

# Deljena podešavanja i rad sa datotekama

## Mobilne aplikacije

# Pregled sadržaja

1 Deljena podešavanja

2 Datoteke

# Deljena podešavanja

- Deljena podešavanja (SharedPreferences) olakšavaju perzistentno skladištenje prostih tipova podataka
- Ti podaci se skladište u datoteci kao uređeni parovi (ključ, vrednost)

# Deljena podešavanja

- Deljenim podešavanjima se može pristupiti metodom `SharedPreferences.getSharedPreferences(String name, int mode)`
- Ova metoda je definisana u klasi `Context`, pa je samim tim dostupna i u okviru njenih naslednica, kao što su `Activity`, `Service` i `IntentService`.
- Moguće je koristiti više skupova deljenih podešavanja čija imena se navode kao parametar (`name`).

# Deljena podešavanja

Konstanta	Opis
MODE_PRIVATE	The created file can only be accessed by the calling application

Table 1: Vrednosti parametra "mode".

Ostali režimi su označeni kao deprecated.

# Deljena podešavanja

- U klasi Activity je definisana metoda `SharedPreferences getPreferences(int mode)` koja omogućava pristup skupu podešavanja te aktivnosti.
- Za razliku od metode `getSharedPreferences`, ova metoda koristi podrazumevan naziv skupa podešavanja koji čine naziv paketa i naziv klase u kojoj je aktivnost implementirana.

# Deljena podešavanja

- Uz korišćenje paketa `androidx.preferences` na raspolaganju je i pristup podrazumevanom skupu podešavanja:

```
SharedPreferences settings = PreferenceManager.getDefaultSharedPreferences(this);
```

pri čemu je naziv tog skupa sastavljen od naziva paketa i sufiksa `"_preferences"`.

Metoda	Naziv datoteke	Opis
<code>getSharedPreferences("some_name",MODE_PRIVATE)</code>	<code>some_name.xml</code>	Poizvoljan naziv skupa podešavanja
<code>getPreferences(MODE_PRIVATE)</code>	<code>activityname.xml</code>	Podešavanja koja se odnose na jednu aktivnost
<code>PreferenceManager.getDefaultSharedPreferences(this)</code>	<code>packagename_preferences.xml</code>	Podešavanja koja se odnose na kontekst (podrazumevano na paket)

Table 2: Načini pristupa podešavanjima.

# Deljena podešavanja - upis vrednosti

Vrednosti prostog tipa T mogu se zapisati u tri koraka:

- 1 Pozvati `edit()` metodu koja započinje transakciju
- 2 Dodati vrednost(i) tipa T metodama oblika `SharedPreferences.Editor putT(String key, T value)`
- 3 Pozvati `commit()` metodu koja završava transakciju u sinhronom modu (i vraća rezultat o uspehu zapisivanja podataka) ili `apply()` koja to radi u asinhronom modu.

U zavisnosti od tipa T, na raspolaganju su metode:

`putBoolean`, `putFloat`, `putInt`, `putLong`, `putString`,  
`putStringSet`.

Ove metode vraćaju tip `SharedPreferences.Editor` tako da je olakšano ulančavanje višestrukih upisa vrednosti.



# Deljena podešavanja

- Zapisane vrednosti mogu se pročitati metodama oblika `T getT(String key, T defaultValue)`
- U zavisnosti od tipa `T`, na raspolaganju su metode: `getBoolean`, `getFloat`, `getInt`, `getLong`, `getString`, `getStringSet`
- S obzirom na to da su nazivi skupova podešavanja i nazivi pojedinačnih vrednosti predstavljeni stringovima, može biti korisno njihovo ekternalizovanje (`R.string.xyz`) jer se u projektu mogu koristiti po više puta.

