



APLIKACJE MOBILNE

Wykład 09

dr Artur Bartoszewski

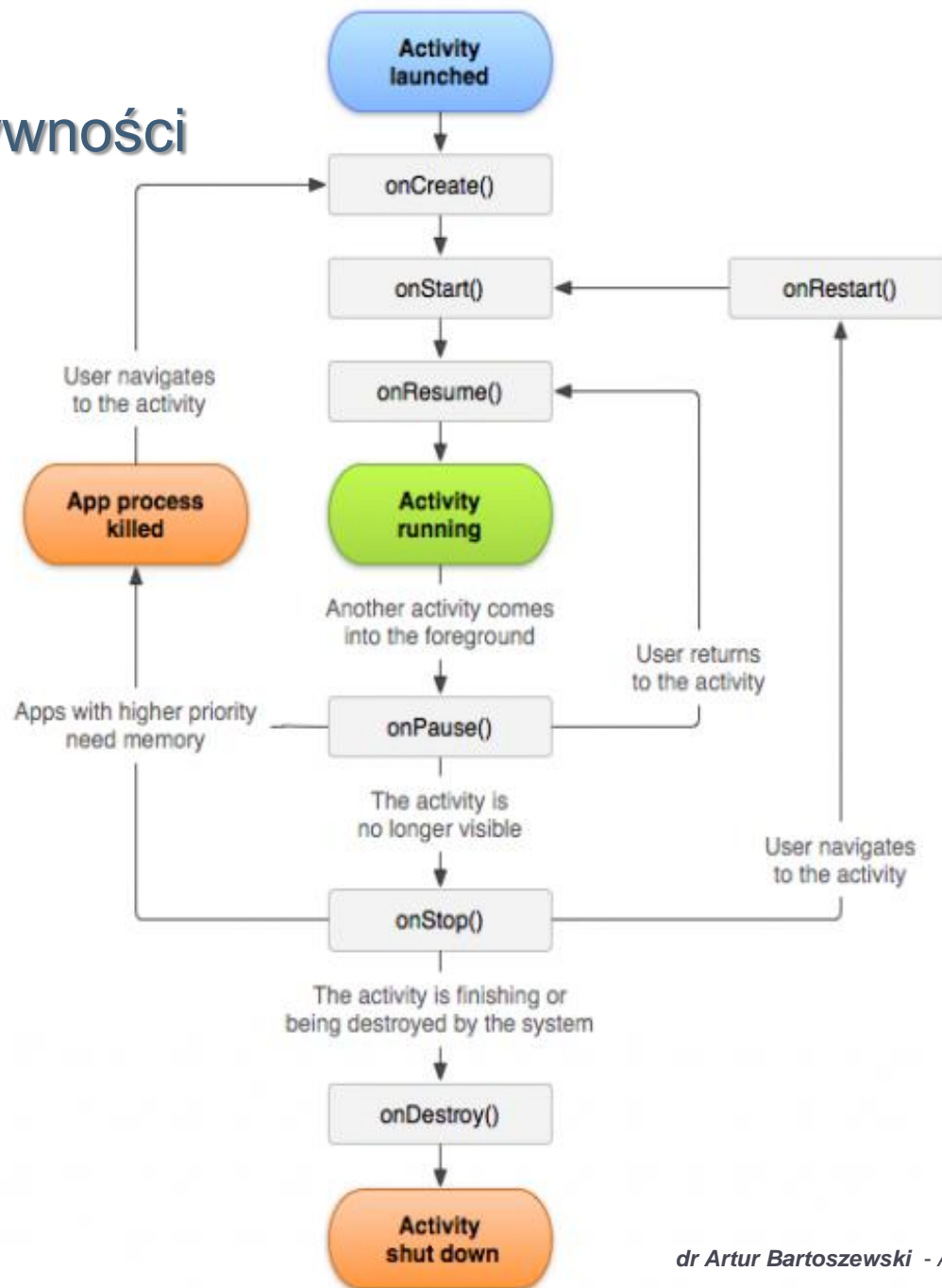


1. Cykl życia aktywności

Cykl życiowy Aktywności w systemie Android składa się z 7 stanów.



Cykl życia aktywności



Cykl życia aktywności

Aktywność aplikacji (od stworzenia do zamknięcia)

1. Aplikacja zostaje uruchomiona, czyli zostaje utworzona instancja obiektu Aktywność(system tworzy obiekt aktywności poprzez uruchomienie jego konstruktora).
2. Po uruchomieniu obiektu aktywności zostaje wywołana metoda onCreate(), w której ciele należy umieścić kod który ma się wykonać wraz ze startem aplikacji.
3. Następnym cyklem jest działanie aktywności, w trakcie kiedy jest włączona na ekranie i użytkownik może prowadzić z nią interakcję (jest to główny stan aplikacji).
4. Metoda onDestroy() w aktywności jest wywoływana tuż przed zamknięciem aplikacji (pozwala ona na zwolnienie używanych przez nią zasobów).
5. Po wykonaniu metody onDestroy() aktywność przestaje istnieć, a aplikacja zostaje zamknięta.

