Mobile applications

Master SDBIS/SIA

Octavian Dospinescu

2021

General topics

- Graphical controls in Android
- Properties, methods, events, listeners
- Graphical interfaces
- Static controls vs. dynamic controls

Frequent graphical controls

- TextView
- EditText
- Button, ImageButton, ToggleButton
- CheckBox
- RadioButton, RadioGroup
- Spinner
- Lists

TextView

- It’s the simplest widget. It is used to display fixed texts.

- Because of its usefulness, we can look like a label, although we can show active links to web pages, phone numbers or email addresses.

TextView - attributes

- android:typeface – set the type of the characters that will be used to display the text, for example *serif*;

- android:textStyle – set the style of the font: **bold**, *italic* or ***combinate***;

TextView - attributes

- android:textColor – set the color of the text in RGB format, for example #fd3099 for a pink☺;
- more details here: http://developer.android.com/reference/android/graphics/Color.html
- android:autoLink – we can set if there will be activated the web addresses, email addresses or phone numbers from the displayed text

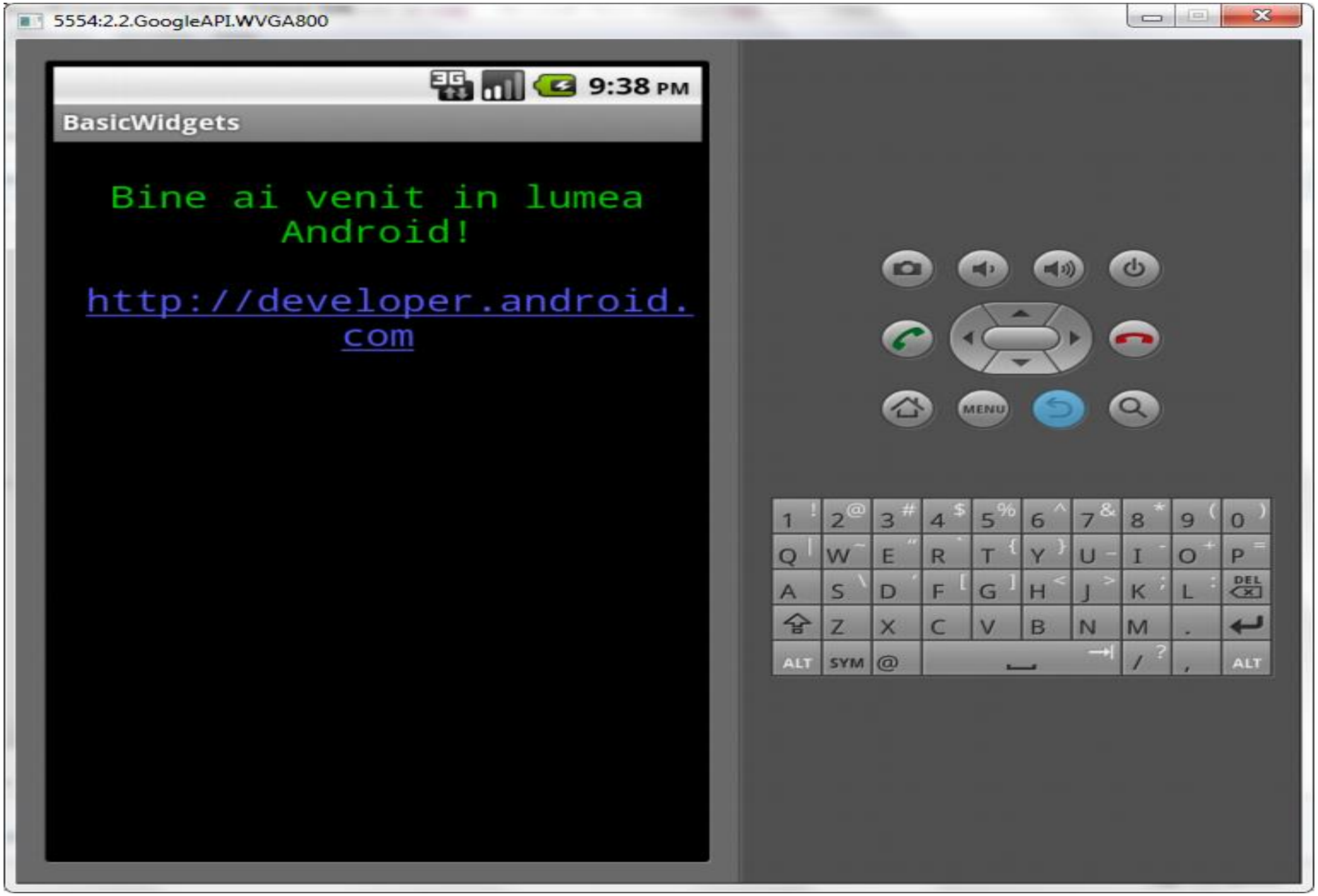
TextView – attributes in xml

```xml
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/bun_venit"
    android:typeface="monospace"
    android:textColor="#00cd00"
    android:textSize="20sp"
    android:autoLink="all"
    android:gravity="center"
    />
```

TextView – adding in dynamic mode

```
TextView label = new TextView(this);
    label.setText(R.string.bun_venit);
    label.setTextColor(Color.GREEN);
    label.setTextSize(TypedValue.COMPLEX_UNIT_SP, 20);
    label.setGravity(Gravity.CENTER);
    label.setTypeface(Typeface.MONOSPACE);
    label.setAutoLinkMask(Linkify.ALL);

    myLayout.addView(label);
```

TextView – graphical appearances



BasicWidgets

Bine ai venit in lumea Android!

http://developer.android.com

EditText

- It’s a subclass of TextView, the only facility to the latter is that it allows us to edit the text that it displays;

- Assimilated with the “classic” term of TextBox.

