


Curs Android

Braşov, 2013.10.05
Marius Mailat

Curs dezvoltat pentru  **LMS**[®]
A Siemens Business

Agenda prima zi

1. Prezentare
2. Program curs
3. Conținut
4. Partea 1 curs (Istorie, stack, ADT, debug, deployment)
5. Partea 2 curs (Anatomia aplicațiilor, controale UI)
6. Partea 3 curs (Liste, SQLite, Services, interactions)
7. Partea 4 curs (Exemple, exemple 😊, QA session)

Agenda prima zi

1. Prezentare
2. Program curs
3. Conținut
4. Partea 1 curs (advance stack)
5. Partea 2 curs
6. Partea 3 curs
7. Partea 4 curs

Program curs

- 08.30 - 09.00 – Înregistrare
- 09.00 - 11.00 - Curs partea I
- 11.00 - 11.15 - Pauză de cafea
- 11.15 - 13.00 - Curs partea II
- 13.00 - 14.00 - Pauză de pranz
- 14.00 - 16.00 - Curs partea III
- 16.00 - 16.15 - Pauză de cafea
- 16.15 - 18.00 - Curs partea IV

Prezentare

Marius Mailat - marius.mailat@gmail.com

- 12 years coding enterprise applications for Xerox and Sabre Travel Network.
- Desktop projects: IDSecuritySuite.com, SubmitSuite.com
- Android trainer for Marakana (now Twitter University) serving clients like Intel, Real Networks, Cisco, etc
- Delivered online Android trainings through www.androider.ro/cursuri-android
- CTO for Appseleration - Düsseldorf, a startup in north Germany with over 1000 apps developed for Android, iOS, WP and BB.
- Partner and co-founder in Appsrise SRL – Cluj-Napoca, a 2013 startup in middle Transylvania focused on hand-crafted mobile apps for Android and iOS



Continut curs Android pentru incepatori

- **Lecția 1 – Concepții de bază Java și Android**

- 1.1 Privire de ansamblu asupra platformei Android
- 1.2 Arhitectura sistemului de operare Android
- 1.3 Instalarea Android SDK și Android Development Tools (ADT)
- 1.4 Emulatorul și simularea unei versiuni de OS

- **Lecția 2 – Componentele unei aplicații Android**

- 2.1 Construcția internă a unei aplicații Android
- 2.2 Activități (Activities)
- 2.3 Intenții (Intents)
- 2.4 Servicii (Services)
- 2.5 Furnizori de conținut (Content Providers)
- 2.6 Receptoare de mesaje (Broadcast Receivers)

- **Lecția 3 – Anatomia unei aplicații Android**

- 3.1 AndroidManifest.xml
- 3.2 R.java
- 3.3 Fișierele resursă
- 3.4 Assets și Raw
- 3.5 Context

- **Lecția 4 – Sfaturi practice pentru dezvoltarea aplicațiilor Android**

- 4.1 Compilarea și testarea unei aplicații HelloWorld
- 4.2 Perspectiva DDMS în Eclipse
- 4.3 Internaționalizarea aplicației
- 4.4 Preferințele unei aplicații Android

- **Lecția 5 – Controale UI**

- 5.1 Dezvoltarea interfeței grafice
- 5.2 View și Layout-uri
- 5.3 Meniuri
- 5.4 Dialoguri
- 5.5 Notificări
- 5.6 Interacțiunea cu butoanele și alte controale

- **Lecția 6 – ListView și ListActivity**

- 6.1 ListView
- 6.2 ListActivity
- 6.3 ListActivity cu layout personalizat
- 6.4 Adaptere pentru un ListView

- **Lecția 7 – Baza de date SQLite**

- 7.1 SQLiteOpenHelper
- 7.2 Servicii REST
- 7.3 JSON

- **Lecția 8 – Webservices și comunicare prin rețea**

- 8.1 HTTP Client
- 8.2 Servicii REST
- 8.3 JSON
- 8.4 Parsare XML

- **Lecția 9 – Location API și Google Maps**

- 9.1 Android Location API
- 9.2 Google Maps

- **Lecția 10 – Publicarea unei aplicații în Google Play**

- 10.1 Pregătirea aplicației pentru publicare
- 10.2 Semnarea aplicației cu o cheie privată
- 10.3 Înregistrarea unui cont în Google Play
- 10.4 Publicarea aplicației în Google Play

