

akademia androida



Pierwsze kroki w Androidzie

część I

agenda

- Środowisko do pracy + emulator
- Struktura projektu z omówieniem
- Po co nam AndroidManifest.xml ?
- Cykl życia aplikacji
- Zadanie 1. Kod, symulacja, omówienie
- Zadanie 2.
- Podsumowanie.

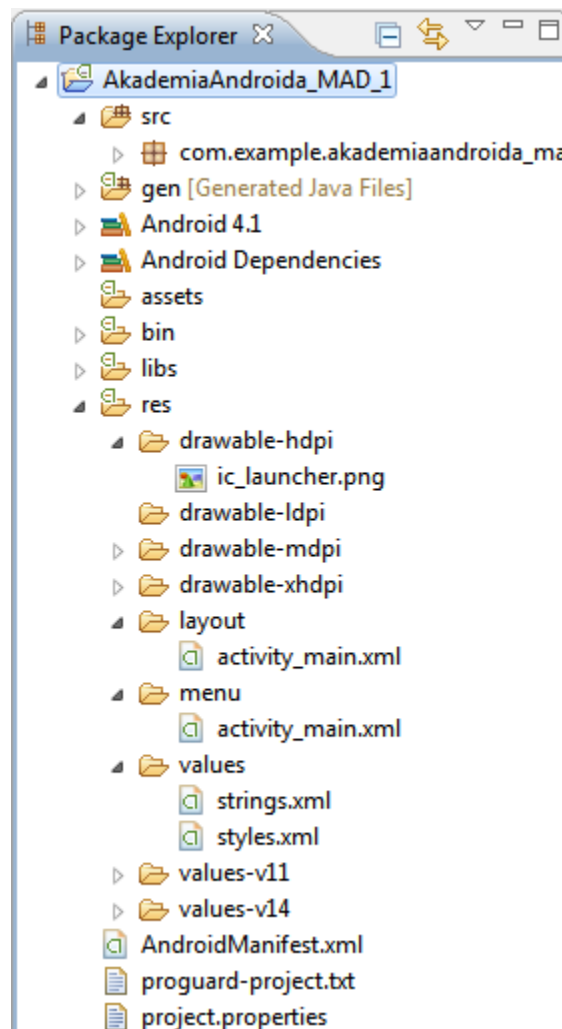
1. Środowisko

Eclipse + ADT + SDK = Wszystko czego potrzebujemy

<http://developer.android.com/sdk/index.html><http://developer.android.com/sdk/index.html>

Utworzymy projekt, AVD, SDK, LogCat, Debugger, DDMS.

2. Struktura z omówieniem



Rysunek 1. Struktura projektu Android

/assets	dowolne dane (nieskompresowane) dostępne przez AssetManager metodą getAssets(), nie traktowane jak zasoby, możliwe podkatalogi
/bin	pliki binarne *.class, *.dex, *.apk
/gen	pliki generowane (plik R.java) posegregowane na pakiety
/res	zasoby aplikacji
/res/drawable	pliki aplikacji, mapy bitowe, ikony dla aplikacji
/res/layout	opis ekranów (widoków) i rozkładu UI (plik activity_main.xml)
/res/menu	plik opisujący listy menu
/res/value	w pliku strings.xml są łańcuchy znaków do każdego elementu w aplikacji, plik styles.xml zawiera formatowanie stylu wyglądu dla aplikacji
/src	w tym katalogu znajdują się pliki źródłowe *.java aplikacji
/libs	dodatkowe potrzebne nam biblioteki, wykorzystywane w aplikacji
AndroidManifest.xml	plik deskryptora aplikacji, definiuje wszystkie występujące w aplikacji aktywności, usługi, dostawców treści, adresatów intencji, odbiorniki komunikatów, uprawnienia dla aplikacji

3. Po co nam AndroidManifest.xml?

- **Każda aplikacja musi go posiadać** (!)
- Zawiera niezbędne informacje o aplikacji dla systemu
- Zawiera:
 - Nazwę pakietu Javy, rozumianą jako unikalną nazwę
 - Opisuje komponenty aplikacji (aktywności, usługi etc.)
 - Opisana jest każda aktywność naszej aplikacji w sekcji (sekcjach) <activity> </activity>
 - Generuje się automatycznie wraz z projektem, w trakcie rozwoju aplikacji dopisujemy potrzebne nam komponenty
 - Jest najważniejszym plikiem projektu.

