



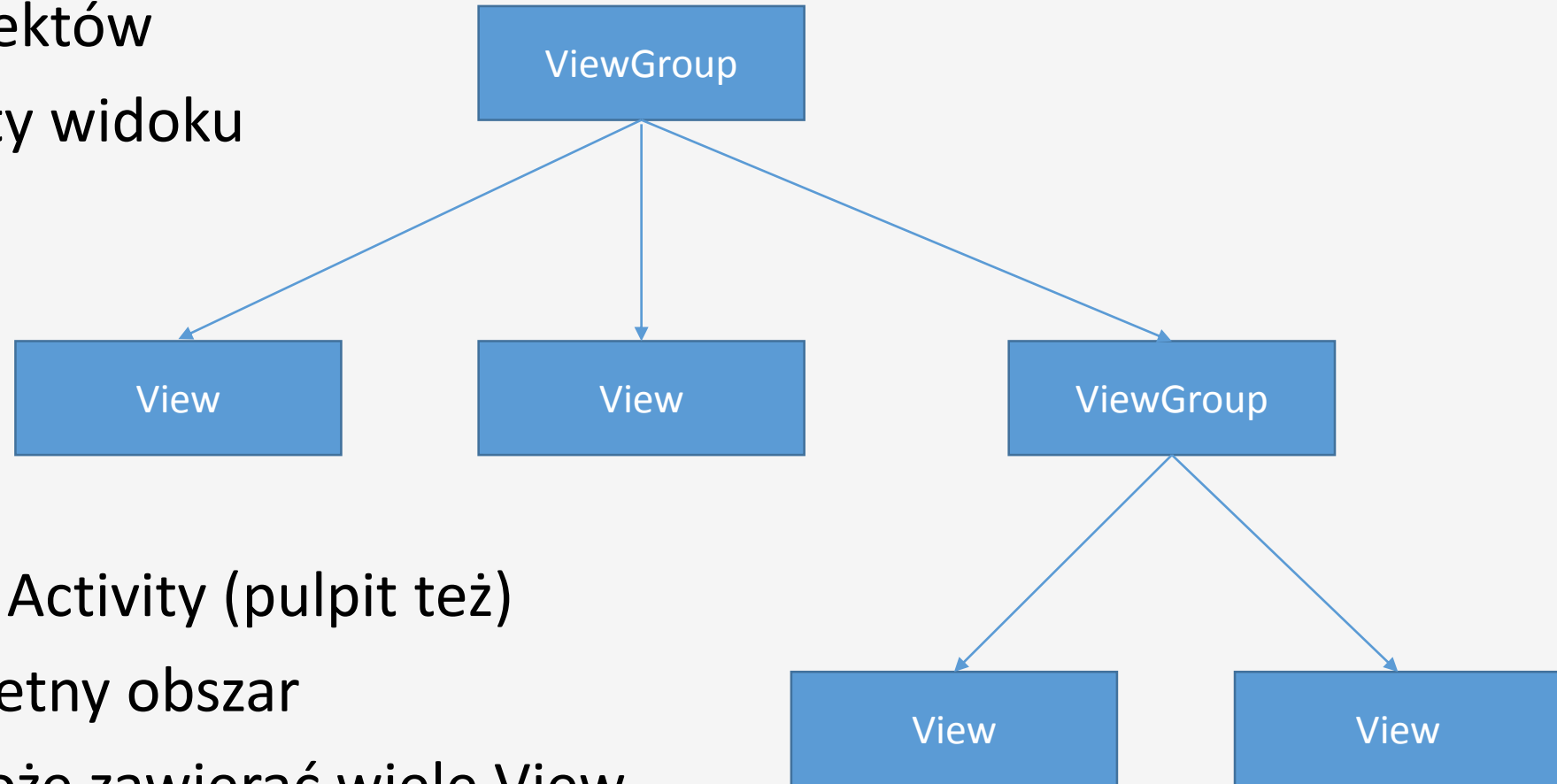
Aplikacje typu single Activity

Rafał Zientara



View vs ViewGroup

- Hierarchia obiektów
- Bazowe obiekty widoku



- Każde okno to Activity (pulpit też)
- Zajmuje konkretny obszar
- ViewGroup może zawierać wiele View