EditText - Attributes

- android:inputType, with its help we specify the type of data to be retrieved; so the keyboard will adjust itself for easier data input. For example, if we give the value *phone*, when the user will press the box, a numeric keypad will be shown;

- android:autoText, with this attribute can set the system to detect and correct grammatical errors;

EditText - Attributes

- android:singleLine, we can set the text box to be expandable depending on the text entered , or stay on one line regardless of it.
- android:hint, we can offer a suggestion for completing the text box when it is empty ;
- android:digits, if we want to introduce only certain numbers, we can specify this limit with this attribute.

EditText - Examples

```
<!-- Exemplu simplu, fiecare propozitie va incepe cu litera mare -->
    <TextView android:text="@string/label_simplu"
            android:layout_height="wrap_content"
            android:layout_width="match_parent" />
    <EditText android:id="@+id/etSimplu"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:capitalize="sentences"
            android:inputType="textMultiLine"
            android:gravity="top"
            android:lines="4">
    </EditText>
```

EditText – Examples...examples...

```
<!-- Vom oferi un hint pentru acest EditText -->
<TextView android:text="@string/label_hint"
      android:layout_height="wrap_content"
      android:layout_width="match_parent" />
<EditText android:id="@+id/etHint"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="@string/et_label_hint"
      >
</EditText>
```

EditText – Examples…examples…examples…

```
<!-- Vom configura aceasta casuta pentru
a introduce mai usor un numar de telefon -->

<TextView android:text="@string/label_telefon"
        android:layout_height="wrap_content"
        android:layout_width="match_parent" />
<EditText android:id="@+id/etTelefon"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone"
        >
</EditText>
```

EditText – Examples…examples…examples…☺

```
<!-- Casuta pentru introducerea unei parole -->

<TextView android:text="@string/label_parola"
        android:layout_height="wrap_content"
        android:layout_width="match_parent" />
<EditText android:id="@+id/etParola"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        >
</EditText>
```

EditText – Adding it in a dynamic way

```
EditText etAdresa = new EditText(this);
    // setam lungimea si inaltimea
    etAdresa.setLayoutParams(new LinearLayout.LayoutParams(
                    LinearLayout.LayoutParams.MATCH_PARENT,
                    LinearLayout.LayoutParams.WRAP_CONTENT));
    // adaugam un indiciu
    etAdresa.setHint("Introduceti adresa dvs.");
    // aceasta casuta de text va fi pe mai multe linii
    // si fiecare propozitie va incepe cu majuscula
    etAdresa.setInputType(InputType.TYPE_TEXT_FLAG_MULTI_LINE
                    | InputType.TYPE_TEXT_FLAG_CAP_SENTENCES);
    // textul va fi aliniat stanga sus
    etAdresa.setGravity(Gravity.TOP);
    // casuta de text va avea 2 linii
    etAdresa.setLines(2);

    // adaug casuta de text in view
    myLayout.addView(etAdresa);
```