Conceptii de bază Java și Android

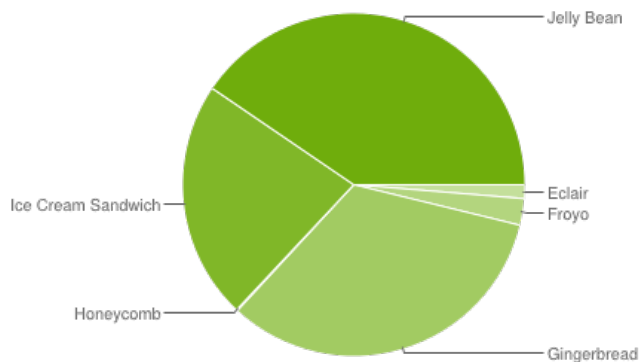
1. **Privire de ansamblu asupra platformei Android**
2. Arhitectura sistemului de operare Android
3. Instalarea Android SDK și Android Development Tools (ADT)
4. Emulatorul și simularea unei versiuni de OS

Privire de ansamblu asupra platformei Android

Android este un ecosistem dezvoltat inițial de Google și mai târziu de Open Handset Alliance, construit pentru dispozitive mobile, organizat în jurul nucleului Linux, mașinii virtuale Dalvik și a librăriilor externe open-source.

Din acest ecosistem fac parte sistemul de operare (sub licență Apache) și platforma software (SDK-ul Android) care permite programatorilor să scrie cod Java.

Versiunile Android lansate si distribuția lor



Platformă	Nume de cod	Data lansare	API level	Nucleu Linux	Distribuție (01.11.2012)
Android 1.0	Angel Cake	23.09.2008	1		
Android 1.1	Battenberg	09.02.2009	2		
Android 1.5	Cupcake	30.04.2009	3	2.6.27	
Android 1.6	Donut	15.09.2009	4	2.6.29	0,1%
Android 2.0	Eclair	26.10.2009	5	2.6.29	
Android 2.0.1	Eclair	03.12.2009	6	2.6.29	
Android 2.1	Eclair	12.01.2010	7	2.6.29	1,2%
Android 2.2	Froyo	20.05.2010	8	2.6.32	2,5%
Android 2.3	Gingerbread	06.12.2010	9	2.6.35	0,1%
Android 2.3.3	Gingerbread	09.02.2011	10	2.6.35	33,0%
Android 3.0	Honeycomb	22.02.2011	11	2.6.36	
Android 3.1	Honeycomb	10.05.2011	12		
Android 3.2 Android 3.2.1 Android 3.2.2 Android 3.2.4 Android 3.2.6	Honeycomb	15.07.2011	13		0,1%
Android 4.0.1	Ice Cream Sandwich	19.10.2011	14	3.0.1	
Android 4.0.3-4.0.4	Ice Cream Sandwich	16.12.2011	15		22,5%
Android 4.1	Jelly Bean	27.06.2012	16	3.0.31	34,0%
Android 4.2	Jelly Bean	13.11.2012	17		6,5%
Android 4.3	Jelly Bean	24.07.2013	18		

