Android – elementy interfejsu

Anna Gogolińska

Kluczowe Pojęcia

- **Widżety (Widgets):** Interaktywne elementy, takie jak przyciski, pola tekstowe.
- **Layouty (Layouts):** Układy służące do organizowania widżetów (np. LinearLayout).
- **XML (eXtensible Markup Language):** Język używany do deklaratywnego opisu struktury interfejsu (pliki w /res/layout).
- **Atrybuty:** Właściwości konfiguracyjne elementów (np. android:layout_width, android:text).

LinearLayout

- Najprostszy menedżer układu, organizujący elementy albo **horyzontalnie** (w rzędzie), albo **wertykalnie** (w kolumnie).
  - Kluczowy Atrybut: **android:orientation** (możliwe wartości: vertical lub horizontal).
- Ważne Atrybuty dla ustawianie wielkości:
  - **android:layout_width**: szerokość elementu/układu.
  - **android:layout_height**: wysokość elementu/układu.
  - Element wypełnia całą szerokość okna/kontenera:
    - Ustawienie atrybutu **android:layout_width** na **match_parent**.
  - Elementy dopasowuje się do zawartości:
    - Ustawienie atrybutu **android:layout_width** na **wrap_content.**
  - **Domyślne opcje ustawiane automatycznie!**

Przykład

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

    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Przycisk Szeroki" />

    <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Przycisk Mały" />

</LinearLayout>
```

Porada praktyczna

- Domyślny układ to ConstraintLayout. Jak zmienić go na LinearLayout?
  - Kliknąć PPM na widok aplikacji albo blueprint.
  - Wybrać CoverView.
  - Z listy wybrać LinearLayout.
  - Potem, gdy znowu kliknie się na widok lub blueprint PPM u góry listy będzie pozycja: w niej można zmienić vertical na horizontal i na odwrót.

Zagnieżdżanie

- Jeden Układ w Drugim (Zagnieżdżanie):
  - Możesz umieszczać jeden layout (np. <LinearLayout>) wewnątrz drugiego, aby tworzyć bardziej złożone kompozycje.
- Np. pionowy układ zawiera dwa zagnieżdżone układy poziome.

Przykład

```
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical">
   <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Nagłówek sekcji" />
   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal">
      <Button
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="OK" />
      <Button
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Anuluj" />
   </LinearLayout>
</LinearLayout>
```

Obrazki

- Komponent: **<ImageView>**
- Wyświetlanie Obrazka:
  - Atrybut: **android:src**
  - Wartość: odniesienie do pliku graficznego umieszczonego w katalogu /res/drawable, np. @drawable/moja_ikona.
- Skalowanie Obrazka:
  - Atrybut: **android:scaleType** (np. centerCrop, fitCenter).

Przykład

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical">

  <ImageView
      android:id="@+id/iv_logo"
      android:layout_width="200dp"
      android:layout_height="200dp"
      android:src="@drawable/logo_firmy"
      android:contentDescription="Logo firmy"
      android:scaleType="fitCenter" />

</LinearLayout>
```

Podstawowe Widżety

- Przycisk: **<Button>**
  - Używany do wyzwalania akcji.
  - **android:text** - etykieta przycisku.
- Pole Tekstowe (do wyświetlania): **<TextView>**
  - Wyświetla tekst dla użytkownika.
- Pole Edycyjne (do wprowadzania): **<EditText>**
  - Umożliwia użytkownikowi wprowadzanie danych.
  - Pole Edycyjne z Podpowiedzią: Użyj atrybutu **android:hint** - tekstu wyświetlanego, gdy pole jest puste.

Przykład

```
<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Wprowadź dane:" />
   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Wpisz tutaj swoje imię" />
    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center_horizontal"
      android:text="Wyślij" />
</LinearLayout>
```

Specjalistyczne Pola Edycyjne

- Atrybut **android:inputType** jest używany w elemencie <EditText> w pliku XML, aby określić, jakiego typu danych użytkownik ma wprowadzić (np. inne klawiatura).
  - Możliwe wartości dla **android:inputType:**
    - **text** (standardowy tekst), **number** (liczby całkowite) , **numberDecimal** (cyfry z możliwością użycia separatora dziesiętnego), **textEmailAddress** (adres e-mail), **textPassword** (wprowadzane znaki są ukrywane), **numberPassword** (cyfry, ale wprowadzane znaki są ukrywane), **textUri** (adres URL/URI), **datetime** lub **date** (data lub data i czasu), **time** (czas), …

