

Universitatea Politehnica din Bucuresti, Facultatea de Electronica,  
Telecomunicatii si Tehnologia Informatiei

Tema de casa

Platforma de laborator. Sistemul de operarea Android

Student: Șerbănescu George Daniel  
Grupa : 431 A

- București -  
2012



# Cuprins

## 1. Istoric

### 1.1 Istoricul versiunilor

## 2. Design. Suport hardware

## 3. Procese. Organizarea proceselor

## 4. Instalarea SDK. Pasi

## 5. Aplicatii rulate pe emulatorul virtual

### 5.1 Aplicatia " Hello,Android "

### 5.2 Trimiterea unui SMS din o aplicatie Android

### 5.3 Rularea unui fisier MP3 in o aplicatie Android

### 5.4 Aplicatie Android care face telefonul sa vibreze

### 5.5 Activarea device-ului wi-fi. Activarea semanlului pentru scanarea de retele wifi

## 6. Bibliografie



## 1. Istoric

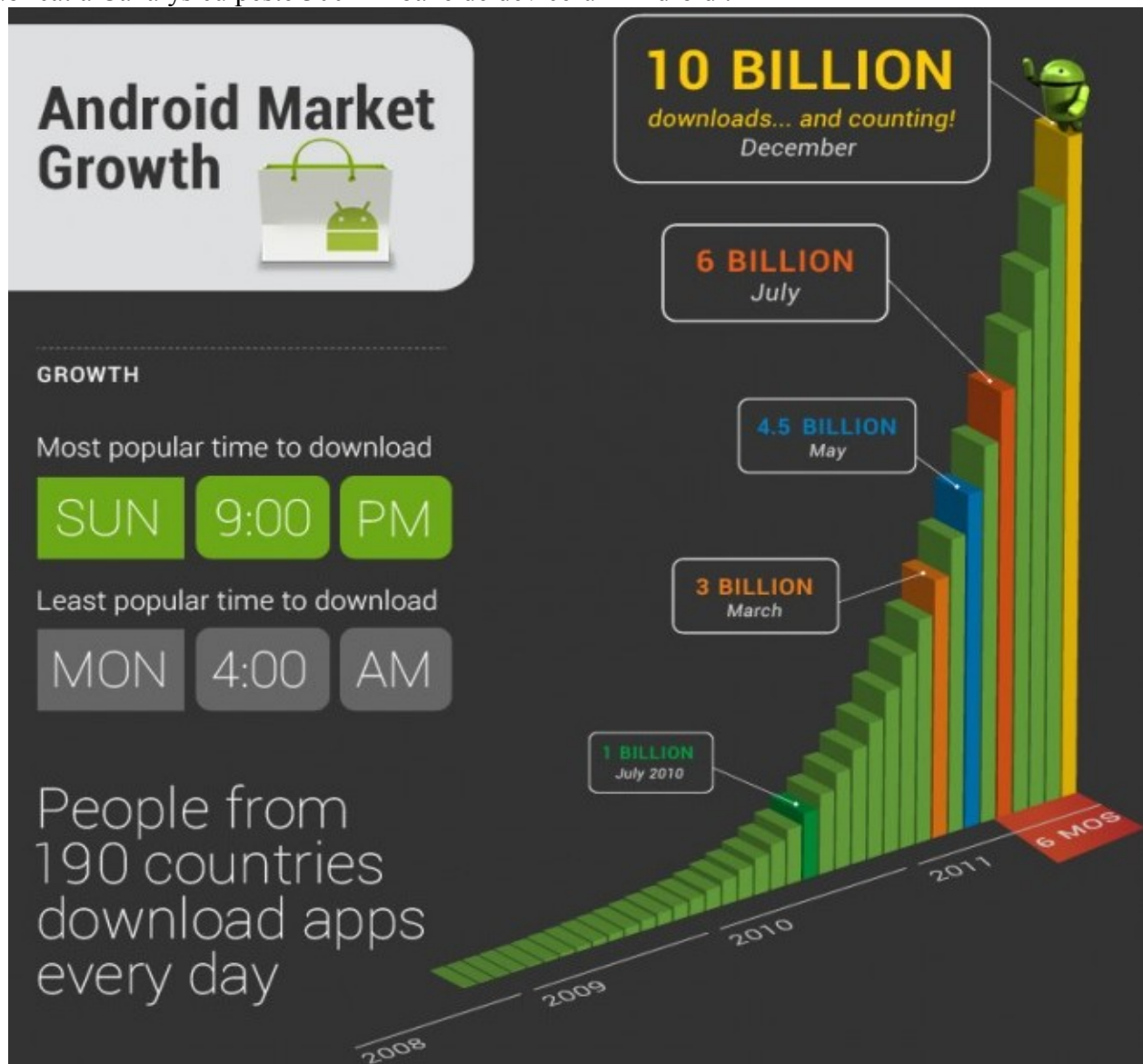
Android este un sistem de operare bazat pe Linux ce este folosit pentru device-uri mobile precum si pentru tablete sau smartphone-uri. A fost dezvoltat de Open Handset Alliance, condus de Google, si alte companii.