Conceptii de bază Java și Android

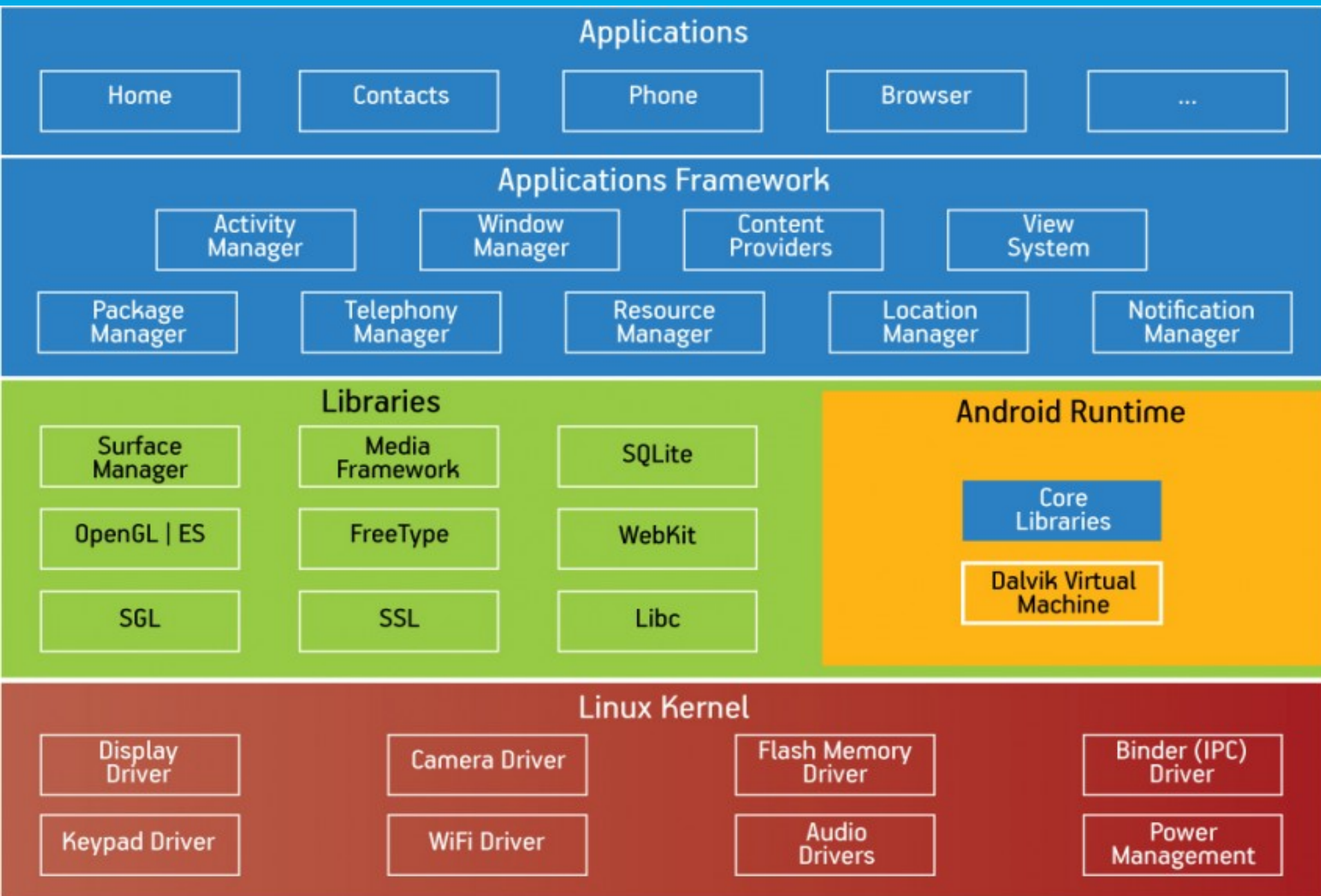
1. Privire de ansamblu asupra platformei Android
- 2. Arhitectura sistemului de operare Android**
3. Instalarea Android SDK și Android Development Tools (ADT)
4. Emulatorul și simularea unei versiuni de OS

Arhitectura sistemului de operare Android

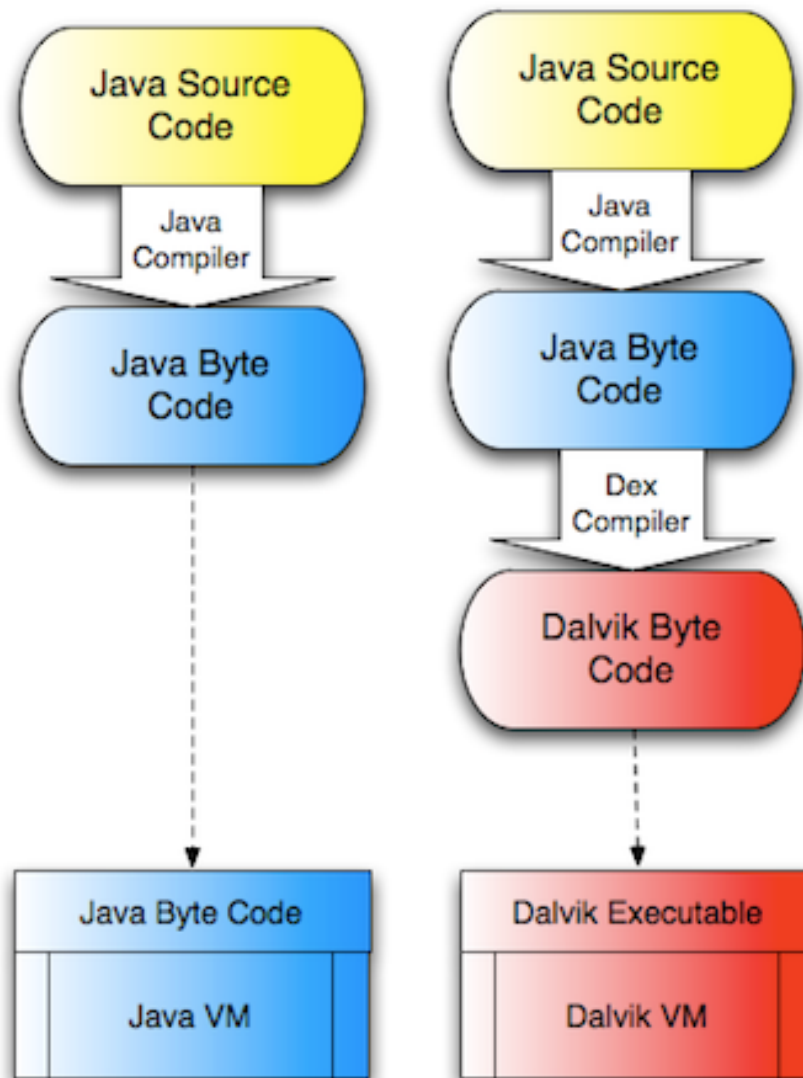
Ecosistemul Android este construit din mai multe straturi. Fiecare strat (layer) are un anumit scop și este construit diferit. Straturile care alcătuiesc Android se interesează și interacționează de multe ori între ele.