EditText – Using in practice

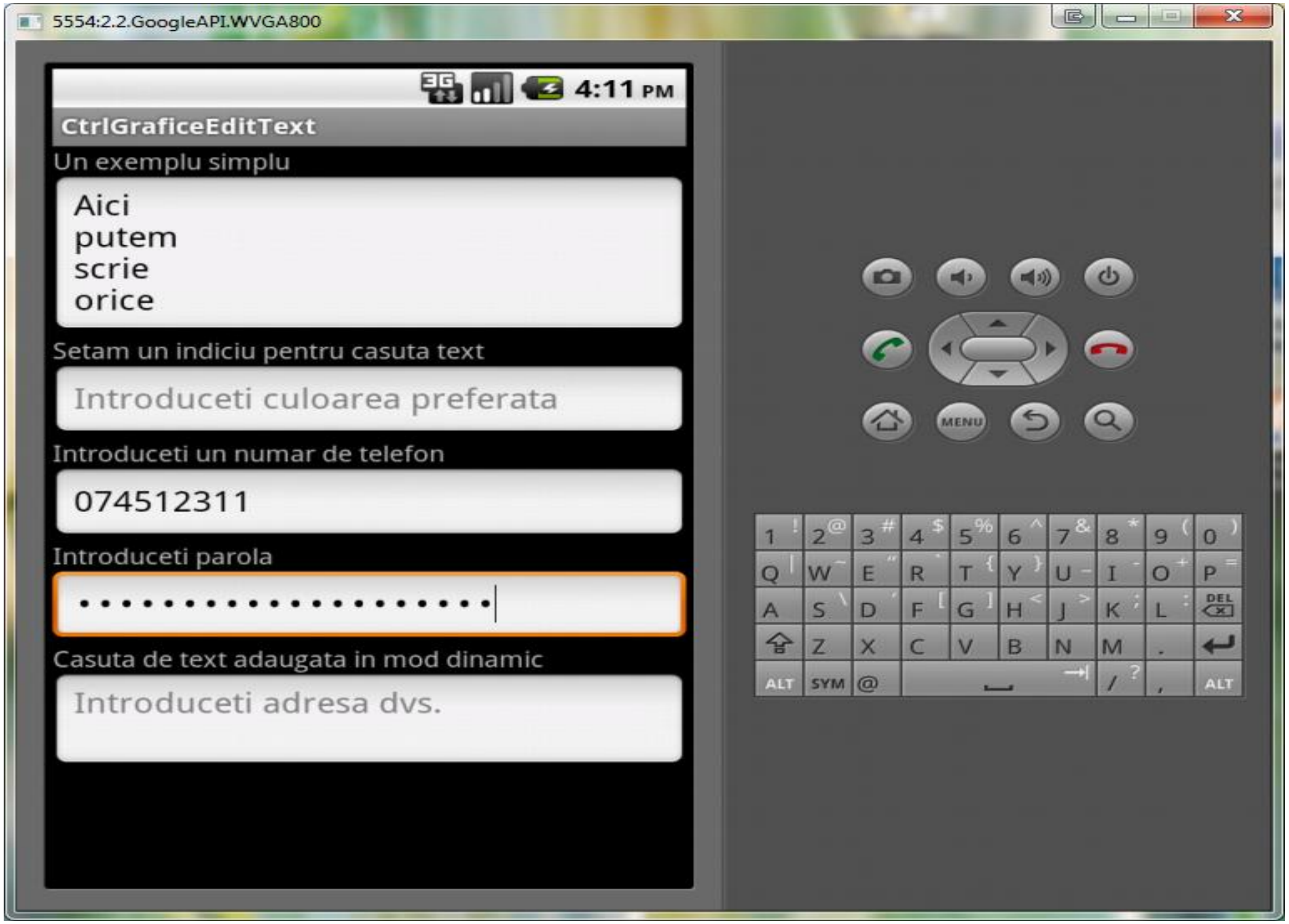
- Most often the values in text boxes will be retrieved and analyzed when the user will press a button, such as ***Send***, but there are times when you want to take a specific action when the user interacts with these text boxes.

- To capture the click event we will have to use the method ***setOnClickListener()*** or to capture the focus event will have to use the method ***setOnFocusChangeListener().***

EditText – Using in practice

```
final EditText  etTelefon = (EditText)findViewById(R.id.etTelefon);

etTelefon.setOnFocusChangeListener(new onFocusChangeListener() {
        public void onFocusChange(View v, boolean hasFocus) {
                //preluam valoarea din casuta text
                String adresaPreluata = etTelefon.getText().toString();
                //afisam in log-ul pentru debug
                Log.d("#### InfoEc Debug ###", adresaPreluata);
        }
    }
);
```

EditText – Example of the displayed result



5554:2.2.GoogleAPI.WVGA800

CtrlGraficeEditText

Un exemplu simplu

Aici
putem
scrie
orice

Setam un indiciu pentru casuta text

Introduceti culoarea preferata

Introduceti un numar de telefon

074512311

Introduceti parola

••••••••••••••••••••

Casuta de text adaugata in mod dinamic

Introduceti adresa dvs.

Button – general presentation

- 3 types of buttons:
  - Button;
  - ImageButton;
  - ToggleButton.

Among the most commonly used attributes of a button there are the ones to specify the id , width, height and the text to be displayed within it .

Button – specific attributes

```
<Button android:id="@+id/btnApasa"
        android:text="@string/label_apasa"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="@color/button_text_color">
  </Button>
```

```
Am creat in prealabil fisierul colors.xml in directorul res/values.
<?xml version="1.0" encoding="utf-8"?>
<resources>
            <color name="button_text_color">#ffcc0005</color>
</resources>
```

Button – how to usually use it

```
// creez butonul pe baza declaratiilor din xml
Button btnApasa = (Button) findViewById(R.id.btnApasa);

btnApasa.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
                // afisez un mesaj atunci cand utilizatorul apasa pe buton
                Toast.makeText(
                CtrlGraficeButtonActivity.this,
                "De ce ma apesi? Tu nu stii ca sunt mic si ca ma doare?“,
                                Toast.LENGTH_LONG).show();
                                }
        });
```

ImageButton – model of implementation

We can specify the image from the xml file as follows:

```
<ImageButton android:id="@+id/imageButton"
             android:layout_width="wrap_content"
             android:layout_height="wrap_content"
             android:src="@drawable/feaa"
      />
```

Or we can specify the image in a dynamic mode using the method ***setImageResource()***:

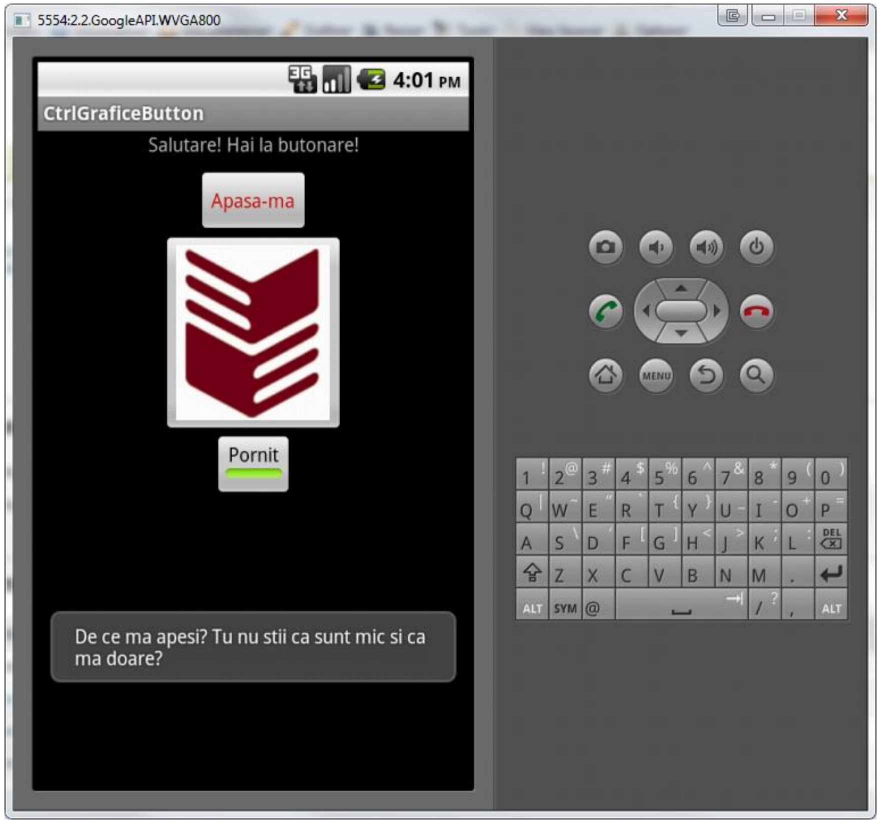
```
ImageButton imageButton = (ImageButton)findViewById(R.id.imageButton);
      imageButton.setImageResource(R.drawable.feaa);
```

ToggleButton – general presentation

- This type of button is more like a checkbox because it is a two-state button, On/Off .
- Depending on the state in which it is, the LED displays green to **On** or gray to **Off**.
- Important notice: you can customize the text displayed for each state in part using the attributes *android:textOn* and *android:textOff.*

ToggleButton – implementation

```
<ToggleButton android:id="@+id/toggleButton"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:textOn="Pornit"
              android:textOff="Oprit"
              android:layout_gravity="center"  />
```



Salutare! Hai la butonare!

De ce ma apesi? Tu nu stii ca sunt mic si ca ma doare?

Checkbox – general presentation

- This widget has two states: checked and unchecked .
- When we click on it, the states will be automatically swapped .
- Since TextView is his "ancestor", the checkbox inherits all its methods.
- We can control the states from the code by using the methods **setChecked()** or **toggle()**. We can also get the current status using the method **isChecked()**.

Checkbox – an implementation

We will create 3 checkboxes in main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
       android:orientation="vertical"
       android:layout_width="fill_parent"
       android:layout_height="fill_parent">
       <TextView android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="@string/alege_culori" />
       <CheckBox android:id="@+id/ckbRosu"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Rosu"        />
       <CheckBox android:id="@+id/ckbGalben"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Galben" />
       <CheckBox android:id="@+id/ckbAlbastru"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Albastru" />
</LinearLayout>
```

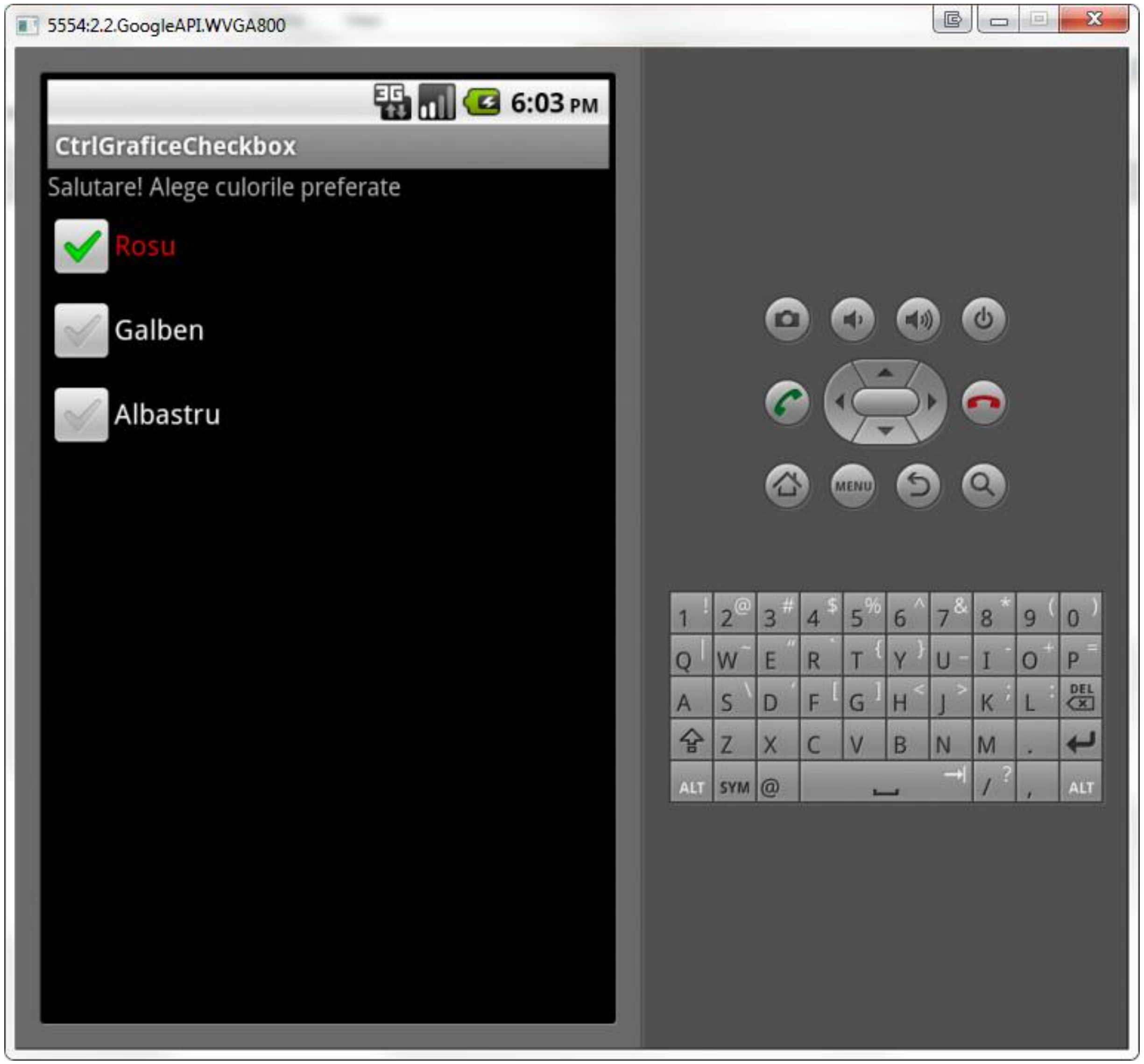
Checkbox – an implementation

If the user picks one of the listed colors, we will highlight the corresponding text.

