



# Android Programing

Gavrilut Dragos



# Storage

- 5 methods that can be used for storage on Android based systems:
  - **Shared Preferences** (stores informations as a Dictionary)
  - **Internal Storage** (store data into device internal memory)
  - **External Storage** (store data into device external memory – SD-Card)
  - **SQLite DataBases**
  - **Via network**



# Shared Preferences

- Use the SharedPreferences class
- It can be obtain from an Activity in the following way:
  - Call **getSharedPreferences()** with the name of the preferences you need
  - Call **getPreferences()** if you only need one preference file for each Activity
  - Does not require any special permissions

# Shared Preferences

- Values that can be written in a SharedPreferences class

Getter	Setter
getBoolean	putBoolean
getFloat	putFloat
getInt	putInt
getLong	putLong
getString	putString
getStringSet	putStringSet
getAll	

# Shared Preferences - Example

```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        SharedPreferences pref = getSharedPreferences("my_prefs", 0);
        int someValue = pref.getInt("my_value", 123);
        // use someValue

        setContentView(...);
    }

    protected void SavePrefs() {
        SharedPreferences pref = getSharedPreferences("my_prefs", 0);
        SharedPreferences.Editor edit = pref.edit();
        edit.putInt("my_value", 123456);
        edit.commit();
    }
}
```



/data/data/com.example.surfaceviewtest/shared\_prefs → my\_prefs.xml

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <int name="my_value" value="123456" />
</map>
```

# Shared Preferences - Example

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        SharedPreferences pref = getPreferences(0);  
        int someValue = pref.getInt("my_value", 123);  
        // use someValue  
  
        setContentView(...);  
    }  
  
    protected void SavePrefs() {  
        SharedPreferences pref = getPreferences(0);  
        SharedPreferences.Editor edit = pref.edit();  
        edit.putInt("my_value", 123456);  
        edit.commit();  
    }  
}
```

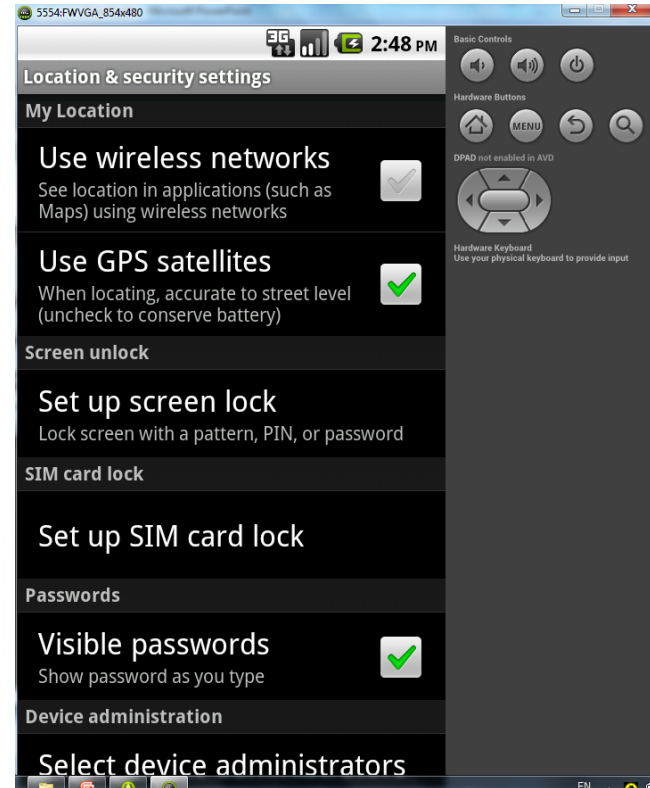
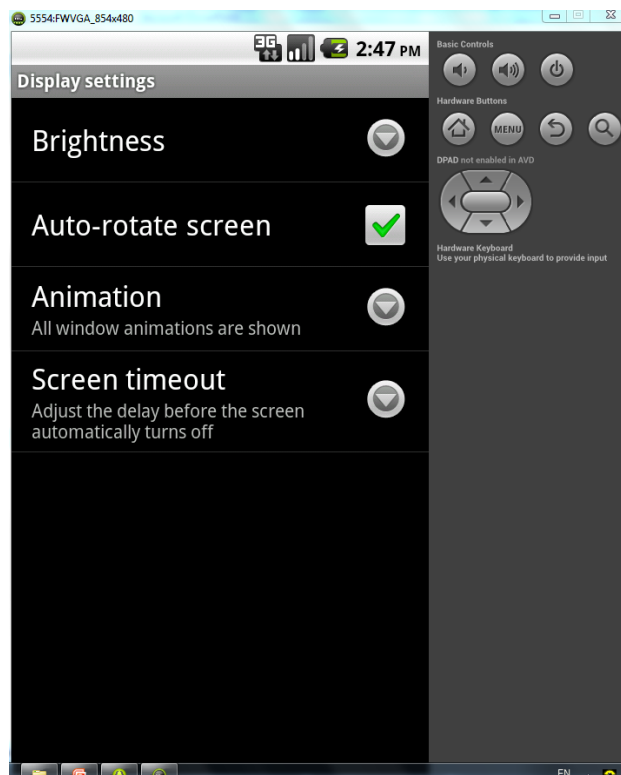


