

# Aktywności i Fragmenty

Mateusz Kluska, Mateusz Nowak, Piotr Słowikowski

# Z czego składają się ekrany

- plik XML (widok)
- plik Java (aktywności)

# Aktywności

- Aktywności służą do definiowania zwykłych funkcjonalności jak generowanie okna aplikacji czy komunikacja z użytkownikiem
- Klasa aktywności musi dziedziczyć z klasy `android.app.Activity`
- Aktywności należy powiązać z widokami. Dzięki temu mają one wpływ na wygląd konkretnych ekranów.

# Cykl życia aktywności

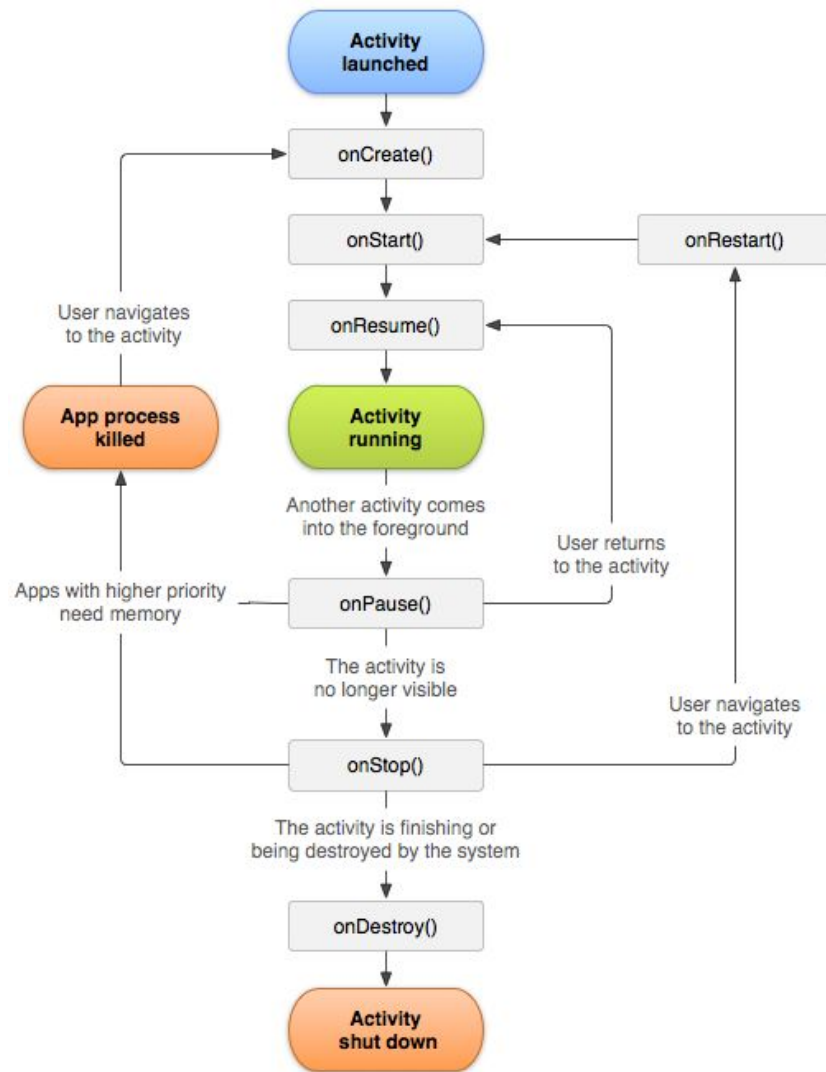
4 stany aktywności:

- Aktywność nie istnieje
- Aktywność w stanie foreground
- Aktywność w stanie background
- Aktywność w stanie pauzy

# Cykl życia aktywności

7 głównych metod w klasie aktywności:

- onCreate()
- onStart()
- onResume()
- onPause()
- onStop()
- onRestart()
- onDestroy()



