Aplicatii mobile pentru afaceri

Master SIA

Lect. Octavian Dospinescu

2012

Tematica generala

- Functionalitati GSM in Android
- Primire/interceptare mesaj SMS in mod programatic
- Clasa **BroadcastReceiver** – anatomie generala
- Conceptul de **Intent**
- Drepturi specifice in AndroidManifest.xml
- Directii de viitor

Cadru de lucru - premise

- interceptarea mesajelor primite de către telefonul nostru de la un alt dispozitiv;
- iniţiativa nu mai aparţine utilizatorului nostru, ci unui terţ care îşi foloseşte propriul dispozitiv mobil;
- nu avem un control cert asupra acestei operaţiuni, ci trebuie mai degrabă „să fim pe fază” la eventuala primire a unui SMS.

Notiuni de avut in vedere

- **Intent** = un eveniment care apare in viata aplicatiei ;
- un **BroadcastReceiver** este capabil să intercepteze diverse evenimente de tip **Intent** care apar din mediul extern sau din sistemul de operare;
- putem face asocierea cu notiunea de trigger;

BroadcastReceiver – anatomie generala

- Un **BroadcastReceiver** implementează metoda abstractă **onReceive()** pentru a procesa **Intent-**urile care sosesc.
- Argumentele metodei sunt un **Context** şi un **Intent.**
- Metoda returnează **void**, dar se poate folosi metoda **setResult** pentru a pasa înapoi un anumit rezultat.

BroadcastReceiver – utilizare uzuala

- în practicã, implementarea unui BroadcastReceiver se realizeazã prin **definirea unei clase derivate** din clasa de bazã BroadcastReceiver, în care **definim metoda onReceive() în funcţie de necesitãţile aplicaţiei**

BroadcastReceiver – precizari

Spre deosebire de clasa Activity (care **are drept corespondent grafic** un formular afişat pe dispozitivul mobil), clasa **BroadcastReceiver** *nu are o reprezentare graficã vizibilã* pentru utilizator;

Ea funcţionează „în fundalul” aplicaţiei, **declanşându-se atunci când apare un Intent corespunzãtor**.

