

Dispozitive și Aplicații Mobile

Cursul 1

Prezentare disciplină

- Obiective
- Elemente de conținut
- Modalitatea de evaluare

Objective

- Utilizarea eficientă a tehnologiilor mobile în cadrul societății informaționale actuale
- Asimilarea conceptelor specifice dezvoltării aplicațiilor destinate dispozitivelor mobile
- Însușirea modelului de programare pentru platforma Android
- Dezvoltarea capacității studenților de rezolvare a problemelor practice prin dezvoltarea de aplicații eficiente pentru platforma Android

Elemente de conținut

- Introducere:
 - Dispozitive mobile (DM)
 - Sisteme de operare pentru DM
 - Aplicații mobile
- Programarea aplicațiilor Android
 - Instrumente de dezvoltare
 - Interfața utilizator
 - Accesul la rețea
 - Stocarea persistentă a datelor
 - Grafică bidimensională
 - Introducere servicii, furnizori de conținut și receptori de mesaje
 - Utilizarea Google Maps în aplicații
 - Publicarea aplicațiilor în magazinul virtual

Sumar

- Dispozitive mobile
- Modelul de programare Android
- Instrumente de dezvoltare
- Structura aplicațiilor Android; componente
- Structura unui proiect
- Activități

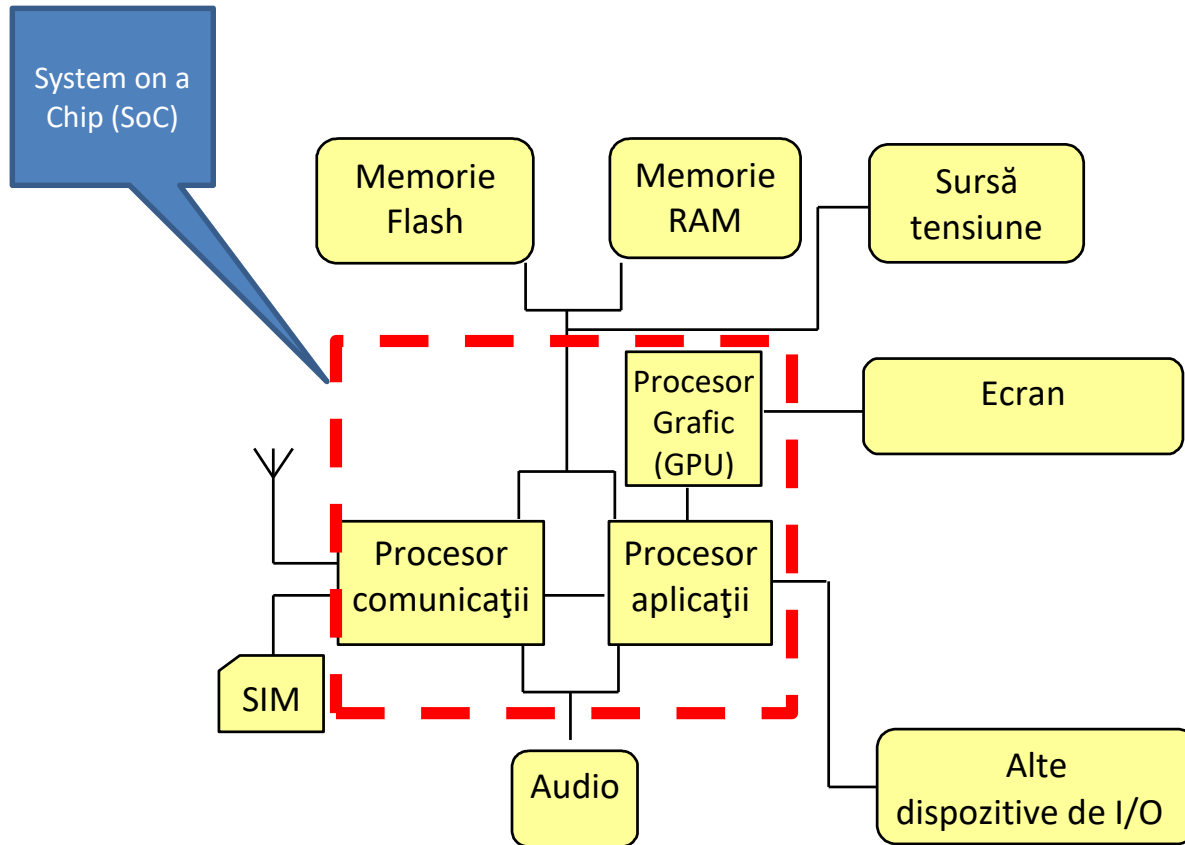
Dispozitive mobile (DM)

- Telefoane mobile
- Smartphone-uri
- Tablete
 - cu/fără suport telefonie
- PDA
 - fără suport telefonie
- Dispozitive portabile (*wearable*)
 - cu/fără suport telefonie

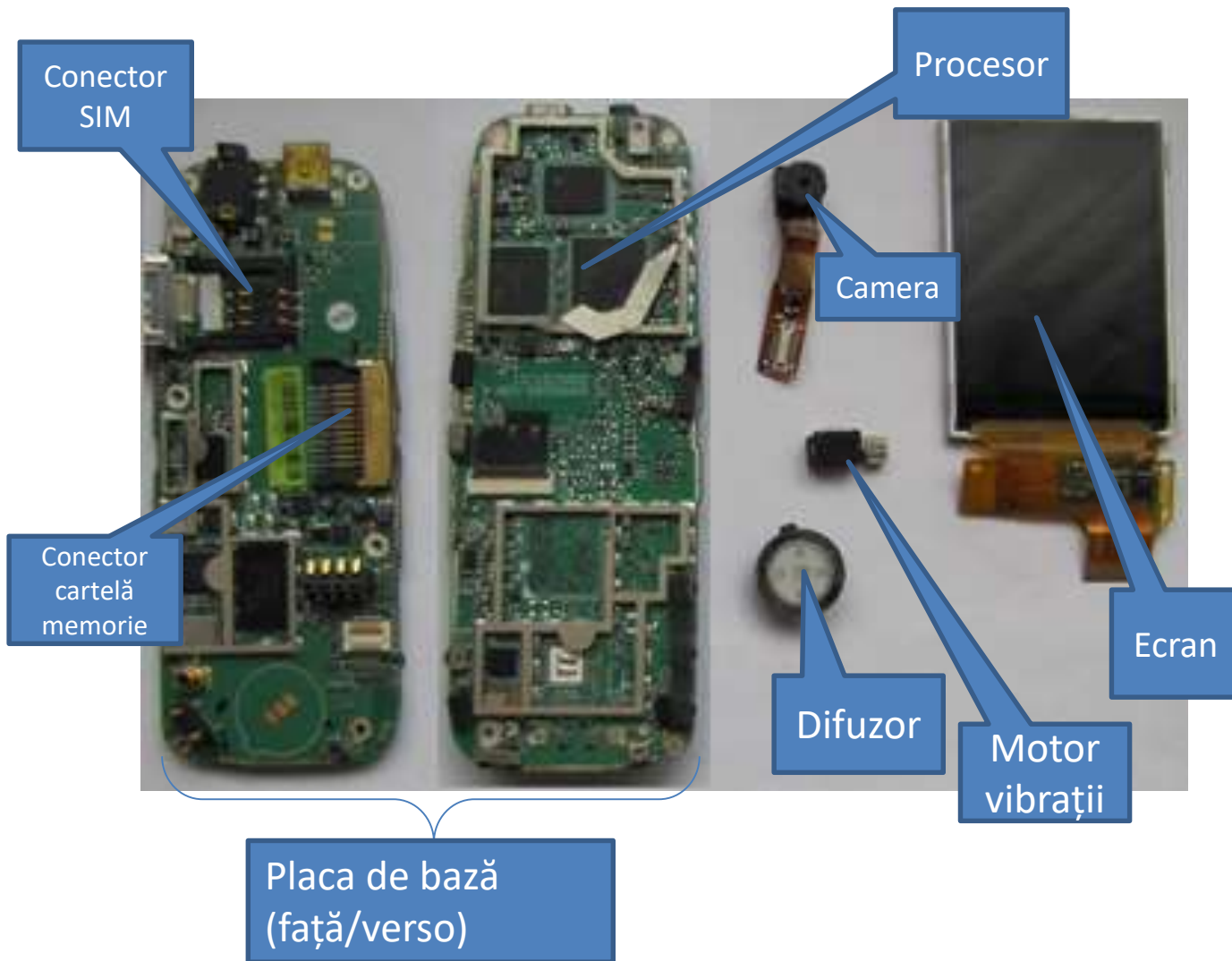
DM Limitări (vs. PC)