# ExampleActivity.java

```
1  SharedPreferences settings = getPreferences(Context.MODE_PRIVATE);  
2  SharedPreferences.Editor editor = settings.edit();  
   editor.putBoolean("silentMode", silentMode);  
4  editor.putInt("refreshInterval", refreshInterval);  
   editor.commit();  
6
```

Ili kraće (sa ulančanim pozivima metoda):

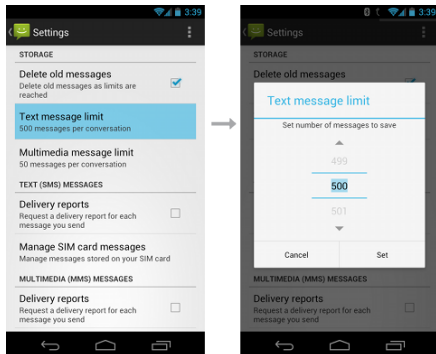
```
   getPreferences(MODE_PRIVATE).edit()  
2       .putBoolean("silentMode", silentMode)  
       .putInt("refreshInterval", refreshInterval)  
4       // ...  
       .commit();  
6
```

# ExampleActivity.java

```
    SharedPreferences settings = getPreferences(Context.MODE_PRIVATE);  
2  boolean silentMode = settings.getBoolean("silentMode", false);  
    int refreshInterval = settings.getInt("refreshInterval", 1000);
```

4

# PreferenceActivity



- Mnoge aplikacije omogućavaju korisnicima konfiguiranje
- U tu svrhu treba koristiti PreferenceFragmentCompat kako bi korisnici imali konzistentan grafički korisnički interfejs (i da bi sebi olakšali posao)

Figure 1: PreferenceFragmentCompat.

# Deljena podešavanja

- Za upravljanje podešavanjima putem korisničkog interfejsa je namenjena klasa `PreferenceFragmentCompat` čijim nasleđivanjem definišemo sopstvene fragmente za rad sa podešavanjima.
- `PreferenceFragmentCompat` je definisan u paketu koji je potrebno dodati u Gradle skriptu projekta kao zavisnost:  
`implementation 'androidx.preference:preference:1.2.0'`
- Ovakav fragment dinamički kreira svoj izgled ako pri njegovom kreiranju pozovemo metodu `setPreferencesFromResource` kojom se učitava XML resurs sa definisanim podešavanjima (obrađeno na predavanjima GUI III).
- Postupak kreiranja pomenute aktivnosti i fragmenta Android Studio olakšava pomoću wizard-a: File -> New -> Activity -> Settings Activity.

# AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
    <application ... >
4         <activity android:name=". SettingsActivity">
            <!-- ... -->
6         </activity>
    </application>
8 </manifest>
```

# SettingsActivity.java

```

1  public class SettingsActivity extends AppCompatActivity {
2
3      @Override
4      protected void onCreate(Bundle savedInstanceState) {
5          super.onCreate(savedInstanceState);
6          setContentView(R.layout.settings_activity);
7          if (savedInstanceState == null) {
8              getSupportFragmentManager()
9                  .beginTransaction()
10                 .replace(R.id.settings, new SettingsFragment())
11                 .commit();
12          }
13          // display back arrow in title bar
14          ActionBar actionBar = getSupportActionBar();
15          if (actionBar != null) {
16              actionBar.setDisplayHomeAsUpEnabled(true);
17          }
18      }
19      // implementation of fragment that displays preferences as GUI
20      public static class SettingsFragment extends PreferenceFragmentCompat {
21          @Override
22          public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
23              // preferences name can be specified by:
24              // getPreferenceManager().setSharedPreferencesName("some_settings_name");
25              // or default name will be used (package name with "_preferences" suffix)
26
27              setPreferencesFromResource(R.xml.app_preferences, rootKey);
28          }
29      }
30 }

```

# layout/settings\_activity.xml

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent">
4
5     <FrameLayout
6         android:id="@+id/settings"
7         android:layout_width="match_parent"
8         android:layout_height="match_parent" />
9
10 </LinearLayout>
```



# xml/app\_preferences.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <PreferenceScreen xmlns:app="http://schemas.android.com/apk/res-auto">

4   <CheckBoxPreference
5     app:key="pref_sync"
6     app:title="@string/pref_sync"
7     app:summary="@string/pref_sync_summ"
8     app:defaultValue="true" />

10  <ListPreference
11    app:dependency="pref_sync"
12    app:key="pref_syncConnectionType"
13    app:title="@string/pref_syncConnectionType"
14    app:entries="@array/pref_syncConnectionTypes_entries"
15    app:entryValues="@array/pref_syncConnectionTypes_values"
16    app:defaultValue="@string/pref_syncConnectionTypes_default" />