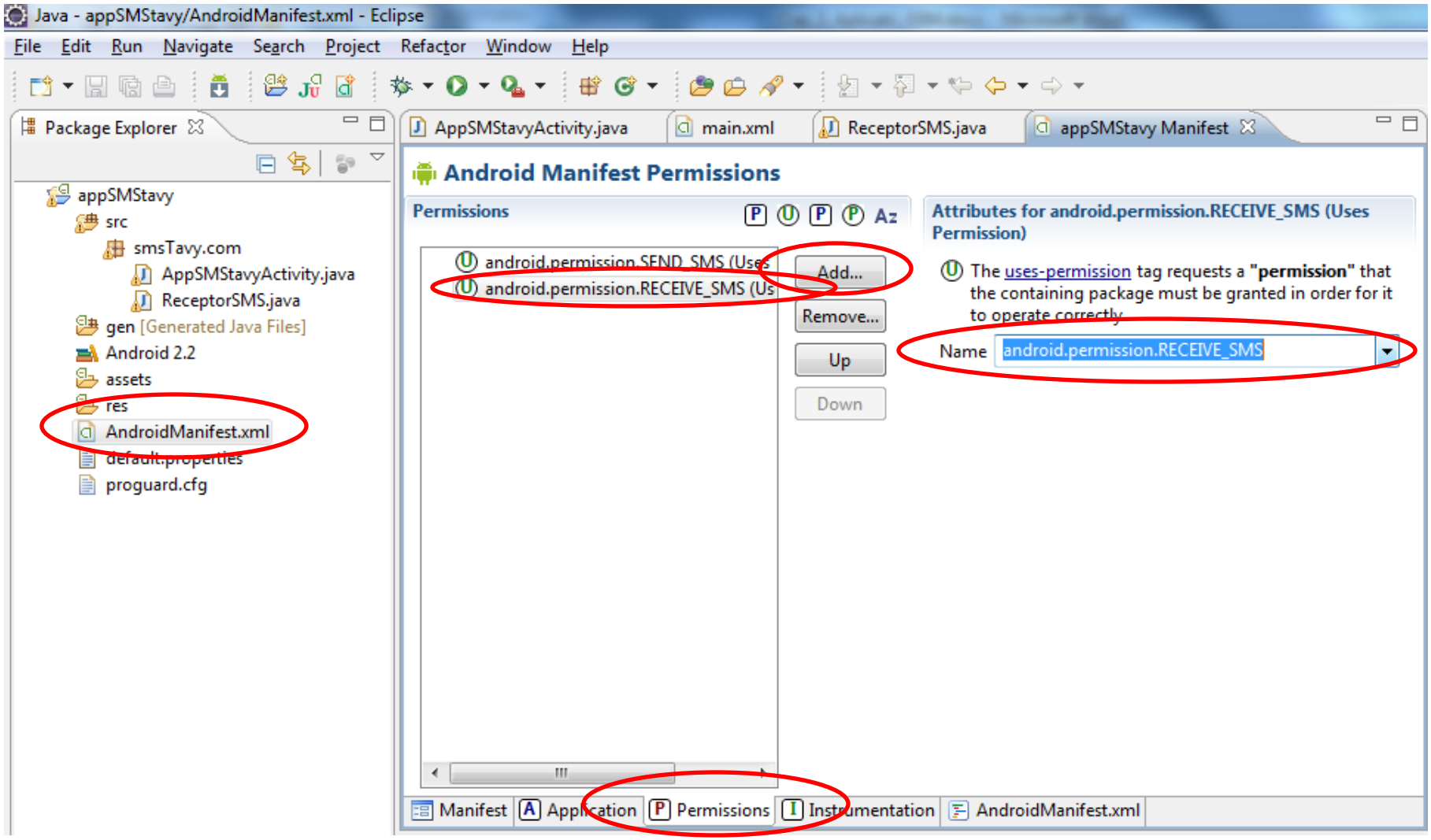
BroadcastReceiver – model de implementare

```
public class ReceptorSMS extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
                // TODO Auto-generated method stub
                Log.i("din receiver","s-a declansat metoda onReceive");
                if(intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED"))
                {
                                Log.i("din receiver", "Am primit un SMS!!!");
                                //aici urmeaza sa completam cu cod de prelucrare

                }
        }
}
```

BroadcastReceiver – precizari importante!!!

- pentru ca aplicaţia noastrã sã poatã **intercepta** mesaje SMS, este nevoie sã **specificãm acest drept** în fişierul AndroidManifest.xml printr-o **clauzã de tip <uses-permissions>** similarã cu cea care ne-a dat dreptul sã trimitem SMS-uri