Straturile sunt astfel construite pentru a permite publicarea capabilităților lor către alte aplicații care doresc să le folosească (cu anumite restricții impuse de către framework). Aplicațiile dezvoltate de programatori au acces la aceleași API-uri ca și aplicațiile platformă.

Arhitectura sistemului de operare Android



Diferența între codul Android și Java?



Conceptii de bază Java și Android

1. Privire de ansamblu asupra platformei Android
2. Arhitectura sistemului de operare Android
- 3. Instalarea Android SDK și Android Development Tools (ADT)**
4. Emulatorul și simularea unei versiuni de OS

Conceptii de bază Java și Android

1. Privire de ansamblu asupra platformei Android
2. Arhitectura sistemului de operare Android
3. Instalarea Android SDK și Android Development Tools (ADT)
- 4. Emulatorul și simularea unei versiuni de OS**

Componentele unei aplicații Android

1. Construcția internă a unei aplicații Android
2. Activități (Activities)
3. Intenții (Intents)
4. Servicii (Services)
5. Furnizori de conținut (Content Providers)
6. Receptoare de mesaje (Broadcast Receivers)

Privire de ansamblu asupra platformei Android

În această lecție vom învăța despre componentele unei aplicații Android. Vom parcurge ciclul de viață al activităților și serviciilor și vom descoperi câteva elemente specifice Android: intenții, furnizori de conținut și receptoare de mesaje.

Aplicațiile Android sunt scrise în limbajul de programare Java. Android SDK tool compilează codul sursă împreună cu resursele aplicației într-o arhivă cu terminația APK. Acest APK este considerat a fi aplicația în sine.

Componentele unei aplicații Android

- 1. Construcția internă a unei aplicații Android**
2. Activități (Activities)
3. Intenții (Intents)
4. Servicii (Services)
5. Furnizori de conținut (Content Providers)
6. Receptoare de mesaje (Broadcast Receivers)

Context

Componentele unei aplicații: activitățile (Activities), intențiile (Intents), serviciile (Services), furnizorii de conținut (Content Providers), receptoarele de mesaje (Broadcast Receivers) împreună cu elementele de legătură: AndroidManifest.xml și fișierele resursă traiesc în același context al aplicației și realizează aplicația în sine.

Contextul aplicației se referă la aplicație și la procesul în care rulează toate componentele sale. Contextul permite unei aplicații să partajeze resursele și datele între componentele sale.

Construcția internă a unei aplicații Android

Componentele unei aplicații Android sunt blocuri esențiale cu roluri specifice. Există 5 tipuri esențiale de componente Android, fiecare dintre ele având un ciclu de viață propriu.

1. **Activitățile** (Activities) sunt, prin analogie cu paginile web, ecrane cu o interfață utilizator.
2. **Intențiile** (Intents) permit activarea activităților, serviciilor și receptoarelor de mesaje.
3. **Serviciile** (Services) reprezintă componente care rulează în fundal și realizează operații care necesită mult timp de procesare.
4. **Furnizorii de conținut** (Content Providers) oferă accesul altor aplicații la datele unei aplicații și la interogarea acestora.
5. **Receptoarele de mesaje** (Broadcast Receivers) interceptează anunțurile/mesajele sistem sau trimise de alte aplicații.