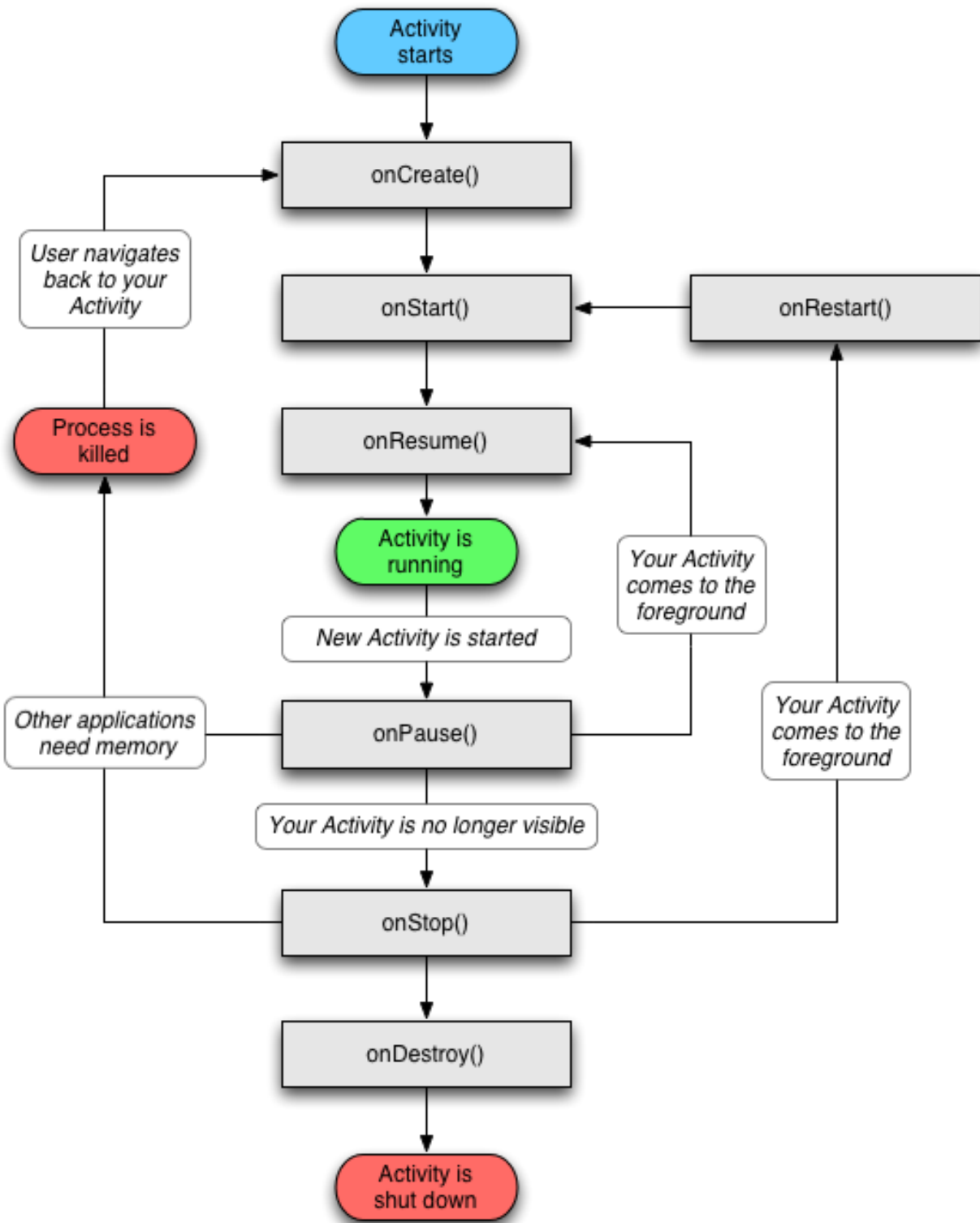
```
AkademiaAndroida_MAD_1 Manifest X
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.akademiaandroida_mad_1"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk
8         android:minSdkVersion="9"
9         android:targetSdkVersion="16" />
10
11 <application
12     android:allowBackup="true"
13     android:icon="@drawable/ic_launcher"
14     android:label="@string/app_name"
15     android:theme="@style/AppTheme" >
16     <activity
17         android:name="com.example.akademiaandroida_mad_1.MainActivity"
18         android:label="@string/app_name" >
19         <intent-filter>
20             <action android:name="android.intent.action.MAIN" />
21
22             <category android:name="android.intent.category.LAUNCHER" />
23         </intent-filter>
24     </activity>
25 </application>
26
27 </manifest>
```