```
  public      class     CtrlGraficeCheckboxActivity     extends     Activity     implements
OnCheckedChangeListener {
      @Override
      public void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.main);

          CheckBox ckbRosu = (CheckBox)findViewById(R.id.ckbRosu);
          ckbRosu.setOnCheckedChangeListener(this);

          CheckBox ckbAlbastru = (CheckBox)findViewById(R.id.ckbAlbastru);
          ckbAlbastru.setOnCheckedChangeListener(this);

          CheckBox ckbGalben = (CheckBox)findViewById(R.id.ckbGalben);
          ckbGalben.setOnCheckedChangeListener(this);
      }
          @Override
          public void onCheckedChanged (CompoundButton buttonView, boolean isChecked) {
                    if(isChecked){
                              buttonView.setTextColor(Color.RED);
                    } else {
                              buttonView.setTextColor(Color.WHITE);
                    }
          }
  }
```

Checkbox – an implementation

The result:



Salutare! Alege culorile preferate

- ✓ Rosu
- Galben
- Albastru

RadioButton – general presentation

- Very similar to the Checkbox widget , RadioButton has as the "ancestor" the TextView so that we can customize these elements using all the methods provided by TextView .

- This widget also has two states: checked/unchecked .

- We can use the method **isChecked()** to verify the current status of a RadioButton and **toggle()** to change the state .

RadioButton – general presentation

- Most times radio buttons are used in a **RadioGroup**. When multiple buttons are placed in a RadioGroup, only one of them can be checked at a time.

- If we give an id to the group of radio buttons, we have access to the following methods :
  - *check()* – we can check the status of individual RadioButton within the group, for example *radioGrup.check(R.id.radioButton1)*;
  - *clearCheck()* – uncheck all items from RadioGroup;
  - *getCheckedRadioButtonId()* – returns the id of the element checked in the RadioGroup community. If any item is unchecked, it will return -1.

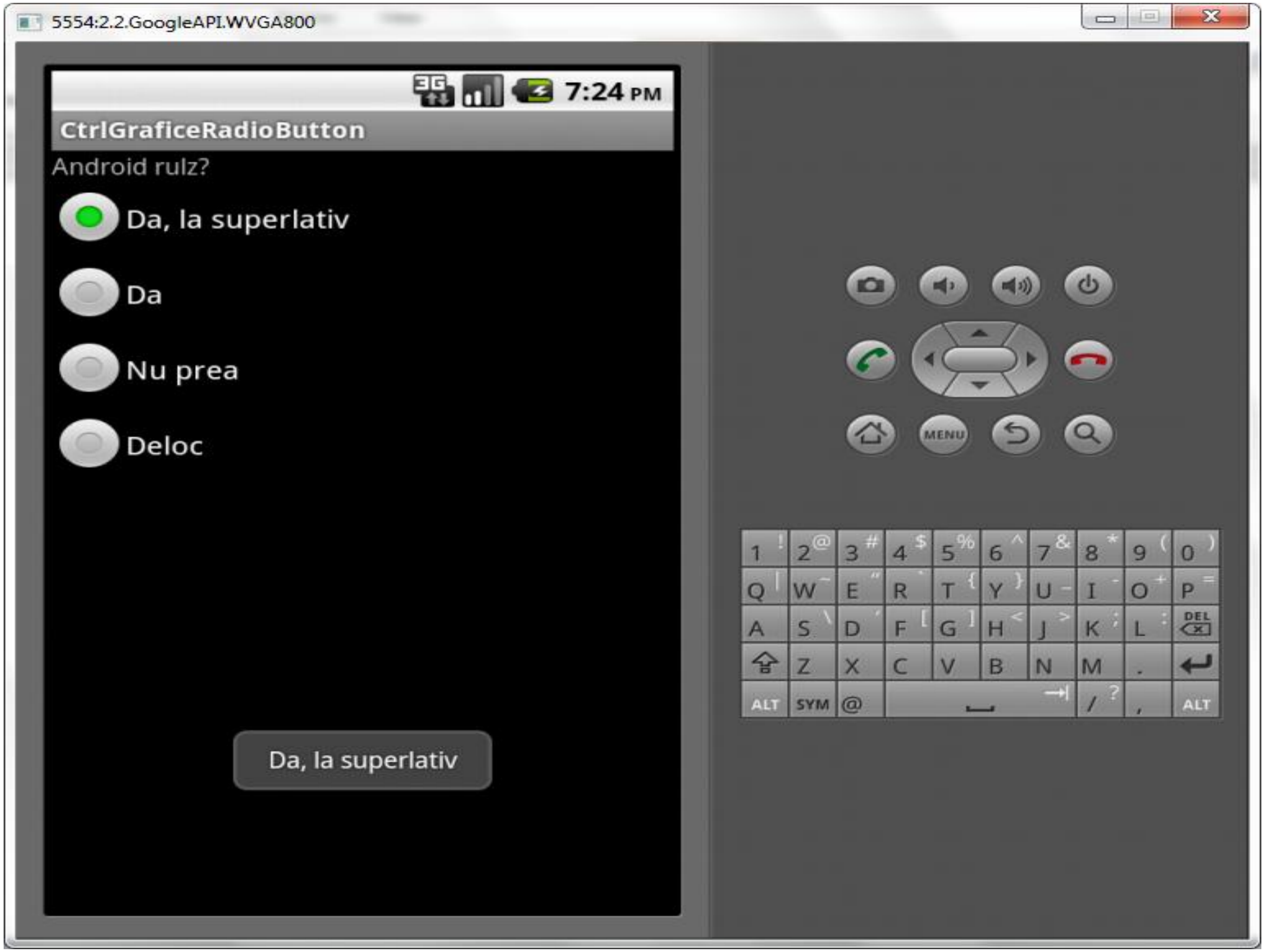
RadioButton – implementation (model)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical" android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TextView android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:text="@string/intrebare"
        />
        <RadioGroup android:id="@+id/rdGroup"
                  android:layout_width="wrap_content"
                  android:layout_height="wrap_content">
                  <RadioButton android:id="@+id/rdButton1"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:text="Da, la superlativ" />