- Autonomie (baterie)
- Fragmentare
- Bandă de transfer
 - acoperirea
- Modalități de interacțiune
- Dimensiuni
 - ecran
- Putere de calcul
- Memorie (RAM și ROM)

Dispozitive mobile



Dispozitive mobile



Dispositivo mobile



Sisteme de operare

- Sisteme de operare proprii
- Sisteme de operare smartphones/tablete/portabile
 - Posibilitatea dezvoltării de aplicații pe baza unui SDK

Sisteme de operare

- **Android** (Google)
- Tizen (Tizen Association)
- **iOS** (Apple)
- Linux Mobile
- **Windows 10 Mobile/Windows Phone/Windows CE/Windows Mobile** (Microsoft)

Aplicații mobile

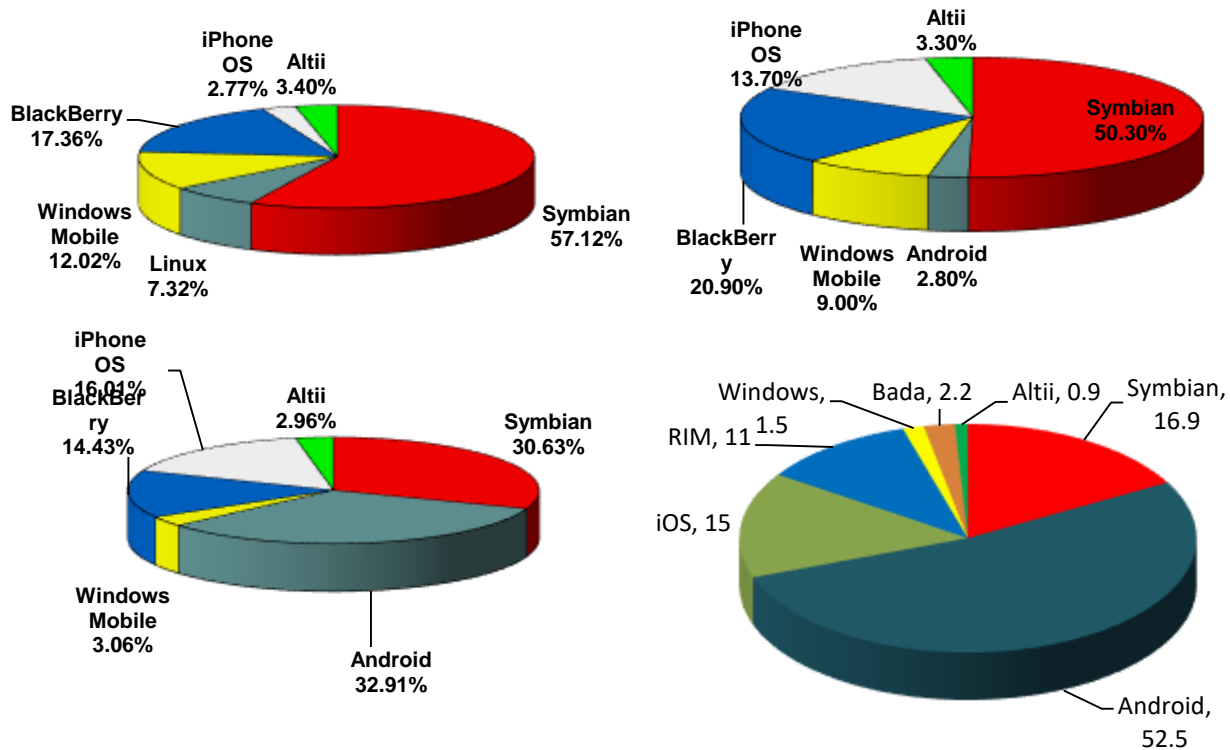
- Aplicații destinate dispozitivelor mobile
- Implementare
 - Aplicații Native
 - Aplicații Hibride
 - Cod binar interpretabil sau compilat JIT
 - Utilizează un nivel intermediar
 - Aplicații Web mobile
- Cu/fără acces la rețea

ANDROID

Android

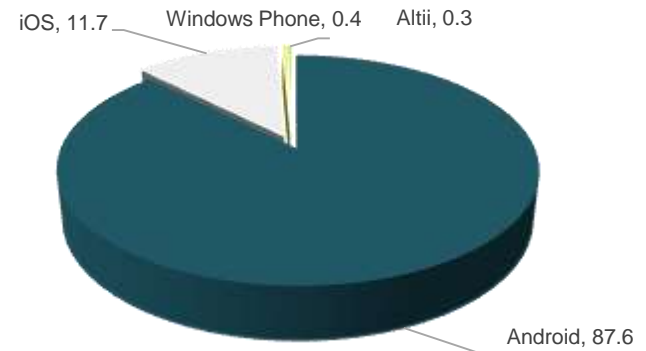
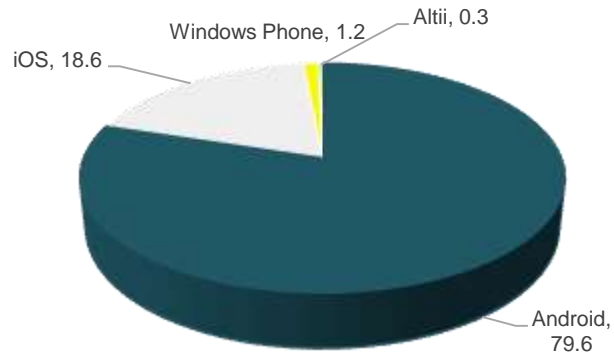
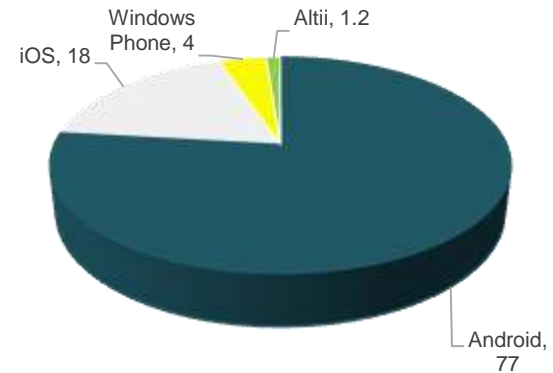
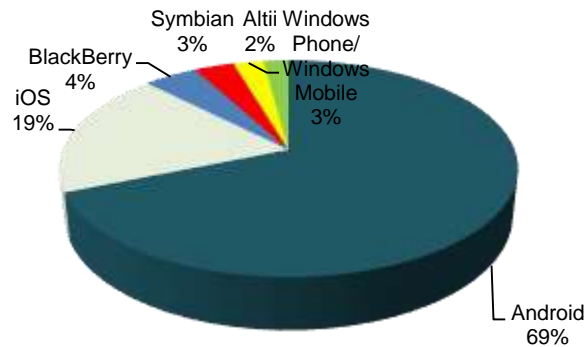
- Open Handset Alliance
- Proiect inițiat de Google
- Nucleul Linux 2.6.x/3.x
- Aplicații bazate pe Java/Kotlin
- Disponibilitate
 - Telefoane mobile
 - Tablete
 - Dispozitive portabile
 - Televizoare/STB
 - Auto

Smartphone (2008-2011)



Sursa: Canals/Gartner

Smartphone 2012/13/15/16+



<http://ctstech.net/blog/2013/02/14/idcs-smartphone-stats-for-4q-2012-and-a-review-of-their-mobile-os-share-prediction-for-2015/>
<http://gadgets.ndtv.com/mobiles/news/windows-phone-grows-104-percent-year-over-year-in-q4-2013-abi-research-478672>
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

De ce Android?