Google a preluat firma care a dezvoltat softul initial, Android Inc. , in 2005 .

Lansarea pe piata a Android a avut loc in 2007 de catre Open Handset Alliance, un consortiu format din 86 de companii din domeniu hardware, software si telecomunicatii devotate unui nou standard mai performant pentru device-urile mobile. Google a lansat codul Android drept open-source, sub licenta Apache . Android Open Source Project ( AOSP ) se ocupa cu mentenanta si dezvoltarea ulterioara a sistemului de operare Android.

Exista o comunitate mare de dezvoltatori care scriu aplicatii care maresc functionalitatea deviceurilor care au Android drept sistem de operare. Dezvoltatorii scriu codul intr-o versiune customizata de Java . Aplicatiile pot fi downloadate de pe magazinul online Google Play ( formal Android Market ). In Octombrie 2011, existau mai mult de 500.000 aplicatii pentru Android si numarul de download-uri estimat pana in Decembrie 2011 este de peste 10 milioane .

Android a fost ales cel mai vandut sistem de operare pentru mobile in 2010 de catre publicatia Canalys cu peste 300 milioane de device-uri Android .





## 1.1 Istoricul versiunilor

Dupa lansare au aparut numeroase upgarade-uri, fiecare eliminad bug-uri si adaugand noi functionalitati. Fiecare versiune este numita in ordine alfabetica dupa un desert .

Versiuni recente :

A. **2.3 GingerBread** : redefineste interfata utilizatorului, imbunatateste tastatura touchscreen-ului si functionalitatiile copy/paste, suport native pentru cod imbunatait ( cu imbunatari in zona performantelor jocurilor ), adaugarea suportului pentru apeluri VoIP si suport pentru Near Field Communication.

B. **3.0 Honeycomb** : este o versiune creata pentru tablete avand functionalitati pentru device-urile cu ecran mare si introduce multe functionalitati in zona interfetei utiliztorului, suport pentru procesoare multi-core, hardware acceleration pentru aplicatiile grafice si o intreaga criptare a sistemului. Primul device care a rulat pe aceasta versiune a fost tableta Motorola Xoom

C. **4.0 Ice Cream Sandwich** : anuntata pe 19 octombrie 2011 aduce functionalitatiile versiunii Honeycomb, creata pentru tablete, pe smartphone-uri si adauga aspecte functionale incluzand recunoasterea faciala, monitorizarea si controlul datelor din retea, face legatura contactlor dinagenda cu persoanele din reteau sociala, noi aplicatii privind camera foto, cautare in mailurile primite offline, directoarele aplicatiilor si share-uirea informatiilor folosind Near Field Communication.

Android 4.0.4 este ultima versiune , codul sursa fiind publicat pe 14 Noiembrie 2011.





## 2.Design . Suport hardware

Sistemul de operare Android are la baza kernel Linux cu librării și API-uri scrise în C, iar aplicațiile software rulează pe un framework care include librării compatibile cu Java bazate pe Apache Harmony. Principala platformă hardware pentru Android este arhitectura ARM . Arhitectura x86 pentru proiectul x86 , Google TV folosește o versiune specială de x86 pentru Android.