/data/data/com.example.surfaceviewtest/shared\_prefs → MainActivity.xml

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>  
<map>  
    <int name="my_value" value="123456" />  
</map>
```

# Shared Preferences

- Android has an option to create a special Activity desing for viewing and editing the preferences.



# Shared Preferences

1. Create a new XML file that contains every item that should be modified. The XML will be located in `res/xml/my_preferences.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
    <EditTextPreference
        android:key="my_value_2"
        android:defaultValue="123"
        android:title="Some value ..."/>
</PreferenceScreen>
```



# Shared Preferences

2. Create a new activity and derive it from the PreferenceActivity class

```
public class MyPrefActivity extends PreferenceActivity{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getPreferenceManager().setSharedPreferencesName("my_prefs");
        addPreferencesFromResource(R.xml.my_preferences);
    }
}
```

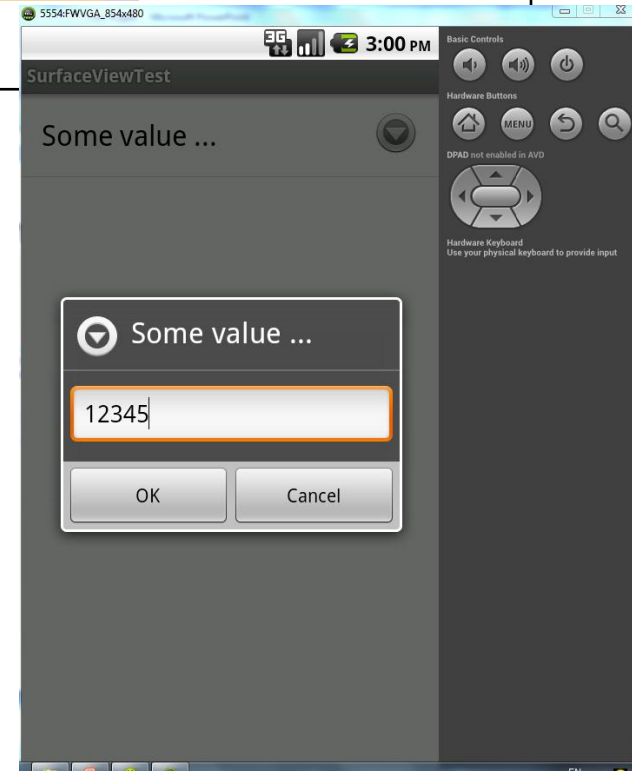
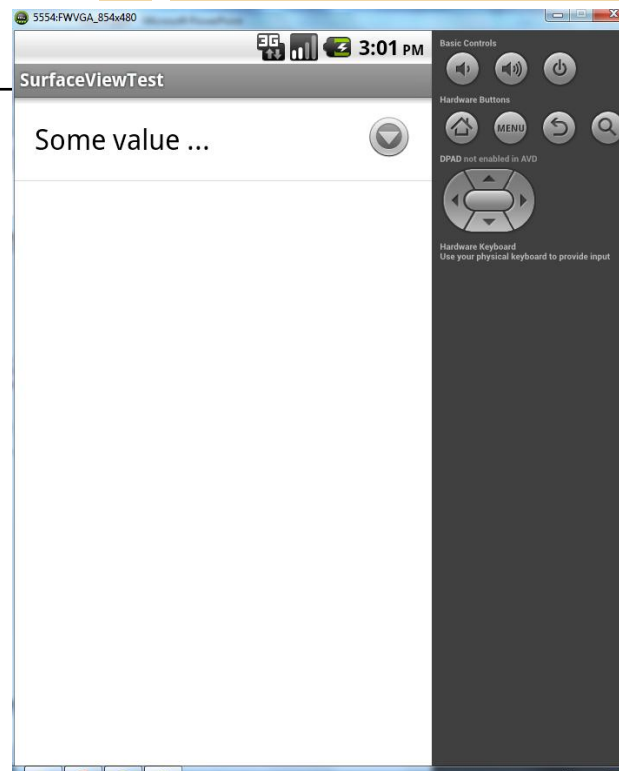
3. Add the new activity in the AndroidManifest.xml

```
<activity android:name="com.example.studenti.MyPrefActivity" />
```