18 </PreferenceScreen>
```

# Datoteka sa vrednostima podešavanja

- Podešavanja aplikacije se na uređaju čuvaju u folderu:  
data/data/package\_name/shared\_prefs

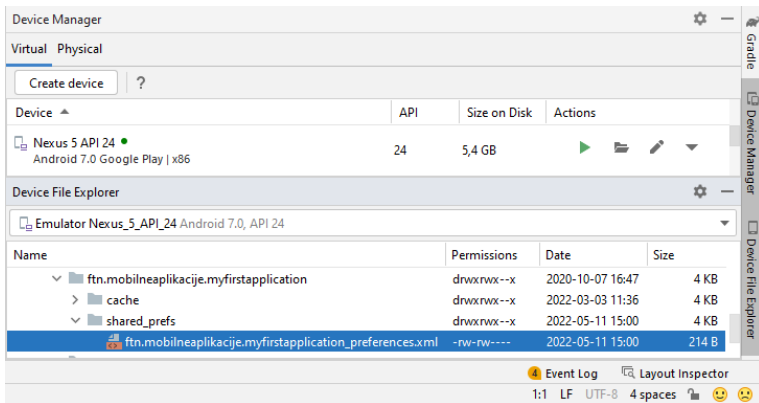


Figure 2: Datoteka sa podešavanjima

# Pristup datotekama putem ADB

- Alat ADB omogućava pristup fajl sistemima uređaja (fizičkim i virtuelnim) koji su povezani na računar.

```

C:\Users\Marko\AppData\Local\Android\Sdk\platform-tools>adb devices
List of devices attached
emulator-5554    device

C:\Users\Marko\AppData\Local\Android\Sdk\platform-tools>adb -s emulator-5554 shell
generic_x86:/ $ run-as ftn.mobilneaplikacije.myfirstapplication
generic_x86:/data/data/ftn.mobilneaplikacije.myfirstapplication $ ls
cache shared_prefs
generic_x86:/data/data/ftn.mobilneaplikacije.myfirstapplication $ cat shared_prefs/ftn.mobilneaplikacije.myfirstapplication_preferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <boolean name="pref_sync" value="true" />
  <string name="reply">reply</string>
</map>
generic_x86:/data/data/ftn.mobilneaplikacije.myfirstapplication $
  
```

Build Variants

Run Problems Terminal Logcat Build TODO Profiler App Inspection

Your anti-virus program might be impacting your build performance. Android Studio checked the following directories: // C:\Users\Marko\gradle // C:\Users\Marko\AppData\Local\Android\Sdk // C:\Users\Mar

Figure 3: Izlistavanje sadržaja datoteke sa podešavanjima

# Pregled sadržaja

1 Deljena podešavanja

2 Datoteke

# Datoteke

- Podaci koji se nalaze u operativnoj memoriji se ne čuvaju kada se uništi proces
- Komponente koje se nalaze u različitim procesima ne mogu da razmenjuju podatke koji se nalaze u operativnoj memoriji (ne dele adresni prostor)
- Najjednostavniji način da se prevaziđu ova ograničenja je korišćenje datoteka

# Datoteke

- Za rad sa datotekama koriste se klase iz java.io paketa (na isti način na koji se koriste u Java SE)
- Međutim, mogu se koristiti metode klase Context koje olakšavaju pristup internom i/ili eksternom skladištu podataka, rad sa privremenim datotekama i upravljanje pravima pristupa
  - `FileInputStream openFileInput(String name)`
  - `FileOutputStream openFileOutput(String name, int mode)`
  - `String[] fileList()` // vraća listu datoteka u folderu kojem samo tekuća aplikacija ima pristup (privatan folder)
  - `boolean deleteFile(String name)` // briše navedenu datoteku u privatnom folderu
  - `File getDir(String name, int mode)` // kreira ili pristupa folderu unutar privatnog foldera
  - `File getCacheDir()` //folder za privremene fajlove
  - `File getExternalCacheDir()` //folder za privremene fajlove na eksternom skladištu
  - `File getFilesDir()` //privatan folder - kome samo tekuća aplikacija ima pristup
  - `File getExternalFilesDir(String type)` // folder kome i druge aplikacije mogu pristupiti

## ExampleService.java

