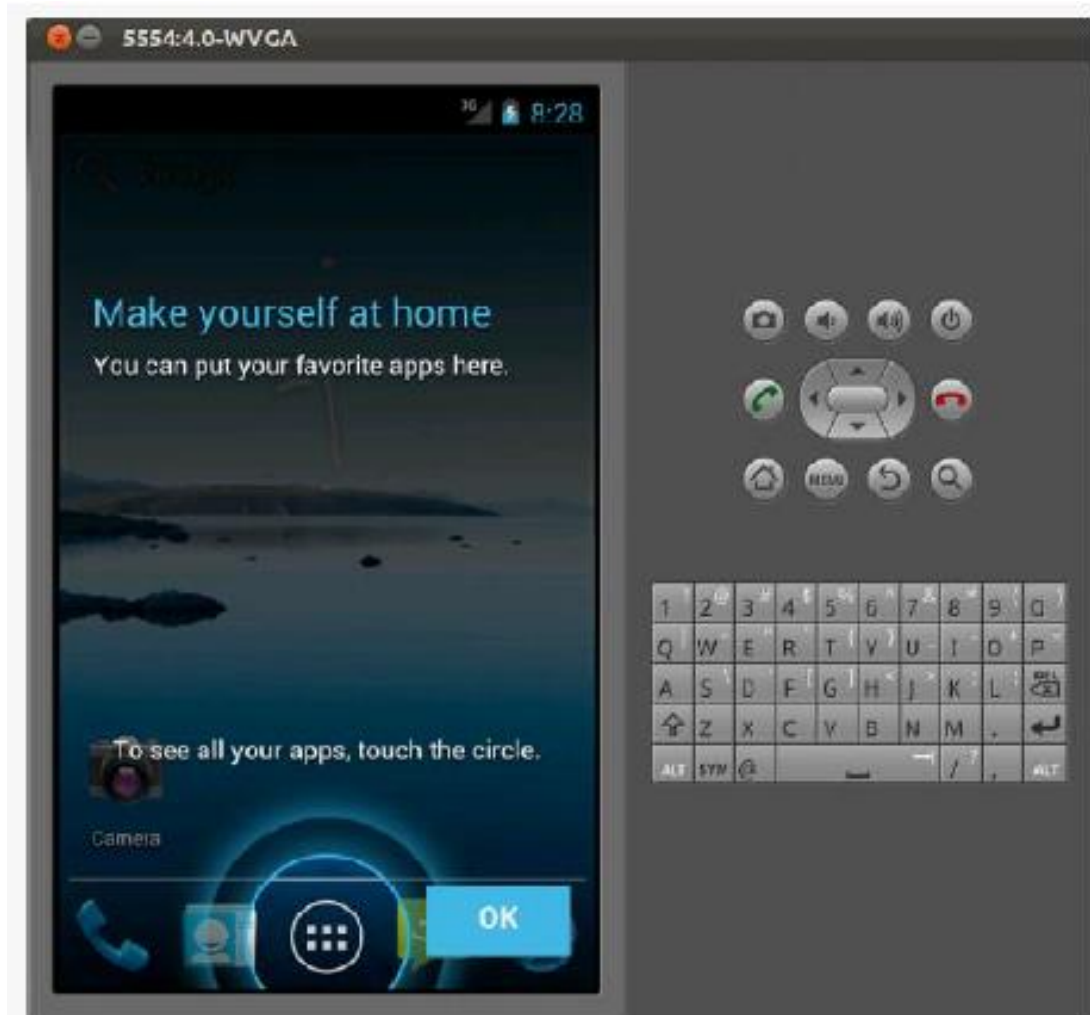
Dalvik

- Pošto VM Dalvik zahteva drugačiji bajtkod koji će interpretirati, to znači da ne podržava standardne Java class fajlove, već se oslanja na sopstveni format: Dalvik Executable (DEX)
- Android platforma za programiranje sadrži skup alata za postprocesiranje kompajliranih Java class fajlova u format DEX
- Za razliku od standardnih Java aplikacija koje sadrže više class fajlova, DEX spaja sve class fajlove u jedinstveni DEX fajl