Rysunek 2. Struktura pliku AndroidManifest.xml

4. Cykl życia aplikacji.

Nazwa	Opis	Pomiąć?
OnCreate()	Wywoływana, gdy Aktywność jest uruchamiana. Tutaj powinno nastąpić inicjowanie wszystkich widoków, list i innych zmiennych. Do metody przekazywany jest obiekt klasy Bundle zawierający zapisany stan aktywności z poprzedniego uruchomienia, o ile taki stan został przechwycony. Po tej metodzie następuje zawsze onStart() . Metoda ta jest wywoływana tylko raz w całym cyklu życia aplikacji.	Nie
OnRestart()	Wywoływana po onStop() , gdy Aktywność wraca na ekran. Po niej zostanie wywołana metoda onStart() .	Nie
OnStart()	Wywoływana po onCreate() lub po onRestart() . Wywoływana tuż przed pojawieniem się Aktywności na ekranie. Po niej następuje zawsze onResume() .	Nie
OnResume()	Wywoływana po onStart() lub po onPause() . W czasie realizacji tej metody, Aktywność jest już na ekranie. Jest to dobre miejsce na rozpoczęcie odtwarzania animacji lub otwarcie urządzeń wymagających wyłączności, takich jak kamera. Po niej następuje normalna praca aplikacji.	Nie
OnPause()	Wywoływana, gdy system zamierza uruchomić inną Aktywność. Jest to dobre miejsce na zapisanie wszystkich ważnych danych. Gdy brakuje pamięci to system może zdecydować się na uśmiercenie procesu tuż po zakończeniu przetwarzania tej metody i inne już nie zostaną wywołane. W tej metodzie powinno również nastąpić zatrzymanie wszystkich animacji i zwalnianie niepotrzebnych zasobów. Inna aktywność nie zostanie uruchomiona do czasu, gdy nie nastąpi powrót z tej metody. Po tej metodzie następuje onResume() lub onStop() .	Nie
OnStop()	Następuje po onPause() , gdy aplikacja nie jest już widoczna. Należy pamiętać, że w przypadku brak pamięci ta metoda może w ogóle nie być wywołana, gdyż proces zostanie uśmiercony po wywołaniu onPause() . Po niej następuje onRestart() lub onDestroy() .	Tak
OnDestroy()	Wywoływana przed uśmierceniem aplikacji. Tutaj powinno nastąpić ostateczne zwalnianie wszystkich zasobów. Tak jak w przypadku onPause() , nie można mieć pewności, że ta metoda zostanie w ogóle wywołana.	Tak

Tabela 1. Opis aktywności. Źródło: <http://programowanie.opole.pl/archives/755>



Rysunek 3. Diagram cyklów życia aplikacji

5. Zadanie 1.

Stworzyć nowy projekt, dodać główną aktywność , dodać do głównej aktywności metody onResume, onPause.... w każdej z nich dodać do textView informację o tym, że ta metoda została wywołana.

TIP1.

Tworzenie logów w androidzie. Dobrym zwyczajem jest zadeklarowanie sobie zmiennej typu String z zawartością, która będzie tagiem dla logów. Same zaś logi tworzymy w następujący sposób: `Log.d(TAG, WIADOMOŚĆ);`

Kod źródłowy zostanie omówiony na spotkaniu, po spotkaniu dostępny będzie na stronie koła naukowego M.A.D w dziale „Akademia Androida”.

Zadanie 2 - domowe

Stworzyć aplikację, która po uruchomieniu, zamknięciu przyciskiem Home, a następnie ponownym uruchomieniu wyświetli przycisk z etykietą "Zamknij". Zadaniem przycisku będzie zakończyć działanie aplikacji, tak aby zaistniał stan onDestroy().

TIP2.

Strings.xml. Byłoby dobrze gdyby etykieta przycisku znajdowała się w pliku strings.xml

Dziękuję za uwagę !

Kontakt:

- mwojtalewicz@wi.zut.edu.pl
- <http://mad.zut.edu.pl>
- „na żywo” w budynkach WI