# Shared Preferences

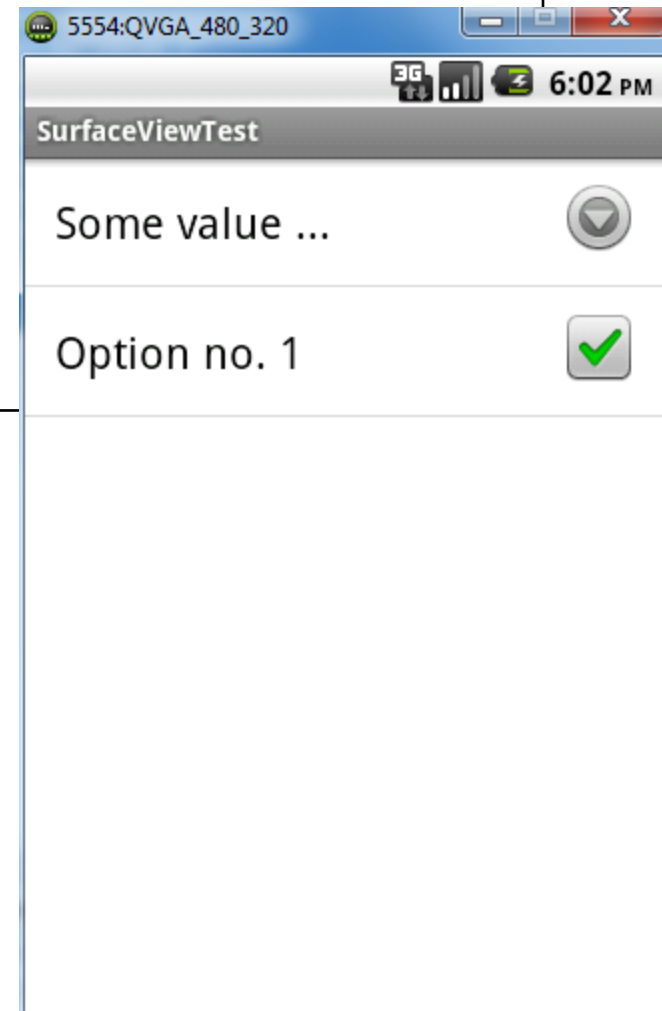
4. Start the Preference activity from one point in the main activity

```
public class MainActivity extends Activity {  
    protected void ShowPreferences() {  
        startActivity(new Intent(this, MyPrefActivity.class));  
    }  
}
```



# Shared Preferences - XML

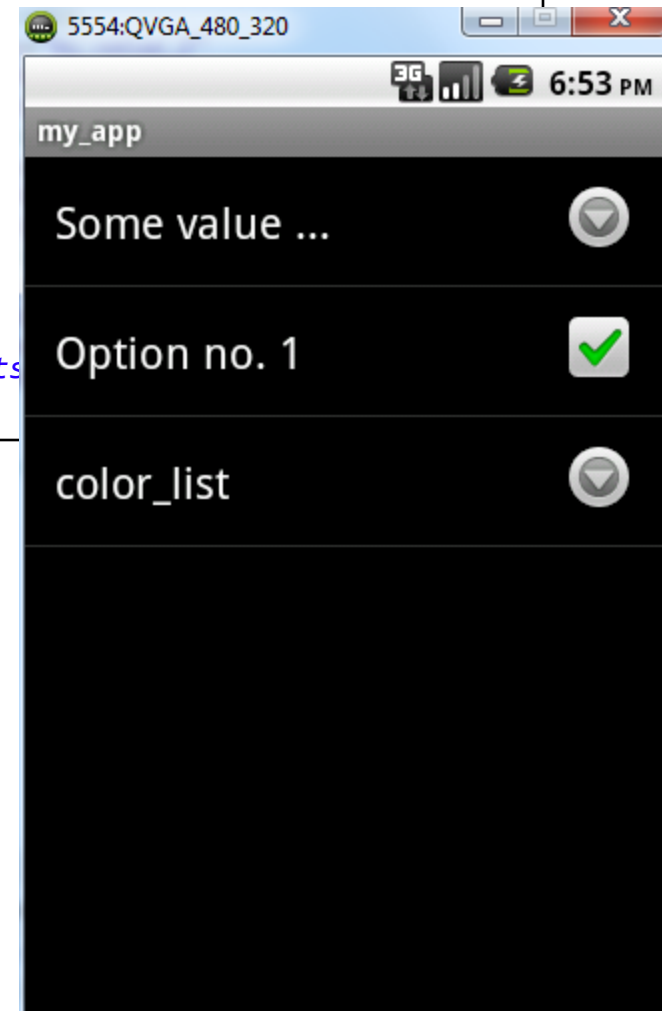
```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
    <EditTextPreference
        android:key="my_value_2"
        android:defaultValue="123"
        android:title="Some value ..."/>
    <CheckBoxPreference
        android:key="option_1"
        android:defaultValue="true"
        android:title="Option no. 1" />
</PreferenceScreen>
```