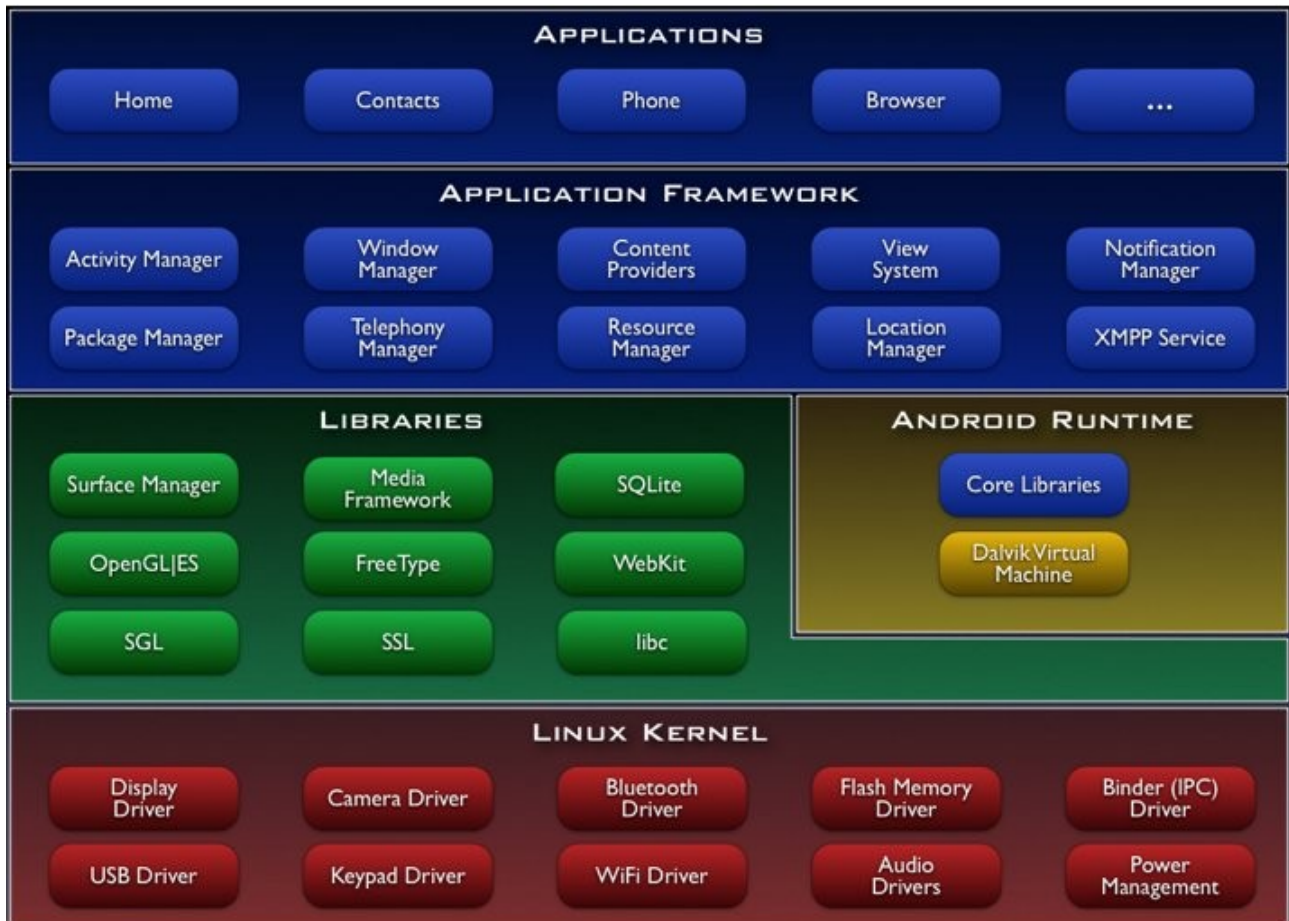


Fig. Arhitectura sistemului Android

Oferind o platformă „open development” , Android furnizează dezvoltatorilor de aplicații abilitatea de a construi un număr mare de aplicații inovative. Dezvoltatorii pot folosi informații despre hardware, informații despre locație, să ruleze aplicații din background, să seteze alarme, să adauge notificări în bara de stare .

La baza tuturor aplicațiilor stă un set de servicii și sisteme, printre care :

**I.** Un set bogat și extensibil de „Views” care pot fi utilizate pentru a construi o aplicație, incluzând liste, text box-uri, butoane și chiar un browser web .

**II.** „Content Providers” care permit accesul aplicațiilor la date din alte aplicații sau să poată da share propriilor date.

**III.** „Resource Manager”, permite accesul la resurse non-code precum string-uri de localizare, fișiere grafice sau fișiere layout .

**IV.** „Notification Manager” permite tuturor aplicațiilor să afișeze alerte în bara de stare.

**V.** „Activity Manager” organizează ciclul de funcționare al aplicațiilor .



### 3. Procese. Organizarea proceselor

În mod implicit, toate componentele ale aceleși aplicații rulează în același proces. Sistemul Android este conceput astfel încât să mențină procesul unei aplicații cât mai mult cu putință, dar eventual are nevoie să anuleze procesele vechi pentru a remprospăta memoria pentru procese noi sau mai importante. Pentru a decide care proces să fie păstrat și care să fie închis, sistemul plasează fiecare proces într-o ierarhie a importanței bazată pe componentele care sunt folosite în cadrul unui proces și pe starea fiecărei dintre aceste componente. Procesele cu cea mai neînsemnată importanță vor fi eliminate primele, apoi cele cu următoarea valoare puțin importantă și ciclul continuă până când se atinge valoarea necesară de resurse a sistemului.