Android emulator

- Služi za testiranje i debugovanje aplikacija
- To je implementacija Dalvik VM; pošto nije vezana za određeni hardver, odlična je za testiranje programa
- Podržava mrežu, podešavanje brzine Interneta, simulira slanje i primanje poziva i SMS poruka
- Ne implementira kameru, LED-ove, akcelerometar, USB povezivanje, snimanje zvuka, indikator baterije

Android emulator



Android aplikacija

- APK (Android Package) fajl koji sadrži tri glavne komponente:
 - Java kod kompajliran u Dalvik izvršni fajl
 - Resurse (sve što nije kod)
 - Opcione izvorne (C/C++) fajlove
- APK su ZIP arhive koje se delimično poklapaju sa JAR formatom
- Da bi mogle da se instaliraju na uređaju, Android aplikacije moraju da budu potpisane (ali se može koristiti self-signed sertifikat)

Android manifest

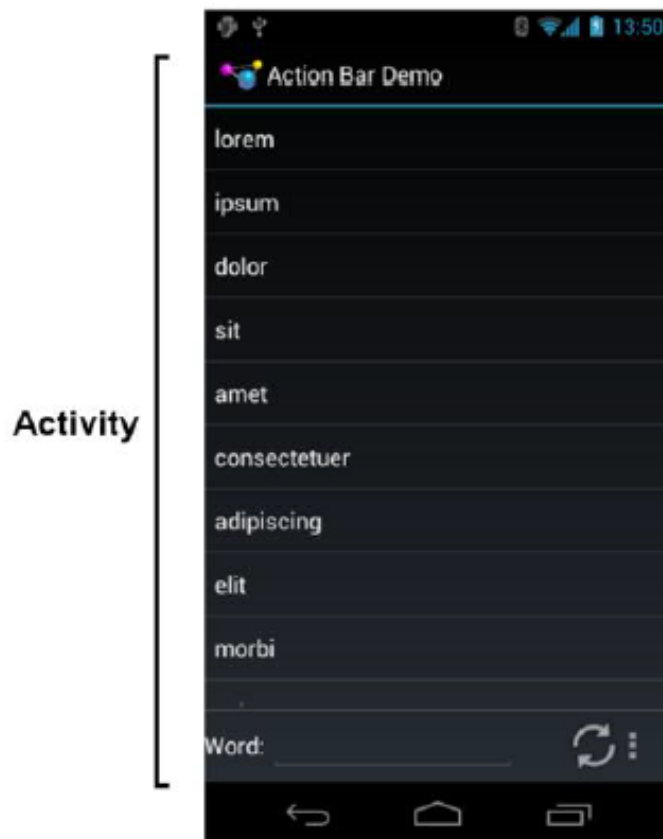
- Sve Android aplikacije u korenskom direktorijumu imaju fajl **AndroidManifest.xml**
- Ovaj fajl obezbeđuje sistemu sledeće informacije:
 - Ime aplikacije, ime paketa, broj verzije aplikacije
 - Minimalnu verziju API-ja neophodnu da bi se aplikacija izvršavala
 - Komponente aplikacije
 - Neophodne dozvole za samu aplikaciju, ali i za druge aplikacije koje žele da komuniciraju sa njenim komponentama
 - Biblioteke sa kojima se aplikacija mora povezati tokom izvršavanja da bi radila

Struktura Android aplikacije

- Android aplikacija u Javi se pravi tako što se klase izvode iz predefinisanih Android osnovnih klasa
- Pored toga, prave se metapodaci koji Android obaveštavaju o tim izvedenim klasama
- Pomenute izvedene klase u Android terminologiji zovu se **komponente** i ima ih četiri tipa:
 - Activities
 - Services
 - Content providers
 - Broadcast receivers

Activity

- Osnovni blok korisničkog interfejsa
- Android analog prozora ili dijaloga u desktop aplikaciji, odnosno strane u klasičnoj Web aplikaciji
- Aktivnost po pravilu zauzima veći deo ekrana, ostavljajući mesto samo za sat, indikator baterije i sl.