# Shared Preferences - XML

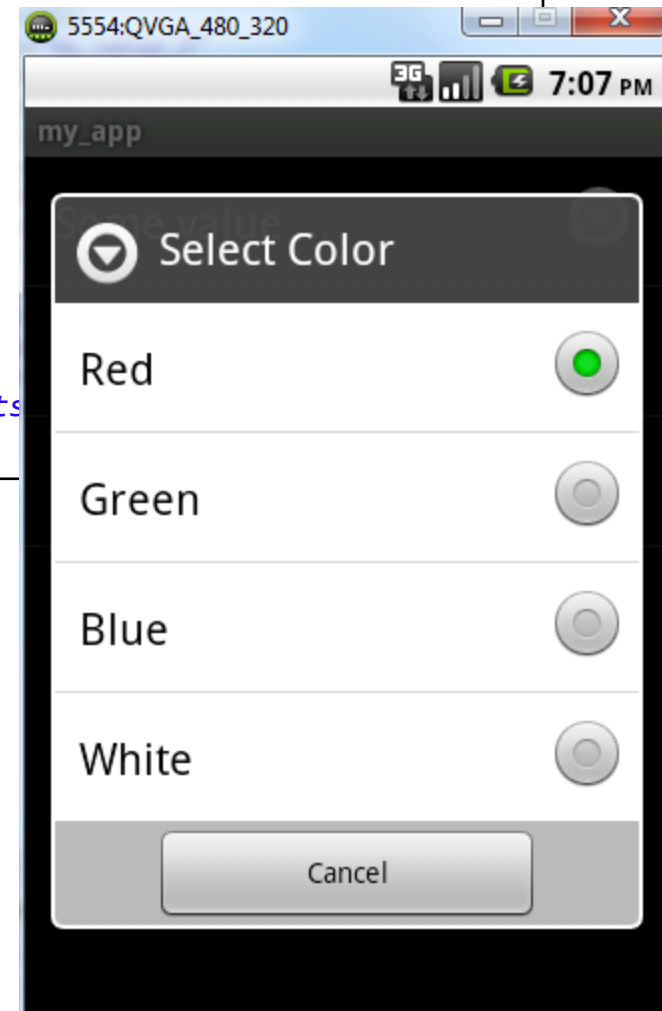
```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
    <EditTextPreference
        android:key="my_value_2"
        android:defaultValue="123"
        android:title="Some value ..."/>
    <CheckBoxPreference
        android:key="option_1"
        android:defaultValue="true"
        android:title="Option no. 1" />
    <ListPreference
        android:key="list_color"
        android:title="color_list"
        android:entries="@array/color_list"
        android:entryValues="@array/color_list_results"
    />
</PreferenceScreen>
```

```
<resources>
    <string-array name="color_list">
        <item>Red</item>
        <item>Green</item>
        <item>Blue</item>
        <item>White</item>
    </string-array>
    <string-array name="color_list_results">
        <item>0</item>
        <item>1</item>
        <item>2</item>
        <item>3</item>
    </string-array>
</resources>
```



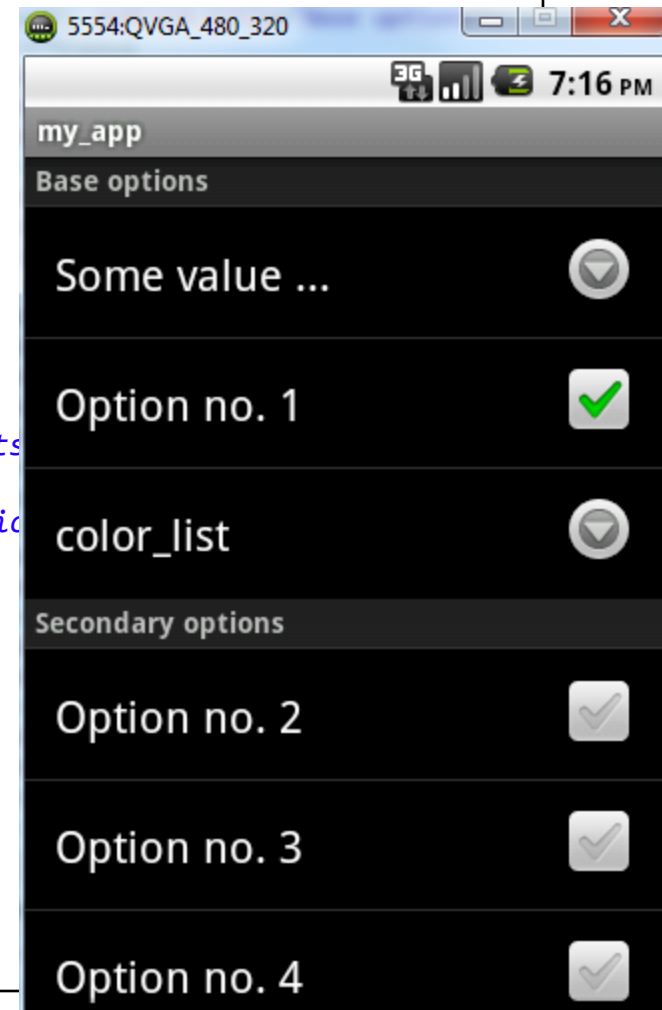
# Shared Preferences - XML

```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
    <EditTextPreference
        android:key="my_value_2"
        android:defaultValue="123"
        android:title="Some value ..."/>
    <CheckBoxPreference
        android:key="option_1"
        android:defaultValue="true"
        android:title="Option no. 1" />
    <ListPreference
        android:key="list_color"
        android:title="color_list"
        android:dialogTitle="Select Color"
        android:entries="@array/color_list"
        android:entryValues="@array/color_list_results"
    </ListPreference>
</PreferenceScreen>
```