View vs ViewGroup

- Pliki XML w zasobach Layout mogą zawierać wiele atrybutów
- Aktywność korzysta z funkcji setContentView(R.layout.xml_file) by wyświetlić zawartość okna

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hello!"
        android:padding="8dp" />

</LinearLayout>
```



- Rozmiar elementów widoku można ustawiać na parę różnych sposobów
- Ustawianie rozmiaru jest łatwiejsze przez XML
- Obsługiwane atrybuty zależą od użytego elementu
- Możliwości View można rozszerzać, więc obsługują też atrybuty poprzedniej klasy

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:padding="8dp"
```



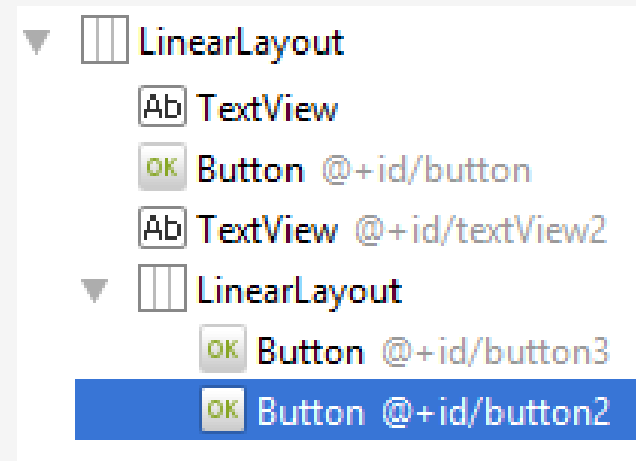
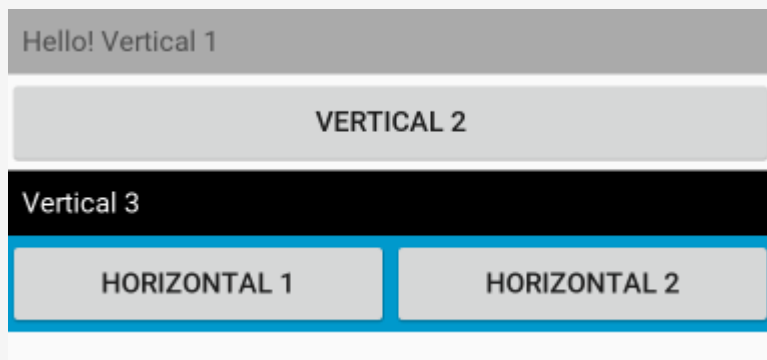
Przykładowe GroupView

- Nazwa powinna kończyć się na Layout
- Podwidokom można przypisywać obiekty LayoutParams
- Czasem ważna jest kolejność widoków w GroupView
- Widoki mają różny sposób rozmieszczania elementów
- Atrybuty można ustawiać w GroupView oraz w zawieranych elementach