- Numărul mare de instalări existente
- Limbajul Java
- Posibilitatea de testare a aplicațiilor direct pe telefonul mobil
- Varietatea dispozitivelor mobile
- Costuri mai mici pentru dezvoltatori
- Magazinul virtual Google Play

ANDROID: MODELUL DE PROGRAMARE

Modelul de programare

- Nucleul Linux
 - Biblioteci native C
- Limbaje de programare
 - Java, Kotlin
 - C/C++ (interfața de programare nativă)
- Mașină virtuală proprie

Medii de execuție

- **ART – Android RunTime**
 - Mediul de execuție curent al aplicațiilor Android
 - Începând cu Android 4.4
 - Compilare înainte de execuție
- **Mașină virtuală proprie (Dalvik VM)**
 - Cod binar executabil incompatibil Java SE
 - Fișiere *dex*
 - Fiecare aplicație rulează într-un proces separat
 - Compilarea JIT

Facilități de programare

- Interfața utilizator
- Baze de date
 - SQLite
- Media API
 - Audio
 - Video
- Camera
- Grafică 2D, Animație
- Grafică 3D (OpenGL)
- Informații personale
- Rețea și comunicații
 - Socket
 - HTTP
 - Bluetooth
 - NFC
- Telefonie
- Senzori
 - GPS
 - Accelerometru
 - Giroscop
 - Busolă

INSTRUMENTE DE DEZVOLTARE

Instrumente de dezvoltare

- Java SE Development Kit (JDK)
- Android SDK
- Mediu de dezvoltare integrat (IDE)

Medii de dezvoltare integrate

- Android Studio
 - Furnizat de Google
- Eclipse
 - plugin: Android Development Toolkit (ADT)
 - fără suport oficial
- IntelliJ IDEA

Instrumente de dezvoltare

- Visual Studio + Xamarin
- Dezvoltare
 - Multiplatformă
 - Nativă
- Limbaje de programare
 - C#
 - F#
- Bazat pe biblioteci .Net (Mono) și SDK nativ (Android, iOS etc.)

Dezvoltare multiplatformă

- Apache Cordova
 - Adobe PhoneGap
 - Ionic
- Appcelerator Titanium
- React Native
- NativeScript
- Xamarin

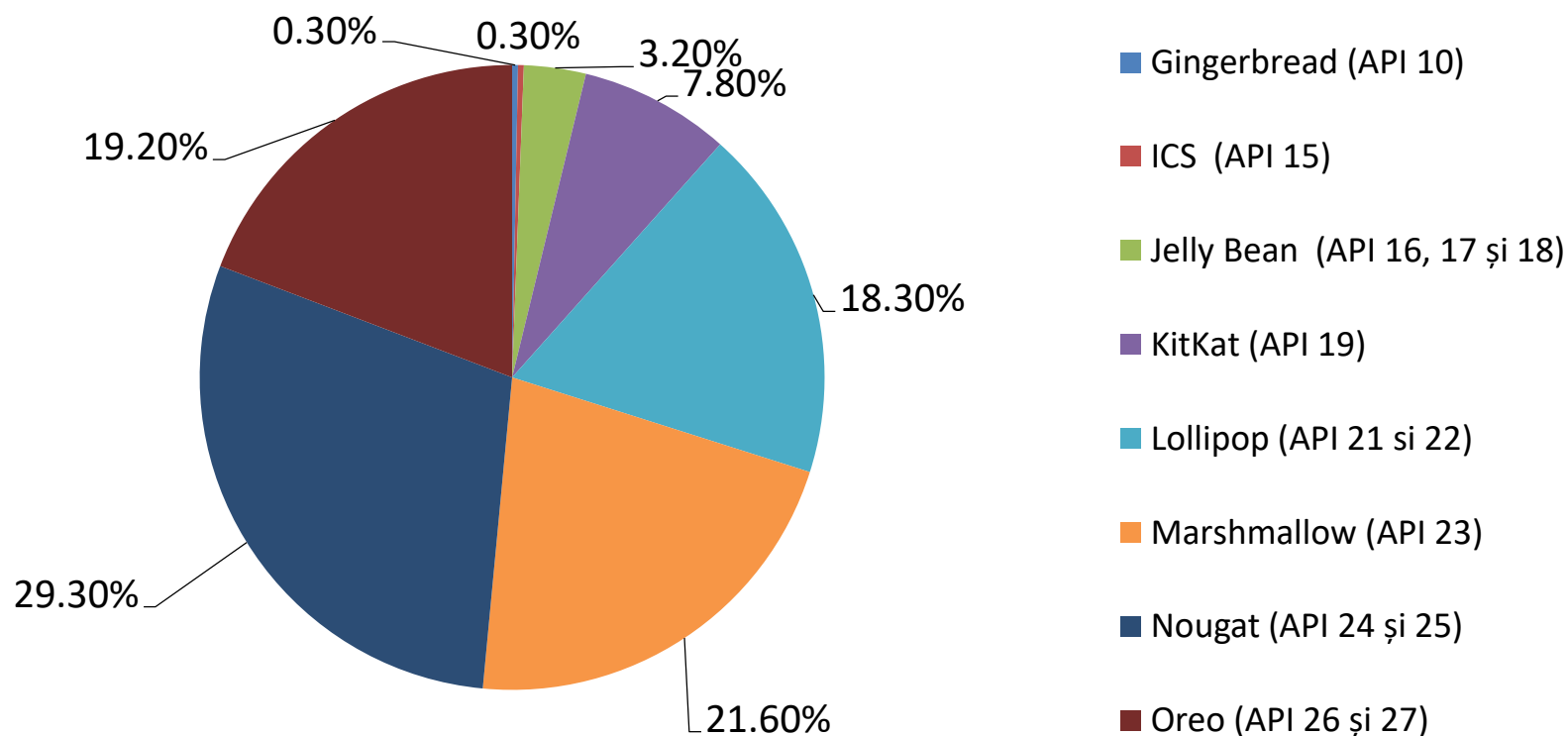
API Android

Versiuni și nume de cod	Nivel API
Android 2.2 (Froyo)	8
Android 2.3.3 – 2.3.7 (Gingerbread)	10
Android 3.x (Honeycomb)	11-13 (tablete)
Android 4.0.x (Ice Cream Sandwich)	14, 15
Android 4.1, 4.2, 4.3 (Jelly Bean)	16, 17, 18

API Android

Versiuni și nume de cod	Nivel API
Android 4.4, 4.4W (KitKat)	19, 20
Android 5.0, 5.1.x (Lollipop)	21, 22
Android 6.0 (Marshmallow)	23
Android 7.0, 7.1 (Nougat)	24, 25
Android 8.0, 8.1 (Oreo)	26, 27
Android 9.0 (Pie)	28