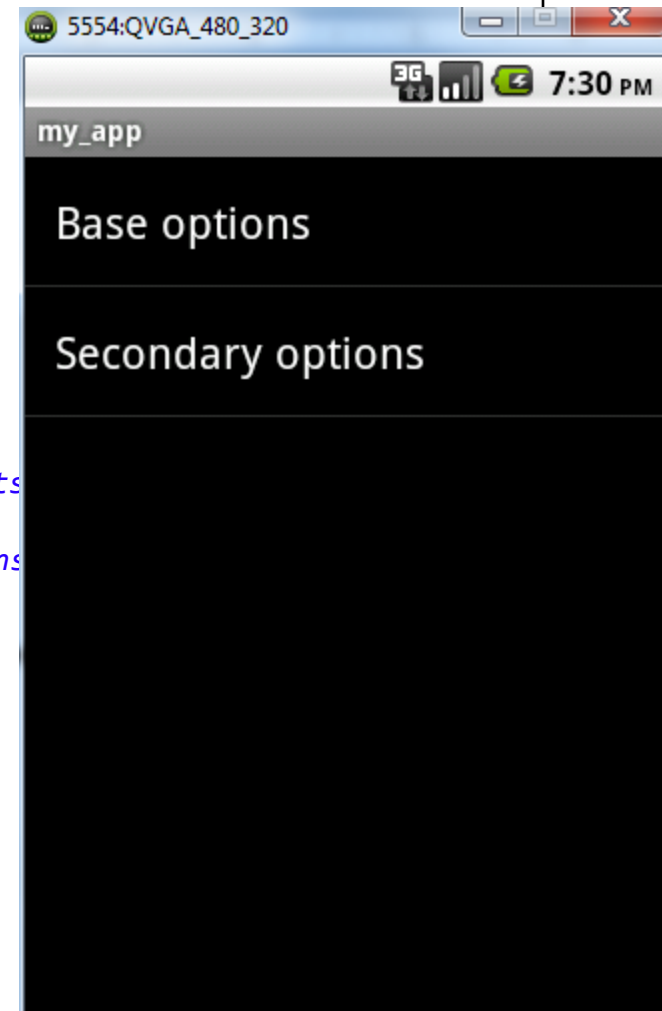
# Shared Preferences - XML

```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
  <PreferenceCategory android:title="Base options">
    <EditTextPreference
      android:key="my_value_2"
      android:defaultValue="123"
      android:title="Some value ..." />
    <CheckBoxPreference
      android:key="option_1"
      android:defaultValue="true"
      android:title="Option no. 1" />
    <ListPreference
      android:key="list_color"
      android:title="color_list"
      android:dialogTitle="Select Color"
      android:entries="@array/color_list"
      android:entryValues="@array/color_list_results" />
  </PreferenceCategory>
  <PreferenceCategory android:title="Secondary options">
    <CheckBoxPreference
      android:key="option_2"
      android:title="Option no. 2" />
    <CheckBoxPreference
      android:key="option_3"
      android:title="Option no. 3" />
    <CheckBoxPreference
      android:key="option_4"
      android:title="Option no. 4" />
  </PreferenceCategory>
</PreferenceScreen>
```



