

Android alkalmazásfejlesztés

UI megismerése - 2. rész
Adattárolás

Sicz-Mesziár János

sicz.mj@gmail.com

2011. március 10.

OE-NIK



AlertDialog és ProgressDialog

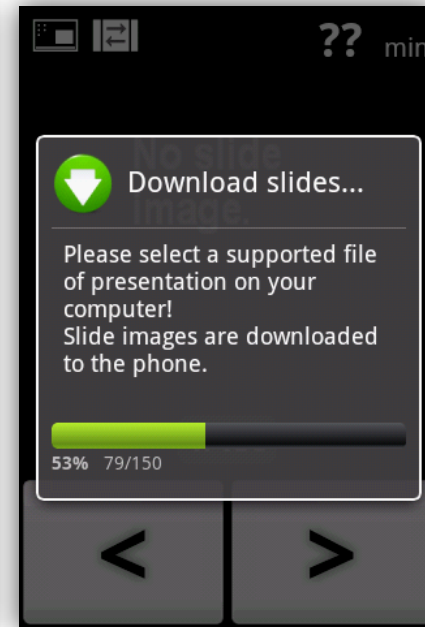
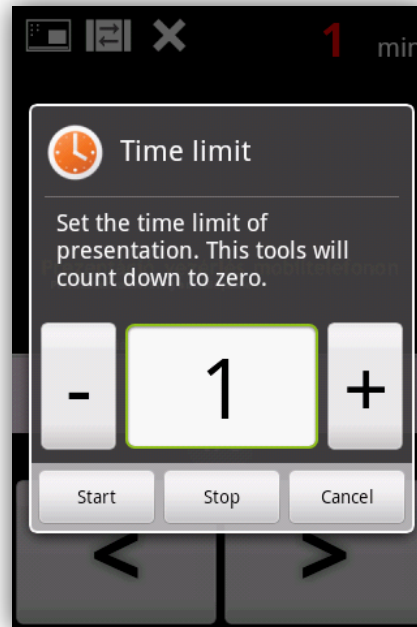
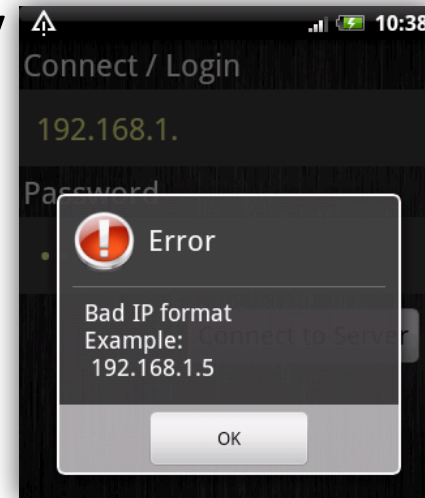
- ◉ Egy párbeszédablak, ami az aktuális Activity előtt jelenik meg.
- ◉ Dialog osztály leszármazottja.
- ◉ Testreszabhatjuk (Ikon, szöveg, gombok)
- ◉ XML fájlal egyéni felületet adhatunk hozzá
- ◉ Beépített gombok:
 1. Pozitív (*PositiveButton*)
 2. Negatív (*NegativeButton*)
 3. Semleges (*NeutralButton*)

AlertDialog példa:

<http://developer.android.com/guide/topics/ui/dialogs.html#AlertDialog>

ProgressDialog példa:

<http://developer.android.com/guide/topics/ui/dialogs.html#ProgressDialog>



Néhány XML leíró bemutatása

- ◉ Layer list

Több kép együttes rétegkezelése.



- ◉ State list: egy állapotlista

Állapotokhoz kapcsolt képek meghatározása.

Pl.: Focus, Press

- ◉ Level list

Különböző szintekhez képek kapcsolása.

Pl.: Fényerő (%), Wi-Fi jelerősségek (dB)

- ◉ Shape drawable: alakzatok leírása XML-ben

Pl.: alak meghatározása, sarkok lekerekítése,
színátmenetek, margók, méret, ...



Menü (Options menu)

☉ Képernyő alján megjelenő opciók.

Több, mint 6 elem esetén a 6. elem „More” lesz és mögötte található a maradék menüpont.

1. Menü elemeinek definiálása XML-ben (*res/menu/*.xml*)

■ **<menu>**

■ **<item>**

android:id

android:icon

android:title >



■ **<group>** csak fejlesztőknek szól, megjelenés nem változik

2. onCreateOptionsMenu() metódus felülírása.

(Csak első alkalommal hívódik meg)

3. onOptionsItemSelected() metódus felülírása.

Adattárolás



Shared Preference

- ⊙ Egy egyszerű megoldás primitív értékek perzisztens tárolására.
- ⊙ Az alábbiakat lehet: boolean, float, int, long, string.
- ⊙ **Nem felhasználói beállítások céljából! (arra van jobb)**

- ⊙ **Tárolás:**

```
String PrefFileName = "MyPrefName";  
SharedPreferences settings = getSharedPreferences(PrefFileName, 0);  
SharedPreferences.Editor editor = settings.edit();  
editor.putInt("test", 1027);  
editor.commit();
```

- ⊙ **Visszaállítás:**

```
SharedPreferences settings = getSharedPreferences(PrefFileName, 0);  
int szam = settings.getInt("test", 0);
```

Belső tároló

- ◉ Fájlmentés közvetlenül a belsőtárolóra.
- ◉ Ha az alkalmazást törlik, akkor ezek a fájlok is törlődnek!
- ◉ *Alapértelmezetten* más alkalmazás nem fér hozzá.