Distribuție versiuni Android



Sursa: <https://developer.android.com/about/dashboards/index.html>

Ultima actualizare: 28 Septembrie 2018

Android SDK

- Biblioteci și resurse specifice fiecărei platforme Android
- Resurse și imagini emulatoare
- Instrumente pentru compilare și generarea conținutului binar executabil
 - Cod sursă
 - Resurse

Android SDK

- **Build-tools**

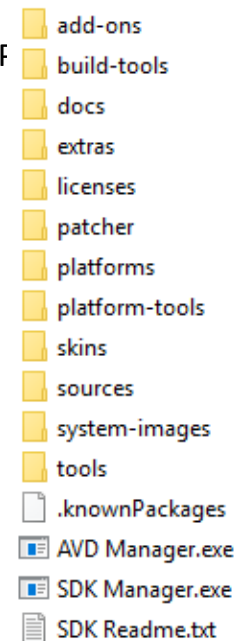
- Director: *sdk/build-tools/versiune/*
- Instrumente
 - aapt – compilare resurse, generarea fișierului R.java, fișiere AF
 - dx - conversie cod binar Java la cod binar Dalvik

- **Platform-tools**

- Specifice fiecărei platforme
- Director: *sdk/platform-tools*
- Instrumente:
 - adb – comunicarea cu dispozitivele Android
 - sqlite3

- **SDK Tools**

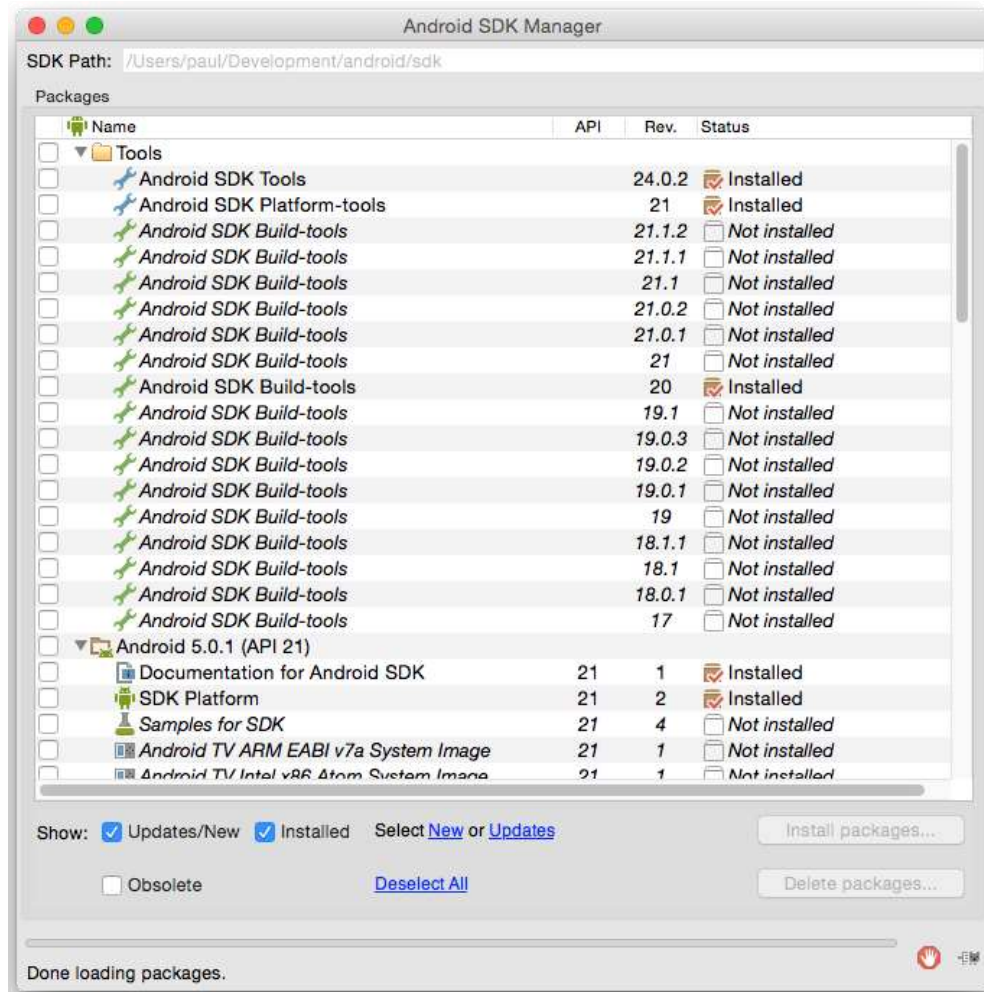
- Independente de platformă
- Subdirector: *sdk/tools*
- Instrumente:
 - script-uri ant pentru obținerea pachetului binar al aplicației
 - monitor(ddms)
 - emulator-arm, emulator-x86



Android SDK Manager

- Gestiunea platformelor și a instrumentelor necesare
- Acces direct sau din mediul de dezvoltare
- Platforma
 - Biblioteci
 - Cod sursă
 - Documentație
 - Imagini emulator

Android SDK Manager



Android Virtual Device (AVD)

- Dispozitive virtuale Android
 - Emulatoare
- Caracteristici
 - Procesor (ARM, x86_x64)
 - Ecran (rezoluție ,dimensiune)
 - Memorie (RAM, internă persistentă, externă),
 - Versiune API
 - Camere
- Comunicare prin aplicația *adb.exe*

Emulatoare

- Emulare
 - ARM
 - X86, x64
 - Necesită Intel HAXM și procesor cu suport virtualizare
- API
 - Standard
 - Google API
 - Google Play

Android Debug Bridge (ADB)

- Asigură comunicarea între mașina de dezvoltare și dispozitivele Android (reale sau virtuale)
- Componente
 - Client
 - Permite transmiterea de comenzi
 - Server
 - Mașina de dezvoltare
 - Dispozitivul Android

Exemple de comenzi


- adb devices
- adb kill-server
- adb install
- adb pull
- adb push
- adb shell

AVD Manager

- Dispozitive virtuale
 - Definire
 - Modificare
 - Ștergere
- Accesat direct sau din mediul integrat de dezvoltare
- Pot fi definite oricâte dispozitive virtuale
 - Se va alege unul la lansarea în execuție a aplicației


AVD Manager


Virtual Device Configuration

 **Android Virtual Device (AVD)**
Verify Configuration

AVD Name:

AVD Id: Nexus_One

 3.7" WVGA (Nexus One) 3.4" 480x800 hdpi Change...

 KitKat Android 4.4.2 armeabi-v7a Change...