4 stany aktywności

1. **Aktywny** (eng. active) – aktywność jest widoczna na pełnym ekranie i jest to jedyna aplikacja jaka jest obecnie otwarta
2. **Zapauzowany** (eng. paused) – użytkownik uruchomił inną aplikację, jednak nie zasłania ona całego ekranu (zapauzowana aplikacja jest nadal częściowo widoczna)
3. **Zatrzymany** (eng. stopped) – użytkownik zminimalizował aplikację lub otworzył inną aplikację, która zajmuje cały ekran.
4. **Zniszczony** (eng. destroyed) – system Android dynamicznie zarządza pamięcią. System może zniszczyć (usunąć z pamięci) aktywność, która jest w stanie zatrzymanym lub zapauzowanym (bardzo rzadko) w celu uzyskania dodatkowej pamięci.

Cykl życia aktywności

Metody `onStop()`, `onPause()` i `onResume()`

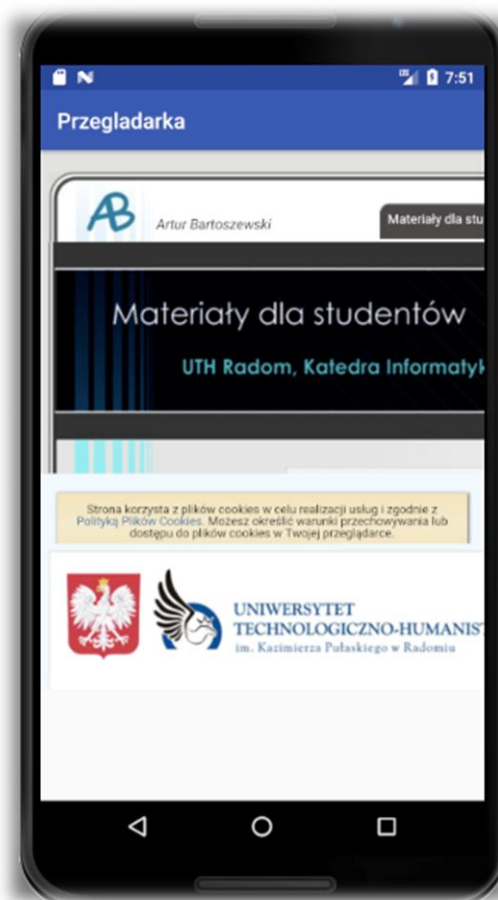
1. Metoda `onStop()` jest uruchamiana, podczas gdy aplikacja jest zamykana.
2. Metoda `onPause()` istnieją trzy momenty jej wywołania:
 - jest wywoływana przed metodą `onStop()` w przypadku zamykania aplikacji.
 - jest wywoływana, gdy użytkownik przejdzie do innej aktywności
 - jest wywołania w momencie, w którym obracamy ekran smartfona.
3. Metoda `onResume()` jest wywoływana tak jak metoda `onPause()` w momencie, gdy obracamy ekran smartfona, lub przy powrocie do działania po wcześniejszym wywołaniu metody `onPause()`

onCreate()	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	onStart()
onRestart()	Wywoływana po onStop(), gdy Aktywność wraca na ekran. Po niej zostanie wywołana metoda onStart().	Nie	onStart()
onStart()	Wywoływana po onCreate() lub po onRestart(). Wywoływana tuż przed pojawieniem się Aktywności na ekranie. Po niej następuje zawsze onResume().	Nie	onResume() ()
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()
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 następować zatrzymywanie 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	onResume() (), onStop()
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	onRestart() (), onDestroy() ()
onDestroy()	Wywoływana przed uśmierceniem aplikacji. Tutaj powinno następować ostateczne zwalnianie wszystkich zasobów. Tak jak w przypadku onPause(),		

Cykl życia aktywności

```
1  @Override
2  public void onCreate(Bundle savedInstanceState) {
3      super.onCreate(savedInstanceState);
4      //Aktywność jest tworzona
5  }
6  @Override
7  protected void onStart() {
8      super.onStart();
9      //Aktywność jest przed ukazaniem się na ekranie
10 }
11 @Override
12 protected void onResume() {
13     super.onResume();
14     //Aktywność jest już na ekranie
15 }
16 @Override
17 protected void onPause() {
18     super.onPause();
19     //Aktywność jest przed przysłonięciem przez inną Aktywność
20 }
21 @Override
22 protected void onStop() {
23     super.onStop();
24     //Aktywność nie jest już widoczna
25 }
26 @Override
27 protected void onDestroy() {
28     super.onDestroy();
29     //Aktywność jest przed usunięciem z pamięci
30 }
```

Wyświetlanie stron www bezpośrednio w aplikacji



Dodawanie obiektu WebView

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context="bartoszewski.artur.przegladarka.MainActivity">
8
9      <WebView
10         android:id="@+id/webView01"
11         android:layout_width="match_parent"
12         android:layout_height="300dp">
13
14      </WebView>
15
```

Obsługa obiektu WebView

```
1 package bartoszewski.artur.przeglądarka;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13        WebView webView = (WebView) findViewById(R.id.webView01);
14        webView.getSettings().setJavaScriptEnabled(true);
15        webView.loadUrl("http://www.bartoszewski.uniwersytetradom.pl");
16    }
17 }
```

Do załadowania strony służy metoda `.loadURL(http://...)`

Uprawnienia

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="bartoszewski.artur.przeglادarka">
4     <uses-permission android:name="android.permission.INTERNET" />
5
```

W pliku AndroidManifest dodajemy uprawnienia do korzystania z Internetu

ZADANIE PRAKTYCZNE:

