



# Wprowadzenie do systemu Android

Rafał Zientara



Co już wiecie?



## Android - koncepcje

- Otwarty system (różne wersje i modyfikacje)
- Modyfikacja systemu Linux
- Urządzenie mobilne (różne podzespoły)
- System uprawnień
- Otwieranie aplikacji przez wiele ścieżek
- Różne urządzenia (wearable, TV itd.)



## Android - fragmentacja

- Wsparcie różnych urządzeń
- UI (różne layouty)
- Resources (różne zasoby zależnie od sprzętu, przypadku użycia itd..)



## Komponenty

- Activity
- Service/IntentService
- Broadcast Receivers (przechwytywanie zdarzeń z systemu)
- Content Providers (np. intencje dla innych aplikacji)



## AndroidManifest.xml

- Aplikacja <application>
- Komponenty aplikacji <activity> <service> <receiver> <provider>
- Uprawnienia wymagane przez aplikacje <uses-permission>
- Z jakich funkcji systemu korzysta aplikacja <uses-feature>
- Metadane <meta-data>



# AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pl.wroc.sda.helloapp">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="HelloApp"
        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>

        <receiver android:name=".PhoneReceiver" >
            <intent-filter>
                <action android:name="android.intent.action.PHONE_STATE" >
            </action>
            </intent-filter>
        </receiver>
    </application>

</manifest>
```



## Zadanie 1

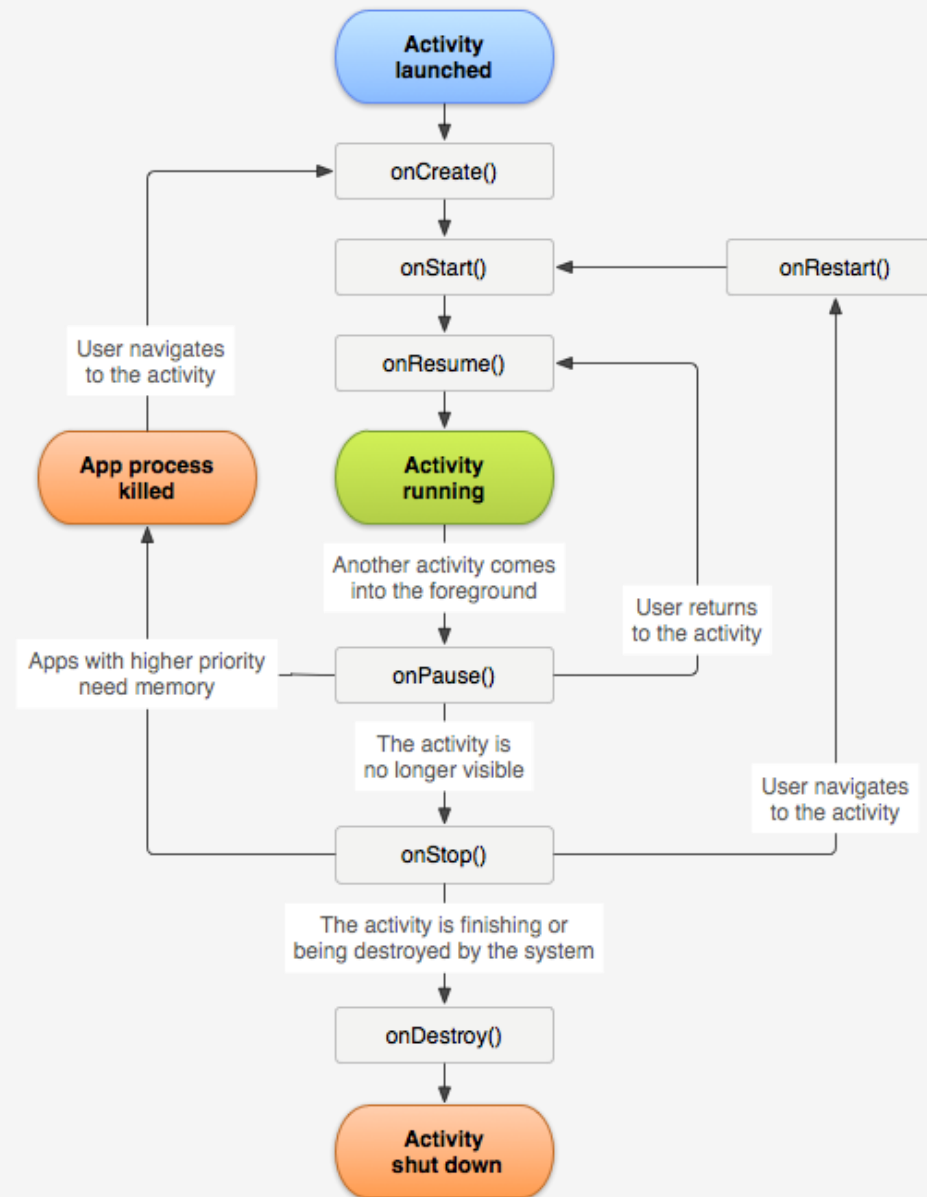


Utwórz aplikację która:

- Otwiera aktywność MainActivity
- Wyświetla tekst „Witaj <Imię> <Nazwisko>!”
- Wyświetla godzinę uruchomienia Activity



# Cykl życia Activity





## Intent

- Wywołanie aktywności, serwisu, wysłania akcji broadcast
- Wywołanie żądania o usługę
- Przekazanie danych do innego komponentu (Bundle)
- Może wywoływać różnego typu akcje (np. ACTION\_VIEW, ACTION\_EDIT)
- Zawiera informacje o kategorii



## Intent