Startup size and orientation

Scale:

Orientation: ☒ Portrait ☐ Landscape

Camera

Front:

Back:

Network

Speed:

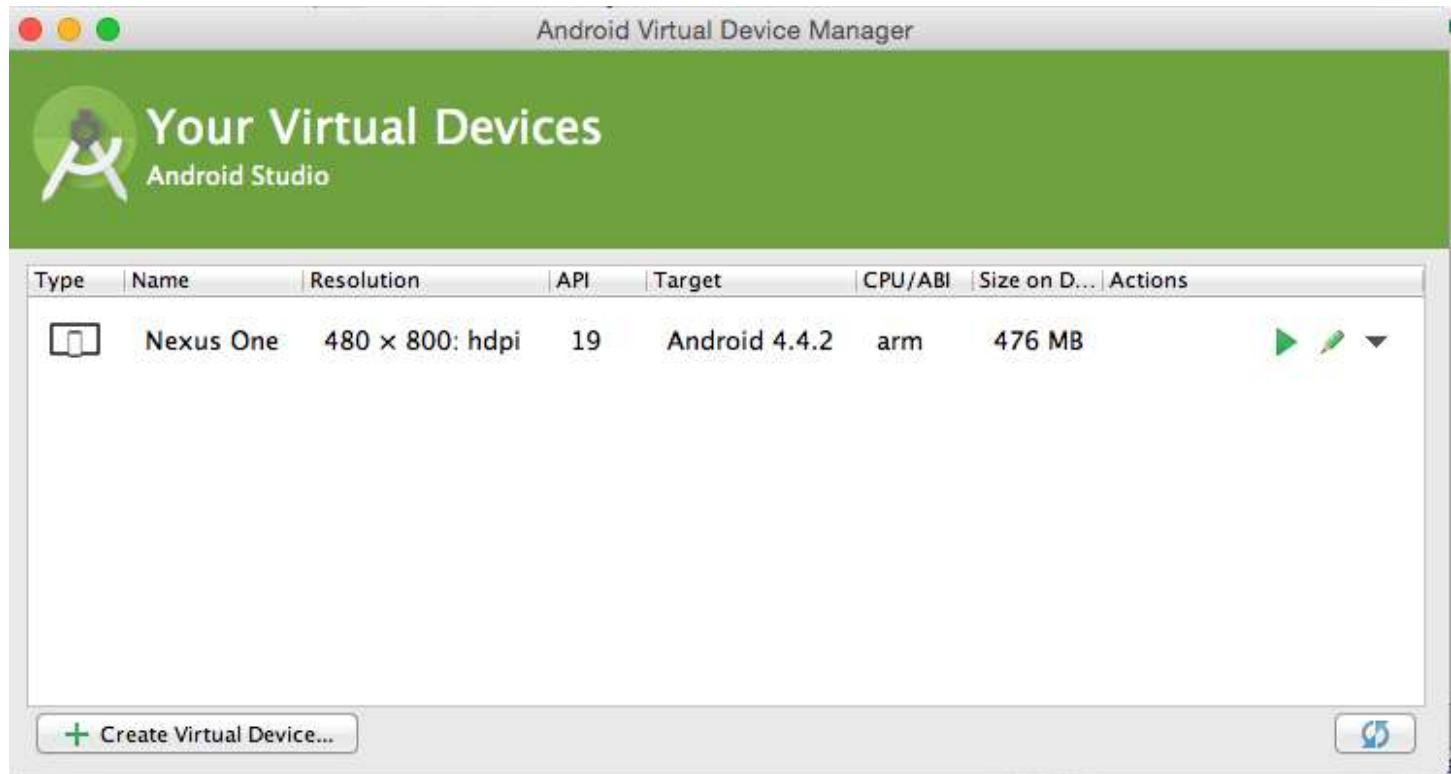
Hide Advanced Settings

AVD Name

The name of this AVD.

Cancel Previous Next Finish

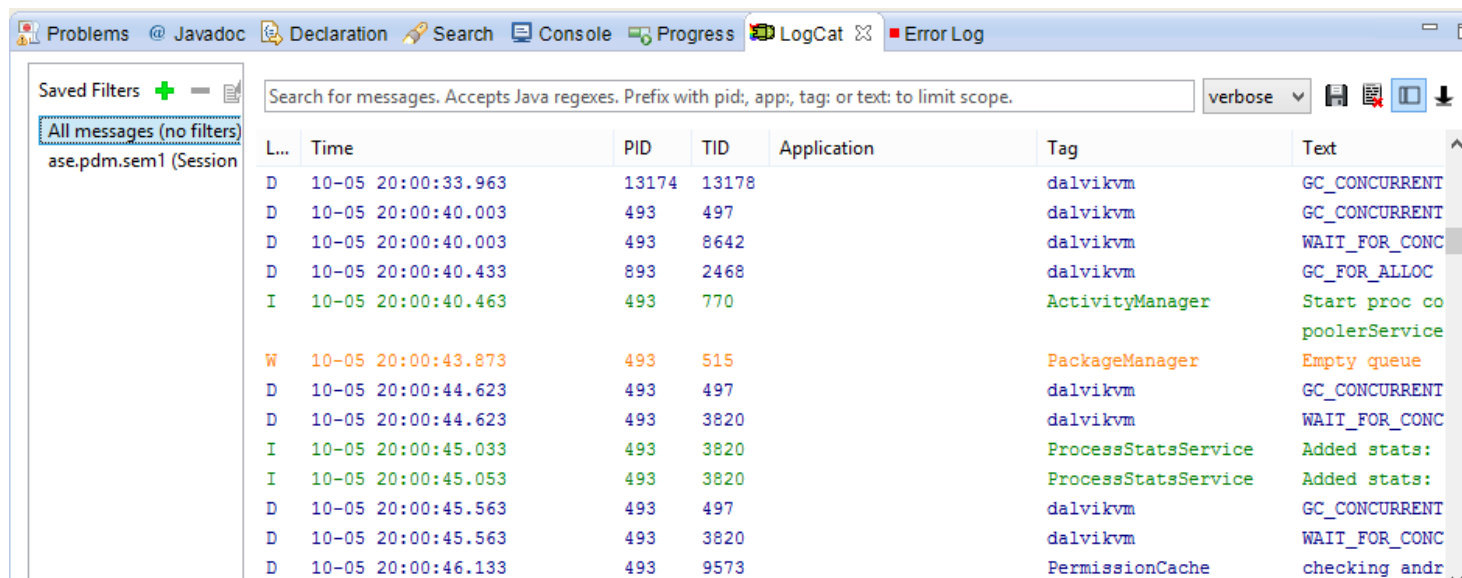
AVD Manager



Consola de mesaje (LogCat)

- Afișează mesaje transmise din aplicații
 - Utilizator
 - Sistem
- Mesaje
 - Avertizare (w)
 - Depanare (d)
 - Eroare (e)
 - Informare (i)
 - Informare detaliată (v)
 - Eroare excepțională (wtf)

Consola de mensaje (LogCat)



The screenshot shows the LogCat window in Android Studio. The top bar includes tabs for Problems, Javadoc, Declaration, Search, Console, Progress, LogCat, and Error Log. The LogCat tab is active, displaying a search bar with the text "Search for messages. Accepts Java regexes. Prefix with pid, app, tag, or text: to limit scope." and a dropdown menu set to "verbose". Below the search bar is a table of log messages. The table has columns for L... (Log level), Time, PID, TID, Application, Tag, and Text. The messages are filtered by "All messages (no filters)" and "ase.pdm.sem1 (Session)".

L...	Time	PID	TID	Application	Tag	Text
D	10-05 20:00:33.963	13174	13178		dalvikvm	GC_CONCURRENT
D	10-05 20:00:40.003	493	497		dalvikvm	GC_CONCURRENT
D	10-05 20:00:40.003	493	8642		dalvikvm	WAIT_FOR_CONC
D	10-05 20:00:40.433	893	2468		dalvikvm	GC_FOR_ALLOC
I	10-05 20:00:40.463	493	770		ActivityManager	Start proc co poolerService
W	10-05 20:00:43.873	493	515		PackageManager	Empty queue
D	10-05 20:00:44.623	493	497		dalvikvm	GC_CONCURRENT
D	10-05 20:00:44.623	493	3820		dalvikvm	WAIT_FOR_CONC
I	10-05 20:00:45.033	493	3820		ProcessStatsService	Added stats:
I	10-05 20:00:45.053	493	3820		ProcessStatsService	Added stats:
D	10-05 20:00:45.563	493	497		dalvikvm	GC_CONCURRENT
D	10-05 20:00:45.563	493	3820		dalvikvm	WAIT_FOR_CONC
D	10-05 20:00:46.133	493	9573		PermissionCache	checking andr

Consola de mesaje (LogCat)

- Clasa `android.util.Log`
- Metode statice asociate tipurilor de mesaje:
 - `e()`, `w()`, `i()`, `d()`, `v()`, `wtf()`
- Parametri:
 - Identificator sursă mesaj (`String`)
 - Numele clasei, aplicației, activității etc.
 - Posibilitatea de filtrare
 - Mesajul care va fi afișat (`String`)
- Metoda statică generală
 - `println()`
 - În plus, primul parametru include tipul mesajului
 - `Log.ASSERT`, `Log.ERROR`, `Log.INFO` etc.