# Tworzenie Aktywności

- Stworzyć podklasę klasy `android.app.Activity`
- Zaimplementować metodę `onCreate()`
- Powiązać layout z aktywnością poprzez metodę `setContentView(R.layout.main)`, gdzie:
  - R oznacza zasób (Resource)
  - layout oznacza ułożenie elementów
  - main jest plikiem xml który stworzyliśmy w katalogu `res->layout->main.xml`

# Szkielet aktywności

```
public class MainActivity extends Activity {
```

```
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }
```

```
    @Override  
    protected void onStart() {  
        super.onStart();  
    }
```

```
    @Override  
    protected void onRestart() {  
        super.onRestart();  
    }
```

```
    @Override  
    protected void onResume() {  
        super.onResume();  
    }
```

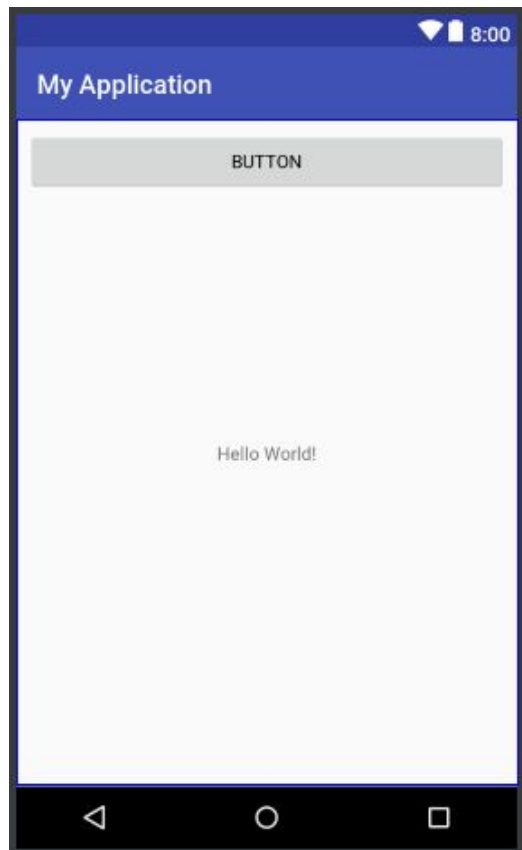
```
    @Override  
    protected void onPause() {  
        super.onPause();  
    }
```

```
    @Override  
    protected void onPause() {  
        super.onPause();  
    }
```

```
    @Override  
    protected void onStop() {  
        super.onStop();  
    }
```

```
    @Override  
    protected void onDestroy() {  
        super.onDestroy();  
    }
```

```
}
```



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android=
    "http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.piot.myapplication.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <LinearLayout
        android:layout_width="368dp"
        android:layout_height="231dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:orientation="vertical"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <Button
            android:id="@+id/button2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Button" />