```
Intent intent = new Intent(this, MainActivity.class);  
intent.putExtra("login", "student");  
startActivity(intent);
```

```
Intent intent = new Intent();  
intent.setAction(Intent.ACTION_SEND);  
intent.putExtra(Intent.EXTRA_TEXT, "Wiadomość");  
intent.setType("text/plain");  
  
if(intent.resolveActivity(getPackageManager()) != null) {  
    startActivity(intent);  
} else {  
    Log.d(TAG, "Brak komponentu obsługującego zapytanie");  
}
```



## Zadanie 2



- W aktywności MainActivity dodaj przycisk „Dalej” i pole na wprowadzenie nazwiska
- Przekaż nazwisko do kolejnej aktywności HelloActivity
- Wyświetl tekst „Witaj na następnej aktywności <Imię> <Nazwisko>!”

## Popularne uprawnienia używane w aplikacjach

ACCESS\_COARSE\_LOCATION  
ACCESS\_FINE\_LOCATION  
ACCESS\_NETWORK\_STATE  
INTERNET  
READ\_EXTERNAL\_STORAGE  
WRITE\_EXTERNAL\_STORAGE

# Uprawnienia

Od Api 23 (Android 6.0) wymagana zgoda użytkownika dla poszczególnych uprawnień poprzez okno dialogowe

READ\_CALENDAR  
WRITE\_CALENDAR  
CAMERA  
READ\_CONTACTS  
WRITE\_CONTACTS  
GET\_ACCOUNTS  
ACCESS\_FINE\_LOCATION  
ACCESS\_COARSE\_LOCATION  
RECORD\_AUDIO  
READ\_PHONE\_STATE  
CALL\_PHONE  
READ\_CALL\_LOG

WRITE\_CALL\_LOG  
ADD\_VOICEMAIL  
USE\_SIP  
PROCESS\_OUTGOING\_CALLS  
BODY\_SENSORS  
SEND\_SMS  
RECEIVE\_SMS  
READ\_SMS  
RECEIVE\_WAP\_PUSH  
RECEIVE\_MMS  
READ\_EXTERNAL\_STORAGE  
WRITE\_EXTERNAL\_STORAGE



# Uprawnienia

```
private void checkPermission() {  
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {  
        if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.SEND_SMS)) {  
            Log.d(TAG, "Zezwolono - można bezpiecznie korzystać z uprawnienia");  
        } else {  
            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.SEND_SMS}, MY_REQUEST_CODE);  
        }  
    }  
}
```

```
@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
    switch (requestCode) {  
        case MY_REQUEST_CODE:  
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {  
                Log.d(TAG, "Zezwolono - można bezpiecznie korzystać z uprawnienia");  
            } else {  
                Log.d(TAG, "Nie zezwolono - wyłącz funkcjonalność");  
            }  
        }  
    }  
}
```



## Zadanie 3



- Utwórz aktywność `PermissionActivity` z przyciskiem „Akcja”
- Poproś użytkownika o udzielenie przynajmniej dwóch niebezpiecznych uprawnień
- Zablokuj przycisk „Akcja” jeśli użytkownik nie wyraził zgody na użycie uprawnień przez aplikację





## Broadcast Receiver

- Odbieranie zdarzeń z systemu
- Może służyć jako punkt wejścia do aplikacji
- Nie posiada interfejsu użytkownika
- Krótkie operacje w tle
- Może służyć do komunikacji między komponentami



## Broadcast Receiver