Consola de mesaje (LogCat)

- `Log.i("Activitate1", "Mesaj de informare");`
- `Log.println(Log.ASSERT, "Activitate 1",
"Aserțiune invalida!");`

STRUCTURA APLICAȚIILOR ANDROID

Structura aplicațiilor Android

- Includ una sau mai multe componente
- Componentele sînt înregistrate de sistem
 - Pot fi și componente locale
- Componentele pot fi activate
 - Local, în cadrul aplicației
 - Global, la nivelul sistemului

Componente de bază ale aplicațiilor Android

- **Activități**
 - clasa de bază *android.app.Activity*
- **Servicii**
 - clasa de bază *android.app.Service*
- **Furnizori de conținut**
 - clasa de bază *android.content.ContentProvider*
- **Receptori de mesaje**
 - clasa de bază *android.content.BroadcastReceiver*
 - **mesaje**
 - clasa de bază *android.content.Intent*

Activități

- Asociate ferestrelor aplicației
- O aplicație poate avea una sau mai multe activități
 - O singură activitate principală
- Componente vizuale asociate
 - Derivate din clasa View

Activități



Servicii

- Rutine care rulează în paralel cu firul principal
- Nu prezintă interfață grafică
- Permit derularea unor acțiuni în fundal fără a bloca:
 - firul principal de execuție
 - Interacțiunea cu aplicațiile

Furnizori de conținut

- Suport pentru partajarea datelor între aplicații
- Datele partajate sunt stocate în diferite surse de date (fișiere, baze de date etc.)
- Pun la dispoziție o modalitatea standard pentru accesul la date și actualizarea acestora
- Accesul se realizează printr-un URI de forma **content://**

Mesaje (obiecte de tip Intent)

- Pentru activarea componentelor se utilizează mesaje asincrone
 - încapsulate în obiecte de tip **Intent**
- Invocare componente
 - Deschidere navigator, inițiere aplicație apeluri telefonice, afișarea hărții la o anumită poziție geografică etc.
- Comunicare între componente

Recepționarea evenimentelor

- Aplicațiile pot reacționa la apariția unor evenimente la nivelul sistemului prin utilizarea claselor derivate din **BroadcastReceiver**
 - Apel telefonic, modificarea nivelului bateriei, recepționarea unui mesaj, mesaje transmise de alte aplicații etc.
- Nu prezintă interfață grafică
- O aplicație poate include mai multe componente pentru recepționarea de evenimente

STRUCTURA PROIECTELOR ANDROID

Aplicații Android cu interfață grafică

- Includ una sau mai multe activități
- Fiecare activitate are asociată o componentă vizuală părinte
 - Definită procedural, în fișiere XML
 - Definită în cod
- Includ resurse:
 - Asociațe interfeței utilizator
 - Fișiere prelucrate în cadrul aplicației

Structura unui proiect Android Studio

- Proiecte și module
- Proiect
 - Include unul sau mai multe module
- Modul
 - Fișiere sursă, resurse, fișiere de configurare
 - Permit împărțirea proiectului în module funcționale
 - Dependente între module
 - Utilizare: biblioteci, tipuri de dispozitive etc.

Structura unui proiect Android Studio

- Fișiere sursă (**src**)
 - Fișiere sursă Java (**java**)
 - Resurse (**res**)
 - drawable
 - layout
 - menu
 - mipmap
 - raw
 - values
 - xml
- Resurse preluate ca fluxuri de date (**assets**)
- Fișier de configurare (**AndroidManifest.xml**)
- Proprietățile sistemului de compilare (proiect și modul)
 - **build.gradle**
- Fișiere generate (**gen**)
 - **R.java**

AndroidManifest.xml

- Componentele aplicației
 - declarare activități, servicii, furnizori de conținut, receptori de mesaje
 - denumirile claselor asociate
 - Proprietăți
- Versiunile SDK (minim, maxim, dorită)
 - Preluate de fișierul *build.gradle*
- Informații pachet (denumire, versiune)
- Atribute aplicație (denumire, pictograma asociată, tema, opțiuni memorie, restricții, permisiuni etc.)

AndroidManifest.xml

- Filtrele de mesaje
 - definite în cadrul aplicației/componentelor
- Permișiunile de acces
 - **<uses-permission** android:name="permisiune"/>
- Cerințe hardware și software
 - **<uses-feature** android:name="cerință"
android:required="true/false"/>

Exemple de permisiuni

Pentru ...	este necesară permisiunea <i>android.permission. ...</i>
acces la Internet/rețea	INTERNET
scriere și citire date de contact	READ_CONTACTS, WRITE_CONTACTS
scriere și citire în Calendar	READ_CALENDAR, WRITE_CALENDAR
trimitere SMS-uri, scriere și citire SMS-uri	SEND_SMS, READ_SMS, WRITE_SMS
utilizarea telefoniei	CALL_PHONE
accesare mediu extern de stocare	READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE
identificare poziție geografică	ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION

Permisuni

- API 23 (Marshmallow)
- Normale
 - Accesul este acordat automat
 - Exemple: Internet, Bluetooth, NFC, Vibrații etc.
- Cu risc/Periculoase
 - Accesul este acordat individual de către utilizator
 - Aplicațiile controlează accesul la execuție
 - Exemple: Calendar, Camera, Contacts, SMS, Location, Phone, Storage etc.

Cerințe hardware și software pentru dispozitive

- `android.hardware.camera`
- `android.hardware.camera.autofocus`
- `android.hardware.camera.flash`
- `android.hardware.nfc`
- `android.hardware.sensor.gyroscope`
- `android.hardware.Bluetooth`
- `android.software.live_wallpaper`
- `android.software.home_screen`

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ase.pdm.sem1"    android:versionCode="1"    android:versionName="1.0" >
    <uses-permission android:name="android.permission.INTERNET"/>
    <application android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity> ... </activity>
        <service>...</service>
        <provider>...</provider>
        <receiver>...</receiver>
    </application>
</manifest>
```

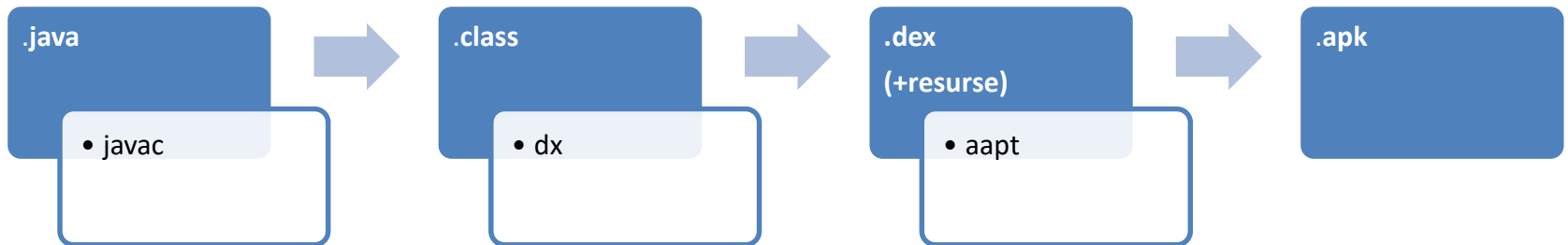

build.gradle