    </LinearLayout>

</android.support.constraint.ConstraintLayout>
```



# Włączanie aktywności

Przed skorzystaniem ze stworzonej przez nas aktywności należy zadeklarować ją w pliku AndroidManifest.xml za pomocą tagu `<activity>`. Dzięki takiej modyfikacji pliku manifest aktywność zacznie być widoczna w projekcie.

# Deklaracja aktywności

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.piotr.myapplication">

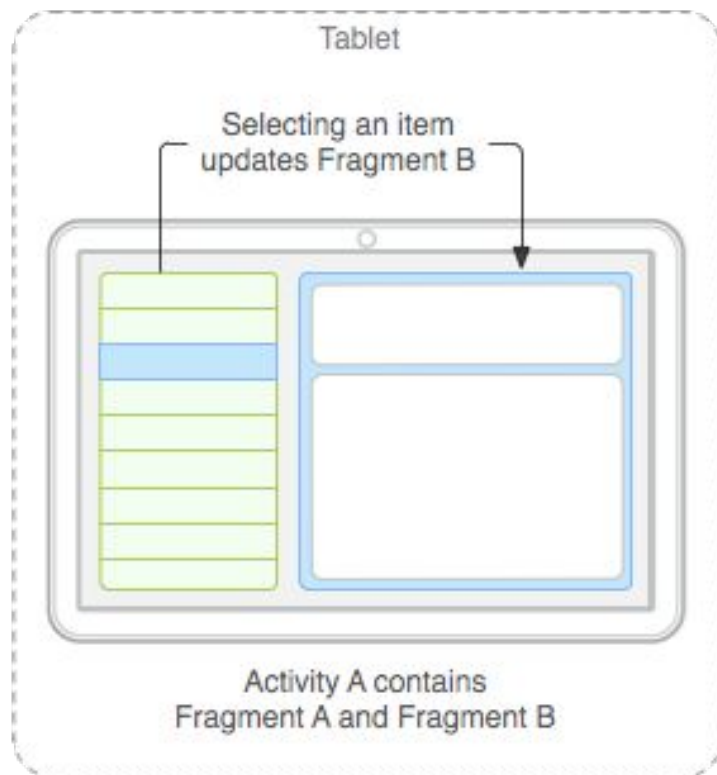
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

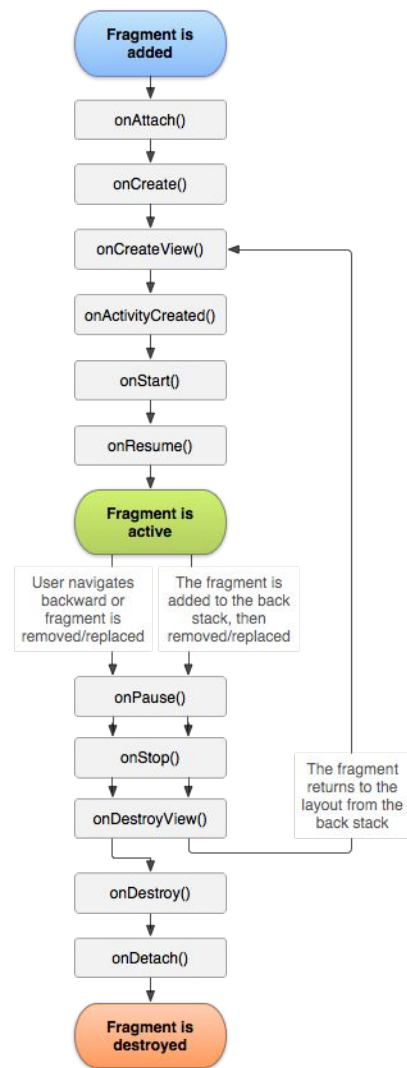
# Fragmenty

Aby ułatwić projektowanie UI na urządzenia o różnych szerokościach ekranu, wprowadzono fragmenty. Fragment to komponent, który może być użyty przez aktywność. Jest kontenerem dla danych elementów UI oraz dla funkcjonalności, które są im przypisane.



# Cykl życia fragmentu

- **onAttach()** - sygnalizuje, kiedy fragment został dodany do Activity i nadaje się do użytku w aplikacji
- **onCreateView()** - wywoływana w celu utworzenia hierarchii widoku powiązanej z fragmentem
- **onActivityCreated()** - metoda wywoływana kiedy metoda onCreate() z Activity się wykona
- **onStart()** - wywoływana, kiedy hierarchia widoku powiązana z fragmentem jest usuwana.
- **onDestroyView()** - wywoływana, kiedy fragment przestaje być powiązany z Activity



# Przykładowy layout dla fragmentu

```
<LinearLayout ...>

    <!-- ... -->

    <TextView android:id="@+id/text"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="Hello, I am a TextView" />

    <Button android:id="@+id/button"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:text="Hello, I am a Button" />

    <!-- ... -->

</LinearLayout>
```

# Przykład klasy rozszerzającej Fragment

```
import android.os.Bundle;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.ViewGroup;

public class ExampleFragment extends Fragment {

    @Override

    public View onCreateView(LayoutInflater inflater, ViewGroup container,

        Bundle savedInstanceState) {

        // Inflate the layout for this fragment

        return inflater.inflate(R.layout.example_view, container, false);