Przykładowe GroupView – LinearLayout

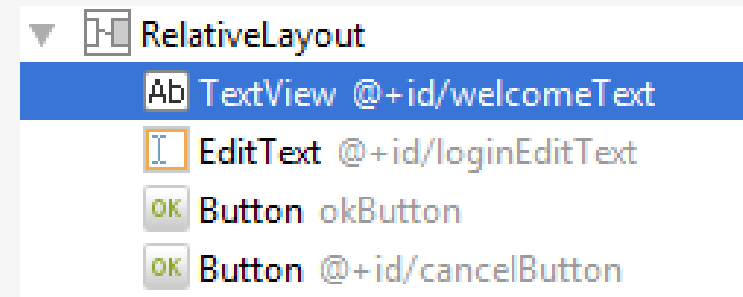
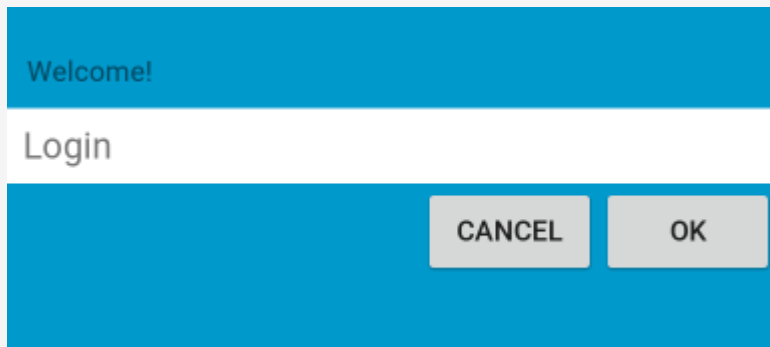
- Lista pozioma lub pionowa
- Kolejność elementów ma znaczenie
- Elementom można ustawiać wagi (layout_weight)
- Jeśli wertykalny to widoki są jedne pod drugim





Przykładowe GroupView – RelativeLayout

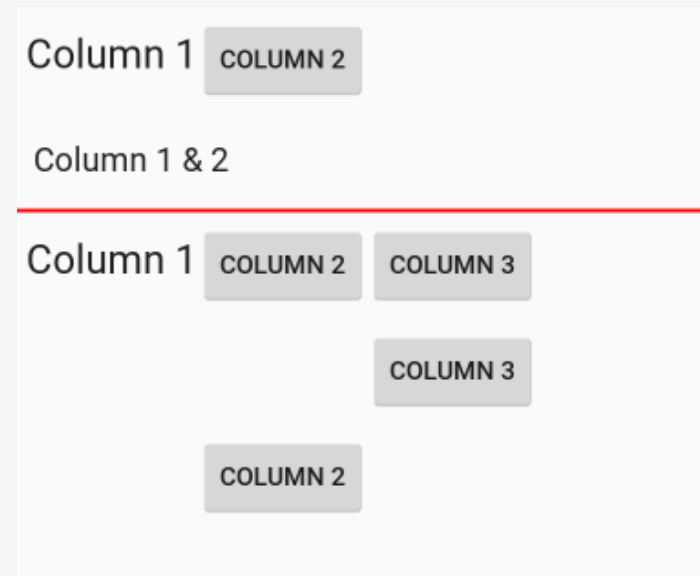
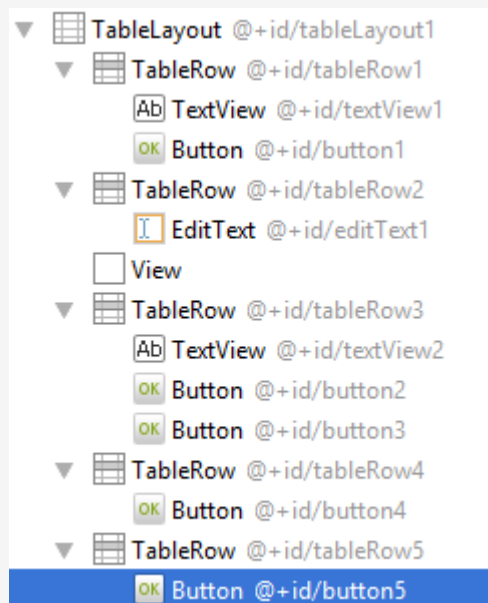
- Obiekty w dowolnej kolejności
- Obiekty są ułożone względem siebie i/lub względem rodzica