Importanța proceselor este împărțită în cinci nivele. Următoarea listă prezintă diferențele dintre procese în ordinea importanței (primul proces este cel mai important și închis ultimul) :

- I. Procese din prim plan .** Un proces care este necesar pentru ce face utilizatorul în momentul de față . Există puține astfel de procese care rulează la orice moment. Sunt închise ca ultimă opțiune , dacă memoria atinge un prag atât de jos astfel încât nu mai pot continua să funcționeze.
- II. Procese vizibile .** Un proces care nu are nici un o componentă de prim plan, dar care este vizibilă utilizatorului. Pot fi întâlnite , spre exemplu, dacă activitatea din prim plan pornește un dialog, care permite activității precedente să văzată după cea actuală.
- III. Procese serviciu .** Un proces care rulează un serviciu și a fost pornit cu metoda `startService()` și care nu se încadrează în celelalte două categorii superioare. Nu au corespondență cu un aspect pe care utilizatorul îl vede direct dar care are o importanță majoră , precum rularea unui music player în background sau download-  
area unor date din rețea.
- IV. Procese din background .** Un proces care menține o activitate ce nu este vizibilă momentan utilizatorului . Aceste procese nu au un impact asupra experienței utilizatorului sistemul putând să le închidă în orice moment în favoarea unui proces vizibil, din prim plan sau unul care rulează un serviciu.
- V. Procese goale .** Un proces care nu are nici o componentă activă a unei aplicații. Singurul motiv pentru care sunt păstrate active este pentru a îmbunătăți timpul de pornire pentru următoarele vor trebui să ruleze. Sistemul închide adesea aceste procese pentru a obține un echilibru al resurselor.





## 4. Instalarea SDK

SDK vine de la *Software Development Kit* si este o colectie de unelte pentru dezvoltarea aplicatiilor. O astfel de colectie pentru Android permite dezvoltarea aplicatiilor pentru sistemul de operare Android. SDK-ul se poate instala pe un sistem de operare Windows, compatibilitatea este garantata pentru versiunile XP, Vista si restul versiunilor aparute ulterior.

Pentru rulara aplicatiilor pe emulator este necesara instalarea unei masini virtuale Java. Se poate descarca de pe site-ul oficial : <http://java.com/en/download/index.jsp>

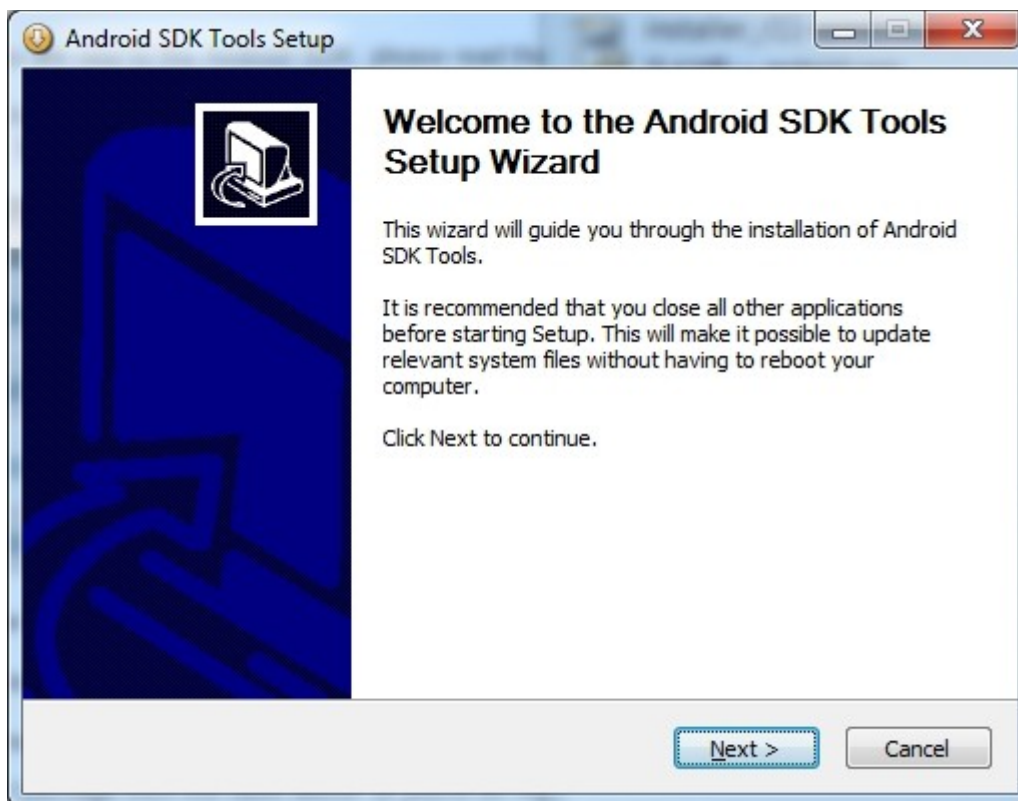
### Pasul 1. Descarcarea de pe internet a fișierelor necesare