    }

}
```

# Dodanie fragmentu do activity przez plik XML

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

    android:orientation="horizontal"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent">

    <!-- ... -->

    <fragment android:name="com.example.android.fragments.ExampleFragment"

        android:id="@+id/example_fragment"

        android:layout_weight="2"

        android:layout_width="0dp"

        android:layout_height="match_parent" />

    <!-- ... -->

</LinearLayout>
```



# Dodanie fragmentu z kodu

- Pobranie instancji klasy `FragmentManager`

```
FragmentManager fragmentManager = getSupportFragmentManager();  
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
```

- Następnie możemy np. dodać fragment w następujący sposób

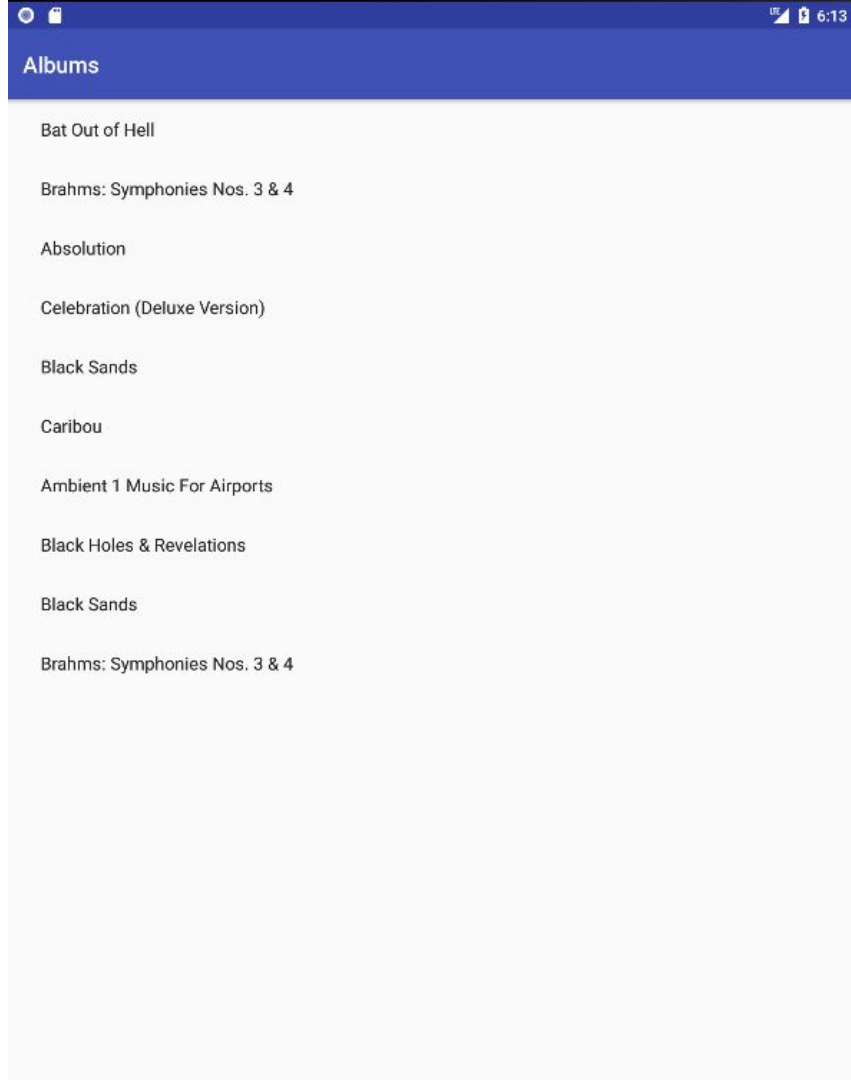
```
ExampleFragment fragment = new ExampleFragment();  
fragmentTransaction.add(R.id.fragment_container, fragment);  
fragmentTransaction.commit();
```