Activity

- Aplikacija obično ima više aktivnosti, a korisnik se kreće napred nazad između njih
- Slično kao što Web aplikacija ima više strana, i Android aplikacija ima više aktivnosti
- Kao što Web sajt ima “home page,” Android aplikacija ima “main” aktivnost, i to je ona koja se obično pojavi kada se pokrene aplikacija
- Kao što Web sajt mora da obezbedi neku vrstu navigacije između različitih strana, to mora da ima i Android aplikacija

Životni ciklus aktivnosti

- Pokretanje aktivnosti može da bude prilično skupo u pogledu resursa
 - Pokretanje novog Linux procesa
 - Alociranje memorije za sve UI objekte
 - Kreiranje objekata iz XML layouta
 - Podešavanje celog ekrana
- Pošto je pokretanje aktivnosti skupo, ne bi bilo pametno da se ona adbacira kada korisnik napusti ekran
- Zbog toga životnim vekom upravlja Activity Manager

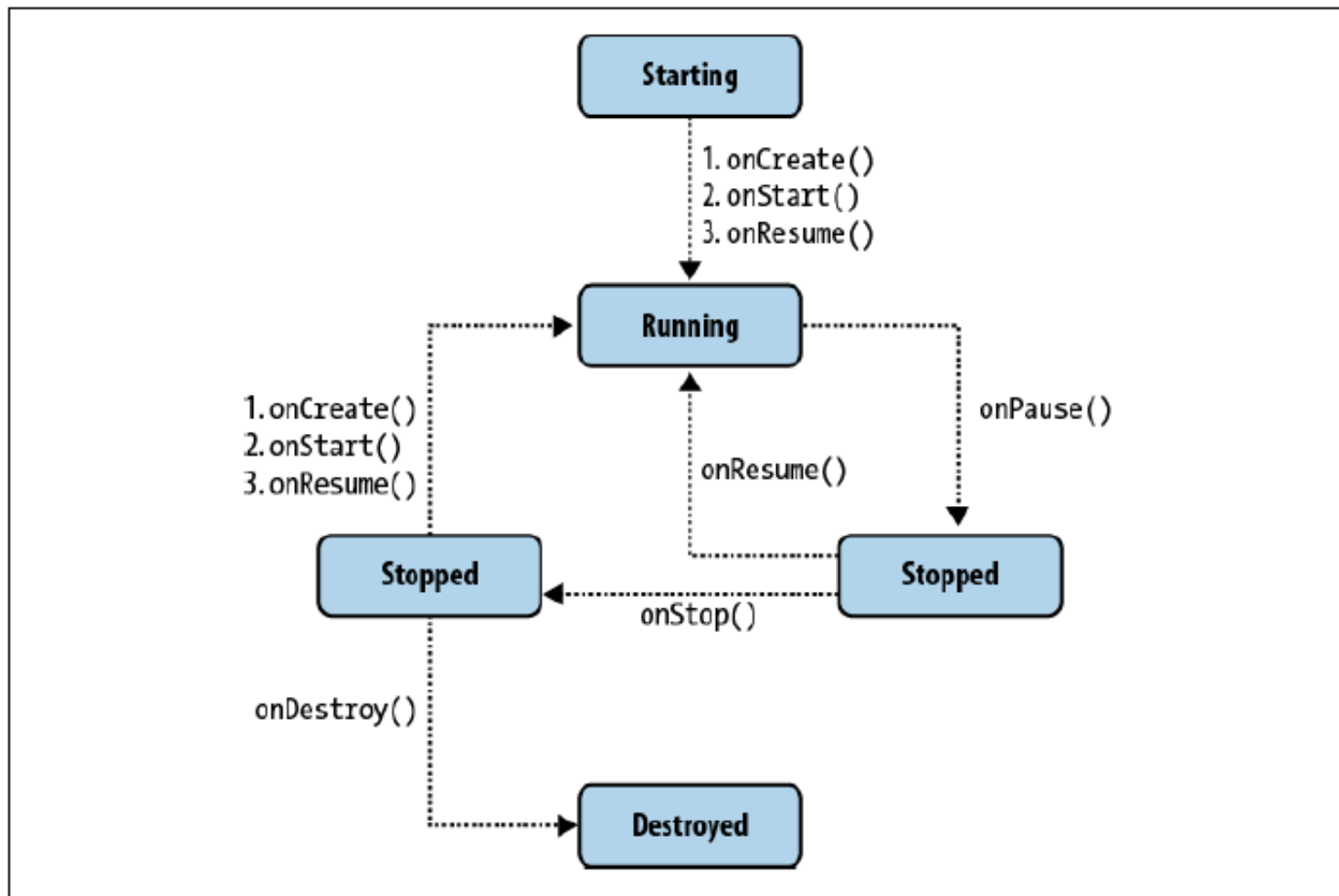
Activity Manager

- Odgovoran za pravljenje, uništavanje i upravljanje aktivnostima
 - Na primer, kada korisnik prvi put pokrene aplikaciju, Activity Manager će kreirati aktivnosti i prikazaće je na ekranu
 - Kasnije, kada korisnik promeni ekran, Activity Manager će premestiti tekuću aktivnost u magacin, odakle će je brže ponovo pozvati
 - Starije aktivnosti koje izvesno vreme nisu korišćene biće uništene da bi se oslobodio prostor za tekuću

Životni ciklus aktivnosti

- Programiranje za Android svodi se više na odgovaranje na promene stanja u aplikaciji nego na prouzrokovanje promena.
