

ANDROID III.

4. týden, KIV/MKZ 2016

L. Pešička

OBSAH

- ◉ case study: Kvíz
- ◉ notifikace
- ◉ UI: webový prohlížeč
- ◉ ukládání dat
- ◉ grafika

CASE STUDY - APLIKACE KVÍZ

- ukázka jednoduché aplikace
- prvky, co jsme již probrali
- semestrální práce - daleko komplexnější

URL: http://mobil.idnes.cz/naucime-vas-programovat-aplikace-pro-android-zaciname-prave-dnes-phe-/aplikace.aspx?c=A120410_125436_aplikace_ham

třída	popis
MainActivity	Aktivita, zobrazí hlavní nabídku - 3 tlačítka
AboutActivity	Aktivita, Info o aplikaci
Question	Otázka - text, správná odpověď, 3 odpovědi
QuizActivity	Aktivita, zobrazení a vyhodnocení kvízu

CASE STUDY - KVÍZ

⦿ uložení otázek

- `ArrayList<Question>`
- všechny otázky uložené v array listu

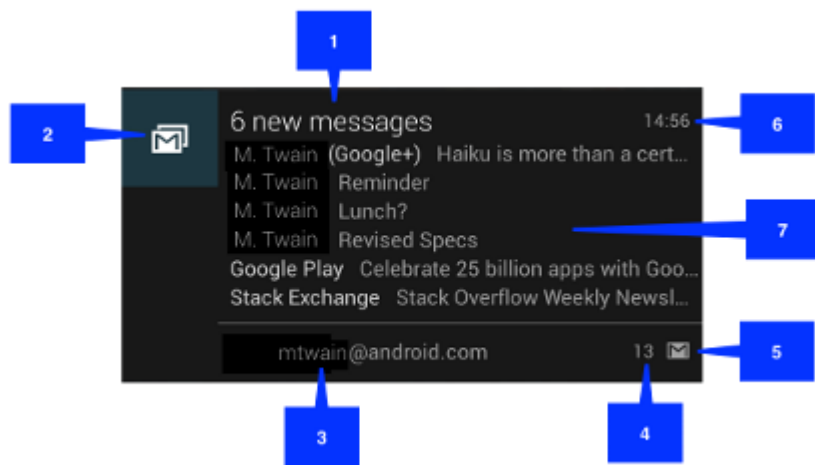
⦿ čísla položených otázek

- `Set<Integer> qs = new HashSet<Integer>()`
- ukládá čísla položených otázek, aby se neopakovala při losování čísel
- použití ve funkci `whichQuestion()`

NOTIFIKACE



1. titulek
2. velká ikona
3. text
4. počet
`.setNumber()`
5. malá ikona
6. čas notifikace
7. podrobné info



obrázky z Android SDK

NOTIFIKACE

- ◉ notifikační objekt
 - malá ikona
 - titulek
 - text
 - další volitelné prvky
- ◉ notifikační manager
 - `NotificationManager.notify()`
 - provede notifikaci s příslušným ID
- ◉ více notifikací od **stejně** aplikace se stejným id = updatuje položku

NOTIFIKACE A AKCE

- ◉ notifikace může obsahovat akce - otevření aktivity
- ◉ akce je definována jako `PendingIntent`
- ◉ nastavení akce, když uživatel klikne na notifikaci: nastavit pomocí metody `setContentIntent()` notifikačního manageru

PENDING INTENT

⦿ Běžný intent

- Vytvořím intent a zavolám `startActivity()`

⦿ Pending intent

- Chci něco udělat v budoucnu
- Nedělám to hned, tj. nevolám `startActivity`
- Vytvořím pending intent a předám ho cizí aplikaci (NotificationManager, AlarmManager, HomeScreen,...) která jej vykoná

Podrobnější vysvětlení:

<http://stackoverflow.com/questions/2808796/what-is-pending-intent>

PENDING INTENT

A `PendingIntent` is a token that you give to a foreign application (e.g. `NotificationManager`, `AlarmManager`, Home Screen `AppWidgetManager`, or other 3rd party applications), which **allows the foreign application to use your application's permissions to execute a predefined piece of code.**

If you give the foreign application an `Intent`, and that application sends/broadcasts the `Intent` you gave, they will execute the `Intent` with their own permissions. But if you instead give the foreign application a `PendingIntent` you created using your own permission, that application will execute the contained `Intent` using your application's permission.

Regular Intent -> DOES NOT REQUIRE PENDING INTENT TO BE MADE

Pending Intent -> REQUIRES A REGULAR INTENT TO BE CREATED

Intents are of two types- **Explicit** and **Implicit**

Citace z odkazu na předchozím slidu

NOTIFIKACE - V ŘÁDKU

```
int N_ID = 55;    //id notifikace
Notification n;
NotificationCompat.Builder builder = new
    NotificationCompat.Builder(this);
n = builder
    .setContentTitle("titulek")
    .setContentText("text")
    .setSmallIcon(R.drawable.ic_launcher)
    .build();

NotificationManager notificationManager =
    (NotificationManager)
    this.getSystemService(this.NOTIFICATION_SERVICE);

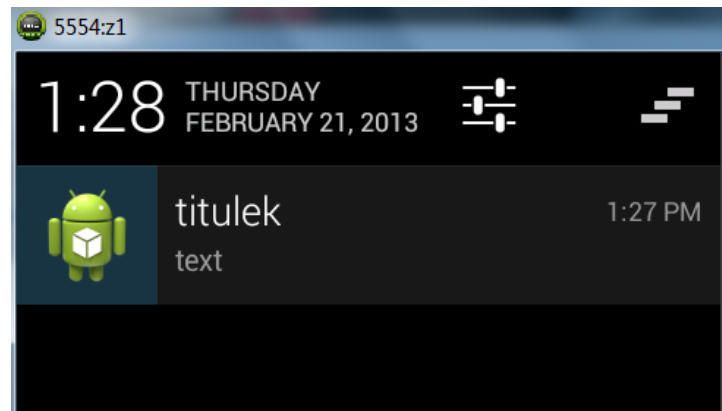
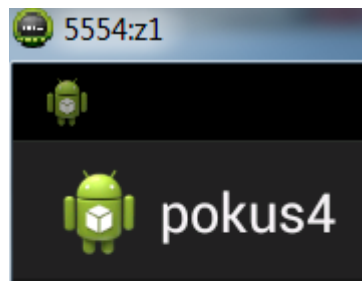
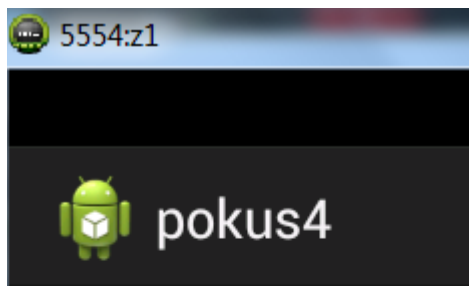
notificationManager.notify(N_ID, n);
```

notifikace

služba

zobrazení

NOTIFIKACE



<http://www.zdrojak.cz/clanky/vyvijime-pro-android-notifikace-broadcast-receivery-a-internet/>

POZNÁMKA - VYUŽITÍ PENDING INTENTU

- ◉ na Android 2.x bez PendingIntentu nezobrazil notifikaci !!
- ◉ chceme např. když klepneme na notif. zpátky do Aktivitu
- ◉ Místo emptyIntent můžeme dát vlastní akci, např. otevřít webovou stránku

```
final Intent emptyIntent = new Intent();
```

```
PendingIntent pi = PendingIntent.getActivity(this, 0,  
    emptyIntent, PendingIntent.FLAG_UPDATE_CURRENT);
```

...

do Builderu přidat:

```
.setContentIntent(pi).build();
```

DALŠÍ MOŽNOSTI NOTIFIKACE

zvuk:

```
n.defaults |= Notification.DEFAULT_SOUND;
```

nebo

```
n.sound = Uri.parse("file:///myfilepath/mysound.mp3");
```

vibrace:

```
n.defaults |= Notification.DEFAULT_VIBRATE;
```

nebo

```
{delay, length of vibration, delay, length of vibration, ...}
```

```
long[] vibration = {0,100,100,100};
```

```
n.vibrate = vibration;
```

<http://chrisrisner.com/31-Days-of-Android--Day-16%E2%80%93Notifications>

POZNÁMKA

```
notification.sound = Uri.parse("android.resource://"
    + context.getPackageName() + "/" + R.raw.sound_file);
```

DALŠÍ MOŽNOSTI NOTIFIKACE

LED dioda:

```
n.defaults |= Notification.DEFAULT_LIGHTS;
```

nebo

```
// červená 1s on a pak 1s off
```

```
n.ledARGB = 0xffff0000; // alfa, red, green, blue
```