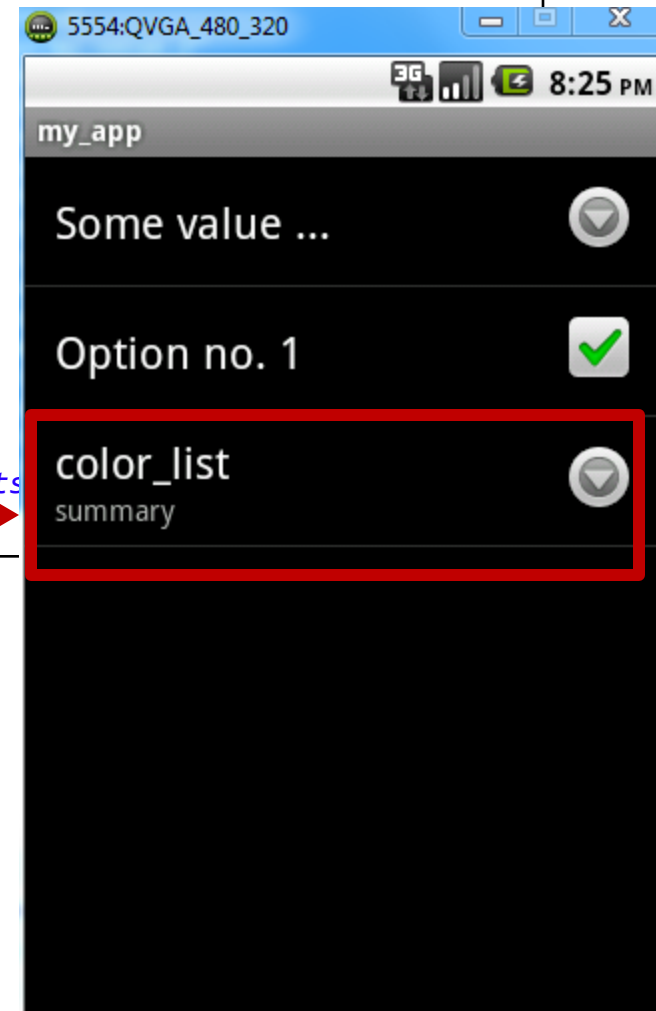
# Shared Preferences - XML

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
  <PreferenceScreen android:title="Base options" >
    <EditTextPreference
      android:key="my_value_2"
      android:defaultValue="123"
      android:title="Some value ..." />
    <CheckBoxPreference
      android:key="option_1"
      android:defaultValue="true"
      android:title="Option no. 1" />
    <ListPreference
      android:key="list_color"
      android:title="color_list"
      android:dialogTitle="Select Color"
      android:entries="@array/color_list"
      android:entryValues="@array/color_list_results" />
  </PreferenceScreen>
  <PreferenceScreen android:title="Secondary options" >
    <CheckBoxPreference
      android:key="option_2"
      android:title="Option no. 2" />
    <CheckBoxPreference
      android:key="option_3"
      android:title="Option no. 3" />
    <CheckBoxPreference
      android:key="option_4"
      android:title="Option no. 4" />
  </PreferenceScreen>
