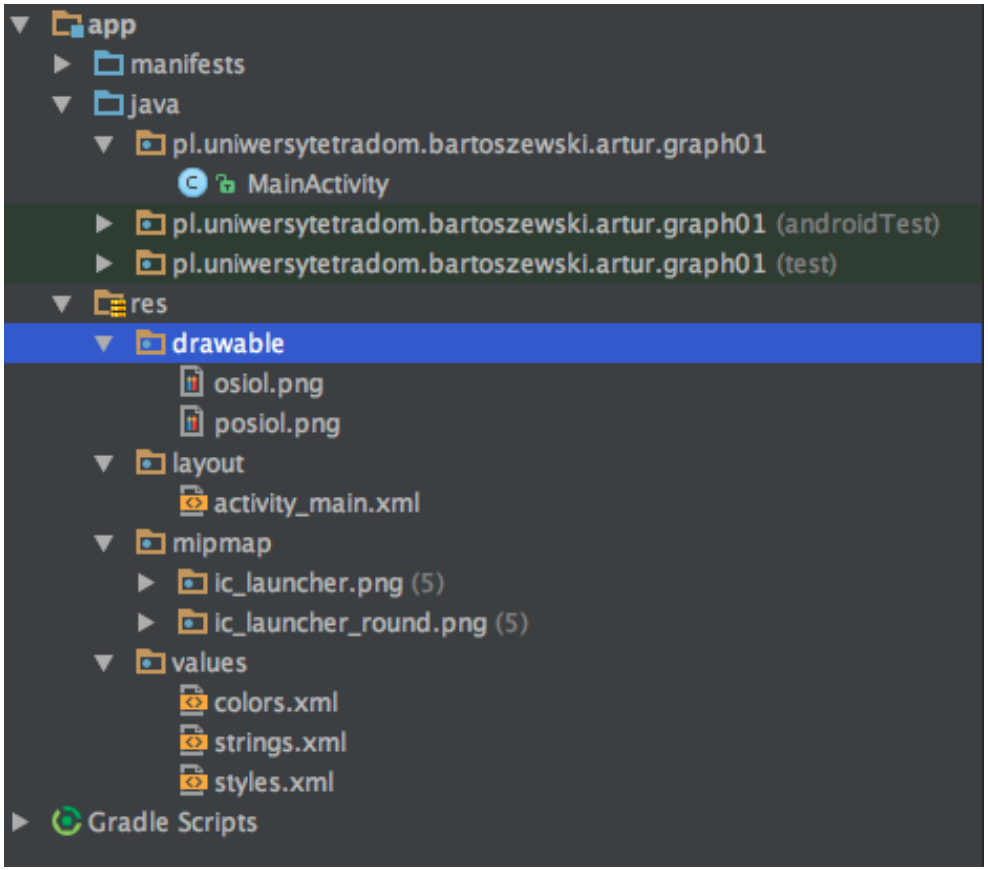
APLIKACJE MOBILNE

Wykład 04

dr Artur Bartoszewski

Zasoby aplikacji

Aby w projekcie skorzystać z zasobów, należy umieścić je podkatalogach folderu **res**.



Aplikacje mobilne

- ✓ Drawable - lokalne obrazy (pliki JPG i PNG,) należy umieścić w folderze **drawable**. Tam również umieszcza się pliki XML, definiujące obiekty wektorowe i kształty.
- ✓ Mipmap - katalog mipmap akceptuje elementy bitmapowe, ale w szczególności przechowuje ikonę aplikacji.
- ✓ Layout - katalog layout zawiera pliki XML, które definiują budowę interfejsu aplikacji.

Źródło: https://www.altkomakademia.pl

Aplikacje mobilne

- ✓ Color - katalog color jest używany do przechowywania plików XML, które kojarzą informacje o stanie z określonymi kolorami. To jest przydatne, np. gdy musisz zmienić kolor tekstu View lub tła po stuknięciu, aby dostarczyć informację zwrotną użytkownikowi. (kolory można zapisać też w katalogu values).
- ✓ Menu - pliki XML menu są używane do definiowania paska akcji, menu nawigacji i podmenu. Te zasoby znajdują się w katalogu zasobów menu.

Źródło: https://www.altkomakademia.pl

Aplikacje mobilne

- ✓ Raw - katalog raw przechowuje pliki surowe dla aplikacji (pliki audio, wideo i tekstowe). To umożliwia łatwy dostęp do plików, jednakże jeśli chcesz uzyskać dostęp do oryginalnej nazwy pliku lub hierarchii katalogów, powinieneś rozważyć umieszczenie plików surowych w katalogu asset Androida.
- ✓ Xml - ten katalog przechowuje dowolne pliki XML używane przez Androida.

Źródło: https://www.altkomakademia.pl

Katalog values - może zawierać wiele plików XML składających się z kluczy wartości, które są używane w aplikacji.

- Arrays - obiekty tablic,
- Colors – kolory (z własnymi nazwami), które mogą być użyte w wielu miejscach w twojej aplikacji.
- Dimens – rozmiary - mogą określać cokolwiek związanego z rozmiarem (np.: tekst, margines).
- Integers - jeśli istnieją konkretne liczby całkowite (stałe), które chcesz wykorzystać w swojej aplikacji, można zebrać je w pliku zasobu

Źródło: https://www.altkomakademia.pl

Aplikacje mobilne

- Strings -. zamiast rozpraszać łańcuchy znaków (teksty) w kodzie, można przechowywać je w pliku strings.xml.
- Plurals - podobne do ciągów, plurals umożliwiają dostarczanie alternatywnych ciągów, gdy liczba jest przekazywana do funkcji wydobywania.
- Styles – definiując style (plik styles.xml) można przygotować domyślny wygląd komponentów View – style wykorzystujemy budując układy.