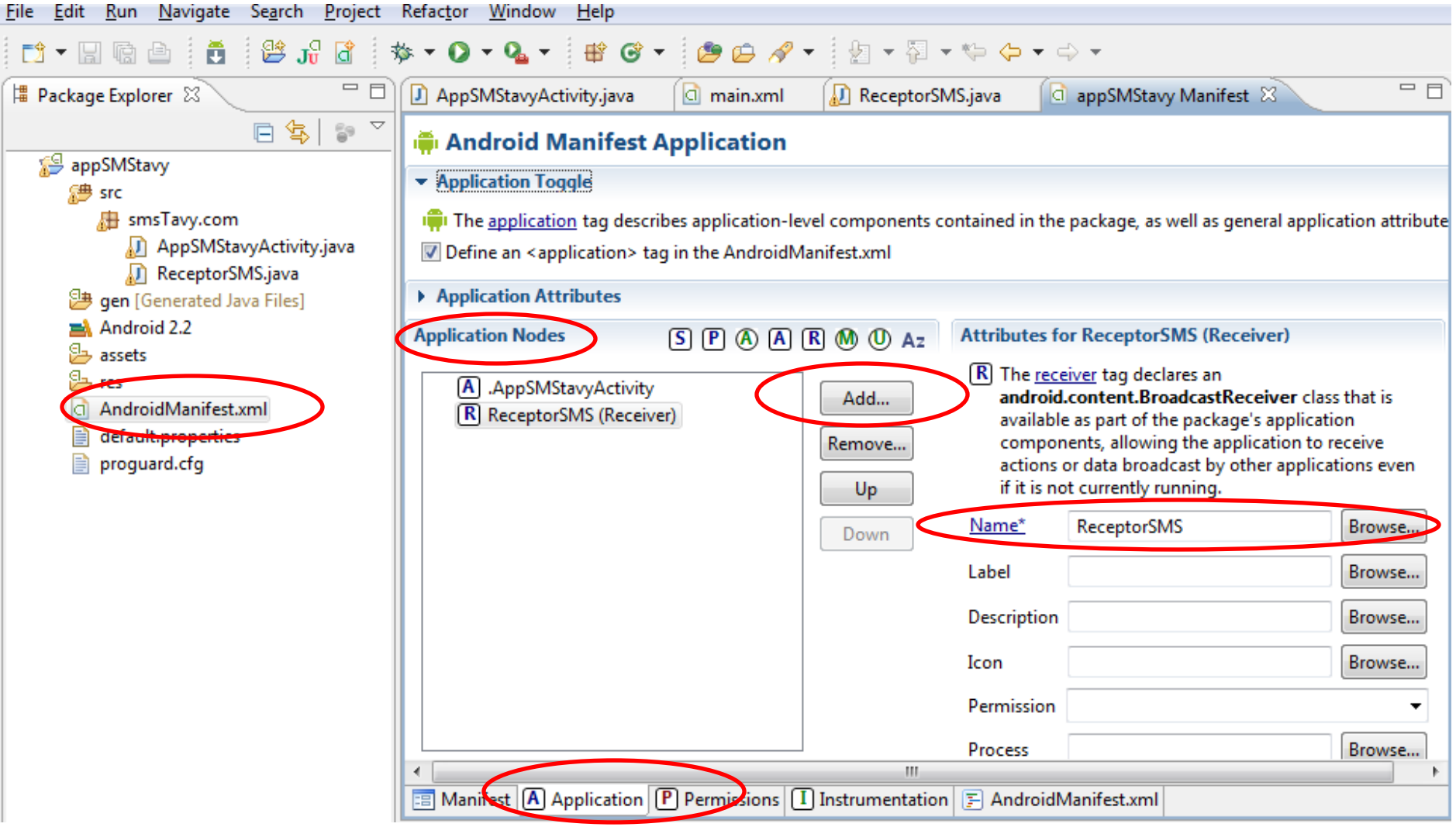
BroadcastReceiver – permisiuni!!!



BroadcastReceiver – precizari importante!!!

- pentru ca **BroadcastReceiver-ul** nostru (ReceptorSMS) să funcţioneze, el **trebuie să fie declarat de asemenea în fişierul AndroidManifest.xml**

BroadcastReceiver – declarare in AndroidManifest



BroadcastReceiver – precizari importante!!!

- pentru acest receiver pe care tocmai l-am declarat în fişierul AndroidManifest.xml trebuie precizãm şi *lista de Intent-uri pe care le „ascultã”*;

- un **BroadcastReceiver** poate intercepta mai multe tipuri de „mesaje” de tip Intent, cum ar fi: primirea unui apel telefonic, primirea unui mesaj SMS, primirea unui mail, descãrcarea bateriei sub o anumitã limitã etc.