- ◉ Tárolás:

```
String FILENAME = "StorageFile,,";  
String saveThis = "Helló Storage!";  
FileOutputStream fos = openFileOutput(FILENAME, MODE_PRIVATE);  
fos.write(saveThis.getBytes());  
fos.close();
```

- ◉ Betöltés:

```
FileInputStream fis = openFileInput(FILENAME);  
byte[] buffer = new byte[1024]; int len;  
while((len = fis.read(buffer)) > 0)  
    Log.d("NIK", new String(buffer, 0, len));  
fis.close();
```

- ◉ Továbbá: `fileList()` , `deleteFile()`, `getDir()`, ...



Külső tároló (SD-kártya)

- ⊙ Bármely másik alkalmazás (v. felhasználó) által olvasható és írható, osztott háttértár.
- ⊙ Bármikor eltávolíthatja a felhasználó.



Ellenőrizni kell az elérhetőségét és állapotát:

```
String state = Environment.getExternalStorageState();
if(state.equals(Environment.MEDIA_MOUNTED)){
    // Elérhető, írható és olvasható
}else if(state.equals(Environment.MEDIA_MOUNTED_READ_ONLY)){
    // Elérhető, de csak olvasható
}else{ // Más állapotban van... DE se nem írható, se nem olvasható }
```

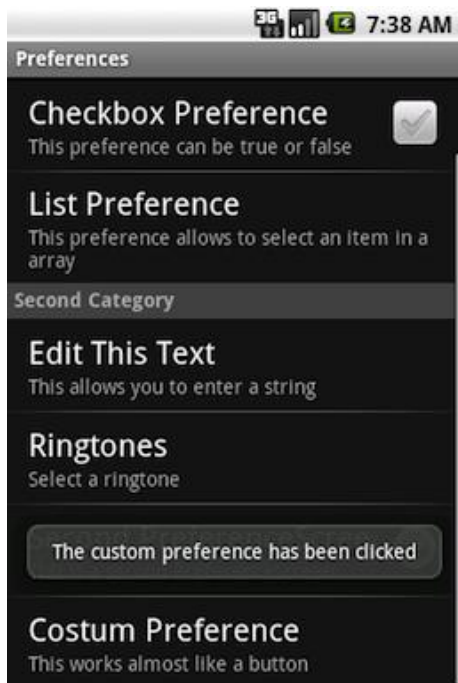
```
String sdcard =
Environment.getExternalStorageDirectory().getAbsolutePath();
```


Adatbázis használata (SQLite)

- ◎ Teljes SQLite támogatás.
- ◎ Alkalmazáson belül bárhonnán elérhető, de azon kívül nem!
- ◎ De ContentProvider-ek segítségével lehetséges az alkalmazások közötti adatcsere is.
Pl.: Böngésző könyvjelzők, előzmények, kontaktok, ...
- ◎ Megszokott SQL kódok használata: INSERT, SELECT, UPDATE, DELETE, CREATE, ...

- ◎ Bővebben:
 - [Alapok, egyszerű SQLite megoldás](#)
 - [SQLiteOpenHelper használata](#)
 - [Content Provider](#)

PreferenceActivity



- ◉ Kimondottan felhasználói beállítások tárolása.
 - ◉ Néhány beépített form:
CheckBoxPreference, EditTextPreference, ListPreference, RingtonePreference + **Egyéni!**
 - ◉ Témák szerint kategorizálhatunk.
 - ◉ Preference XML létrehozása:
 1. File > New > Other... > Android XML File
 2. Layout helyett **Preference** típus!
 3. Add > ChechBoxPreference, ...
 - ◉ Java osztály:
 1. extends PreferenceActivity
 2. onCreate() implementálása
 3. addPreferencesFromResource()
- + **AndroidManifest.xml**

Jöjjön egy kis feladat!



OE-NIK tanárkereső

File kezelés - külső és/vagy belső tároló elérése

SQLite adatbázis használata

Dialog ablakok készítése

Menü létrehozása

Feladat

- ⊙ Készítsük el az OE-NIK tanárkereső alkalmazását, mely az alábbiakra képes:
 - A felület legyen egy TabLayout:
 - 1. fül: rákereshetünk tanárookra (pl.: név és/vagy szoba alapján)
 - 2. fül: pedig frissíthetjük az adatbázist egy gombbal
 - Adatbázis frissítés:

Az alábbi linkről letöltött fájlból kiolvassa az aktuális elérhetőségeket és ezeket feltölti egy SQLite adatbázisba:
<http://nik.uni-obuda.hu/malk/android/segedlet/03 - contact.csv>
 - Keresés:

Minimum egy beviteli mező és egy „Keresés gomb”.

A találatokat egy táblázatba betölti.
 - „Új keresés” indítása a menüből legyen elérhető, és kérdezzen rá egy Dialog ablakban, hogy tényleg szeretnék-e egy új keresést!