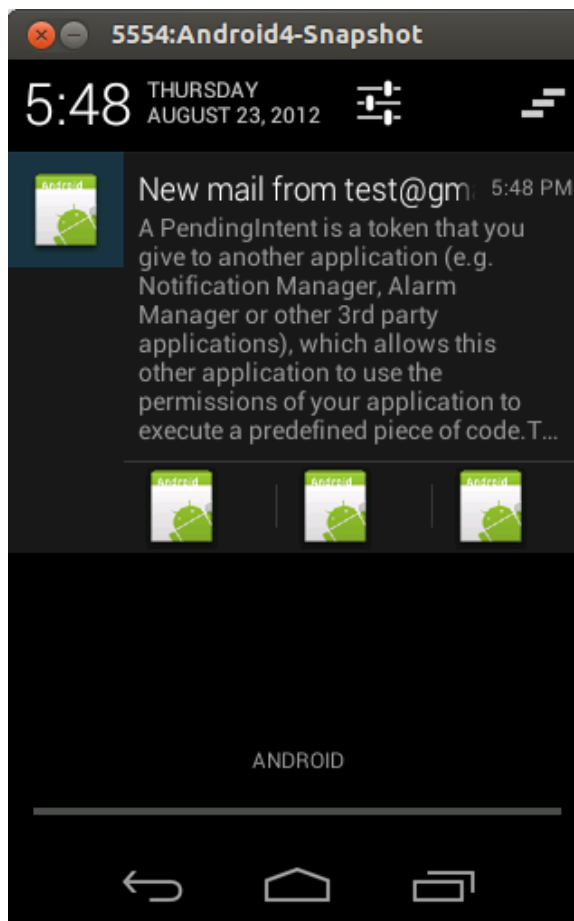
```
n.ledOnMS = 1000; // 1s on
```

```
n.ledOffMS = 1000; // 1s off
```

```
n.flags |= Notification.FLAG_SHOW_LIGHTS;
```

TUTORIÁLY LARSE VOGELA

<http://www.vogella.com/tutorials/AndroidNotifications/article.html>



akční tlačítka v notifikaci
jako id notifikace - aktuální čas 😊

```
// prepare intent which is triggered if the
// notification is selected

Intent intent = new Intent(this, NotificationReceiver.class);
// use System.currentTimeMillis() to have a unique ID for the pending intent
PendingIntent pIntent = PendingIntent.getActivity(this, (int) System.currentTimeMillis(), intent, 0);

// build notification
// the addAction re-use the same intent to keep the example short
Notification n = new Notification.Builder(this)
    .setContentTitle("New mail from " + "test@gmail.com")
    .setContentText("Subject")
    .setSmallIcon(R.drawable.icon)
    .setContentIntent(pIntent)
    .setAutoCancel(true)
    .addAction(R.drawable.icon, "Call", pIntent)
    .addAction(R.drawable.icon, "More", pIntent)
    .addAction(R.drawable.icon, "And more", pIntent).build();

NotificationManager notificationManager =
    (NotificationManager) getSystemService(NOTIFICATION_SERVICE);

notificationManager.notify(0, n);
```

POZNÁMKY

Android Builder classes

Notification.Builder – This class was added in Android 3.0 Honeycomb [API 11]. So if you need to support older SDK's you will need to use NotificationCompact instead.

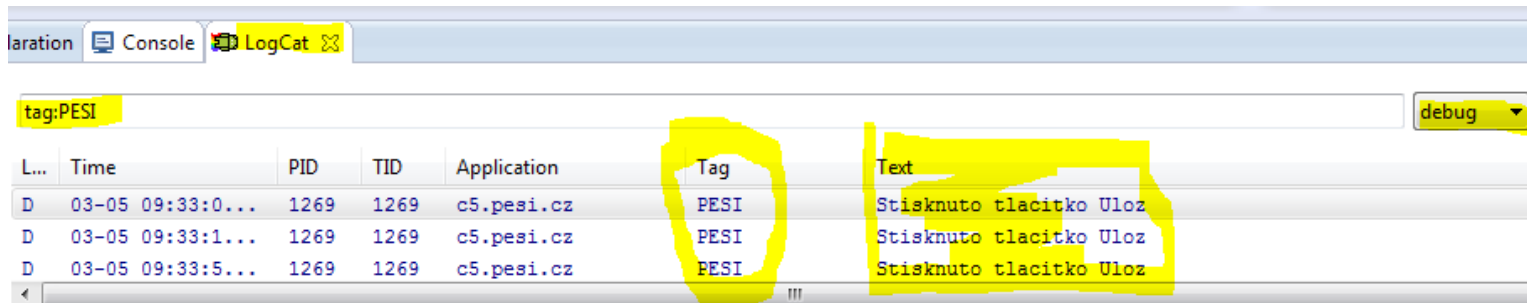
NotificationCompat.Builder – This class is in the version 4 Support Library (compatible Android 1.6 and up).

NotificationCompat.Builder

The NotificationCompat.Builder, similar to other class builders, provides an interface that is used to create a Notification object. NotificationCompat.Builder help you constructing the typical notification layouts. On platform versions that **don't offer expanded** notifications, methods that depend on expanded notifications **have no effect**.

Zdroj: <https://trinitytuts.com/notification-using-notificationcompat-builder-in-android/>

LOGOVÁNÍ PRO VÝVOJÁŘE



L...	Time	PID	TID	Application	Tag	Text
D	03-05 09:33:0...	1269	1269	c5.pesi.cz	PESI	Stisknuto tlacitko Uloz
D	03-05 09:33:1...	1269	1269	c5.pesi.cz	PESI	Stisknuto tlacitko Uloz
D	03-05 09:33:5...	1269	1269	c5.pesi.cz	PESI	Stisknuto tlacitko Uloz

zobrazení v okně LogCat

```
Log.d("button1_click", Integer.toString(5));
```

```
Log.d("PESI", "Až sem to jde");
```

```
Log.v(TAG, TEXT);
```

Log.v, Log.d, Log.i, Log.w, Log.e, Log.a

Verbose, debug, info, warn, error, assert

UI: WEBOVÝ PROHLÍŽEČ

- ◉ widget **WebView** - webový prohlížeč
- ◉ metoda **loadUrl("http://m.idnes.cz")**
 - jakou webovou stránku zobrazit
- ◉ vyžádat v manifestu právo přístupu k Internetu:
`<uses-permission android:name="android.permission.INTERNET" />`

WEBVIEW - KDE DEFINOVAT

- v XML popisu layoutu
 - podobně jako např. tlačítko
- celé okno aktivity jako WebView:

v metodě *onCreate()* {

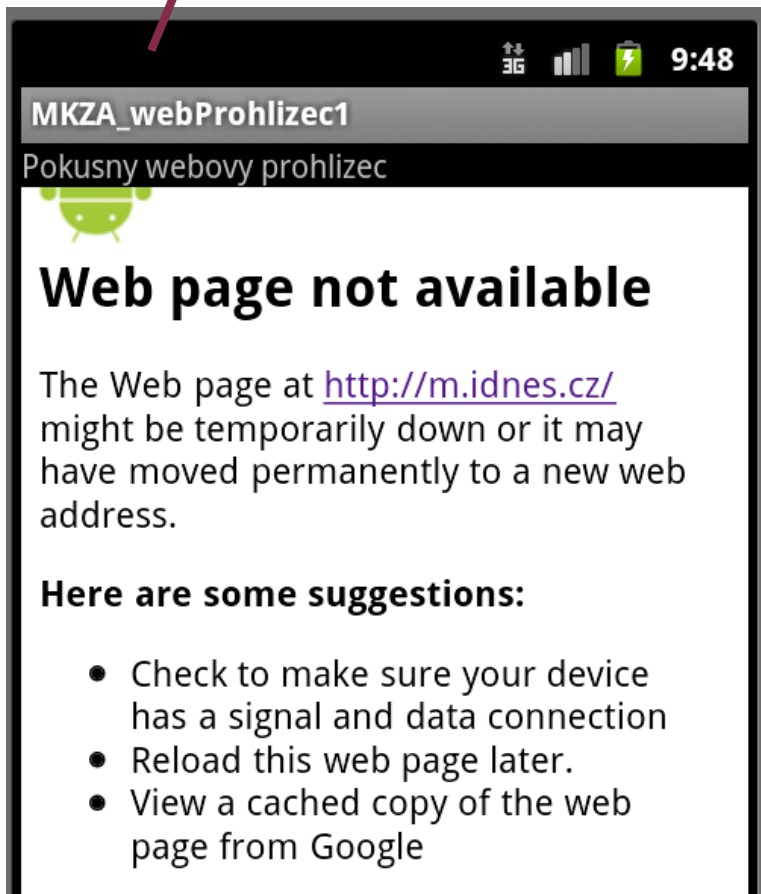
...

```
WebView webview = new WebView(this);  
setContentView(webview);
```

```
}
```

aplikace má vlastní cache, cookie store, požadavky otevřít nové okno z prohlížeče ignorovány

pokud v manifestu není vyžádáno
právo přístupu na Internet



MKZA_webProhlizec1
Pokusny webovy prohlizec

Web page not available

The Web page at <http://m.idnes.cz/> might be temporarily down or it may have moved permanently to a new web address.