- To je upravljano, kontejnersko okruženje slično programiranju servleta
- Kada je reč o životnom ciklusu aktivnosti, nije važno znati u kom je stanju aktivnost, ali postoje razne mogućnosti da se reguliše šta se dešava u prelazima između stanja

Životni ciklus aktivnosti



Starting state

- Kada aktivnost ne postoji u memoriji, ona je u početnom stanju (*starting state*).
- Dok se pokreće, aktivnost će pozivati razne metode koje programer može da popuni
- Na kraju, aktivnost će biti u stanju *running*
- Aktivnost u stanju running state je ona koja je trenutno na ekranu i sa kojom korisnik komunicira; ona reaguje na sve akcije korisnika kao što su kucanje, dodirivanje ekrana ili pritiskanje dugmadi

Paused state

- Kada aktivnost nije u fokusu (tj. ne komunicira sa korisnikom), ali je još uvek vidljiva na ekranu, ona je u stanju *paused*
- Ovo nije čest slučaj, pojavljuje se npr. kada se dijalog pojavi ispred aktivnosti čime je pauzira
- Sve aktivnosti prolaze kroz stanje *paused* dok se zaustavljaju

Stopped state

- Kada aktivnost nije vidljiva, ali se još uvek nalazi u memoriji, ona je u stanju *stopped state*
- Zaustavljena aktivnost može se ponovo pozvati u prvi plan i vratiti u stanje Running, a može se i uništiti i ukloniti iz memorije
- Sistem zadržava aktivnosti u stanju stopped zato što je verovatno da će im se korisnik uskoro vratiti, a ponovno pokretanje stopirane aktivnosti je mnogo jeftinije od ponovnog pokretanja ispočetka.