# Predefiniowane klasy rozszerzające Fragment

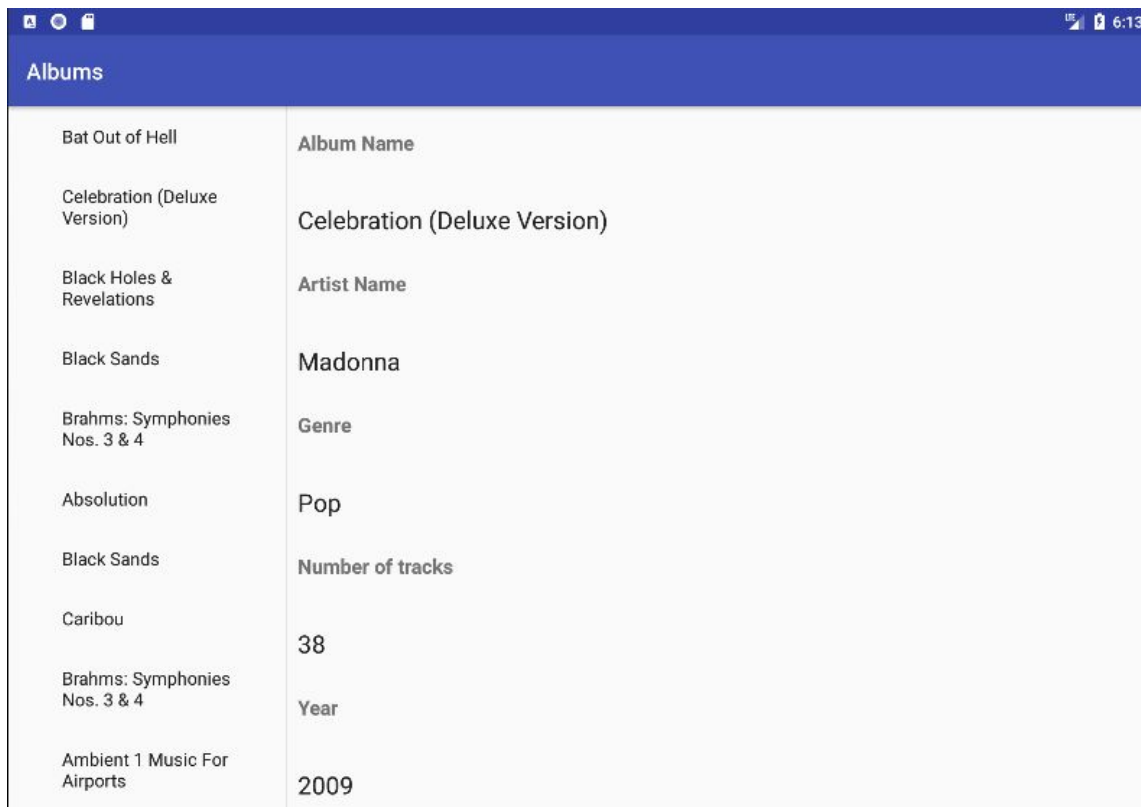
- **ListFragment** - posiada wbudowany ListView i kilka pomocniczych metod, takich jak nieokreślony ProgressBar i TextView do wyświetlania tekstu
- **DialogFragment** - odpowiedzialny za wyświetlanie okna dialogowego
- **MapFragment** - przydatny gdy chcemy wykorzystać Google Maps w swojej aplikacji
- **PreferenceFragment** - wspomaga tworzenie widoku ustawień

# Zadanie

Prosta przeglądarka albumów muzycznych z 2 aktywnościami i fragmentem. Wyświetlane różnie w zależności od rozdzielczości ekranu.



# Zadanie - widok szczegółów 1



Album Name	Artist Name	Genre	Number of tracks	Year
Bat Out of Hell				
Celebration (Deluxe Version)				
Black Holes & Revelations				
Black Sands	Madonna			
Brahms: Symphonies Nos. 3 & 4				
Absolution		Pop		
Black Sands				
Caribou			38	
Brahms: Symphonies Nos. 3 & 4				
Ambient 1 Music For Airports				2009

# Zadanie - widok szczegółów 2

Kod do uzupełnienia:

[github.com/matkluska/android\\_lab3](https://github.com/matkluska/android_lab3)