Here are some suggestions:

- Check to make sure your device has a signal and data connection
- Reload this web page later.
- View a cached copy of the web page from Google



MKZA_webProhlizec1
Pokusny webovy prohlizec

iDNES.cz
Út 10:50, svátek má Leona.

Rubriky | On-line | Hledání

reklama
Vyberte si kartu

 **Vedení Sazky přiznalo úpadek firmy a vyzvalo Hušáka k podání návrhu**

 **Dráhy koupí 30 let staré vagony z Rakouska**

 **Obviněný z vraždy Aničky už možná nepromluví, hrozí mu trvalé následky**

 **Pozdější odchod do penze bude od lidí vyžadovat víc učení i odvahy**

 **Volejte se smartphony zcela zdarma. Víme, jak na to**

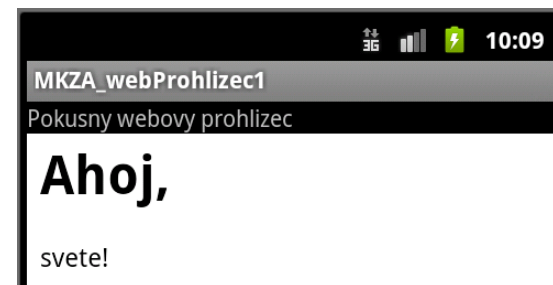
Další články

Nejnovější Nahoru

10:50 **Krteček mřítí do vesmíru jako maskot na poslední misi Endeavouru**

WEBVIEW

- ◉ **povolení JavaScriptu:**
`prohlizec.getSettings().setJavaScriptEnabled(true);`
- ◉ **zobrazení webové stránky:**
`prohlizec.loadUrl("http://m.idnes.cz");`
- ◉ **zobrazení lokálního html kódu:**
`prohlizec.loadData(řetězec, MIME type, kódování);`
`prohlizec.loadData("<html><body><h1>Ahoj,</h1>svete!</body></html>",
"text/html", "UTF-8")`



WEBVIEW - NOVÉ OKNO

Viz:

<http://stackoverflow.com/questions/5561709/opening-webview-not-in-new-browser>

Ya. You must implement `WebViewClient` class and `Override shouldOverrideURLLoading()` method in this class.

Why ? Because `webview` just open your "exactly link", if that link `redirect other links`, android will `open default browser` for this action.

In your example, as you know, when you connecting to `google.com` google will redirects to google at your country. Example, if you are in China, google will go to `google.com.cn` , if in Vietnam, will be `google.com.vn` .

WEBVIEW - NAVIGACE

k dispozici jsou metody pro pohyb v prohlížeči:

- reload() // znovu načte akt. URL
- goBack() //back v historii WebView
- goForward()
- clearCache() // cache je per-aplikaci
// parametr, zda jen RAM nebo
// i diskovou
- clearHistory() // vyčistí back-forward list
- canGoBack()
- ...

WEBVIEW - VLASTNOSTI

- ◉ prohlizec.getSettings()

- .setTextZoom(int) //default je 100
 - .setJavaScriptEnabled() //default je false
 - .setUserAgentString(String)

- ◉ prohlizec.setWebViewClient (..)

- odchyťávat různé další notifikace a události a využít tak prohlížeč jako UI
 - vždy prostudovat metody dle SDK dokumentace
<http://developer.android.com/reference/android/webkit/WebViewClient.html>

- ◉ prohlizec.setWebChromeClient() //JS dialogy atd.

WEBVIEW - DALŠÍ PODROBNOSTI

- ◉ <http://chrisrisner.com/31-Days-of-Android--Day-18%E2%80%93The-Webview>
- ◉ <http://www.giantflyingsaucer.com/blog/?p=1331> -- ukázat kód
- ◉ puštění kódu z JavaScriptu
- ◉ podobným způsobem fungují frameworky jako je PhoneGap

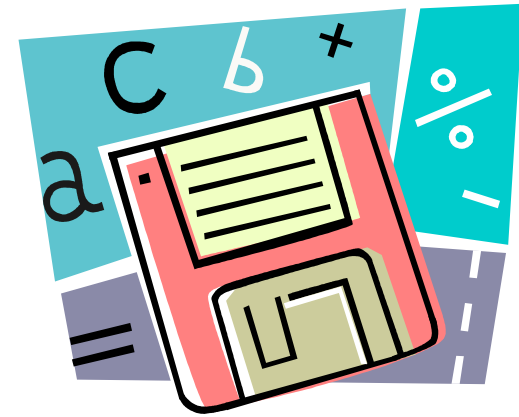
VYVOLÁNÍ WEBOVÉHO PROHLÍŽEČE („VELKÝ PROHLÍŽEČ“)

spuštění plnohodnotného prohlížeče
pomocí intentu:

```
Uri uri = Uri.parse("http://www.zive.cz");  
Intent intent = new Intent(Intent.ACTION_VIEW, uri);  
startActivity(intent);
```

UKLÁDÁNÍ DAT

jak uložit persistentní data



- ⊙ **sdílené preference**
 - pár (klíč, hodnota)
- ⊙ **vnitřní úložiště**
 - v paměti zařízení
- ⊙ **externí úložiště**
 - sdílené vnější úložiště (SD karta)
- ⊙ **SQLite databáze**
- ⊙ **sít'ové spojení**

POZNÁMKA - STACK OVERFLOW

- ⦿ **Internal Storage:** is storage that is **not accessible by the user**, except via installed apps (or by rooting their device). Example: data/data/app_packageName
- ⦿ **Primary External Storage:** In built shared storage which is "accessible by the user by plugging in a USB cable and **mounting it as a drive on a host computer**". Example: When we say Nexus 5 32 GB.
- ⦿ **Secondary External Storage:** Removable storage. Example: SD Card.

ANDROID 6 - ADAPTABLE STORAGE

- ◉ Např. **SD karta**, kryptovaná a chová se jako interní úložiště
- ◉ Pracuje jen s jedním androidím zařízením
- ◉ Lze na něj ukládat aplikace, když to vývojář povolí (android:installLocation atribut)
- ◉ Nově instalované aplikace na úložiště s nejvíce volným místem
- ◉ Uživatel může v nastavení přesouvat
- ◉ Výrobce telefonu nemusí podporovat (obava o rychlost apod.)

ODKAZ NA DALŠÍ INFORMACE

<https://source.android.com/devices/storage/index.html>

- Od Androidu 1.0 vyžadoval `WRITE_EXTERNAL_STORAGE`
- Od Android 4.1 dále `READ_EXTERNAL_STORAGE`
- Od Android 4.4 i bez práva `WRITE_EXTERNAL_STORAGE` může přistupovat aplikace s package name `com.example.foo` do `Android/data/com.example.foo/`

CO KAM INSTALOVAT?

- ⦿ <http://developer.android.com/guide/topics/data/install-location.html>

for your application with the `android:installLocation` manifest attribute. If you **do not declare** this attribute, your application will be **installed on the internal storage only** and it **cannot be moved** to the external storage.

To allow the system to install your application on the external storage, modify your manifest file to include the

`android:installLocation` attribute in the `<manifest>` element, with a value of either "`preferExternal`" or "`auto`". For example:

```
<manifest xmlns:android="http://schemas.android.com/apk/r
  android:installLocation="preferExternal"
  ... >
```

INSTLACE APLIKACE NA EXTERNÍ ÚLOŽIŠTĚ

When your application is installed on the **external storage**:

- There is **no effect** on the application **performance** so long as the external storage is mounted on the device.
- The **.apk file is saved on the external storage**, but all private user data, databases, optimized .dex files, and extracted native code are saved on the internal device memory.
- The unique container in which your application is stored is encrypted with a randomly generated key that can be decrypted only by the device that originally installed it. Thus, an **application installed on an SD card works for only one device.**
- The user **can move y**our application to the internal storage through the system settings.

Warning: When the user enables USB mass storage to share files with a computer or unmounts the SD card via the system settings, the external storage is unmounted from the device and all applications running on the external storage are immediately killed.