Cea mai nouă versiune de Android SDK se descarca de pe site-ul <http://developer.android.com>. Versiunea recomandata este cea cu installer-ul de tip executabil (l [installer\\_r11-windows.exe](#)). De asemenea este nevoie de JDK (Java Development Kit), daca nu aveti deja o versiune instalata. JRE (Java Runtime Environment) nu este de ajuns. Pentru a downloada o versiune puteti accesa pagina [Oracle Java](#). Cea mai nouă versiune este [jdk-6u25](#). Folositor este frameworkul Eclipse, ce fi folosita ca mediu grafic de dezvoltare. <http://www.eclipse.org/downloads/>.

Optional se poate descarca manual si pluginul ADT pentru Eclipse. Descarcarea acestuia se poate face însa direct din Eclipse .

### Pasul 2. Instalarea SDK-ului Android

Se deschide instalatorul din linkul de mai sus. Urmatoarea fereastra ca apare :



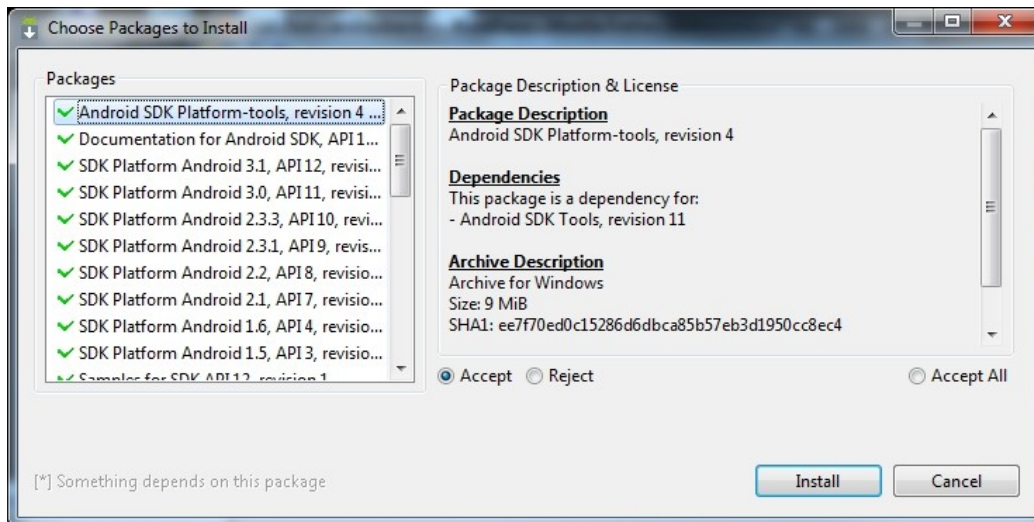
Se instaleaza urmand pasii, fara a avea cerinte speciale.

Prin aceasta instalare, se vor copia si instala fisierele binare necesare dupa

care în ultima fereastră vom avea opțiunea de a porni SDK Manager-ul pentru ca acesta să își descarce librăriile diferitelor versiuni de sistem de operare Android, librăriilor specifice pentru anumite telefoane și exemplelor de programe.

#### **Pasul 4. Descărcarea prin SDK Manager a librăriilor necesare dezvoltării**

La prima pornire a utilitarului *SDK Manager* se va descărca lista pachetelor disponibile. Dacă este nevoie de configurarea unui proxy pentru accesul la internet, acest lucru trebuie făcut în zona *Settings* a *SDK Manager*.



Automat, la prima deschidere a managerului, sunt selectate un număr de pachete, ce conțin documentația, exemple de programe, librăriile diferitelor versiuni ale sistemului de operare, librării pentru introducerea reclamelor în aplicație și librării necesare tranzacțiilor prin Android Market direct din aplicație.

În afara de cele selectate automat, mai sunt librării specifice, cum ar fi cele pentru grafică 3D, librării specifice pentru Samsung Galaxy Tab sau pentru teatru aplicației folosind un telefon conectat prin USB. Pentru selectarea unei alte librării decât cele standard se selectează librăria din lista din stânga și se alege *Accept*.

După selectarea tuturor librăriilor dorite se apasă pe *Install* și va începe descărcarea lor de pe internet.

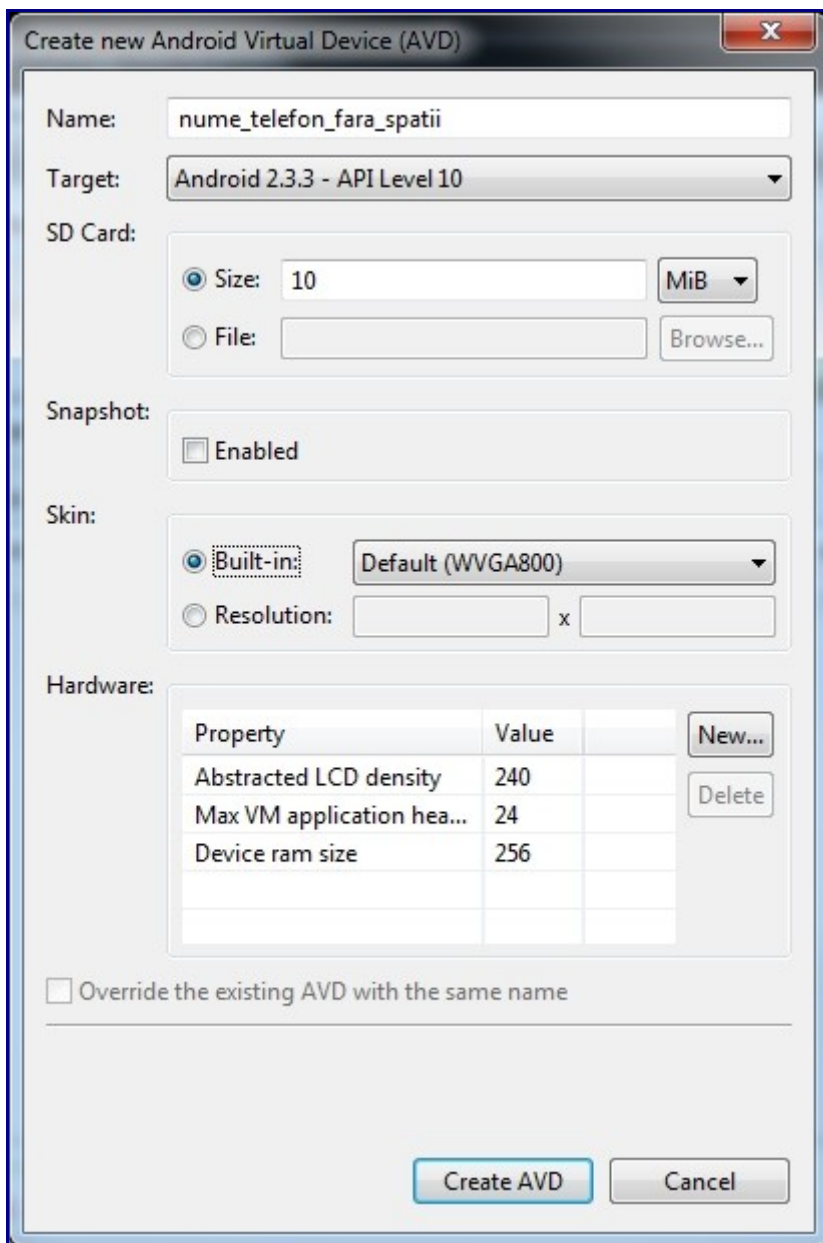
Pentru selectarea ulterioară a altor librării, se pornește *SDK Manager* și se alege *Update All* din *Installed packages* sau se bifează pachete noi dintre cele din *Available Packages*. Fișierele descărcate se instalează automat tot în aceeași locație în care s-au instalat utilitățile SDK-ului.

#### **Pasul 5. Instalarea unui device virtual Android**

Pentru testarea aplicațiilor vom avea nevoie de cel puțin un device virtual Android (AVD). Aplicațiile pot fi testate și cu un device Android adevărat, legat prin USB, dar aceste device-uri virtuale va permite testarea aplicațiilor pe mai multe versiuni ale sistemului de operare și la mai multe rezoluții ale ecranului, fără a fi nevoie să vă cumpărați o multitudine de telefoane.