```
private void write() {  
2   FileOutputStream fos = null;  
   try {  
4       fos = openFileOutput("test.txt", Context.MODE_PRIVATE);  
       fos.write(bytes);  
6   } catch (FileNotFoundException e) {  
       Log.e(Constants.LOG_TAG, "File not found", e);  
8   } catch (IOException e) {  
       Log.e(Constants.LOG_TAG, "IO problem", e);  
10  } finally {  
       try {  
12         fos.close();  
        } catch (IOException e) {  
14        }  
    }  
16 }
```

# Datoteke

Konstanta	Opis
MODE_PRIVATE	The created file can only be accessed by the calling application
MODE_APPEND	if the file already exists then write data to the end of the existing file instead of erasing it
MODE_WORLD_READABLE	Allow all other applications to have read access to the created file (deprecated)
MODE_WORLD_WRITEABLE	Allow all other applications to have write access to the created file (deprecated)

Table 3: Vrednosti parametra "mode".



# ExampleService.java

```
private void read() {
2   FileInputStream fis = null;
   Scanner scanner = null;
4   StringBuilder sb = new StringBuilder();

6   try {
       fis = openFileInput("test.txt");
       scanner = new Scanner(fis);
       while (scanner.hasNextLine()) {
10      sb.append(scanner.nextLine() + LINE_SEP);
       }
12  } catch (FileNotFoundException e) {
       Log.e(Constants.LOG_TAG, "File not found", e);
14  } finally {
       if (fis != null) {
16      try {
           fis.close();
18      } catch (IOException e) {
           }
20      }
       if (scanner != null) {
22      scanner.close();
       }
24  }
}
```

# Datoteke

- Interno skladište podataka se nalazi u mobilnom uređaju
  - Uvek je dostupno
  - Obično je manjeg kapaciteta (i nije ga moguće proširiti)
  - Privatno je
- Eksterno skladište podataka se (obično) nalazi na SD kartici
  - Nije uvek dostupno
  - Obično je većeg kapaciteta (i moguće ga je proširiti)
  - Javno je

# ExampleService.java

```
String state = Environment.getExternalStorageState();
2 if (Environment.MEDIA_MOUNTED.equals(state)) {
    // We can read and write the media
4 } else if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
    // We can only read the media
6 } else {
    // Something else is wrong. It may be one of many other states,
8 // but all we need to know is we can neither read nor write
    }
10
```

# Datoteke

- Privremene datoteke treba skladištiti u cache direktorijumu (Android ih automatski briše kada ponestane slobodnog prostora)
  - `File getCacheDir()`
  - `File getExternalCacheDir()`
- Datoteke koje deli više aplikacija treba snimiti u javni eksterni direktorijum
  - `File getExternalFilesDir(String type)`
  - `File[] getExternalFilesDirs(String type)`

# Datoteke

- Pri korišćenju eksternog skladišta potrebno je obezbediti statičke (putem manifest fajla) i dinamičke (putem dijaloga) permisije:
  - `<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />`
  - `<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />`
- Permisije nisu neophodne kada se koriste direktorijumi namenjeni tekućoj aplikaciji odn. direktorijumi koje vraćaju metode `getExternalFilesDir()` i `getExternalCacheDir()`.

# Konstante klase Environment

Konstanta	Opis
DIRECTORY_ALARMS	Standard directory in which to place audio files for alarms
DIRECTORY_DCIM	The traditional location for pictures and videos when mounting the device as a camera.
DIRECTORY_DOCUMENTS	Standard directory in which to place documents that have been created by the user.
DIRECTORY_DOWNLOADS	Standard directory in which to place files that have been downloaded by the user.
DIRECTORY_MOVIES	Standard directory in which to place movies that are available to the user.

Table 4: Tip javnog eksternog direktorijuma.

# Konstante klase Environment

Konstanta	Opis
DIRECTORY_MUSIC	Standard directory in which to place any audio files that should be in the regular list of music for the user.
DIRECTORY_NOTIFICATIONS	Standard directory in which to place any audio files that should be in the list of notifications that the user can select.
DIRECTORY_PICTURES	Standard directory in which to place pictures that are available to the user.
DIRECTORY_PODCASTS	Standard directory in which to place any audio files that should be in the list of podcasts that the user can select (not as regular music).
DIRECTORY_RINGTONES	Standard directory in which to place any audio files that should be in the list of ringtones that the user can select (not as regular music).

Table 5: Tip javnog eksternog direktorijuma.