</PreferenceScreen>
```



# Shared Preferences - XML

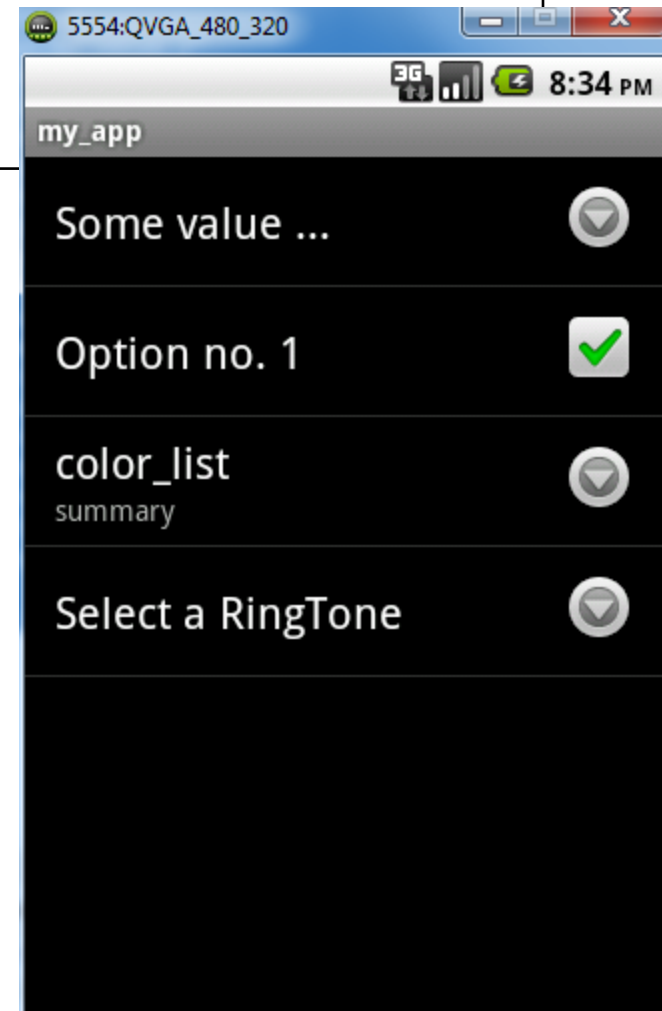
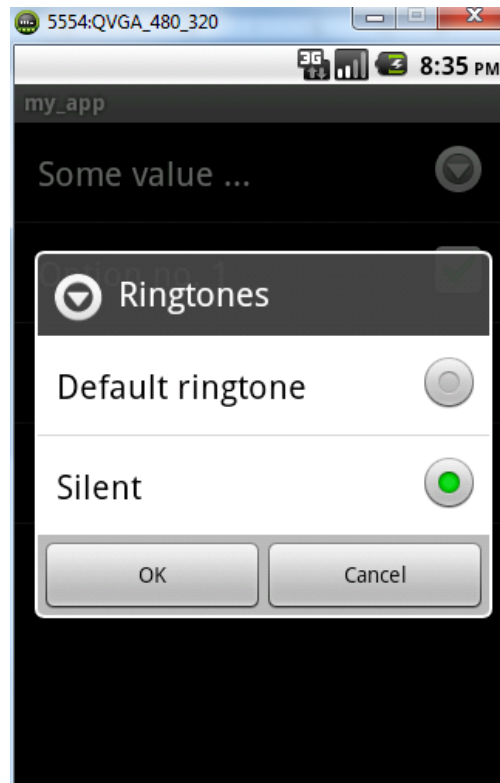
```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
    <EditTextPreference
        android:key="my_value_2"
        android:defaultValue="123"
        android:title="Some value ..."/>
    <CheckBoxPreference
        android:key="option_1"
        android:defaultValue="true"
        android:title="Option no. 1" />
    <ListPreference
        android:key="list_color"
        android:dialogMessage="Test message"
        android:summary="summary"
        android:title="color_list"
        android:entries="@array/color_list"
        android:entryValues="@array/color_list_results"
    </ListPreference>
</PreferenceScreen>
```





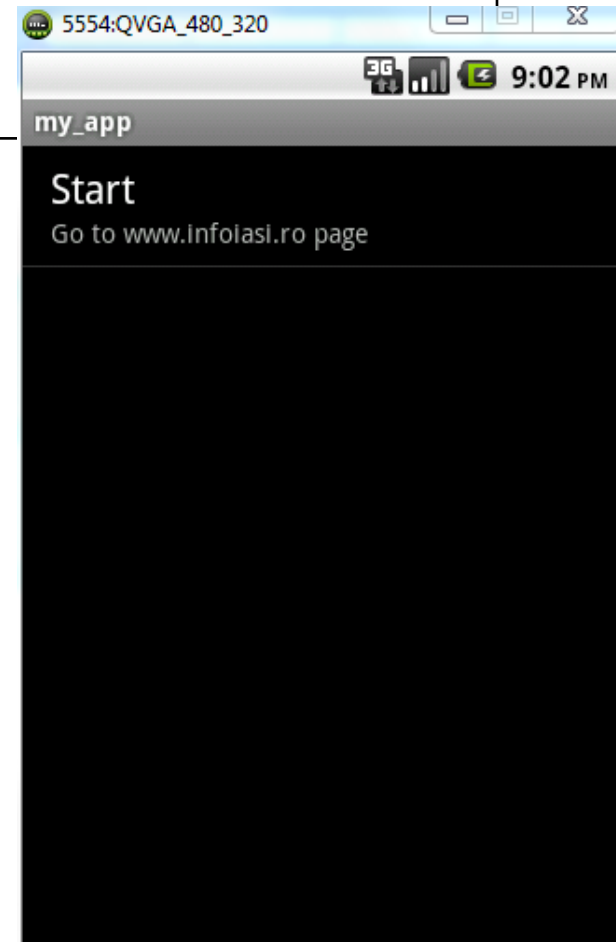
# Shared Preferences - XML

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
...
<RingtonePreference
    android:title="Select a RingTone"
    android:ringtoneType="alarm" />
</PreferenceScreen>
```