Przykład

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical">
   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Wpisz swój email"
      android:inputType="textEmailAddress" />
   <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Wprowadź hasło"
      android:inputType="textPassword" />
     <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Wprowadź cenę (np. 19.99)"
      android:inputType="numberDecimal" />
</LinearLayout>
```

Suwak i Separator

- Suwak (SeekBar/Slider): **<SeekBar>**
  - Umożliwia wybór wartości z określonego zakresu.
  - Ustawianie Wartości Maksymalnej: Atrybut **android:max** (np. android:max="100").
  - Wartość Początkowa: Atrybut **android:progress**.
- Linia Horyzontalna (Separator Wzrokowy):
  - Można ją utworzyć za pomocą prostego <View> z odpowiednią wysokością i kolorem tła.
  - O tym później.

Przykład

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical">

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Ustaw Głośność (0-100):"/>

   <SeekBar
      android:id="@+id/seekBarVolume"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:max="100"
      android:progress="50"/>
</LinearLayout>
```

CheckBox i RadioButton

- Checkbox: wybranie jednej lub więcej opcji:
  - **android:text**: tekst wyświetlany obok pola wyboru.
  - **android:checked**: stan początkowy (true/false).
- RadioButton: wybranie jednej opcji z RadioGroup.
  - **android:text**: tekst wyświetlany obok przycisku.
  - **android:checked**: stan początkowy (true/false).
  - **android:orientation**: orientacja elementów w RadioGroup (vertical/horizontal).

Przykład

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical">
   <CheckBox
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Jabłko"
      android:checked="true" />
   <CheckBox
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Gruszka"/>
 <RadioGroup
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal">
      <RadioButton
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Mężczyzna"
         android:checked="true"/>
      <RadioButton
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Kobieta" />
   </RadioGroup>
</LinearLayout>
```

ToggleButton

- ToggleButton - dwustanowy przełącznik (ON/OFF):
  - Ma dwa stany: zaznaczony (ON) lub odznaczony (OFF).
  - Po naciśnięciu zmienia swój stan i wyświetla odpowiedni tekst.
    - **android:textOn**: tekst wyświetlany, gdy przełącznik jest włączony (ON).
    - **android:textOff**: tekst wyświetlany, gdy przełącznik jest wyłączony (OFF).
    - **android:checked**: stan początkowy (true/false)

Przykład

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Tryb samolotowy:"/>
   <ToggleButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textOn="WŁĄCZONY"
      android:textOff="WYŁĄCZONY"
      android:checked="false" />
</LinearLayout>
```

Spinner

- Spinner - rozwijana lista opcji (wyświetla pojedynczy element, po kliknięciu rozwija się).
  - **android:entries**: Odwołanie do tablicy stringów zdefiniowanej w **res/values/strings.xml** (najprostszy sposób na wypełnienie danymi):

```
<resources>
  <string-array name="lista_krajobrazow">
            <item>Góry</item>
            <item>Morze</item>
            <item>Jeziora</item>
            <item>Las</item>
            <item>Miasto</item>
  </string-array>
</resources>
```

Przykład

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Wybierz ulubiony krajobraz:" />

   <Spinner
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:entries="@array/lista_krajobrazow"
      android:spinnerMode="dropdown" />

</LinearLayout>
```

Lista (ListView)

- Komponent: **<ListView>**
  - Służy do wyświetlania długich list elementów.
  - Wysokość dopasowana do Ilości elementów:
    - **android:layout_height="wrap_content"** na ListView
  - **android:divider**:
    - Określa kolor lub zasób używany jako separator między wierszami listy. Wartość np.: #C0C0C0 lub @drawable/separator_line
    - **android:dividerHeight** - grubość/wysokość linii separatora
  - android:scrollbarStyle:
    - Definiuje, gdzie i jak pasek przewijania ma być rysowany.
    - Możliwe wartości: insideOverlay, insideInset, outsideOverlay, outsideInset (czy wewnątrz czy na zewnątrz paddingu, czy nad zawartością czy obok).

Przykład

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical">

   <ListView
      android:id="@+id/simpleListView"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:divider="#C0C0C0"
      android:dividerHeight="1dp"
      android:scrollbarStyle="outsideOverlay"  />

</LinearLayout>
```