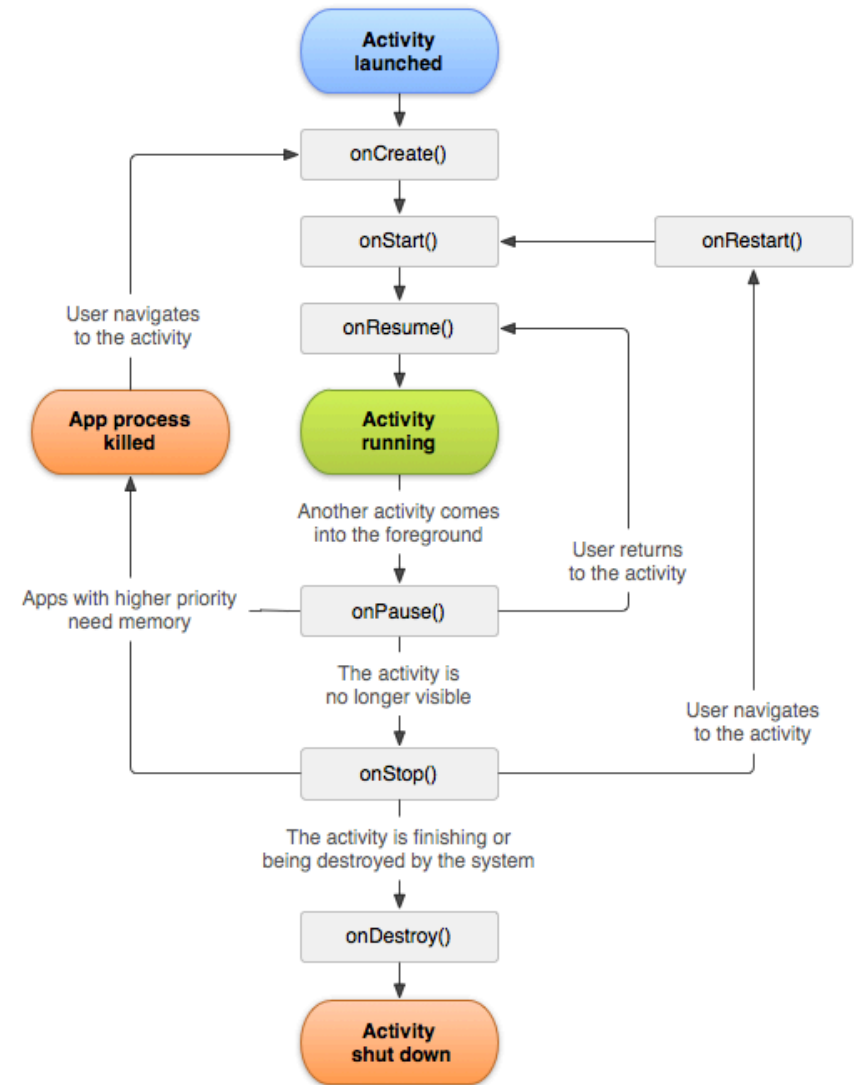
Componentele unei aplicații Android

1. Construcția internă a unei aplicații Android
- 2. Activități (Activities)**
3. Intenții (Intents)
4. Servicii (Services)
5. Furnizori de conținut (Content Providers)
6. Receptoare de mesaje (Broadcast Receivers)

Activitățile (Activities)

O activitate reprezintă un ecran cu o interfață de vizualizare.

O analogie pentru a înțelege mai bine activitățile este comparația cu un site web unde fiecare pagină web poate să fie considerată o activitate și unde este posibilă trecerea de la o pagină (activitate) la alta.



Componentele unei aplicații Android

1. Construcția internă a unei aplicații Android
2. Activități (Activities)
- 3. Intenții (Intents)**
4. Servicii (Services)
5. Furnizori de conținut (Content Providers)
6. Receptoare de mesaje (Broadcast Receivers)

Intenții (Intents)

Activitățile, serviciile și receptoarele de mesaje sunt activate prin mesajele trimise asincron de tip intenții (*Intents*). De exemplu, o activitate poate trimite un *Intent* către sistemul de operare Android pentru a lansa o nouă activitate.

Intențiile pot fi folosite pentru a semnaliza sistemului de operare că un anumit eveniment a apărut. Alte componente în sistemul Android se pot înregistra pentru a fi notificate de apariția unui astfel de eveniment (*BroadcastReceivers*).