```
android {  
    compileSdkVersion 24  
    buildToolsVersion "24.0.4"  
  
    defaultConfig {  
        applicationId "ro.ase.pdm.myapplication"  
        minSdkVersion 16  
        targetSdkVersion 24  
        versionCode 1  
        versionName "1.0"  
    }  
    buildTypes {  
        release {  
            minifyEnabled false  
            proguardFiles getDefaultProguardFile('proguard-android.txt'),  
                           'proguard-rules.pro'  
        }  
    }  
}  
  
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:23.0.1'  
    compile 'com.android.support:design:23.0.1'  
}
```

Fișierele binare Android

- Extensia apk (Android Package)
- Conțin codul binar și resursele
- Resursele pot fi compilate sau nu
- Pe lângă resurse poate fi adăugat orice tip de fișier

Fișierele binare Android

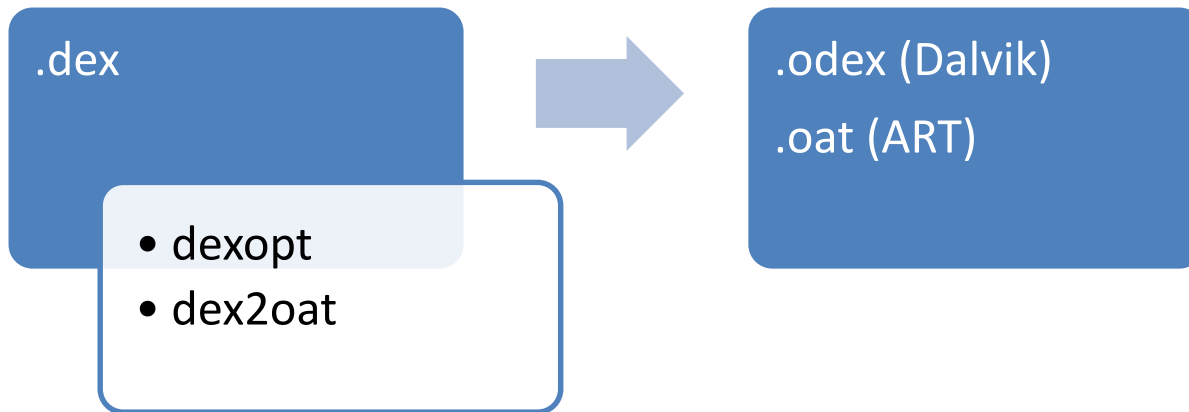


Fișierele binare Android

apk

- /META-INF
- /lib
- /res
- /assets
- AndroidManifest.xml
- classes.dex
- resources.arsc

Fișierele binare Android



ACTIVITĂȚI

Clasa Context

- Clasă abstractă definită în pachetul *android.content*
- Asigură accesul la mediul aplicației
- Acces la resurse
- Lansarea de activități noi
- Acces la servicii de sistem
- Acces la baze de date și fișiere

Clasa aplicație

- Accesul la setările și metodele ale aplicației
- Clasă derivată din **Application**
- Generată implicit
- Poate fi creată o clasă utilizator prin derivare
- Are un context asociat
 - Disponibil pe toată durată rulării aplicației
 - **getApplicationContext()**

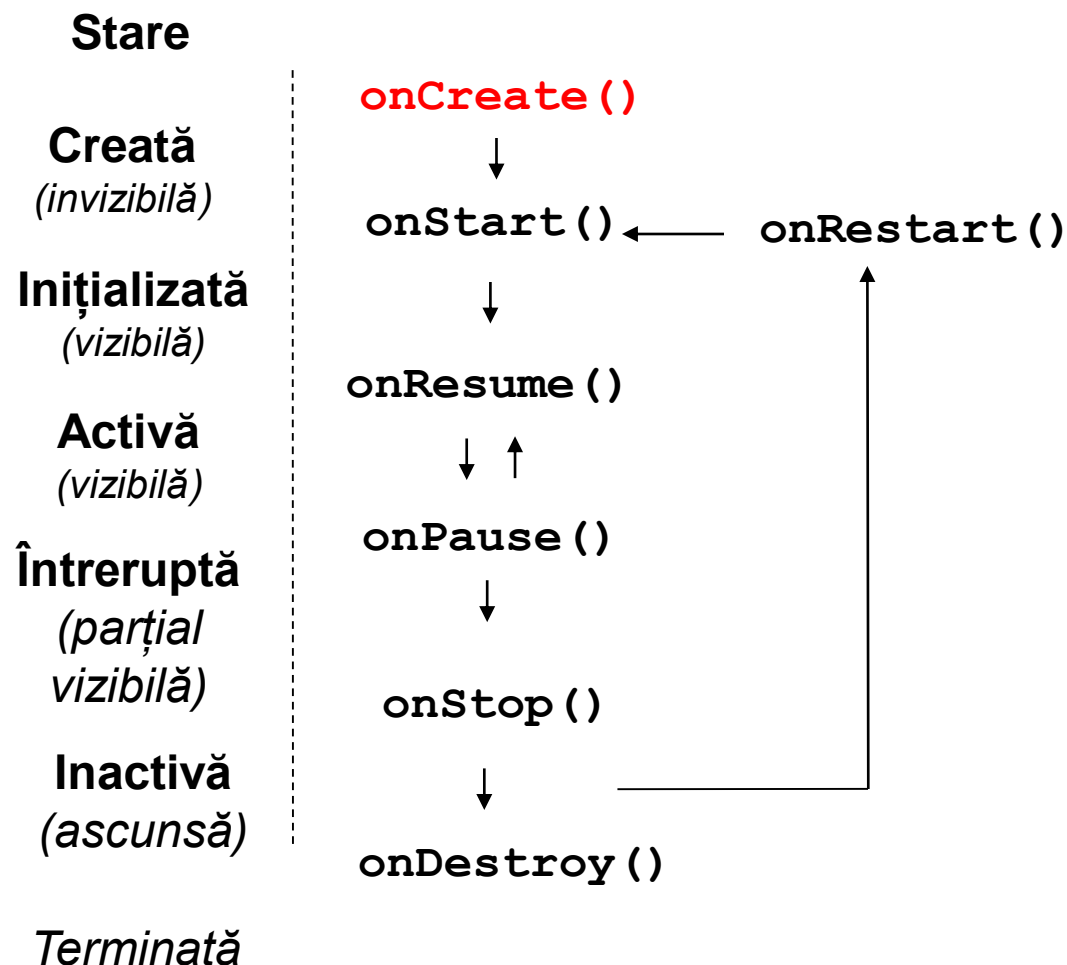
Activități

- Asociaze ferestrelor aplicației
- O aplicație poate avea una sau mai multe activități
- Stiva de activități
 - *Task-uri*
- Derivate din clasa de bază `android.app.Activity`
 - Derivată din clasa `Context` (în vârful ierarhiei)
- Contextul activității = `this`
- Fiecare obiect grafic referă contextul activității din care face parte

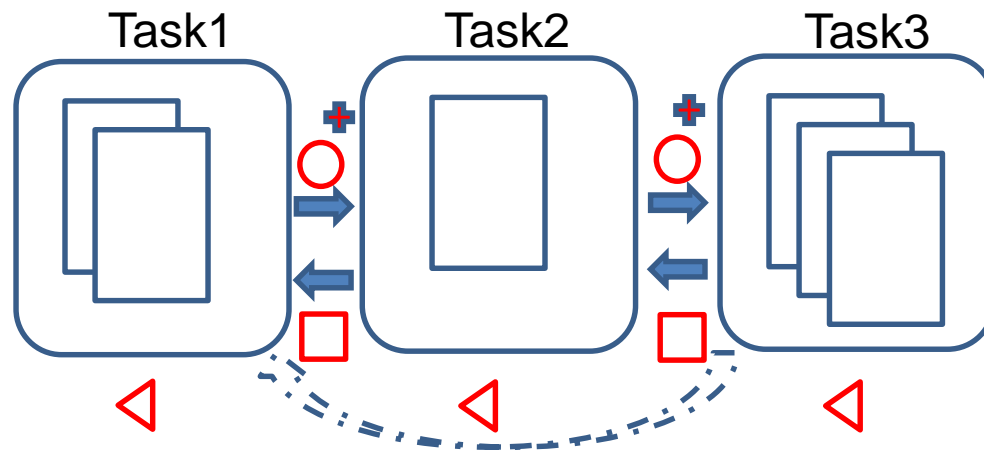
Activități

- Au o fereastră asociată
 - Reprezentarea interfeței grafice
- Dispun de un ciclu de viață
 - Mai multe stări
 - Metode cu apel invers
 - Invocate la trecerea într-o stare
 - Posibilitatea de salvare a stării activității (conținut, poziție componente vizuale, proprietăți etc.)

Ciclul de viață al activităților



Stiva de activități



- ✚ Semnifică lansarea unei noi activități din lista de aplicații recente (☐) sau din lista de programe (după apăsarea butonului Home ☐)

Activități