VNITŘNÍ ÚLOŽIŠTĚ

- ◉ soubory jsou **soukromé** pro naši aplikaci
 - ◉ další aplikace se k nim nedostanou
 - ◉ při odinstalaci aplikace - odstraněny
-
- ◉ soubory v .apk
 - ◉ soubory vytvořené naší aplikací

SOUBORY V .APK

- ⊙ soubory přibalené k naší aplikaci (*.apk*)
 - nelze data upravovat !
(jen při aktualizaci aplikace na novou verzi)
 - při vývoji dáme do složky */res/raw*
 - `InputStream ins =
getResources().openRawResource(R.raw.file1)`

 - Případně do složky *assets*
v Aktivitě: `getAssets().open("my_file.txt")`
ve Fragmentu:
`getActivity().getAssets().open("my_file.txt")`

SOUBORY VYTVOŘENÉ APLIKACÍ

- využití *openFileOutput()*, *openFileInput()*
- lze zadat pouze název souboru, nikoliv cestu k něm
 - pokud 2 aplikace soubor ahoj.txt => 2 různé soubory

```
String FILENAME = "ahoj.txt";
```

```
String retezec = "Uloženo v Androidu";
```

```
FileOutputStream fos =
```

```
    openFileOutput(FILENAME, Context.MODE_PRIVATE);
```

```
fos.write(retezec.getBytes());
```

```
fos.close();
```

čtení analogicky: *openFileInput()*, *read()*, *close()*

KDE JSOU ULOŽENY?

všimněte si kromě
adresáře files
i adresáře cache

Android Device Monitor

📁	cz.pesi.mkz2013.cv04p1	2013-03-06	16:57	drwxr-x--x	
▷	cache	2013-03-06	15:44	drwxrwx--x	
▷	databases	2013-03-06	15:51	drwxrwx--x	
📁	files	2013-03-06	16:24	drwxrwx--x	
📄	ahoj.txt	15	2013-03-06	16:58	-rw-rw----
▷	lib	2013-03-06	16:57	drwxr-xr-x	

```
C:\sw\android-sdk-windows\platform-tools>adb start-server
C:\sw\android-sdk-windows\platform-tools>adb shell
# cd /data/data/cz.pesi.mkz2013.cv04p1/files
cd /data/data/cz.pesi.mkz2013.cv04p1/files
#
# ls
ls
ahoj.txt
#
# cat ahoj.txt
cat ahoj.txt
pokusna radka77#
#
```

VNITŘNÍ ÚLOŽIŠTĚ - PŘÍKLAD

```
public void Uloz(View v) {  
  
    FileOutputStream fos;  
    String retez;  
    String FILENAME = "ahoj.txt";  
    retez = new String(editText1.getText().toString());  
    try {  
        fos = openFileOutput(FILENAME, Context.MODE_PRIVATE);  
        fos.write(retez.getBytes());  
  
        fos.close();  
  
    } catch (Exception e) { }  
}
```

VNITŘNÍ ÚLOŽIŠTĚ - PŘÍKLAD

```
public void Nacti(View v) {  
    FileInputStream fis;  
    byte [] buffer = new byte[1024];  
    String vysledek;  
    String FILENAME = "ahoj.txt";  
  
    try {  
        fis= openFileInput(FILENAME);  
        fis.read(buffer);  
        fis.close();  
  
        vysledek = new String(buffer);  
        editText2.setText(vysledek);  
  
    } catch (Exception e) { }  
}
```


NAČTI - MODIFIKACE

```
BufferedReader inputReader = new BufferedReader(new  
InputStreamReader( openFileInput("s1.txt")));
```

```
String inputString;
```

```
StringBuffer stringBuffer = new StringBuffer();
```

```
while ((inputString = inputReader.readLine()) != null) {
```

```
    stringBuffer.append(inputString + "\n");
```

```
}
```

```
editText2.setText(stringBuffer.toString());
```

MÓDY PŘÍSTUPU K SOUBORU

◉ **MODE_PRIVATE**

- privátní
- pokud existuje, přepíše

◉ **MODE_APPEND**

- neexistující vytvoří
- přidává na konec existujícího souboru

◉ **MODE_WORLD_READABLE** //deprecated API 17

◉ **MODE_WORLD_WRITABLE** //deprecated API 17

```
FileOutputStream fos = openFileOutput("s1.txt",  
    Context.MODE_APPEND | Context.MODE_WORLD_READABLE);
```

VNITŘNÍ ÚLOŽIŠTĚ - DALŠÍ METODY

- ◉ **getFilesDir()**
 - absolutní cesta k našemu úložišti, kde budou vytvářeny soubory přes `openFileOutput`
- ◉ **getDir()**
 - vytvoří nebo otevře existující adresář uvnitř našeho úložiště
- ◉ **deleteFile()**
 - smaže soubor
 - Lze také `myFile.delete()`
- ◉ **fileList()**
 - seznam našich souborů
 - vrací pole řetězců

GETDIR() - POZNÁMKY

Jak přistupovat k souboru v adresáři, když openFileOutput akceptuje jen název bez cesty? (dle StackOverflow)

```
File dir = getDir("myDir", Context.MODE_PRIVATE);
```

```
File myFile = new File(dir, "myFile");
```

```
FileOutputStream out = new FileOutputStream(myFile);
```

```
String hello = "hello world";
```

```
out.write(hello.getBytes());
```

```
out.close();
```

Práce s daty

Zadejte text:

pokusny text pro ulozeni

Přečtený text:

pokusny text pro ulozeni

Save

Load

/data/data/c5.pesi.cz/files

```
Toast.makeText(this,  
getFilesDir().getPath(),  
Toast.LENGTH_LONG).  
show();
```

VNITŘNÍ ÚLOŽIŠTĚ - CACHE

- místo persistentního uložení potřebujeme nějaká data jen nakešovat
- v případě nedostatku místa (<1MB) systém smaže
- přesto bychom sami měli po sobě data uklidit

1. `File file = new File(this.getCacheDir(), "temp.txt");`
2. `file.createNewFile();`
3. `FileWriter fw = new FileWriter(file);`
4. `BufferedWriter bw = new BufferedWriter(fw);`
5. `bw.write("Hello World");`
6. `bw.newLine();`
7. `bw.close();`

A také uživatel může vymazat přes
Settings cache aplikace

ULOŽENÍ DO CACHE - PŘÍKLAD

```
public void Uloz_cache(View v) {
```

```
    String retez;
```

```
    String FILECACHE = „pracovni.txt“;
```

```
    File file = new File(this.getCacheDir(),FILECACHE);
```

```
    retez = new String(editText1.getText().toString());
```

```
    try {
```

```
        file.createNewFile();
```

```
        FileWriter fw = new FileWriter(file);
```

```
        BufferedWriter bw = new BufferedWriter(fw);
```

```
        bw.write(retez);
```

```
        bw.newLine();
```

```
        bw.close();
```

```
    } catch (Exception e) {}
```

```
}
```

ČTENÍ Z CACHE - PŘÍKLAD

```
public void Nacti_cache(View v) {  
    String FILECACHE = „pracovni.txt“;  
    char [] buffer = new char[1024];  
    String vysledek;  
  
    File file = new File(this.getCacheDir(), FILECACHE);  
    try {  
        FileReader fr = new FileReader(file);  
        BufferedReader br = new BufferedReader(fr);  
        br.read(buffer);  
        vysledek = new String(buffer);  
        editText2.setText(vysledek);  
        br.close();  
    }  
    catch (Exception e) { }  
}
```




C5

Práce s daty

Zadejte text:

ahoj cache

Přečtený text:

ahoj cache

Save internal

Load internal

Save cache

Load cache

/data/data/c5.pesi.cz/cache

```
Toast.makeText(this,  
getCacheDir().getPath(),  
Toast.LENGTH_LONG).show();
```

VNĚJŠÍ ÚLOŽIŠTĚ

sdílené externí úložiště

- SD karta, ale i interní (non-removable) úložiště!
- k souborům přístup i další aplikace
- lze je přesunout do PC, smazat bez vědomí atd.
- `<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>`
- Pro čtení - většinou není potřeba, ale může se změnit `<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>`

při použití zkontrolovat, zda je úložiště připojené

VNĚJŠÍ ÚLOŽIŠTĚ

- ◉ `getExternalStorageState()`
 - kontrola dostupnosti média

```
String state = Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED.equals(state)) {
    // můžeme číst a zapisovat
} else if
    (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
    // můžeme pouze číst
} else {
    // máme smůlu
}
```

SOUBORY NA VNĚJŠÍM ÚLOŽIŠTI

⦿ Public files

- Volně dostupné dalším aplikacím, uživateli
- Při odinstalaci aplikace by měli zůstat
- Vyfocené fotky, stažené soubory
- `getExternalStoragePublicDirectory()`

⦿ Private files

- Při odinstalaci aplikace by se měli odstranit
- Jsou na vnějším úložišti, ostatní se na ně tedy můžou dostat, ale nemají pro ně cenu
- `getExternalFilesDir()`

VNĚJŠÍ ÚLOŽIŠTĚ - PŘÍSTUP K SOUBORŮM

⊙ `getExternalFilesDir()`

- API LEVEL ≥ 8
- typ podadresáře, který chceme (DIRECTORY_MUSIC, DIRECTORY_RINGTONES, null)
- při odinstalaci aplikace smaže data