Przykładowe GroupView – TableLayout

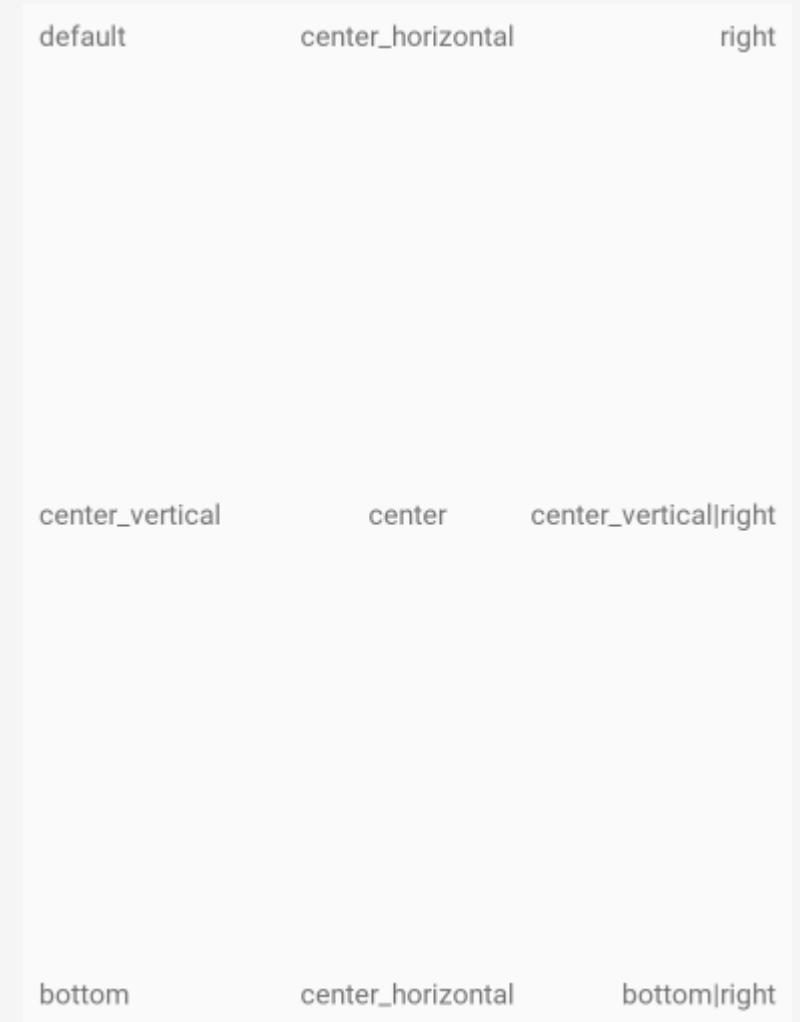
- Widok opierający się na kolumnach
- Dziedziczy po LinearLayout
- Każdy wiersz to GroupView typu TableRow
- Można scalać komórki atrybutem *layout_span*
- Można umieścić element w konkretnej kolumnie *layout_column*





Przykładowe GroupView – FrameLayout

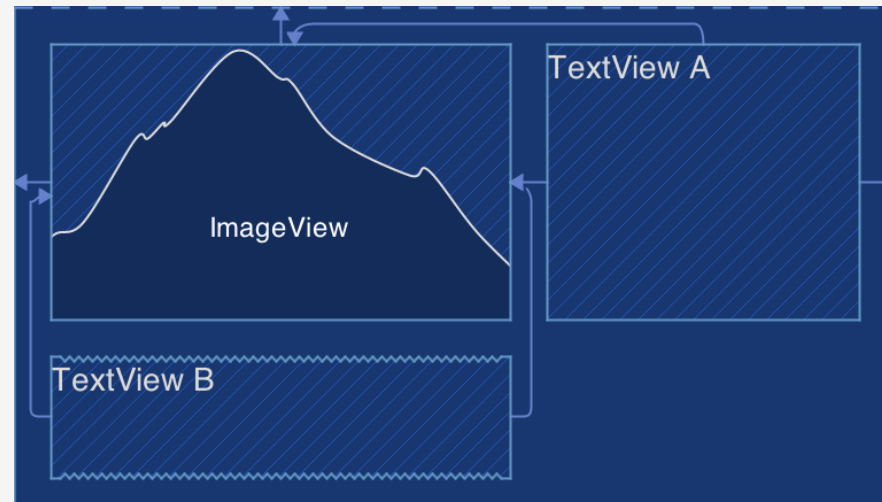
- nieskomplikowany obliczeniowo widok
- Najprostszy typ układu
- Opiera się na grawitacji względem rodzica
- Domyślna grawitacja to górne lewo
- Kolejność obiektów nie ma znaczenia