Destroyed state

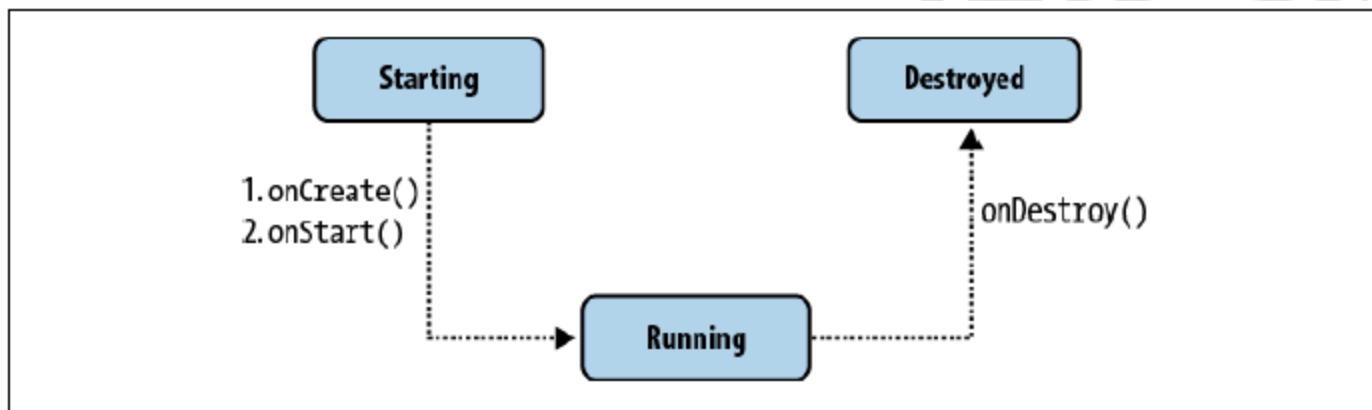
- Uništena aktivnost više nije u memoriji, jer je Activity Manager odlučio da ona više nije potrebna i uklonio ju je
- Pre nego što se aktivnost uništi, mogu se izvesti određene operacije, npr. čuvanje informacija, ali nema garancije da će aktivnost biti stopirana pre nego što bude uništena
- Zbog toga je bolje obaviti važne aktivnosti na putu ka stanju paused nego ka stanju destroyed

Service

- Aktivnosti su kratkog veka i mogu se ukinuti u svakom trenutku, npr. ako se pritisne dugme Back
- Servisi, s druge strane, izvršavaju se po potrebi, nezavisno od bilo koje druge aktivnosti, izvesno kratko vreme
- Servis se može koristiti npr. za proveravanje RSS feeda, ili za reprodukovanje muzike i kada se aktivnost koja ju je pokrenula završena
- Servisi rade u pozadini i ne moraju da imaju komponente korisničkog interfejsa

Service

- Android servisi koji su deo aplikacije nemaju veze sa izvornim Linux servisima (demonima), koji se nalaze mnogo niže, u operativnom sistemu
- Životni vek servisa je mnogo prostiji od aktivnosti, i uglavnom je pod kontrolom programera, a ne sistema