⊙ `getExternalStorageDirectory()`

- API LEVEL do 7
- vrátí kořen externího úložiště
- ukládáme do `/Android/data/<package_name>/files/`

PŘÍKLAD

```
File path =  
getExternalFilesDir(Environment.DIRECTORY_PICTURES);  
    if (path != null)  
        File file = new File(path, "DemoPicture.jpg");
```

ZÁPIS DO VNĚJŠÍHO ÚLOŽIŠTĚ

```
String storageState = Environment.getExternalStorageState();  
  
if (storageState.equals(Environment.MEDIA_MOUNTED)) {  
  
    File file = new File(getExternalFilesDir(null), "soubor.txt");  
  
    BufferedReader inputReader2 = new BufferedReader(  
        new InputStreamReader(new FileInputStream(file)));  
  
    // zde kód na čtení souboru  
  
    }  
  
    editText2.setText(stringBuffer2.toString());
```

VNĚJŠÍ ÚLOŽIŠTĚ

- sdílené soubory - nemazat při odinstalaci aplikace
- veřejné adresáře
 - `getExternalStoragePublicDirectory()` .. `>= API L8`

Název	popis
Music	hudba
Podcast	Podcast (mluvené slovo)
Ringtones	Vyzváněcí tón
Alarms	Zvuk budíku
Notifications	Notifikační zvuky
Pictures	Fotky (mimo vyfocených)
Movies	Filmy (mimo natočených)
Download	Bez komentáře

VNĚJŠÍ ÚLOŽIŠTĚ

```
File path = Environment.getExternalStoragePublicDirectory  
( Environment.DIRECTORY_PICTURES );
```

```
File file = new File(path, "DemoPicture.jpg");
```

```
// pokud adresáře neexistují, vytvořit je  
path.mkdirs();
```

```
// a dále pracujeme s file ...
```

VNĚJŠÍ ÚLOŽIŠTĚ - CACHE

- ◉ **getExternalCacheDir()**

- API LEVEL ≥ 8

- ◉ **getExternalStorageDirectory()**

- API LEVEL ≤ 7

- adresář:

/Android/data/<package_name>/cache/

VNĚJŠÍ ÚLOŽIŠTĚ - PŘÍKLAD

```
File f = new File("/sdcard/file1.txt");
```

```
try {
```

```
    RandomAccessFile ra =  
        new RandomAccessFile(f, "rw");
```

```
ra.setLength(10 * 1024 * 1024); // 10MB
```

```
} catch (Exception e) { }
```

VNĚJŠÍ ÚLOŽIŠTĚ

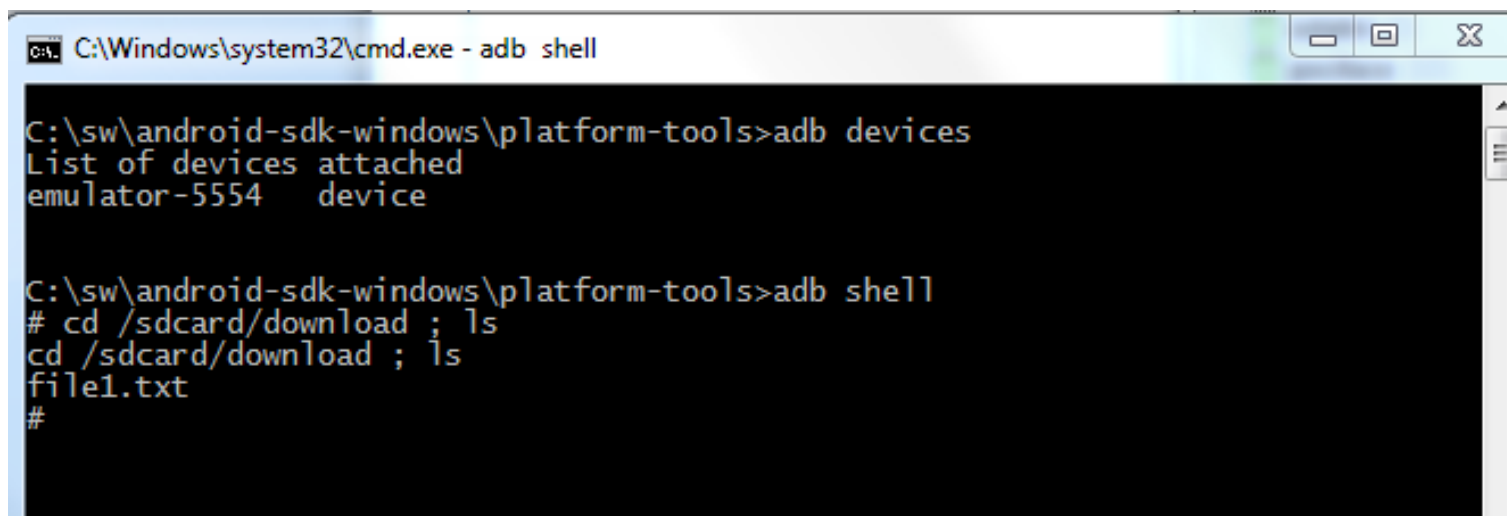
- ◉ nastavení práva v manifestu

```
<uses-permission android:name=  
"android.permission.WRITE_EXTERNAL_STORAGE" />
```

Nenastavení práva se nijak neprojeví při překladu - pouze soubor nebude vytvořen

VERIFIKACE OBSAHU SD KARTY

nástroj adb



```
C:\Windows\system32\cmd.exe - adb shell

C:\sw\android-sdk-windows\platform-tools>adb devices
List of devices attached
emulator-5554    device

C:\sw\android-sdk-windows\platform-tools>adb shell
# cd /sdcard/download ; ls
cd /sdcard/download ; ls
file1.txt
#
```

adb devices - vypíše aktivní zařízení

adb shell - shell zařízení

- výpis obsahu paměťové karty
- zkontrolujeme, zda je file1.txt k dispozici

VÝPIS LS -L

```
# ls -l
```

```
ls -l
```

```
drwxr-xr-x root      root          2012-03-21 18:59 acct
drwxrwx--- system   cache        2012-03-21 13:32 cache
dr-x----- root    root         2012-03-21 18:59 config
lrwxrwxrwx root     root         2012-03-21 18:59 d -> /sys/kernel/debug
drwxrwx--x system  system      2012-03-19 08:39 data
-rw-r--r-- root    root         116 1970-01-01 00:00 default.prop
drwxr-xr-x root    root         2012-03-21 18:59 dev
lrwxrwxrwx root    root         2012-03-21 18:59 etc -> /system/etc
-rwxr-x--- root    root        98676 1970-01-01 00:00 init
-rwxr-x--- root    root         2344 1970-01-01 00:00 init.goldfish.rc
-rwxr-x--- root    root        17040 1970-01-01 00:00 init.rc
drwxrwxr-x root    system      2012-03-21 18:59 mnt
dr-xr-xr-x root    root        1970-01-01 00:00 proc
drwx----- root    root        2011-11-14 19:00 root
drwxr-x--- root    root        1970-01-01 00:00 sbin
lrwxrwxrwx root    root        2012-03-21 18:59 sdcard -> /mnt/sdcard
drwxr-xr-x root    root        1970-01-01 00:00 sys
drwxr-xr-x root    root        2011-12-14 05:41 system
-rw-r--r-- root    root         272 1970-01-01 00:00 ueventd.goldfish.rc
-rw-r--r-- root    root        3825 1970-01-01 00:00 ueventd.rc
lrwxrwxrwx root    root        2012-03-21 18:59 vendor -> /system/vendor
```

LS -L /SDCARD

```
# cd /sdcard
```

```
cd /sdcard
```

```
# ls -l
```

```
ls -l
```

```
d---rwxr-x system  sdcard_rw      2012-03-19 14:42 Alarms
d---rwxr-x system  sdcard_rw      2012-03-19 14:42 DCIM
d---rwxr-x system  sdcard_rw      2012-03-22 08:06 Download
d---rwxr-x system  sdcard_rw      2012-03-19 08:39 LOST.DIR
d---rwxr-x system  sdcard_rw      2012-03-19 14:42 Movies
d---rwxr-x system  sdcard_rw      2012-03-19 14:42 Music
d---rwxr-x system  sdcard_rw      2012-03-19 14:42 Notifications
d---rwxr-x system  sdcard_rw      2012-03-19 14:42 Pictures
d---rwxr-x system  sdcard_rw      2012-03-19 14:42 Podcasts
d---rwxr-x system  sdcard_rw      2012-03-19 14:42 Ringtones
```