Intent-uri si filtre (IntentFilter)

În Android este necesar sã **mapãm un astfel de BroadcastReceiver cu o listã (un filtru) de Intent-uri.**

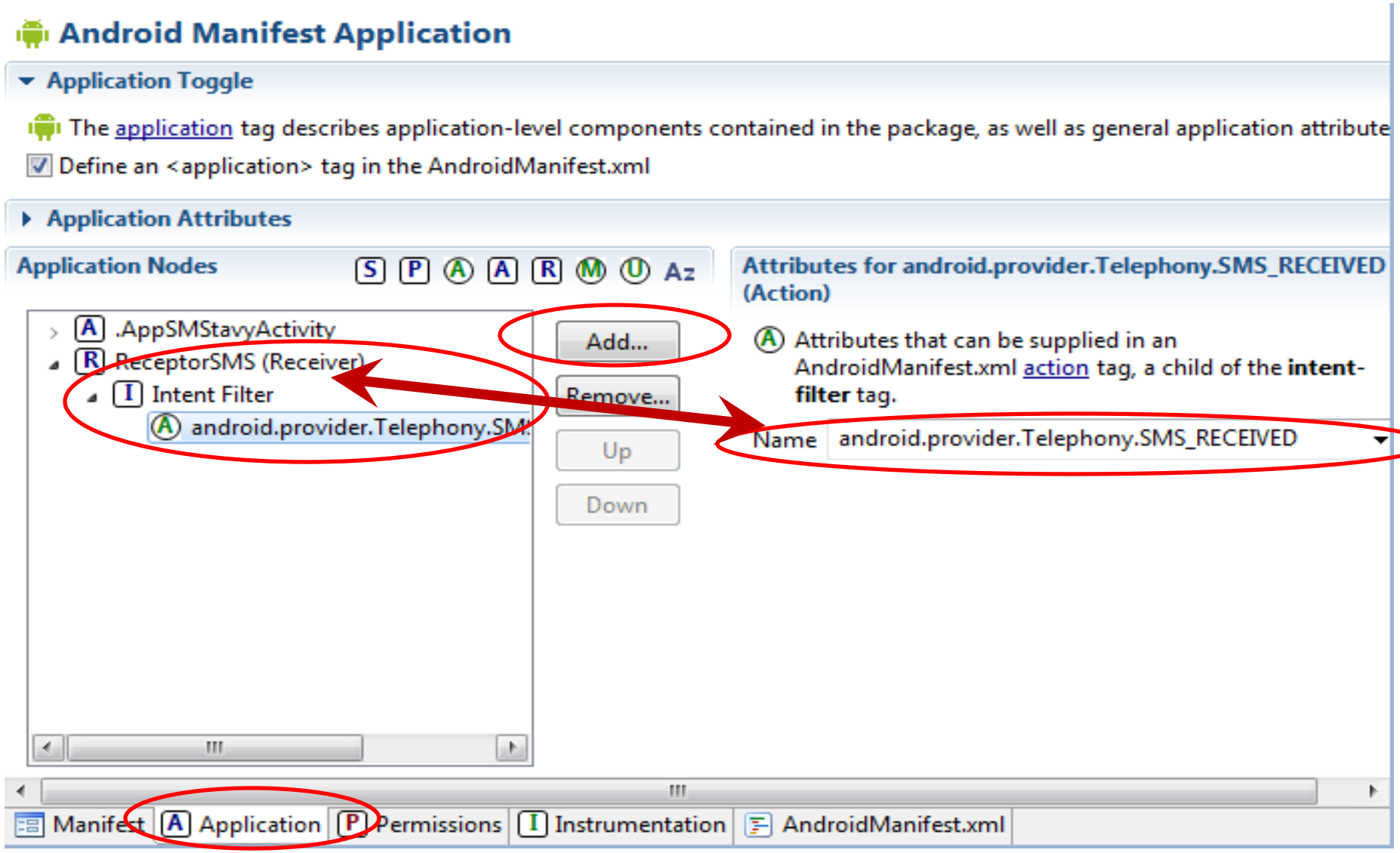
Aceastã punere în corespondenţã se realizeazã prin intermediul conceptului de **IntentFilter**, care trebuie declarat tot în fişierul AndroidManifest.xml.