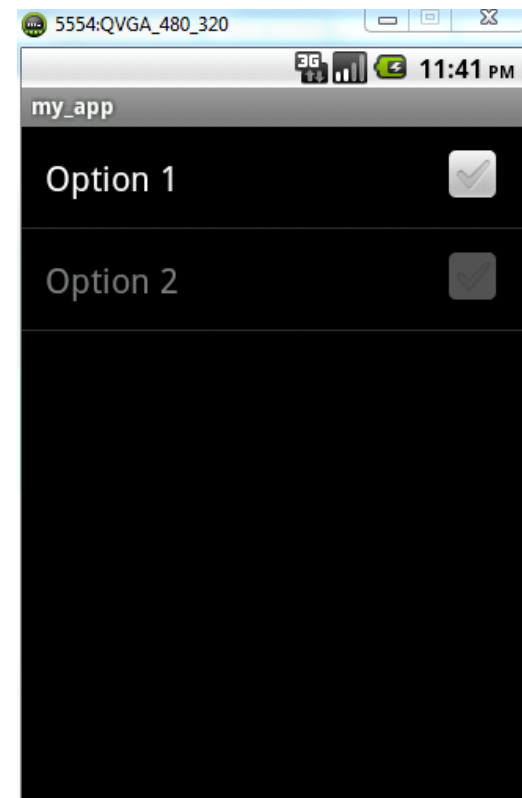
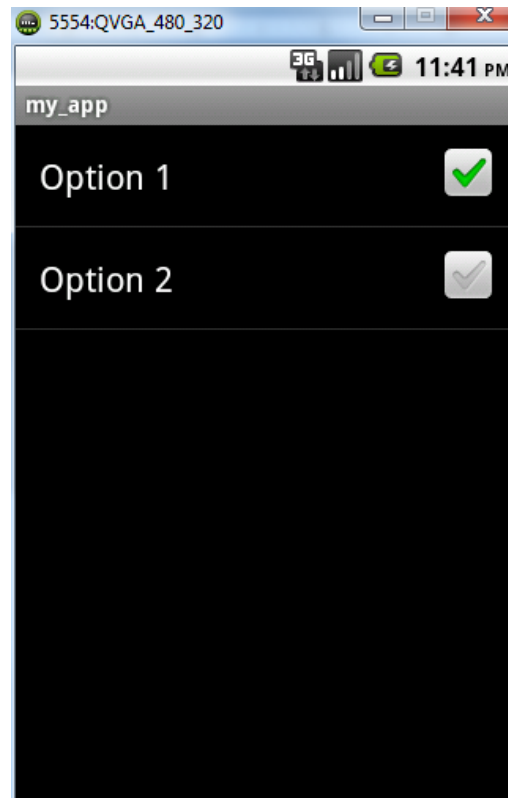
# Shared Preferences - XML

```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
    <Preference
        android:title="Start"
        android:summary="Go to www.infoiasi.ro page" >
        <intent
            android:action="android.intent.action.VIEW"
            android:data="http://www.infoiasi.ro" />
    </Preference>
</PreferenceScreen>
```



# Shared Preferences - XML

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
    <CheckBoxPreference
        android:key="opt_1"
        android:title="Option 1" />
    <CheckBoxPreference
        android:title = "Option 2"
        android:dependency="opt_1" />
</PreferenceScreen>
```





# Internal storage

- Represented by a location in the device internal memory
- The files saved into the internal storage can only be used by the application that saves them
- Once the application is uninstalled, these files are deleted

# Internal storage - Example

```
public class MainActivity extends Activity {  
  
    protected void SaveFileToInternalStorage() {  
        FileOutputStream fos;  
        try {  
            fos = openFileOutput("internal_file.txt", Context.MODE_PRIVATE);  
            fos.write("test".getBytes());  
            fos.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```



/data/data/com.example.studenti/files → internal\_file.xml

- Other functions associated with internal storage (members of the **Context** class)
  - openFileInput → opens a private file for reading
  - getFilesDir → returns the directory where the internal files are kept
  - getDir → to create/retrieve a directory
  - deleteFile → deletes an internal storage file

# External storage

- SD-Card (requires `WRITE_EXTERNAL_STORAGE` permissions)
- Obtained using: `getExternalStorageDirectory`, `getExternalStoragePublicDirectory`,
- `getExternalStoragePublicDirectory` can be used to obtain the location of some special directories such as:
  - Music
  - Podcasts
  - Ringtones
  - Alarms
  - Notifications
  - Pictures
  - Movies
  - Downloads

# SQLite databases

```
public class MainActivity extends Activity {
    private SQLiteDatabase myDB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(...);
        OpenDB();
    }
    public void OpenDB() {
        SQLiteOpenHelper sql = new SQLiteOpenHelper(this,"my_db",null,1) {
            @Override
            public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
            }
            @Override
            public void onCreate(SQLiteDatabase db) {
            }
        };
        myDB = sql.getWritableDatabase();
        myDB.execSQL("CREATE TABLE names (name TEXT)");
    }
}
```



/data/data/com.example.surfaceviewtest/databases →my\_db