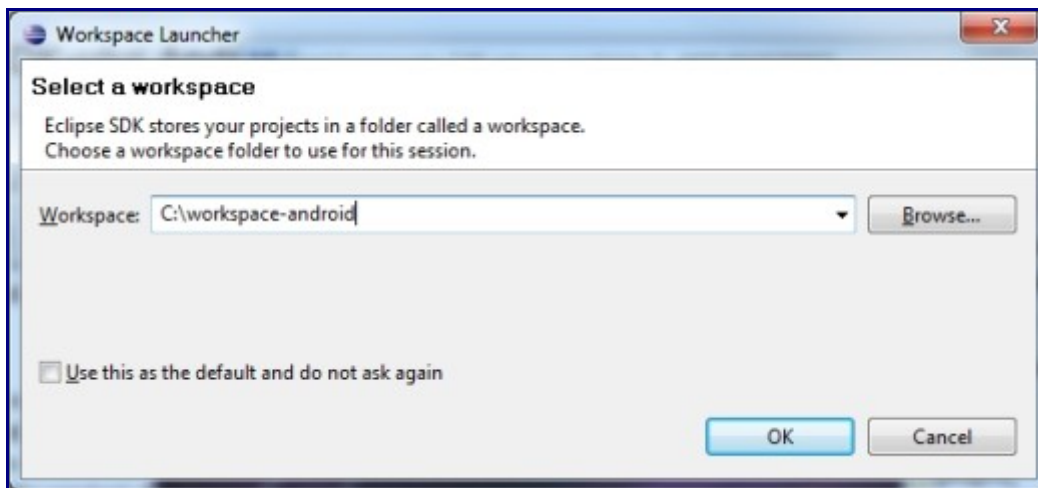


Tot din *SDK Manager* se merge în *Virtual Devices* si se selectează *New*. Apoi se alege configuratia pentru test. Exemplul de mai jos este pentru un telefon Galaxy S.



### **Pasul 5. Instalarea plug-in-ului Eclipse**

La pornirea Eclipse se recomanda folosirea unui workspace dedicat aplicatiilor Android:



Odată pornit Eclipse, pentru descarcarea automata a pluginului si pentru instalarea lui se selecteaza din meniul Help opțiunea Install New Software. Apoi se apasa pe butonul Add, iar în fereastra nou deschisa se adauga <https://dl-ssl.google.com/android/eclipse/> în campul Location si un nume reprezentativ în câmpul Name (ex. ADT Plugin).

Apoi se va lua lista pachetelor disponibile, selectand Developer Tools vor fi selectate toate.



După acceptarea licentelor, Eclipse va verifica cerintele, va descarca fisierele necesare si isi va instala plugin-ul. Este posibil sa fie nevoie sa confirmati instalarea plugin-ului dintr-o locatie care nu este cunoscută de Eclipse. Dacă apar probleme, se mai poate încerca adaugarea locatiei nesecurizate de descarcare a plugin-ului, adica acelasi link, dar cu http în loc de https: <http://dl-ssl.google.com/android/eclipse/>.

Pentru a activa noul plug-in este nevoie de o repornire a aplicatiei Eclipse.



## 5. Aplicatii rulate pe emulatorul virtual

### 5.1 Aplicatia " Hello,Android "

Codul este descris in urmatoarele randuri :

```

package com.example.helloandroid;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class HelloAndroid extends Activity {
    /** apelata cand prima activitate este creata. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

    TextView tv = new TextView(this);
    tv.setText("Hello, Android");
    setContentView(tv);
}
}