ADB SHELL - VYBRANÉ PŘÍKAZY

- mount
- id
- cat /proc/cpuinfo (ARMv7 processor)
- cat /proc/meminfo
- cat /proc/version

```
# cat /proc/version
cat /proc/version
Linux version 2.6.29-g46b05b2 (vchtchetkine@vc-irv.irv.corp.google.com) (gcc ver
sion 4.4.3 (GCC) ) #28 Thu Nov 17 06:39:36 PST 2011
#
# id
id
uid=0(root) gid=0(root)
#
```


OVĚŘENÍ INTERNÍHO ÚLOŽIŠTĚ

```
C:\Windows\system32\cmd.exe - adb shell
#
#
#
# pwd
pwd
/data/data/c5.pesi.cz
# ls -l
ls -l
drwxrwx--x app_42    app_42          2012-03-21 14:04 cache
drwxrwx--x app_42    app_42          2012-03-21 11:42 files
drwxr-xr-x system    system          2012-03-21 11:26 lib
# ls -l files/
ls -l files/
-rw-rw---- app_42    app_42          62 2012-03-21 19:00 ahoj.txt
# cat files/ahoj.txt
cat files/ahoj.txt
ahoj jsem textovy editor a ulozim nejaky obsah souboru interne#
# ls -l cache/
ls -l cache/
-rw----- app_42    app_42          24 2012-03-21 19:01 cache.txt
# cat cache/cache.txt
cat cache/cache.txt
a ted ukladame do cache
#
```

všimněte si cesty: /data/data/c5.pesi.cz/files/ahoj.txt

DATABÁZE - SQLITE

- ◉ vytvoření DB -> přepsat metodu **onCreate** třídy **SQLiteOpenHelper**
- ◉ implementace třídy **SQLiteOpenHelper**
 - **onCreate()** - vytvoření struktury databáze
 - **onUpgrade()** - je-li nutné změnit strukturu databáze
- ◉ na instanci třídy **SQLiteOpenHelper**
 - **getWritableDatabase()**
 - **getReadableDatabase()**

DATABÁZE - VYTVOŘENÍ

```
public class DictionaryOpenHelper extends SQLiteOpenHelper {  
  
    private static final int DATABASE_VERSION = 2;  
    private static final String DICTIONARY_TABLE_NAME = "dictionary";  
    private static final String DICTIONARY_TABLE_CREATE =  
        "CREATE TABLE " + DICTIONARY_TABLE_NAME + " (" +  
        KEY_WORD + " TEXT, " +  
        KEY_DEFINITION + " TEXT);";  
  
    DictionaryOpenHelper(Context context) {  
        super(context, DATABASE_NAME, null, DATABASE_VERSION);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
        db.execSQL(DICTIONARY_TABLE_CREATE);  
    }  
}
```

ukázka z Google SDK

DATABÁZE - VYTVOŘENÍ

podobná ukázka:

```
public void onCreate(SQLiteDatabase sqLiteDB)
{
    String createSql = "CREATE TABLE Tabulka " +
        "(_id integer primary key autoincrement, " +
        "name text not null);";
    sqLiteDB.execSQL(createSql);
}
```

zdroj: <http://chrisrisner.com/31-Days-of-Android--Day-24%E2%80%93Using-SQLite-Databases>

DATABÁZE - UPGRADE

```
public void onUpgrade(SQLiteDatabase sqLiteDB,  
int oldVersion, int newVersion) {
```

```
sqLiteDB.execSQL("DROP TABLE IF EXISTS  
TABULKA");
```

```
onCreate(sqLiteDB);
```

```
}
```

- ◉ upgrade DB na novější verzi, např. ukládající více informací
- ◉ zde pouze tabulku zrušíme a znovu vytvoříme

DATABÁZE - POUŽITÍ

- ◉ DictionaryOpenHelper mDOHelper = new DictionaryOpenHelper(getContext());
- ◉ SQLiteDatabase db = mDOHelper.**getWritableDatabase()**;
- ◉ .. **getWritableDatabase()**
- ◉ cursor - iterace přes záznamy databáze
- ◉ viz další přednáška
- ◉ <http://developer.android.com/training/basics/data-storage/databases.html>

SÍŤOVÉ SPOJENÍ

- ⦿ zde pro doplnění přehledu
- ⦿ *java.net.**
- ⦿ *android.net.**

PC má loopback
127.0.0.1
chceme-li naše PC
kontaktovat z
emulátoru,
použijeme adresu
10.0.2.2

```
s = new Socket("10.0.2.2", 5678);  
reader = new BufferedReader(new InputStreamReader(s.getInputStream()));  
writer = new PrintWriter(new OutputStreamWriter(s.getOutputStream()));
```

```
String line;
```

```
writer.println(„Ahoj, jsem tvuj klient. Věnuj se mi!");
```

```
writer.flush();
```

```
line = reader.readLine();
```

```
tv.setText(tv.getText() + " , "+line);
```

```
reader.close();
```

```
writer.close();
```

PREFERENCE

- ◉ uložení dat ve tvaru (klíč, hodnota)
- ◉ persistentní - i po ukončení aktivity
- ◉ ukládají se do XML souboru

- ◉ typické použití - uživatelská konfigurace
 - úroveň obtížnosti hry, high skóre
 - poslední prohlížená stránka
 - vypnutý / zapnutý zvuk

- ◉ **klíč** - typu String
- ◉ **hodnota** - primitivní datové typy

PREFERENCE

- ◉ pro jednu aktivitu
- ◉ sdílené více aktivitami v aplikaci
- ◉ (na úrovni aplikací)

- ◉ **getPreferences(*mode*)**
 - jen jeden preferenční soubor pro aktivitu
- ◉ **getSharedPreferences(*name*, *mode*)**
 - více preferenčních souborů
- ◉ **getDefaultSharedPreferences(context)**
 - defaultní pro daný kontext (aplikaci)

PREFERENCE - ZÁPIS / ČTENÍ HODNOTY

Zápis:

1. zavoláme **edit()** pro získání *SharedPreferences.Editor*
2. vložíme hodnoty:
putBoolean(), **putString()**, ...
3. potvrdíme **commit()**

Čtení:

1. pro čtení:
getBoolean(), **getString()**, ...

PREFERENCE - PŘÍKLAD Z GOOGLE SDK (1/2)

```
public class Calc extends Activity {  
    public static final String PREFERENCES_NAME = "MyPrefsFile";  
  
    @Override  
    protected void onCreate(Bundle state) {  
        super.onCreate(state);  
        ...  
        // načtení preferencí  
        SharedPreferences settings =  
        getSharedPreferences(PREFERENCES_NAME, 0);  
  
        boolean silent = settings.getBoolean("silentMode", false);  
        // příklad: nastavíme tichý mód jak byl dříve  
        nastavTichyMod(silent);  
    }  
}
```

klíč a defaultní
hodnota

nastavíme tichý mód, tak jak byl „posledně“

PREFERENCE - PŘÍKLAD Z GOOGLE SDK (2/2)

@Override

```
protected void onStop(){  
    super.onStop();
```

```
    // potřebujeme editor, abychom provedli změny
```

```
    SharedPreferences settings =  
    getSharedPreferences(PREFS_NAME, 0);
```

```
    SharedPreferences.Editor editor = settings.edit();  
    editor.putBoolean("silentMode", mSilentMode);
```

```
    // potvrdíme změny  
    editor.commit();
```

```
}
```

```
}
```

když aplikace během životního cyklu zavolá
onStop(), uložíme aktuální preference

EDITOR PREFERENCÍ

◉ `commit()`

- uložení změn
- důležité, nezapomenout na něj

◉ `clear()`

- smaže všechny preference

◉ `remove()`

- smaže jednu preferenci

změní-li aktivita preference, změnu uvidí hned i další části aplikace, např. služba

PREFERENCE Z XML SOUBORU

- ◉ hodnotu preferencí lze také načíst z XML souboru
- ◉ **Deprecated**, PreferenceFragment načte soubory z resource filu

```
public void onCreate(Bundle savedInstanceState) {
```

```
....
```

```
addPreferencesFromResource(R.xml.preferences);
```

```
}
```

PROSTŘEDÍ PRO SPRÁVU PREFERENCÍ

- ◉ v /res/xml
- ◉ getDefaultSharedPreferences()
- ◉ uživatelské rozhraní pro nastavování preferencí dodané Androidem

public class EditPreferences extends
PreferenceActivity {

```
// v onCreate()  
addPreferencesFromResource(R.xml.preferences);  
}
```

PROSTŘEDÍ PRO SPRÁVU PREFERENCÍ

tuto aktivitu pustíme např. z Menu naší aktivity pomocí intentu:

```
startActivity(new Intent(this,  
EditPreferences.class);
```


TVAR XML SOUBORU

```
<PreferenceScreen
```

```
    xmlns:android="http://schemas.android.com/apk/res/android">
```

```
    <PreferenceCategory
```

```
        android:title="@string/inline_preferences">
```

```
        <CheckBoxPreference
```

```
            android:key="checkbox_preference"
```

```
            android:title="@string/title_toggle_preference"
```

```
            android:summary="@string/summary_toggle_preference"
```