Content provider

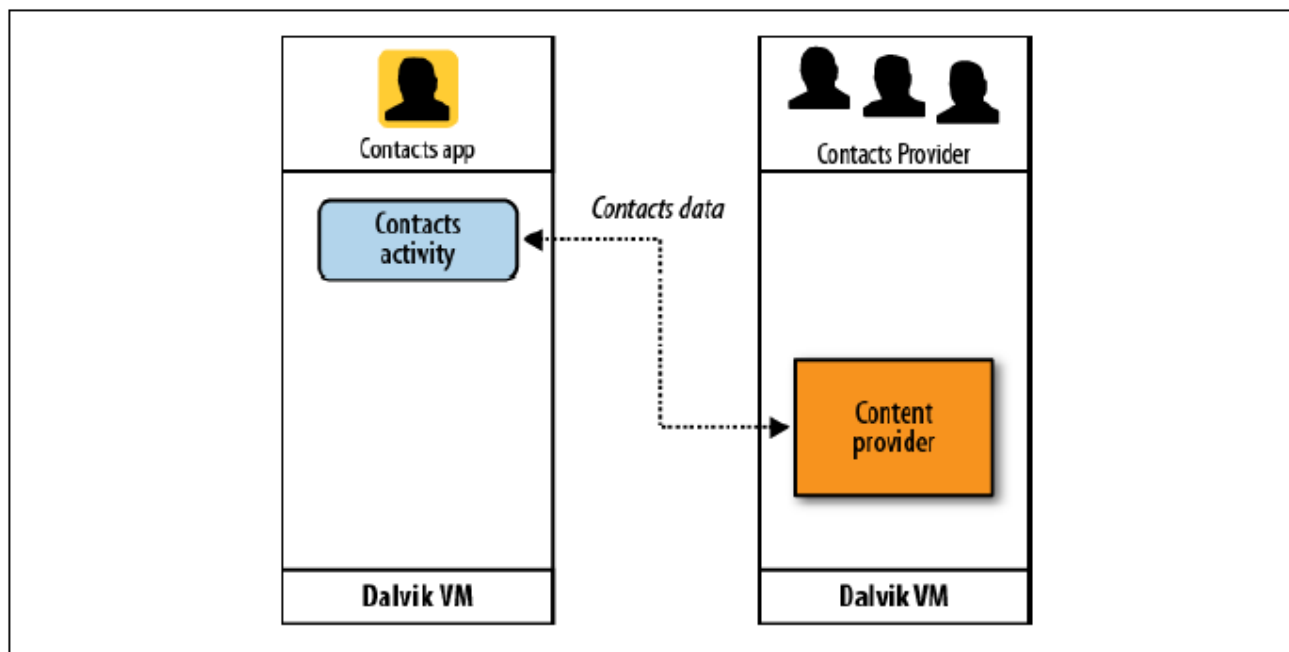
- Content provideri su interfejsi za deljenje podataka između aplikacija
- Android standardno pokreće svaku aplikaciju u sopstvenom sandboxu, što znači da su svi podaci koji joj pripadaju potpuno izolovani od drugih aplikacija u sistemu
- Iako se male količine podataka mogu razmenjivati između aplikacija preko intentova, content providers su mnogo pogodniji za razmenu perzistentnih podataka

Content Provider

- Na primer, Contacts Provider je content provider koji izlaže sve podatke o kontaktima različitim aplikacijama
- Settings Provider pruža sistemska podešavanja različitim aplikacijama, uključujući i ugrađenu aplikaciju Settings
- Media Store je odgovoran za čuvanje i deljenje različitih medija, npr. fotografija i muzike, u različitim aplikacijama

Content Provider

- Aplikacija Contacts koristi Contacts Provider; sama aplikacija Contacts nema nikakve podatke, a Contacts Provider nema korisnički interfejs



Content Provider

- Razdvojeno čuvanje podataka i aplikacija sa interfejsom omogućuje prilagođavanje sistema
 - npr. može se napraviti nova aplikacija za kontakte koja koristi postojeće podatke
 - mnogi proizvođači telefona koristi prednosti content providera da bi dodali sopstvene aplikacije i poboljšali korisnički interfejs, npr. HTC Sense

Content Provider

- Content provideri imaju relativno jednostavne interfejsse, sa standardnim metodama `insert()`, `update()`, `delete()`, i `query()`
- Relativno je lako implementirati content provider kao proksi ka bazi podataka
- Ipak, češće se content provideri koriste nego što se pišu sopstveni