Intent-uri si filtre (IntentFilter)

Operaţiunea presupune poziţionarea pe **ReceptorSMS(Receiver)** şi acţionarea butonului **Add...** urmatã de selectarea opţiunii **IntentFilter**.

Dupã aceastã operaţiune, observãm cã ReceptorSMS a devenit rãdãcina unui mic arbore în care avem o secţiune numitã IntentFilter. Ne poziţionãm pe aceastã secţiune, acţionãm butonul **Add...**, selectãm tipul **Action** şi apoi alegem din listã intent-ul numit **android.provider.Telephony.SMS_RECEIVED**.

Intent-uri si filtre (IntentFilter)



Intent-uri si filtre (IntentFilter)

Prin această operaţiune, am precizat de fapt mediului de lucru următorul fapt:

- BroadcastReceiver-ul cu numele ReceptorSMS va „reacţiona” atunci când apare un „eveniment” (intent) de tip SMS_RECEIVED, adică atunci când dispozitivul primeşte un mesaj de tip SMS.

Structura fisierului AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="smsTavy.com"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.SEND_SMS"></uses-permission>
    <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".AppSMStavyActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name="ReceptorSMS">
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED">
                </action>
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

Programarea aplicatiei

Pentru a definitiva aplicaţia, trebuie sã completãm metoda **onReceive(...)** astfel încât sã obţinem numãrul de telefon al expeditorului, precum şi mesajul text care alcãtuieşte mesajul primit. Metoda onReceive(...) are 2 parametri: un **Context** şi un **Intent**.

Ceea ce ne interesează pe noi este conţinut într-o formã „împachetatã” în acel intent care este primit ca parametru de cãtre metodã☹.

Strategia de lucru

- verificãm dacã intent-ul este de tip SMS_RECEIVED;
- cu ajutorul metodei **getExtras()** obţinem „pachetul” (bundle) din cadrul intent-ului primit ca parametru;
- din „pachet” obţinem lista de obiecte conţinute (Object[]), folosind metoda .get(”pdus”);