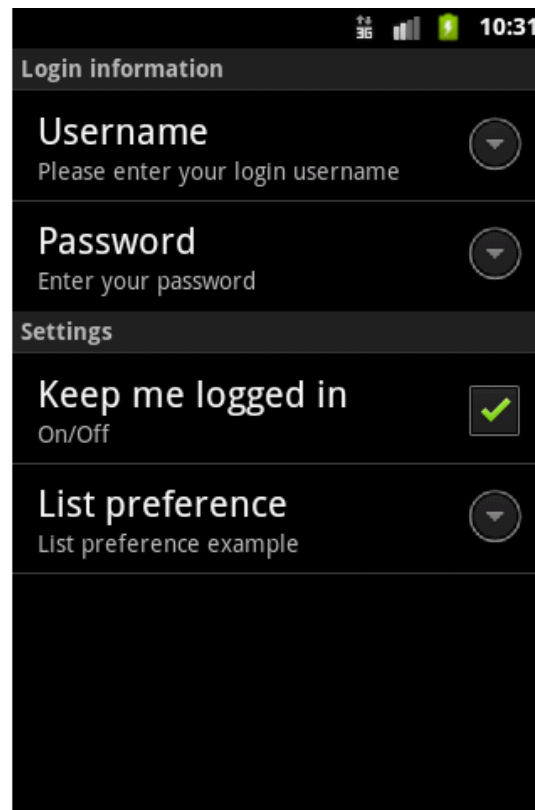
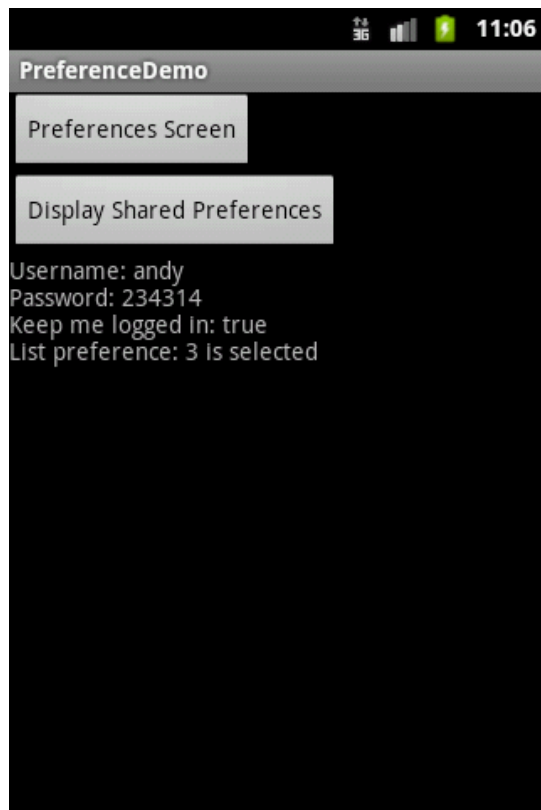
```
        />
```

```
    </PreferenceCategory>
```

```
</PreferenceScreen>
```

NASTAVENÍ PREFERENCÍ

<https://androidresearch.wordpress.com/2012/03/09/creating-a-preference-activity-in-android/>



DATA BACKUP

data backup:
jen pro info, že něco takového je

- ◉ služba umožňující kopírovat data aplikací do **cloudového úložiště** (restore point)
- ◉ není určeno pro synchronizaci dat s dalšími klienty, čtení/zápis na požádání, dostupná jen přes API Backup Managera
- ◉ **cloud storage** a **transport service** se mohou lišit pro různá zařízení => opatrně při ukládání jmen, hesel

DATA BACKUP

- ◉ musíme implementovat **backup agenta**
 - jaká data chceme zálohovat
- ◉ volán **Backup managerem**
 - spravuje datové transakce s cloudem
 - využívá backup transport
- ◉ volán při obnovení dat po reinstalaci
- ◉ registrace aplikace, backup service key

I have read and agree with the Android Backup Service Terms of Service ([printable version](#))

Application package name:

Register

POSTUP

1. Deklarovat backup agenta v manifestu
2. registrovat aplikaci s backup službou
Google nabízí Android Backup Service
můžou být i jiné
3. definovat backup agenta
rozšířit BackupAgent - onBackup(), onRestore()
nebo rozšířit BackupAgentHelper

DATA BACKUP - MANIFEST

```
<application android:label="MyApplication"
    android:backupAgent="MyBackupAgent">
    ...
    <meta-data
android:name="com.google.android.backup.api_key"
    android:value="AEdPqrEAAAAIDaYEVgU6DJnyJdBmU7KLH
3kszDXLv_4DIseIyQ" />
</application>
```

máme-li více aplikací, je třeba registrovat každou zvlášť

CO LZE ZÁLOHOVAT

- ◉ sdílené preference
- ◉ jiné typy objektů

čtení a zápis na interní úložiště není **thread-safe**

(backup agent nesmí číst a zapisovat stejná data ve stejný čas jako jiná naše aktivita)

synchronizované
metody v Javě

```
static final Object[] sDataLock = new Object[0];
synchronized (MyActivity.sDataLock) {
    File dataFile = new File(getFilesDir(), TOP_SCORES);
    RandomAccessFile raFile = new RandomAccessFile(dataFile, "rw");
    raFile.writeInt(score);
}
```

GRAFIKA

- ◉ 2D grafická knihovna
 - balík `android.graphics.drawable`
 - balík `android.view.animation`
- ◉ 3D grafika OpenGL ES 1.0/1.1
 - API 1 a výše
- ◉ 3D grafika OpenGL ES 2.0
 - API 8 a výše
- ◉ 3D grafika OpenGL ES 3.0
 - API 18 a výše
- ◉ 3D grafika OpenGL ES 3.1
 - API 21 (Android 5) a výše

GRAFIKA 2D

1. kreslení uvnitř **View** objektu

- např. graf
- nevyžaduje velký frame-rate (šachy, had)
- Android poskytne předdefinovaný Canvas

2. kreslení přímo s využitím **Canvasu**

- aplikace potřebuje překreslovat často
- video hry

VLASTNÍ WIDGET - VIEW

- ◉ rozšíříme třídu **View**
- ◉ definujeme callback **onDraw**
 - zde je kód, který vykreslí náš widget
 - vykreslení by mělo proběhnout rychle, žádné dlouhé výpočty - vše by mělo být předem připraveno

potřeba překreslit?

Z UI vlákna můžeme zavolat `invalidate()`

SURFACEVIEW

- ◉ podtřída View
- ◉ cílem nabídnout kreslicí plochu sekundárnímu vlákně aplikace; nemusí se čekat, až bude View hierarchie připravená k vykreslení
- ◉ v **odděleném vlákně** spravujeme SurfaceView
- ◉ kreslíme do Canvasu jakou rychlostí naše vlákno chce (nemusíme volat *invalidate()*)
- ◉ SurfaceHolder.lockCanvas()
- ◉ SurfaceHolder.unlockCanvasAndPost()

VYTVOŘENÍ NOVÉHO CANVASU

definice s využitím Bitmapy

```
Bitmap b = Bitmap.createBitmap(100, 100,  
                               Bitmap.Config.ARGB_8888);  
Canvas c = new Canvas(b);
```

Canvas má vlastní množinu metod

`drawBitmap()`, `drawRect()`, `drawText()`

na Canvas lze dát **Drawable** objekty
mají vlastní draw metody, s argumentem Canvas

DRAWABLES

- ⦿ „něco, co lze nakreslit“
- ⦿ knihovna pro kreslení tvarů a obrázků
- ⦿ `android.graphics.drawable`

Typy Drawable

BitmapDrawable

ShapeDrawable

PictureDrawable

LayerDrawable

viz <http://developer.android.com/guide/topics/graphics/2d-graphics.html>

HW AKCELERACE

- ⊙ povolit / zakázat na různých úrovních
 - Application, Activity, Window, View

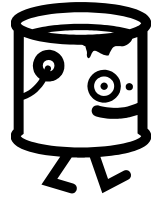
```
<application android:hardwareAccelerated="true">  
  <activity ... />  
  <activity android:hardwareAccelerated="false" />  
</application>
```

- ⊙ pokud používáme vlastní prvky, otestovat, zda nedojde k problémům - viz seznam nepodporovaných operací
<http://developer.android.com/guide/topics/graphics/hardware-accel.html>
- ⊙ od API 14 defaultně zapnutá
- ⊙ View.isHardwareAccelerated(), Canvas.isHardwareAccelerated()