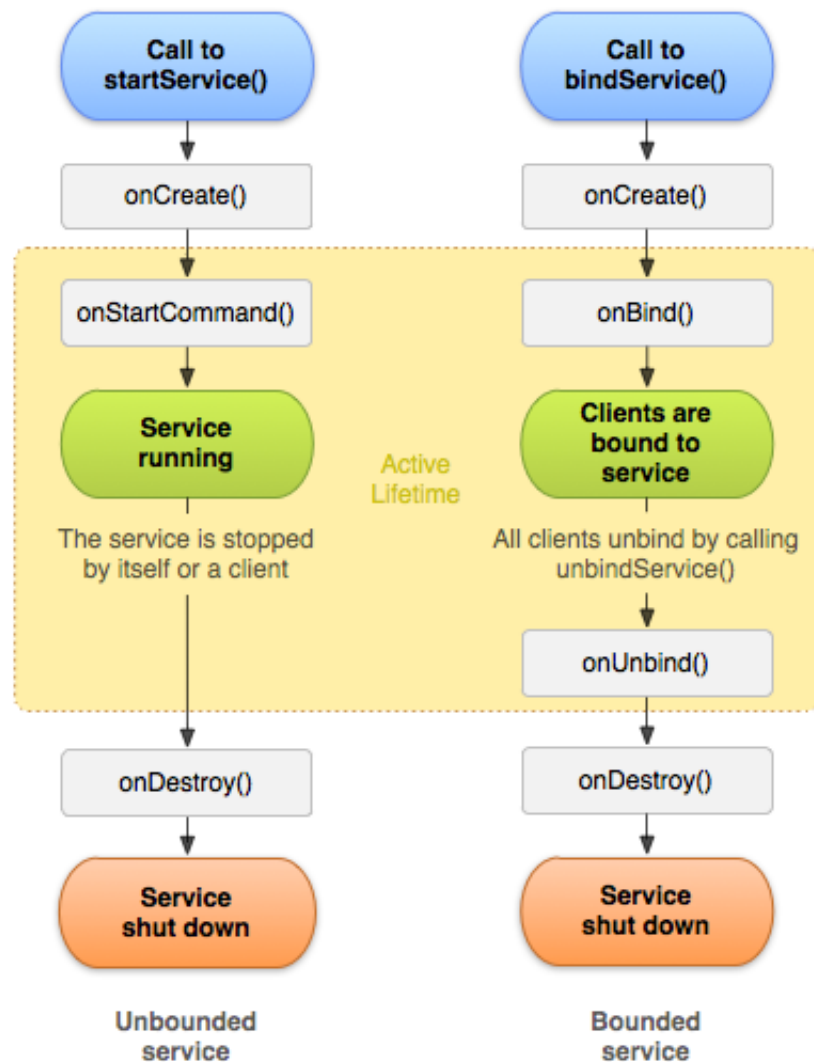
Componentele unei aplicații Android

1. Construcția internă a unei aplicații Android
2. Activități (Activities)
3. Intenții (Intents)
- 4. Servicii (Services)**
5. Furnizori de conținut (Content Providers)
6. Receptoare de mesaje (Broadcast Receivers)

Servicii (Services)

Serviciul este o componentă non vizuală care rulează în fundal și realizează operații de lungă durată sau se conectează la un sistem extern pentru a obține date.

De exemplu, un serviciu poate rula muzică în fundal, în timp ce un al doilea serviciu verifică actualizările aplicației. Serviciul este o subclasă a Service.



Componentele unei aplicații Android

1. Construcția internă a unei aplicații Android
2. Activități (Activities)
3. Intenții (Intents)
4. Servicii (Services)
- 5. Furnizori de conținut (Content Providers)**
6. Receptoare de mesaje (Broadcast Receivers)

Furnizori de conținut (Content Providers)

Folosind un furnizor de conținut permiți celorlalte aplicații accesul la datele stocate de aplicație local. De exemplu, sistemul Android permite accesul la datele din agendă, la fișierele media sau la alte elemente definite în A.provider.