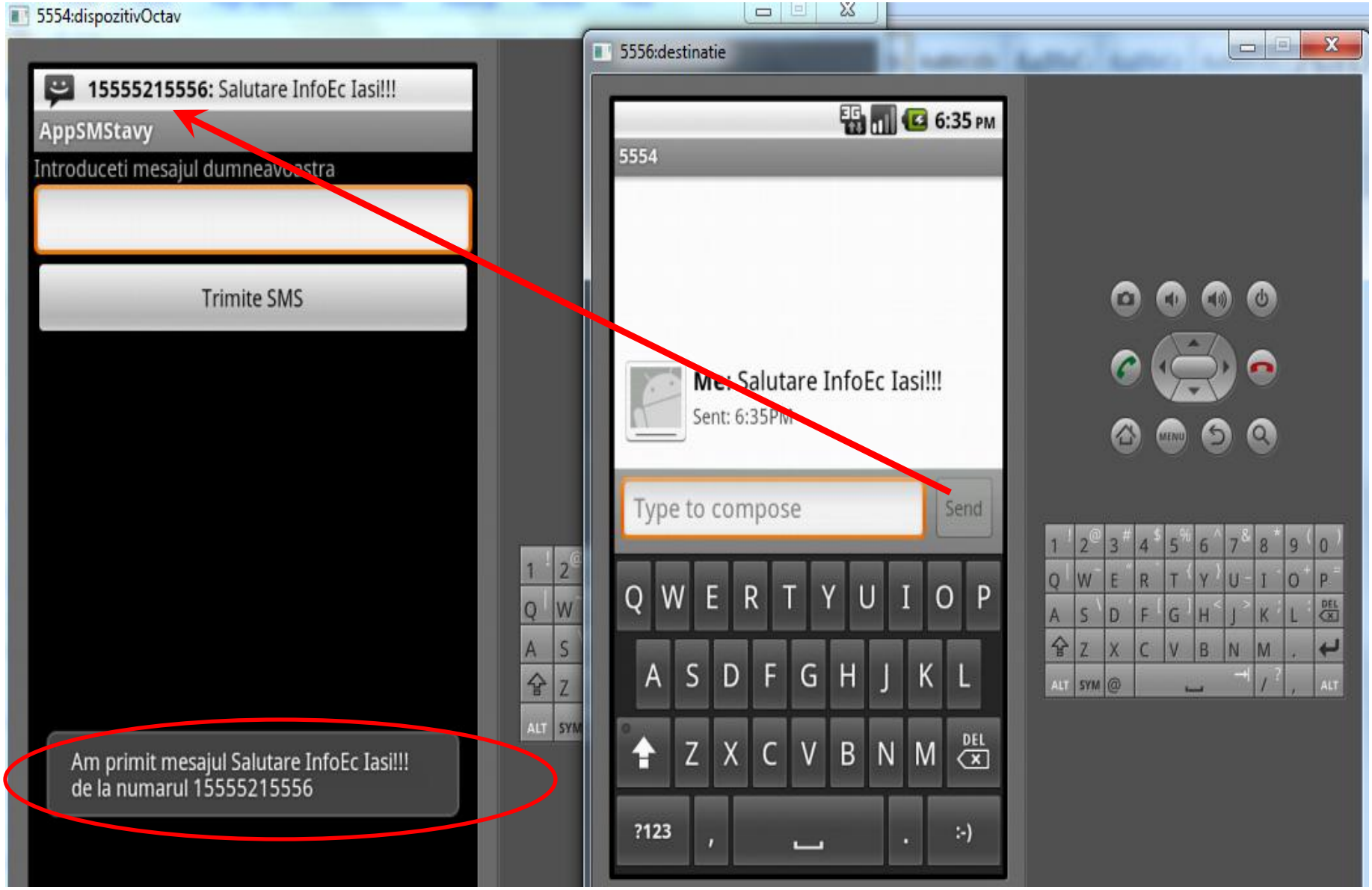
Strategia de lucru - continuare

- definim un **mesaj** de tip **SmsMessage**;
- din clasa **SmsMessage** folosim metoda **createFromPdu** pentru a obţine mesajul efectiv din primul obiect din pachet;
- pentru a obţine numãrul de telefon al expeditorului şi mesajul primit, folosim metodele **getOriginatingAddress** şi **getMessageBody**;
- afişãm pe ecran datele obţinute, prin intermediul unui **Toast**.

Implementare – model☺

```
package smsTavy.com;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.util.Log;
import android.widget.Toast;
public class ReceptorSMS extends BroadcastReceiver {
        @Override
        public void onReceive(Context context, Intent intent) {
                // TODO Auto-generated method stub
                Log.i("din receiver","s-a declansata metoda onReceive");
                if(intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED"))
                {
                        Log.i("din receiver", "Am primit un SMS!!!");
                        Bundle pachet;
                        pachet = intent.getExtras();

                        Object[] mesajSosit =  (Object[]) pachet.get("pdus");

                        SmsMessage mesaj;
                        mesaj = SmsMessage.createFromPdu((byte[]) mesajSosit[0]);

                        String numar;
                        String text;

                        numar = mesaj.getOriginatingAddress();
                        text = mesaj.getMessageBody();

                        Toast notificare;
                        notificare = Toast.makeText(context, "Am primit mesajul " + text + " de la
numarul " + numar, Toast.LENGTH_LONG);

                        notificare.show();

                        Log.i("din receiver", numar + "   " + text);

                }
        }
}
```

Rezultat final



Directii de viitor

- Monitorizarea copiilor de catre parinti☺

1. pãrintele trimite un SMS cu un mesaj special cãtre telefonul copilului;
2. aplicaţia instalatã pe acel telefon trimite pãrintelui un rãspuns SMS în care furnizeazã locaţia geograficã (latitudine şi longitudine) în care se aflã copilul;
3. mai în glumã, mai în serios, ne putem imagina cã aceste locaţii pot fi: şcoala, ~~discoteca~~☺, biblioteca etc.