```
import android.app.Activity;
import android.os.Bundle;

public class TestActivity extends Activity
{
    @Override
    public void onCreate(Bundle stare) {
        super.onCreate(stare);
        //inițializare conținut vizual
        setContentView(view);
        //inițializari activitate
    }
}
```

Modificarea configurației

- Schimbarea orientării dispozitivului
 - rotire
- Activarea/dezactivarea tastaturii fizice
- Schimbare limbii utilizate
 - Setări regionale
- etc.
- Activitățile sunt recreate pentru a corespunde noilor condiții

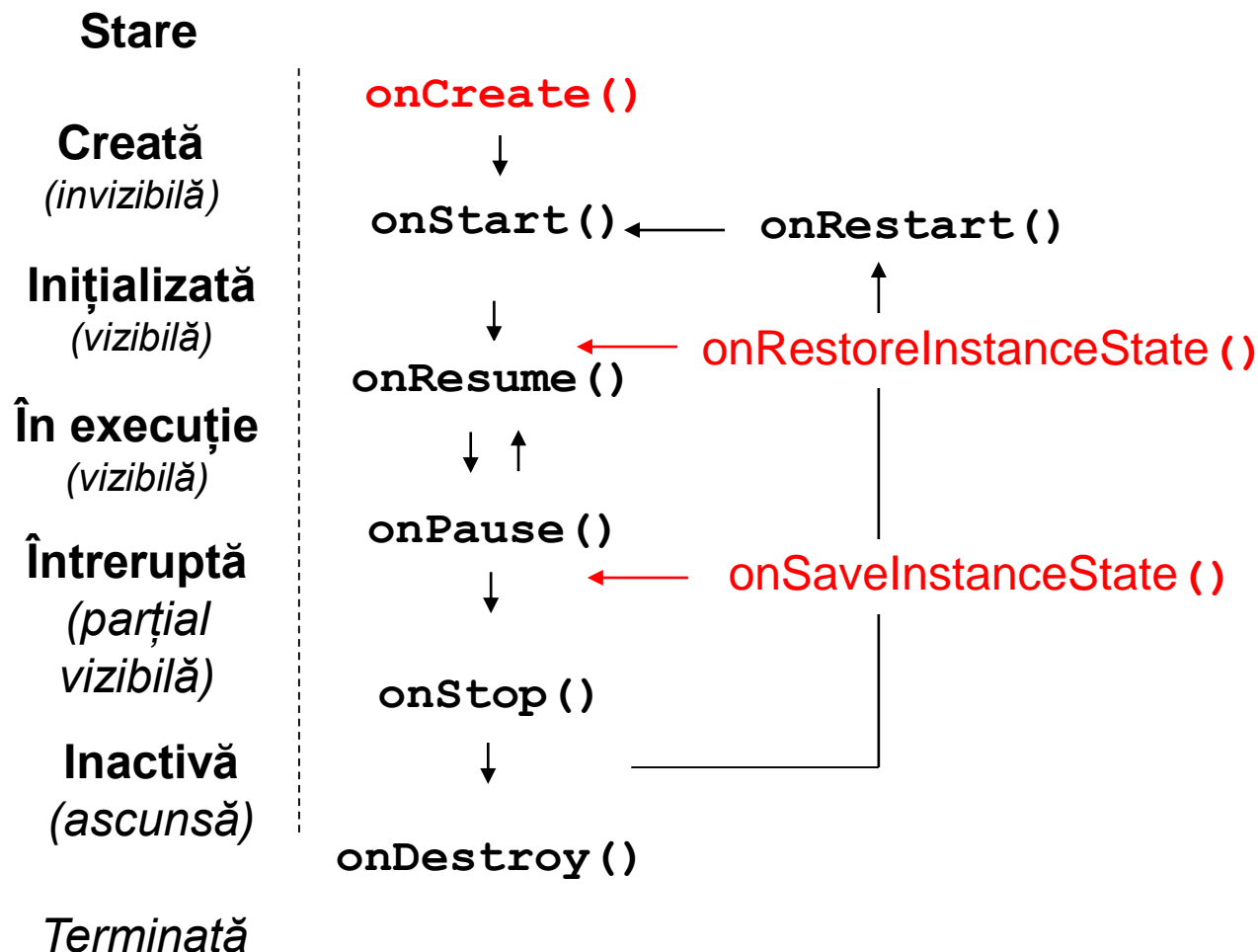
Clasa Bundle

- Permite stocarea și regăsirea de valori după o cheie de tip **String**
- Valorile stocate
 - Tipuri simple
 - Tipuri care implementează interfețele
 - **Parcelable**
 - **Serializable**
- Adăugare valori
 - **putString**(cheie, valoare_String),
 - **putInt** (cheie, valoare_int),
 - **putObject**(cheie, valoare_object),
 - **putLongArray**(cheie, valoare_long_array) etc.
- Regăsire valori
 - **getString**(cheie),
 - **getInt**(cheie),
 - **getObject**(cheie),
 - **getLongArray**(cheie) etc.

Salvarea/restaurarea stării

- Salvarea stării
 - **onSaveInstanceState**(Bundle stare)
 - Apelată la ieșirea forțată din aplicație (modificarea configurației)
- Restaurarea stării
 - **onRestoreInstanceState**(Bundle stare) *sau*
 - **onCreate**(Bundle stare)
 - *stare* poate fi null!

Ciclul de viață al activităților (2)



Proprietățile activităților în AndroidManifest.xml

<activity

android:name=".MainActivity"

android:label="@string/app_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

Bibliografie

- S. Komatineni, D. MacLean – Pro Android 4, Apress, 2012
- E-P Lim, K Siau (ed) – Advances in Mobile Commerce Technologies, Idea Group, 2003
- M. Mallick – Mobile and Wireless Design Essentials, John Wiley & Sons, 2003
- R. Meier – Professional Android 4 Application Development, Wiley, 2012
- P. Pocatilu, Programarea dispozitivelor mobile, Editura ASE, 2012
- P. Pocatilu, I. Ivan ș.a. – Programarea aplicațiilor Android, Editura, ASE, 2015
- M. Sauter – Communication Systems for the Mobile Information Society, John Wiley & Sons, 2006
- <http://developer.android.com>