```
public class SmsReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Log.d(getClass().getSimpleName(), "Dostałeś SMS!");  
    }  
}
```



## Broadcast Receiver

- Przykładowe Broadcast Receivery w klasie Intent

ACTION\_TIME\_TICK  
ACTION\_TIME\_CHANGED  
ACTION\_TIMEZONE\_CHANGED  
ACTION\_BOOT\_COMPLETED  
ACTION\_PACKAGE\_ADDED  
ACTION\_PACKAGE\_CHANGED  
ACTION\_PACKAGE\_REMOVED  
ACTION\_PACKAGE\_RESTARTED  
ACTION\_PACKAGE\_DATA\_CLEARED  
ACTION\_PACKAGES\_SUSPENDED  
ACTION\_PACKAGES\_UNSPENDED  
ACTION\_UID\_REMOVED  
ACTION\_BATTERY\_CHANGED  
ACTION\_POWER\_CONNECTED  
ACTION\_POWER\_DISCONNECTED  
ACTION\_SHUTDOWN



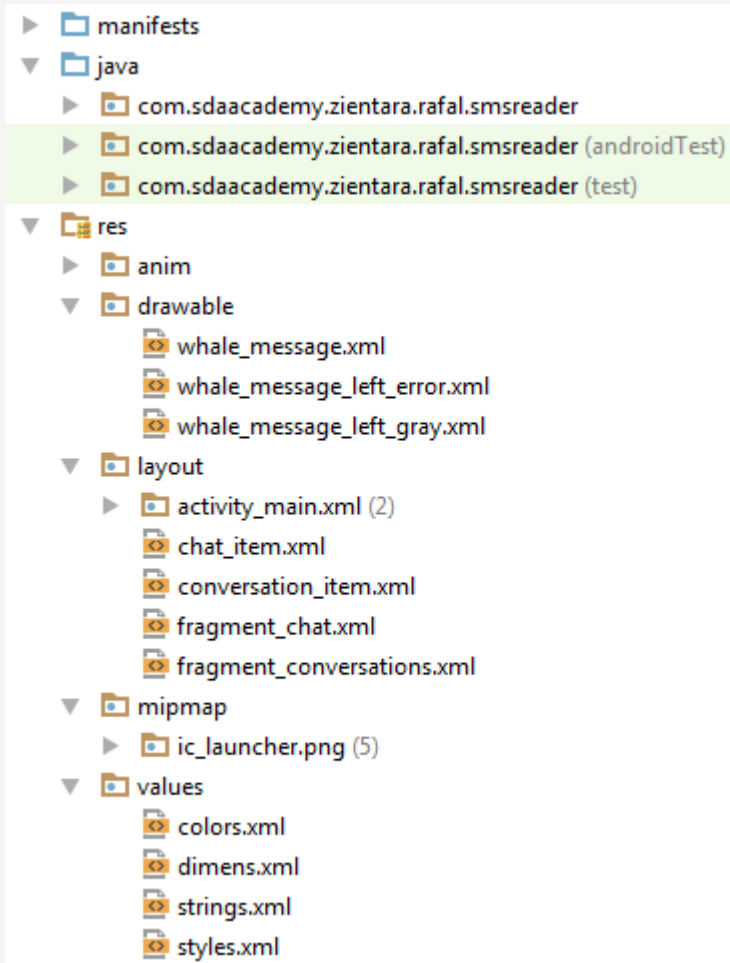
## Zadanie 4



- Utwórz `BatteryReceiver` odbierający wiadomości o stanie baterii
- Przechwyconą wiadomość wyświetl w oknie Toast
- Aby utworzyć receiver należy stworzyć klasę dziedziczącą po `BroadcastReceiver`
- Receiver trzeba zarejestrować przez manifest lub przez kod programu.



# Resources



Do zasobów każdego katalogu poprzez kod aplikacji dostajemy identyfikator przez `R.rodzaj.nazwa`

Poprzez pliki XML identyfikator ustawiamy: `@rodzaj/nazwa`



/res/anim

Pliki XML definiujące animacje widoków. Możliwe sekcje to:

<alpha><scale><translate><rotate>

<set>

<animation-list>



/res/anim

Pliki XML definiujące animacje widoków. Możliwe sekcje to:

<alpha><scale><translate><rotate>

<set>

<animation-list>



## /res/menu

- Menu aplikacji
- Menu kontekstowe
- Navigation drawer menu (hamburger)





## /res/values

- Teksty
- Kolory
- Tablice danych
- Style
- Odległości
- Atrybuty CustomView