Przykładowe GroupLayout – inne

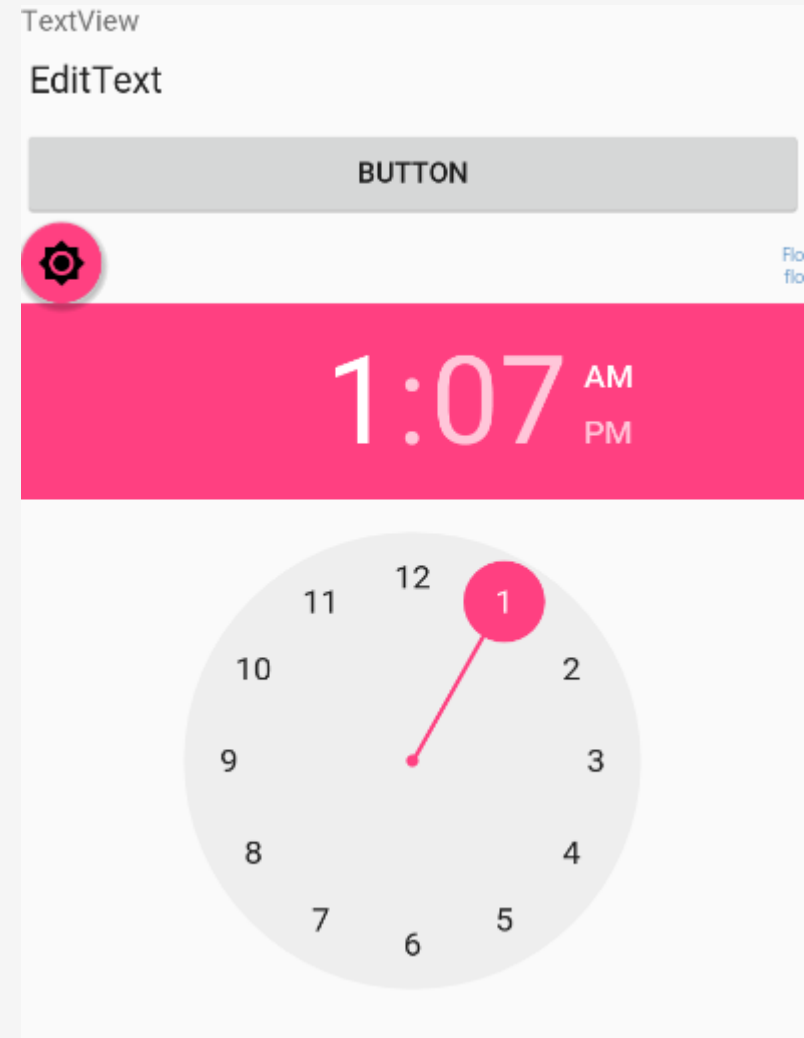
- GridLayout – siatka z określoną liczbą kolumn i/lub wierszy
- ConstraintLayout – jeden z nowszych widoków z zaimplementowanymi animacjami (RelativeLayout 2.0?)
- CardView – AppCompat
- RecyclerView – AppCompat
- RadioGroup