Źródło: https://www.altkomakademia.pl

Dostęp do zasobów

Zasoby przechowywane w aplikacji posiadają identyfikowalną nazwę składającą się z typu zasobu (takiego jak np.: drawable lub layout) oraz
id (w generowanym automatycznie pliku **R)**

Źródło: https://www.altkomakademia.pl

Dostęp do zasobów – w Java

Wiele obiektów w systemie Android akceptuje id zasobu w celu jego wykorzystania.
Np.: ustawienie źródła obrazu dla ImageView:

```
ImageView.setImageResource(R.drawable.nazwa);
```

Jeżeli trzeba uzyskać oryginalny zasób w kodzie. Można to zrobić za pomocą metody getResources( )

```
getResources().getColor(R.color.kolor);
```


Źródło: https://www.altkomakademia.pl

Alternatywne wersje zasobów - rozdzielczości

Dołączając kwalifikatory takie jak: -ldpi, -mdpi, -hdpi do nazw folderów, (jak widać w tworzonym autmatycznie folderze minimaps) możemy przygotować zestawy grafik dla rożnych klas urządzeń

Źródło: https://www.altkomakademia.pl

Alternatywne wersje zasobów - języki

Można stworzyć osobne katalogi wartości dla każdej wersji językowej aplikacji - dołączając kod języka na końcu nazwy katalogu (na przykład values-de dla niemieckiego lub values-pl)

Źródło: https://www.altkomakademia.pl

Alternatywne wersje zasobów - orientacje

Można dodać kwantyfikator -land do katalogu zasobu, aby zdefiniować domyślne wartości, dla pozycji poziomej.

Można zdefiniować również minimalną szerokość urządzenia (-sw600dp lub -sw720dp)

Źródło: https://www.altkomakademia.pl

ZADANIE PRAKTYCZNE:

W pliku layout zdefiniowany jest obraz. Po kliknięciu zamienia się on na inny (zdefiniowany w kodzie Java)

Przykład

1. Tworzenie layout-u

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    android:src="@drawable/posiol"
    android:padding="20dp"
    android:onClick="zmien"
    android:clickable="true"
     />
```

Przykład

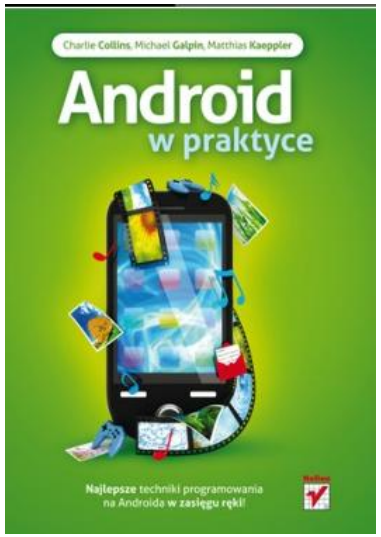
1. Kod Java

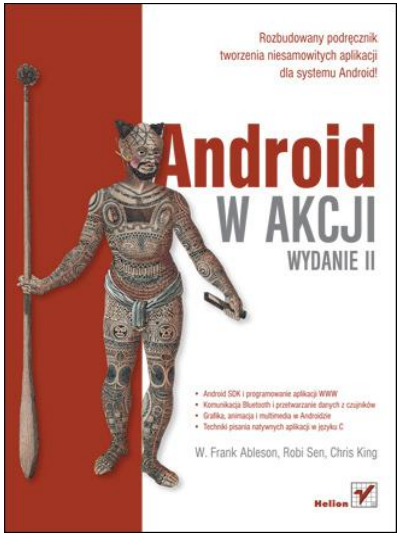
```
public void zmien (View view) {
    ImageView obrazek = (ImageView) findViewById(R.id.imageView);
    obrazek.setImageResource(R.drawable.osiol);

}
```

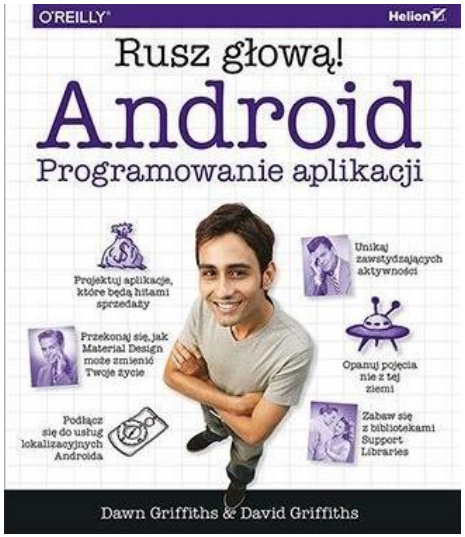
ZADANIE PRAKTYCZNE:

Program wczytuje dane i po kliknięciu przyciski „Zapisz” składa je w jeden tekst i wyświetla poniżej.

Literatura


Charlie Collins, Michael Galpin, Matthias Kaeppler
Android
w praktyce
Najlepsze techniki programowania na Androida w zasięgu ręki!


Rozbudowany podręcznik tworzenia niesamowitych aplikacji dla systemu Android!
Android
W AKCJI
WYDANIE II
W. Frank Ableson, Robi Sen, Chris King
Helion


O'REILLY
Helion
Rusz głową!
Android
Programowanie aplikacji
Projektuj aplikacje, które będą hitami sprzedaży
Unikaj zawstydzających aktywności
Przekonaj się, jak Material Design może zmienić Twoje życie
Opanuj pojęcia nie z tej ziemi
Podłącz się do usług lokalizacyjnych Androida
Zabaw się z bibliotekami Support Libraries
Dawn Griffiths & David Griffiths