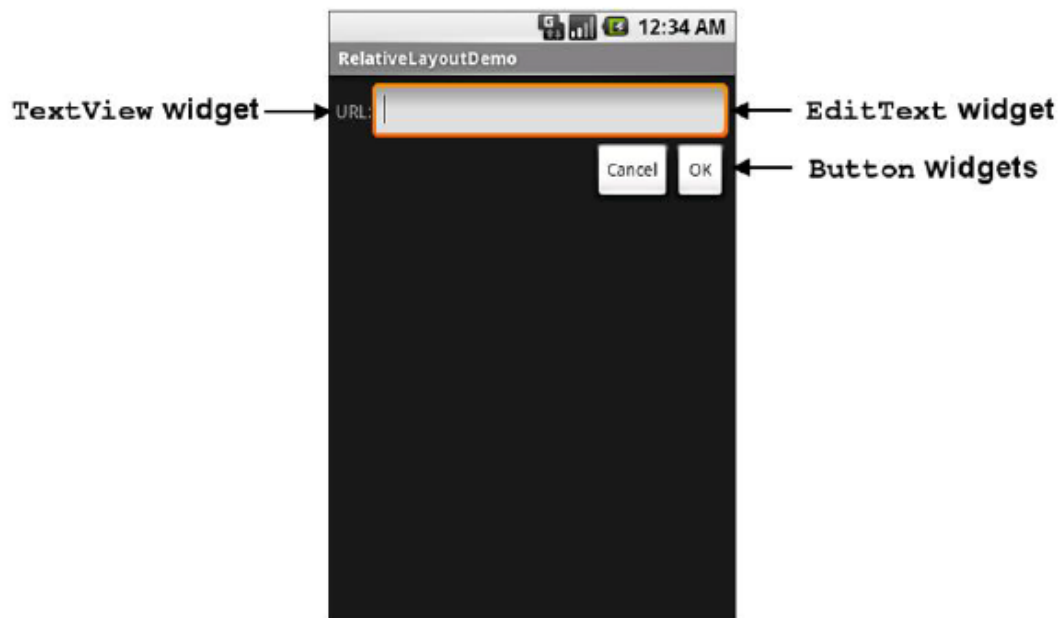


Broadcast Receiver

- Androidova implementacija projektnog obrasca Observer
- Prijemnik (receiver) je “uspavani” kod koji se aktivira kada se desi događaj na koji reaguje
 - Npr. kada stigne SMS poruka, baterija je na izdisaju, ili se sistem podiže
 - Ti događaji se emituju, a na njih mogu da reaguju broadcast receiveri
- Broadcast receiveri nemaju vizuelnu predstavu, niti se aktivno izvršavaju u memoriji

Widgets

- Omogućuju pravljenje dinamičkih, interaktivnih komponenata aplikacije
- Widget je “mikro” jedinica korisničkog interfejsa: polja, dugmad, labele, liste



Notifications

- Obaveštenja koja se koriste da bi nevidljive komponente aplikacije (Services, Broadcast Receivers) obavestile korisnike o događajima koji zahtevaju njihovu pažnju, pri čemu njihova aktivnost nije vidljiva
 - Prikazivanje statusne linije
 - Blinkanje LED-ova
 - Uključivanje vibracije
 - Zvučna upozorenja
 - Prikazivanje dodatnih informacija na paleti

Intents

- Intents (namere) su poruke koje se razmenjuju između glavnih blokova
- Intent je apstraktna predstava operacije koja treba da se izvrši
 - pokreće aktivnost, zaustavlja ili pokreće servis, šalje poruku
- Intentovi su asinhroni, što znači da kod koji ih šalje ne mora da čeka da se završe

Intents

- Intent može da bude eksplicitan ili implicitan
 - U eksplicitnom intentu pošiljalac jasno kaže koja komponenta treba da bude prijemnik
 - U implicitnom intentu pošiljalac zadaje samo tip prijemnika
- Android održava preslikavanje između intentova i aktivnosti i započinje odgovarajuću aktivnost na osnovu intentu.
- Kada postoji više aktivnosti koje mogu da obave intent, Android korisniku predstavlja listu aktivnosti iz koje treba birati

Intents