Przykładowe View



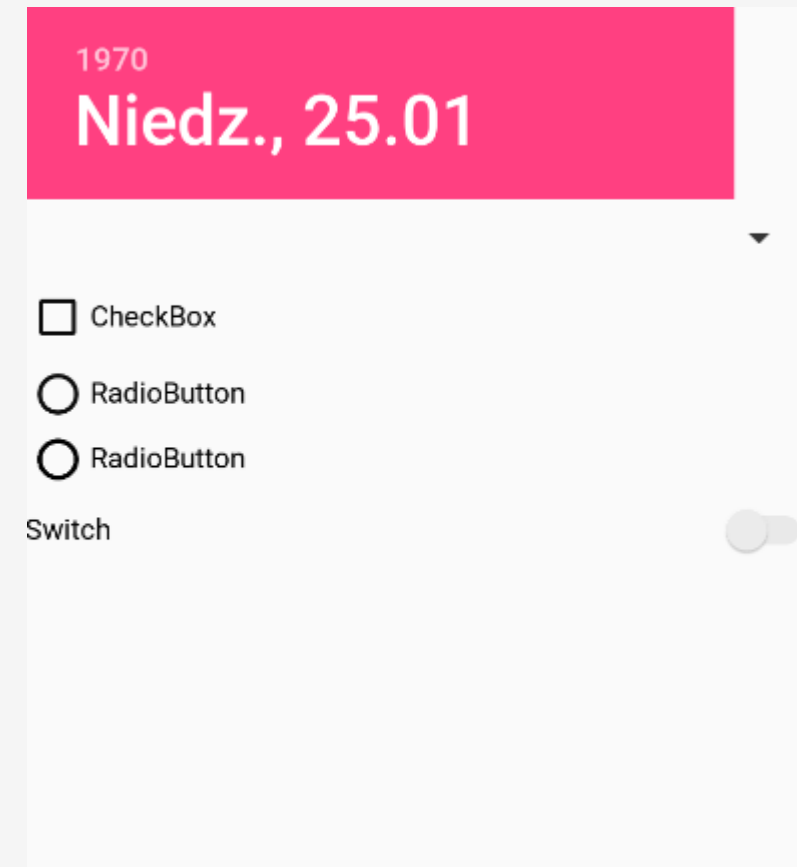
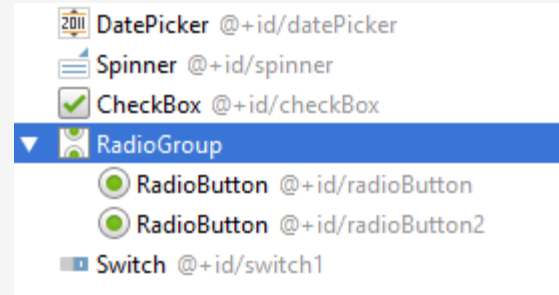
- TextView
- EditText
- Button
- ImageButton
- FloatingActionButton
- TimePicker
- MapView





Przykładowe View

- DatePicker
- Spinner (ComboBox)
- CheckBox
- RadioButton
- Switch
- Itd...