Exemplul de cod de mai jos permite citirea numelui și numărului de telefon din agenda telefonică.

```
public void citesteContactele() {
    ArrayList<String> listaContacte = new ArrayList<String>();
    String[] columns = new String[] { People.NAME, People.NUMBER };
    Uri mContacts = People.CONTENT_URI;
    Cursor mCur = managedQuery(mContacts, columns, null, null, People.NAME);
    if (mCur.moveToFirst()) {
        do {
            listaContacte.add(mCur.getString(mCur.getColumnIndex(People.NAME)));
        } while (mCur.moveToNext());
    }
    Toast.makeText(this, "Am citit " + listaContacte.size(), Toast.LENGTH_LONG).show();
}
```

Receptoare de mesaje (Broadcast Receivers)

1. Construcția internă a unei aplicații Android
2. Activități (Activities)
3. Intenții (Intents)
4. Servicii (Services)
5. Furnizori de conținut (Content Providers)
- 6. Receptoare de mesaje (Broadcast Receivers)**

Receptoare de mesaje (Broadcast Receivers)

Receptoarele de mesaje sunt componente care răspund la mesajele emise pentru întreg sistemul. Multe dintre aceste mesaje au originea în sistem. Exista posibilitatea de a avea un receptor de mesaje care să detecteze pornirea telefonului, primirea unui apel telefonic, primirea unui sms, dacă ecranul a fost închis, dacă bateria este la limită, dacă o poză a fost realizată cu camera foto, etc.

În mod normal, un receptor de mesaje este doar calea de intrare pentru mesaj, procesarea aceluși mesaj realizându-se într-un serviciu dedicat. Receptorul de mesaje definește metoda `onReceive()`. Doar în această metodă obiectul `BroadcastReceiver` va fi activ, din această cauză în `onReceive()` neputându-se realiza operații asincrone.

Continut curs Android pentru avansați

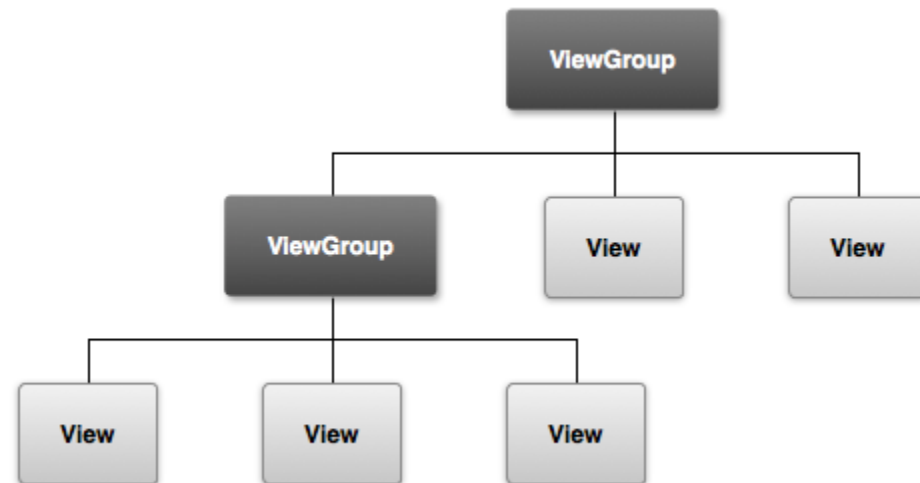
- **Lecția 1 – Concepții avansate Java și Android**
 - 1.1 Privire în detaliu a platformei Android
 - 1.2 Arhitectura în detaliu a sistemului de operare Android
 - 1.3 Android SDK și Android Development Tools (ADT)
- **Lecția 2 – Concepte avansate interfata Android**
 - 2.1 Styles
 - 2.2 Attributes
 - 2.3 Teme
 - 2.4 Folosirea DPI (Device Independent Pixel)
 - 2.5 Fragments
 - 2.6 Exemplu de controale customizate iPhone like
- **Lecția 3 – Android Widgets**
- **Lecția 4 – Android Live Wallpaper**
- **Lecția 5 – Content Providers**
- **Lecția 6 – Android Native Development Kit (NDK)**
- **Lecția 7 – Securitatea Android**
- **Lecția 8 – Compilarea platformei Android din codul sursă**
- **Lecția 9 – Threading în Android**
- **Lecția 10 – Dezvoltarea de jocuri pentru Android**

Controale UI

1. **View și Layout-uri**
2. Meniuri
3. Dialoguri
4. Notificări
5. Interacțiunea cu butoanele și alte controale

View și Layout-uri

O aplicație Android este construită folosind obiectele View și ViewGroup. Obiectele de tip View (AnalogClock, ImageView, ProgressBar, TextView, Button, CalendarView, CheckBox, e sunt unitățile de bază în construcția interfeței unei aplicații.