```
<string name="action_settings">Settings</string>
<color name="white">#FFFFFF</color>
<dimen name="table_padding">8dp</dimen>
<string-array name="sweetArray">
    <item name="cookies">Cookies</item>
    <item name="chocolate">Chocolate</item>
    <item name="ice_cream">Ice cream</item>
</string-array>
```



## Colors

Moga być wpisywane w formatach

#RGB

#ARGB

#RRGGBB

#AARRGGBB



## Strings

- Teksty mogą być przechowywane w resources.
- Łatwo zaimplementować wsparcie dla różnych języków (atrybut folderu np. pl, fr itd.)
  - Część funkcji np `textView.setText()` można użyć od razu podając identyfikator tekstu

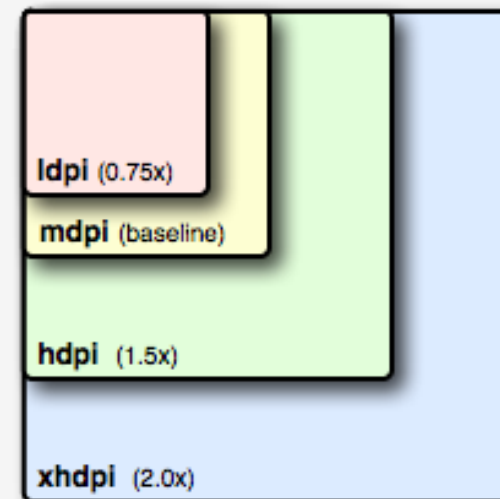
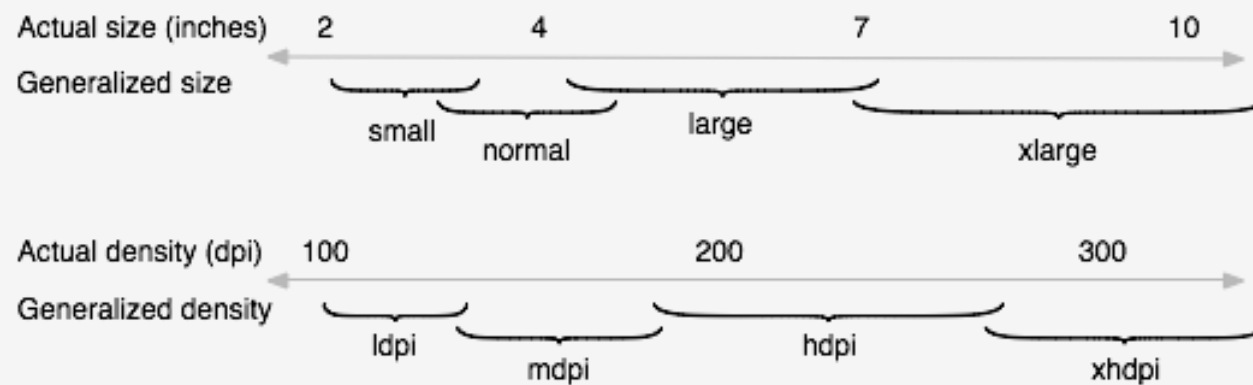


/res/raw

Pliki dowolnego typu, muzyka, pliki binarne, pdf, doc, html itd.



/res/raw





## Modyfikatory

- Gęstość ekranu (ldpi, mdpi, hdpi, xhdpi, xxhdpi, xxxhdpi, nodpi, tvdpi, anydpi)
- Ekran dotykowy (notouch, finger)
- Dostępność klawiatury (keysexposed, keyshidden, keyssoft)
- Rodzaj klawiatury (nokeys, qwerty, 12key)
- Przycisk nawigacji (navexposed, navhidden)
- Rodzaj kontrolera nie dotykowego (nonav, dpad, trackball, wheel)
- Wersja API (np. v8, v18, v25, .... )



## Modyfikatory

- MCC / mobile country code(mcc310, mcc310-mnc004, mcc208-mnc00, .....)
- Język i region (en, pl, .....)
- Kierunek layoutu (ldrtl, ldltr)
- Najmniejsza szerokość (sw<N>dp, np. sw720dp)
- Dostępna szerokość (w<N>dp, np. w720dp)
- Dostępna wysokość (h<N>dp, np. h720dp)



## Modyfikatory

- Rozmiar ekranu (small, normal, large, xlarge)
- Proporcje ekranu (long, notlong)
- Kształt ekranu (round, notround)
- Orientacja ekranu (port, land)
- Tryb interfejsu użytkownika (car, desk, television, appliance, watch)
- Nightmode(night, notnight)





## Zadanie 5



- Utwórz widok dla ustawienia portretowego i horyzontalnego
- Przygotuj grafikę o takiej samej nazwie z inną zawartością dla różnych gęstości ekranu – odkryj gęstość swojego smartfona!
- Zmień ikonę aplikacji
- Przygotuj teksty dla dwóch wersji językowych
- Przygotuj selector w drawable z różnymi stanami przycisku