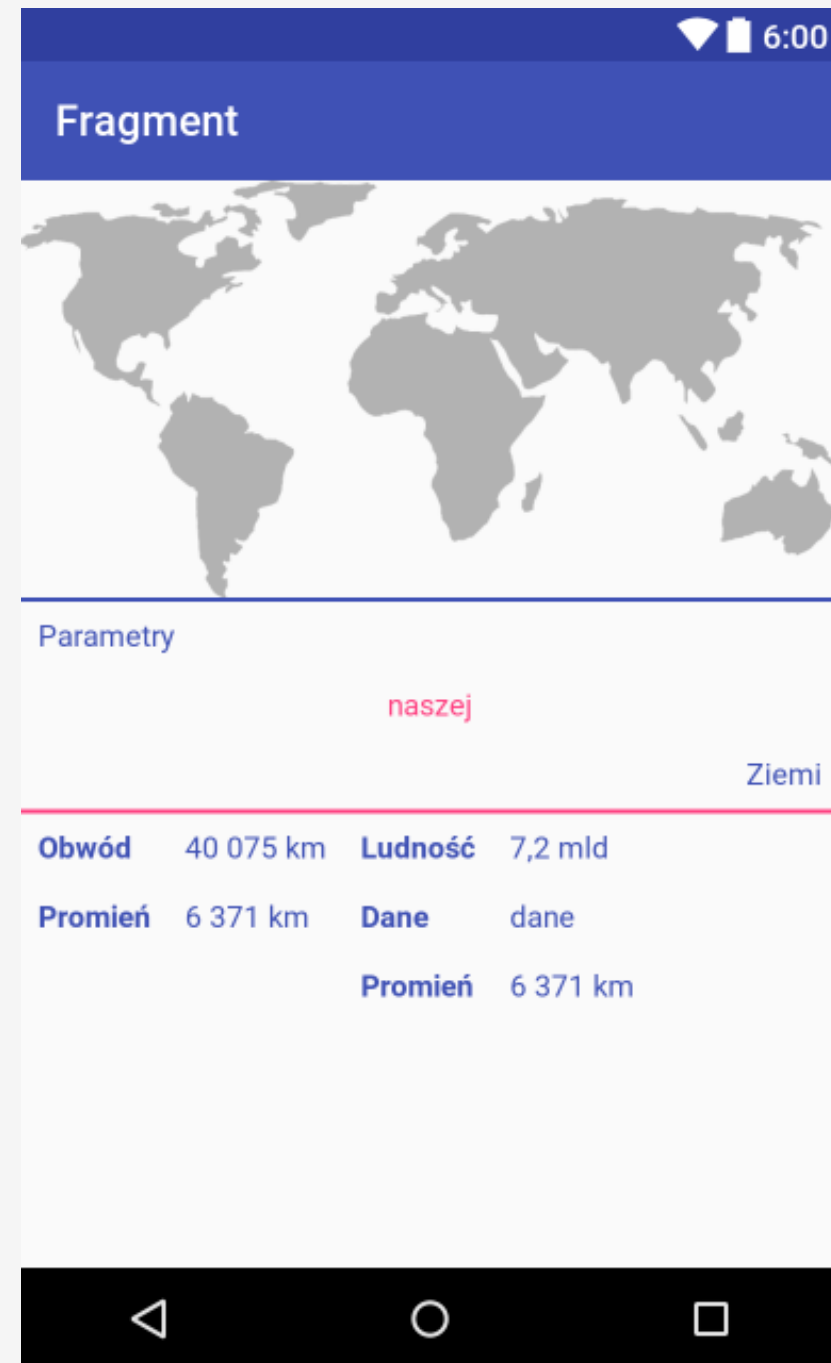


- Podstawowe elementy różnią się wyglądem na różnych wersjach systemu
- ...oraz przez różnych producentów!
- Można pisać swoje własne CustomView dziedzicząc po którymkolwiek elemencie widoku
- Czasem wychodzą nowe ciekawe widoki
- Niektóre frameworki posiadają też gotowe widoki (np. MPAndroidCharts)



Zadanie 1

Stwórz widok według
konceptu





Zadanie 2



Rejestracja użytkownika

Stwórz formularz z opcjami:

- Imię (EditText)
- Wiek (EditText)
- Płeć (RadioButton w RadioGroup)
- Data urodzenia (DatePicker)
- Przycisk „Zarejestruj”

Dodatkowo:

- Adres Email (EditText)
- Walidacja na pola
- Użyj ButterKnife



- Ładowanie grafiki

<https://github.com/bumptech/glide>

<https://github.com/facebook/fresco>

<https://github.com/square/picasso>

- Lokalne bazy danych

<https://github.com/pardom/ActiveAndroid>

<http://greendao-orm.com/>

- Usługi internetowe

<http://square.github.io/retrofit/>

<https://github.com/google/volley>



Przydatne frameworki 2

- Programowanie reaktywne RxJava – na przyszłość 😊

<https://github.com/ReactiveX/RxJava>

- Wykresy

<https://github.com/PhilJay/MPAndroidChart>

- Animacje

<https://github.com/andkulikov/Transitions-Everywhere>