                  <RadioButton android:id="@+id/rdButton2"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:text="Da" />
                  <RadioButton android:id="@+id/rdButton3"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:text="Nu prea" />
                  <RadioButton android:id="@+id/rdButton4"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:text="Deloc" />
        </RadioGroup>
</LinearLayout>
```

RadioButton – implementation (model)

```
public class CtrlGraficeRadioButtonActivity extends Activity {
        /** Called when the activity is first created. */
        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
                RadioGroup rdGroup = (RadioGroup) findViewById(R.id.rdGroup);
                rdGroup.setOnCheckedChangeListener(new OnCheckedChangeListener() {
                        @Override
                        public void onCheckedChanged(RadioGroup rd, int idRadioBtn) {
                                RadioButton rdBtn;
                                String textSelectat = "Nu ai selectat nimic";
                                // iterez printre toate elementele grupului
                                // pentru a verifica care buton a fost selectat
                                for (int j = 0; j < rd.getChildCount(); j++) {
                                        rdBtn = (RadioButton) rd.getChildAt(j);
                                // daca id-ul curent este acelasi cu cel trimis ca
                                // paramentru retin textul
                                        if (rdBtn.getId() == idRadioBtn)
                                        textSelectat = rdBtn.getText().toString();
                                }
                                // afisez un mesaj atunci cand utilizatorul apasa pe buton
        Toast.makeText(CtrlGraficeRadioButtonActivity.this,textSelectat,Toast.LENGTH_SHORT).show();
                        }
                });
        }
}
```

RadioButton – result☺



ListView – general presentation

- Android offers more controls for displaying lists. The most common and most useful is ListView . With its help we can display **very large vertical lists.**

ListView – implementation

- The main class that was generated when creating the project can extends ***ListActivity***. It houses a ListView that can be populated with data from different sources, either an array or a Cursor containing the results of an SQL query .

- *ListActivity* will insert a list that will be expanded on the whole screen.

- If we want to customize the layout, we'll insert in the xml file a ListView element with the attribute **android:id="@android:id/list"** so that *ListActivity* will know which is the element that will display the list.