OPEN GL

- ◉ multiplatformní API pro 3D grafiku
- ◉ podobné J2ME JSR 239, ale rozdíly
- ◉ OPEN GL **ES** určené pro embedded zařízení
- ◉ Android podporuje OpenGL přes framework API i NDK (Native Development Kit)

verze	dostupnost
Open GL ES 1.0/1.1	Od Android 1.0
Open GL ES 2.0	Od Android 2.2 (API 8)
Open GL ES 3.0	Od API 18
Open GL ES 3.1	Od API 21



Kreslení s využitím vlastního widgetu



PŘ.: VLASTNÍ WIDGET

1. public class MujWidget extends View {

oddělené od View

2. public MujWidget(Context context) {

3. super(context);

4. setMinimumWidth(100);

5. setMinimumHeight(50);

6. }

7. @Override

8. protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {

9. setMeasuredDimension(

10. getSuggestedMinimumWidth(),

11. getSuggestedMinimumHeight()); }

minimální rozměry widgetu, nejsou ale garantované

12. @Override

13. protected void onDraw(Canvas canvas) {

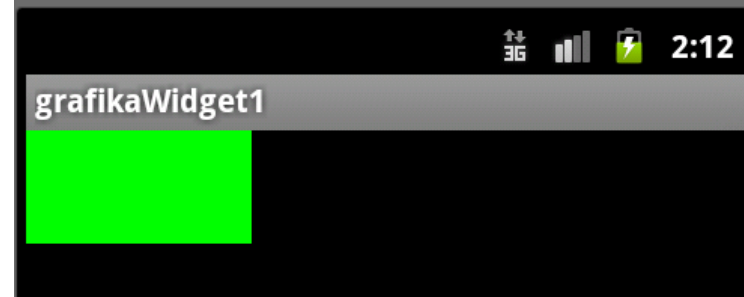
14. canvas.drawColor(Color.GREEN); }

15. }

vlastní kreslení

POUŽITÍ WIDGETU

1. package cz.pesi.grafikaWidget1;
2. import android.app.Activity;
3. import android.os.Bundle;
4. public class grafikaWidget1 extends Activity {
- 5.
6. @Override
7. public void onCreate(Bundle savedInstanceState) {
8. super.onCreate(savedInstanceState);
- 9.
10. **MujWidget widget1 = new MujWidget(this);**
11. **setContentView(widget1);**
- 12.
13. }
14. }



PŘEKRESLENÍ

- ◉ pokud je potřeba aktuální obrazovku překreslit, zavolá se metoda View **invalidate**
- ◉ z non-UI vlákna: **postInvalidate**
- ◉ událost „překresli“ bude přidána do fronty událostí
- ◉ při zpracování události zavolá framework metodu **draw** celé *view hierarchie*
- ◉ každý člen „vykreslí“ sebe a pak zavolá potomka
- ◉ draw metoda zavolá **onDraw()** - naše kreslení

DALŠÍ UKÁZKA ONDRAW()

```
protected void onDraw(Canvas canvas) {
```

```
    canvas.drawColor(Color.WHITE);
```

```
    Paint paint = new Paint();
```

```
    canvas.drawLine(50, 0, 50, 100, paint);
```

```
    canvas.drawLine(0,50,100, 50, paint);
```

```
    paint.setColor(Color.RED);
```

```
    paint.setStrokeWidth(5);
```

```
    paint.setAlpha(255);
```

```
    canvas.drawLine(50, 0, 0, 50, paint);
```

```
    canvas.drawLine(100,50,50,100, paint);
```

```
    paint.setAlpha(20);
```

```
    canvas.drawLine(50, 0, 100, 50, paint);
```

```
    canvas.drawLine(50,100,0,50, paint);
```

```
}
```

styl a barva jak kreslit



RENDEROVÁNÍ TEXTU

- ◉ text daný jako String nebo char []

```
public void drawText(String text, float x, float y, Paint paint)
```

```
public void drawText(char[] text, int index, int count, float  
x, float y, Paint paint)
```

zobrazení textu lze proložit podél křivky, nebo přímo určit pozici každého znaku

PŘÍKLAD

do metody `onDraw()` přidáno:

```
paint.setColor(Color.BLUE);
```

```
paint.setTextSize(20);
```

```
canvas.drawText("MKZ - Android", 50, 120, paint);
```

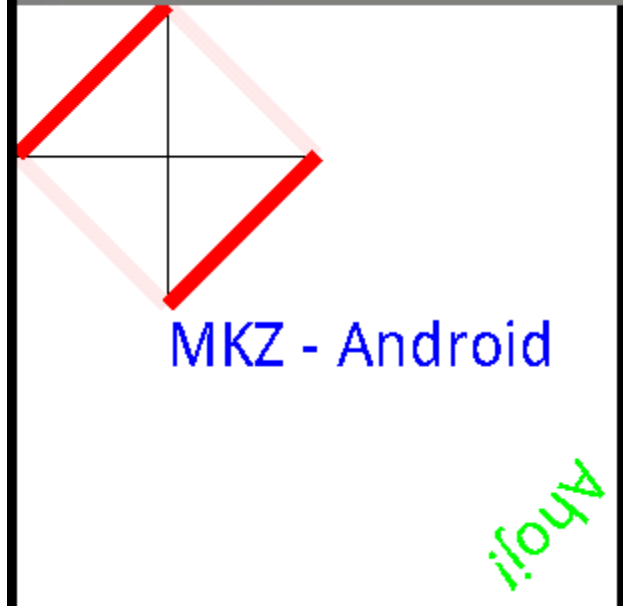
```
Path path = new Path();
```

```
path.addArc(new RectF(0, 0, 200, 200), 30, 100);
```

```
paint.setColor(Color.GREEN);
```

```
canvas.drawTextOnPath("Ahoj!", path, 0, 0, paint);
```

grafikaWidget1



UKÁZKOVÁ APLIKACE - KRESLENÍ

zdroj:

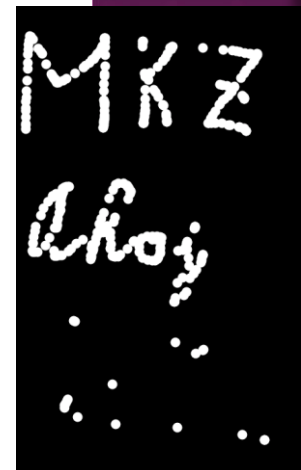
https://thenewcircle.com/s/post/1036/android_2d_graphics_example (pozn. pozor na barvu kreslení 😊)

funkcionalita:

- dotykem obrazovky kreslíme „tlustou čáru“ (kružnice v jednotlivých bodech dotyku obrazovky)
- zpracováváme událost **OnTouch()**
- ukládáme souřadnice bodů, aby bylo možné kdykoliv vše překreslit

vlastní widget DrawView

- oddělený od View
- implementuje rozhraní `OnTouchListener`
- metoda `OnDraw()` .. vykreslení
- metoda `OnTouch()` .. při události dotyku



UKÁZKA - HLAVNÍ AKTIVITA

```
public class grafika2 extends Activity {
    DrawView drawView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        /* přes celou obrazovku */
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        requestWindowFeature(Window.FEATURE_NO_TITLE);

        drawView = new DrawView(this);
        setContentView(drawView);
        drawView.requestFocus();
    }
}
```


UKÁZKA - WIDGET I.

```
public class DrawView extends View implements OnTouchListener {
```

```
List<Point> points = new ArrayList<Point>();  
Paint paint = new Paint();
```

```
public DrawView(Context context) {  
    super(context);  
    setFocusable(true);  
    setFocusableInTouchMode(true);
```

```
    this.setOnTouchListener(this);
```

```
    paint.setColor(Color.WHITE);  
    paint.setAntiAlias(true);
```

```
}
```

budeme
zpracovávat
dotykové události

konstruktor

UKÁZKA - WIDGET II.

@Override

```
public void onDraw(Canvas canvas) {  
    for (Point point : points) {  
        canvas.drawCircle(point.x, point.y, 5, paint);  
    }  
}
```

poloměr

při zavolání metody onDraw()
překreslíme všechny uložené body
každý bude zobrazen jako malá kružnice

UKÁZKA - WIDGET III. (!)

```
public boolean onTouch(View view, MotionEvent event) {
```

```
    // if(event.getAction() != MotionEvent.ACTION_DOWN)
```

```
    // return super.onTouchEvent(event);
```

```
    Point point = new Point();
```

```
    point.x = event.getX();
```

```
    point.y = event.getY();
```

```
    points.add(point);
```

```
    invalidate();
```

```
    return true;
```

```
}
```

získání x a y
souřadnice události

vyžádáme si
překreslení

co se stane, pokud odkomentujeme zelené řádky?

UKÁZKA - WIDGET - TŘÍDA POINT

```
class Point {  
    float x, y;  
  
    @Override  
    public String toString() {  
        return x + ", " + y;  
    }  
}
```

LITERATURA

- https://thenewcircle.com/s/post/1036/android_2d_graphics_example
příklad aplikace
- <http://developer.android.com/guide/index.html>
výchozí materiál