Meniuri

Meniurile erau până la Android 3.0 (API level 11) elemente des întâlnite în aplicațiile Android. Pentru aplicațiile dezvoltate pentru Android 1.5 - 2.3 utilizatorii puteau apăsa butonul meniu al telefonului și aveau anumite opțiuni la dispoziție precum: căutare, despre, contact, etc.

Începând cu Android 3.0 elementele din Options Menu sunt prezentate de Action Bar. Majoritatea dispozitivelor cu Android 4.0 nu mai includ un buton de meniu și este recomandat să faci trecerea către Action Bar sau alte opțiuni.

Dialoguri

Un dialog este o fereastră care apare deasupra unei activități. Activitatea de sub dialog pierde focusul și dialogul modal este pregătit pentru interacțiunea cu utilizatorul. Dialogurile sunt în mod normal folosite pentru confirmări sau alte interacțiuni cu utilizatorul.

Un specialist de UI ar spune că dialogurile sunt ceea ce urăsc cel mai mult utilizatorii la o aplicație mobilă; din această cauză trebuie folosite fără a se abuza de ele.

Notificări

Android îți permite să adaugi elemente de avertizare cu text și icoană în bara de notificare a telefonului. Utilizatorul poate deschide bara de notificare și să aleagă notificarea adăugată de tine programatic. În acest moment este lansat un Intent care este definit în notificare, de obicei o redirectare spre o nouă activitate.

O notificare în status bar poate fi însoțită și de un sunet, vibrație sau semnalizare luminoasă. Notificarea trebuie folosită în cazul în care un serviciu are nevoie de un răspuns sau de o reacție de la utilizator.

Interacțiunea cu butoanele și alte controale

Interacțiunea în Android cu widgeturile și controalele acestora se realizează prin capturarea evenimentelor unui View.

Următoarele metode callbacks sunt cele mai frecvente în interfețele event listener:

- `onClick()`
- `onLongClick()`
- `onFocusChange()`
- `onKey()`
- `onTouch()`
- `onCreateContextMenu()`

1. **SQLiteOpenHelper**
2. Crearea tabelelor
3. Modificare înregistrărilor unei tabele
4. Interogarea SQLite din linia comandă

SQLiteOpenHelper

Metoda recomandată pentru a crea o bază de date este o instanță SQLiteOpenHelper care suprascrie metoda onCreate(). Aici execuți comenzi SQLite pentru a crea baza de date și tabelele incluse.

Odată ce baza de date este inițializată prin SQLiteOpenHelper te poți folosi de metodele getReadableDatabase() sau getWritableDatabase() pentru a obține o referință la ea. Android salvează bazele de date în directorul: /data/data/pachetul aplicatiei/databases. În cazul exemplului de mai jos, baza de date se găsește în: /data/data/com.androider.demo8/databases/cities.db

Crearea tabelelor

Pentru a crea o tabelă în cadrul unei baze de date te folosești de metoda `execSQL()` care acceptă ca argument un string care descrie operația ce trebuie efectuată. Pentru a crea o tabelă cu orașe și locațiile cu latitudine / longitudine folosim:

```
String sql = "CREATE TABLE CITIES ( _ID INTEGER PRIMARY KEY  
AUTOINCREMENT, CITY TEXT NOT NULL, LATITUDE REAL,  
LONGITUDE REAL);";  
db.execSQL(sql);
```


Modificare înregistrărilor unei tabele

Android oferă câteva clase care simplifică accesul la baze de date. Astfel, instanța de ContentValues îți permite să creezi o serie echivalentă a metodei INSERT care mai apoi să fie executată prin metoda insert().

```
//import coordonatele pentru orase prin ContentValues
ContentValues cv=new ContentValues();
cv.put(CititesDataSQLHelper.CITY, "Medias");
cv.put(CititesDataSQLHelper.LATITUDE, 24.36);
cv.put(CititesDataSQLHelper.LONGITUDE, 46.16);
db.insert(CititesDataSQLHelper.TABLE, null, cv);
```

Echivalentul codului de mai sus, folosind doar un SQL query, este:

```
//importa coordonatele pentru orase prin SQL
db.execSQL("INSERT INTO cities values ( null, 'Medias',24.36,46.16);");
```

Interogarea SQLite din linia comandă

Android SDK conține utilitarul ADB care îți permite să ai acces la linia de comandă din emulator.

Ne vom lega la emulator și vom interoga noua bază de date folosind utilitarul sqlite3.

QA session

Marius Mailat
Marius.mailat@gmail.com

www.appsrise.com
www.appselevation.com