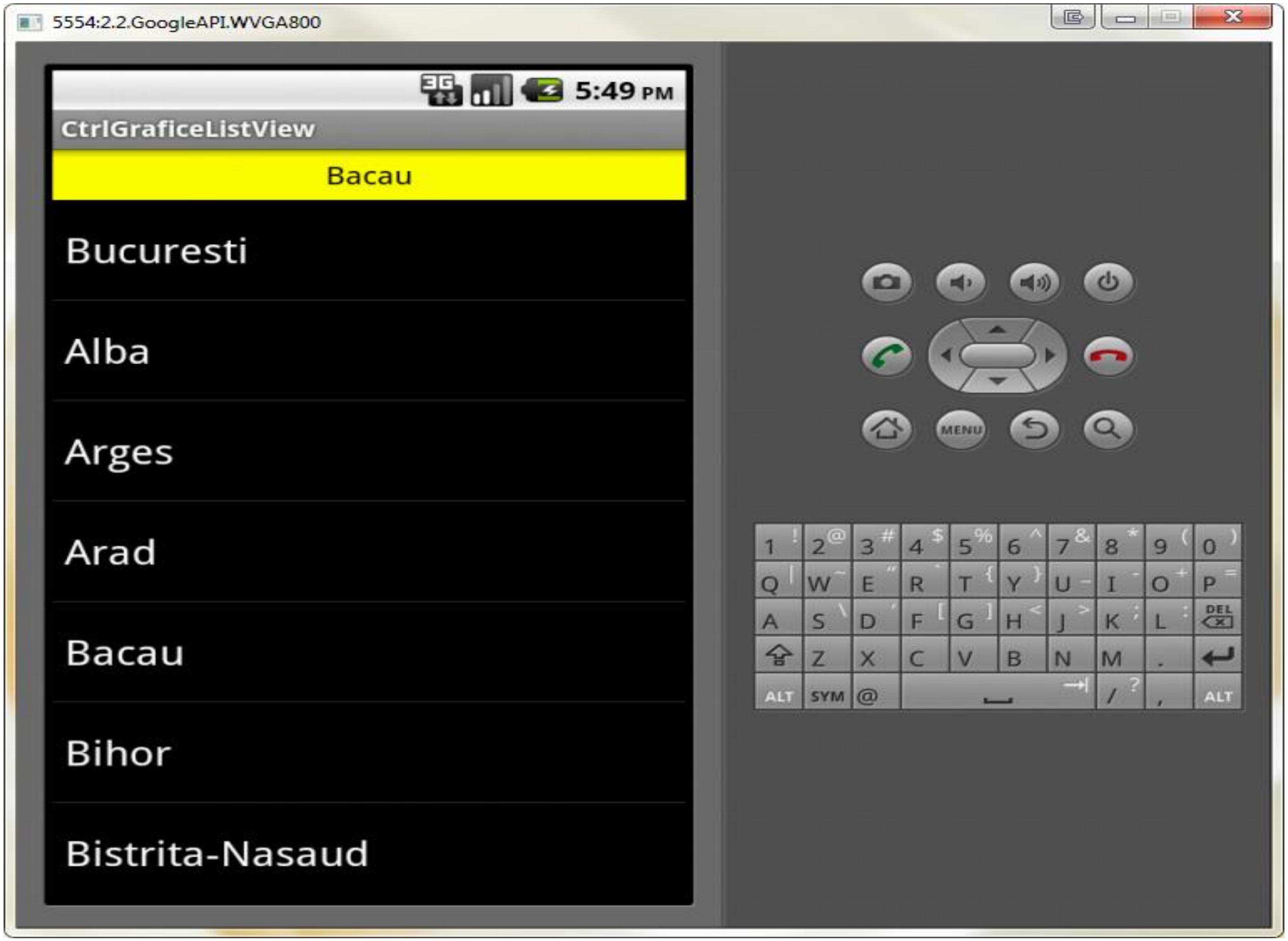
ListView – xml implementation

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
       android:orientation="vertical"
       android:layout_width="fill_parent"
       android:layout_height="fill_parent" >
       <TextView android:id="@+id/judetSelectat"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="#ffffff00"
                android:textColor="#ff000000"
                android:textSize="16dp"
                android:gravity="center"
                android:padding="5dp"
                />
       <ListView android:id="@android:id/list"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:drawSelectorOnTop="false"
       />
</LinearLayout>
```

ListView – java implementation

```
public class CtrlGraficeListViewActivity extends ListActivity {
       final static String[] judete = new String[] { "Bucuresti", "Alba", "Arges",
                "Arad", "Bacau", "Bihor", "Bistrita-Nasaud", "Botosani", "Brasov",
                "Braila", "Buzau", "Caras-Severin", "Calarasi", "Cluj",
                "Constanta", "Covasna", "Dambovita", "Dolj", "Galati", "Giurgiu",
                "Gorj", "Harghita", "Hunedoara", "Ialomita", "Iasi", "Ilfov",
                "Maramures", "Mehedinti", "Mures", "Neamt", "Olt", "Prahova",
                "Satu Mare", "Salaj", "Sibiu", "Suceava", "Teleorman", "Timis",
                "Tulcea", "Vaslui", "Valcea", "Vrancea" };
       TextView judetSelectat;