```

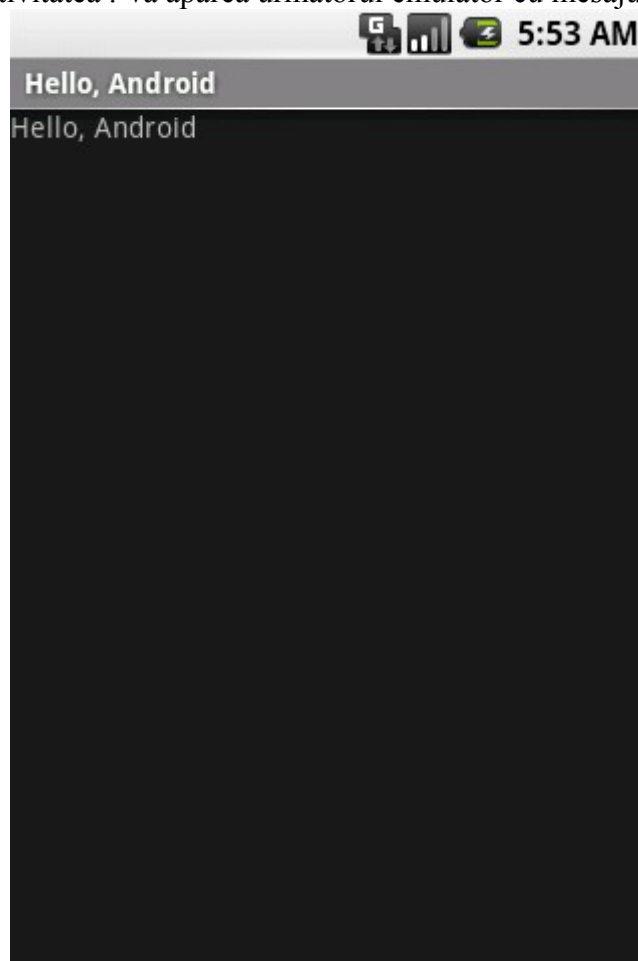
O interfata Android este alcatuita din obiecte numite "Views" organizate ierarhic . Un "view" este un obiect desenat folosit in layout -urile interfetei utilizatorului precum o imagine sau un buton sau in acest caz un text . Fiecare din aceste obiecte este o subclasa a clasei "View" iar subclasa care manipuleaza text-ul este "TextView".

In codul de mai sus, se creaza un TextView cu clasa constructor, care accepta o instanta "Context" drept parametru.

Pentru a rula aplicatia se fac urmatorii pasi :

1. Selectati Run > Run .
2. Selectati "Android Application".

Plugin-ul pentru Eclipse creaza in mod automat o noua configuratie de rulare si lanseaza un emulator Android. Cand emulatorul este boot-atat , se instaleaza aplicatia si se lanseaza in prim plan activitatea . Va aparea urmatorul emulator cu mesajul "Hello,Android" :



## 5.2 Trimiterea unui SMS din o aplicatie Android

Codul de mai jos realizeaza trimiterea unui SMS cu textul "Salut, text trimis din

"cod":

```
SmsManager m = SmsManager.getDefault();
String destination = "0728890520";
String text = "Salutare, functioneaza perfect.!";
m.sendMessage(destination, null, text, null, null);
```

### 5.3 Rularea unui fisier MP3 in o aplicatie Android

Urmatoarea metoda realizeaza rularea unui fisier MP3 :

```
private void playSound(int dResource) {
    try{
        if ( m_objMediaPlayer != null ) {
            m_objMediaPlayer.stop();
            m_objMediaPlayer = null;
        }
        m_objMediaPlayer = MediaPlayer.create(this, dResource);
        m_objMediaPlayer.setLooping(false);
        m_objMediaPlayer.start();
    } catch( Exception ex ) {
        Log.e( "Oops in playsound", "", ex );
    }
}
```

### 5.4 Aplicatie Android care face telefonul sa vibreze

```
Vibrator v = (Vibrator)
getSystemService(Context.VIBRATOR_SERVICE);
// 1. Vibreaza 1000 millisekunde
long milliseconds = 1000;
v.vibrate(milliseconds);
// 2. Vibreaza cu "pattern" timp de 500msn, si 500ms de 5 ori
5 times
long[] pattern = { 500, 300 };
v.vibrate(pattern, 5);
```

### 5.5 Activarea device-ului wi-fi. Activarea semanlului pentru scanarea de retele wi-fi

```
private WifiManager wifiManager;
// se creeaza variabila de tip private WifiManager
    public void onCreate(Bundle icicle) {
        // se creeaza metoda onCreate
        .....
        wifiManager=(WifiManager) this.getSystemService(Context.WIFI_SE
RVICE);
```

```
//se obtine serviciul Wifi din sistem
if(wifiManager.isWifiEnabled()){
//Se verifica starea curenta a wifi-ului, on sau off
// Se activeaza sau dezactiveaza wifi-ul
wifiManager.setWifiEnabled(false);
}
else{
    wifiManager.setWifiEnabled(true);
}
}
```

Pentru fiecare bloc de cod este explicata functionarea folosind liniile de comentariu.



## 6. Bibliografie

<http://www.wired.com/>

<http://developer.android.com/guide/index.html>

<http://www.codeproject.com/Articles/102065/Android-A-beginner-s-guide>

<http://www.ibm.com/developerworks/opensource/tutorials/os-eclipse-android/>