       @Override
       public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
                setListAdapter(new ArrayAdapter<String>(this,
                                  android.R.layout.simple_list_item_1, judete));
                judetSelectat = (TextView) findViewById(R.id.judetSelectat);
       }

       public void onListItemClick (ListView lv, View v, int position, long id) {
                judetSelectat.setText(judete[position]);
       }
}
```

ListView – graphical result ☺



5554:2.2.GoogleAPI.WVGA800

5:49 PM

CtrlGraficeListView

Bacau

Bucuresti

Alba

Arges

Arad

Bacau

Bihor

Bistrita-Nasaud

ListView – a (little) more complex implementation

If we want to change the display mode **to enable multiple selection** in the list, we will have to make the following changes in our code :

- in the method *onCreate()* we replace the layout *simple_list_item_1* cu ***simple_list_item_multiple_choice*** and we add the following line of code:

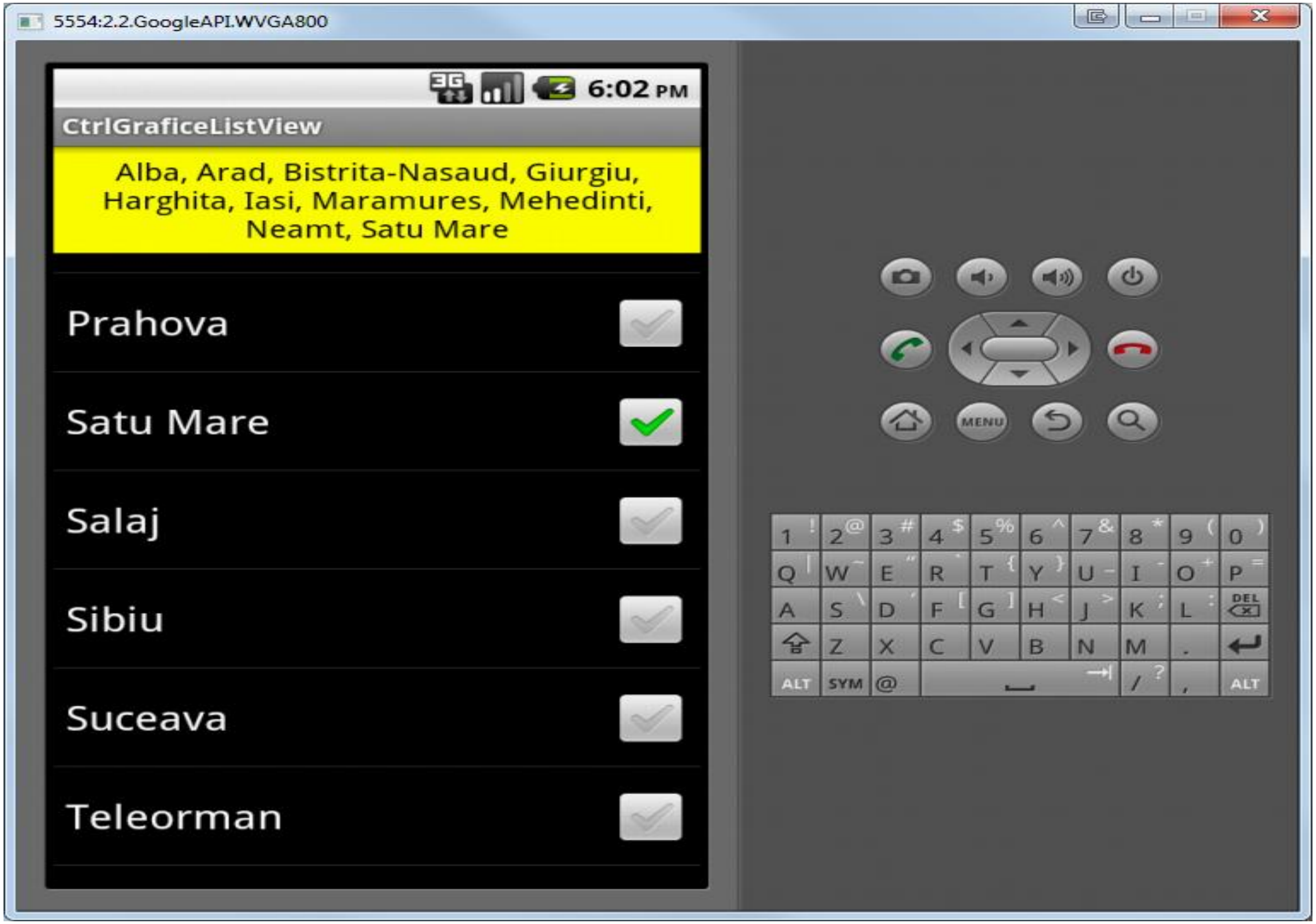
getListView().setChoiceMode(ListView.*CHOICE_MODE_MULTIPLE*);

ListView – a (little) more complex implementation

The Java code will have some changes (we “scroll” through the list of “selections”):

```
public void onListItemClick (ListView parent, View v, int position, long id) {
              SparseBooleanArray arrChecked = parent.getCheckedItemPositions();
              StringBuilder sb = new StringBuilder();
              String delim = "";
              for (int i = 0; i < arrChecked.size(); i++) {
                       if (arrChecked.valueAt(i)) {  //adica daca este selectat
                                sb.append(delim).append(

    parent.getItemAtPosition(arrChecked.keyAt(i)));
                                delim = ", ";
                       }
              }
              judetSelectat.setText(sb);
    }
```

ListView – a result for multiple selections


5554:2.2.GoogleAPI.WVGA800

6:02 PM

CtrlGraficeListView

Alba, Arad, Bistrita-Nasaud, Giurgiu, Harghita, Iasi, Maramures, Mehedinti, Neamt, Satu Mare

Prahova	
Satu Mare	✓
Salaj	
Sibiu	
Suceava	
Teleorman	

More implementations…

…during the lab ☺