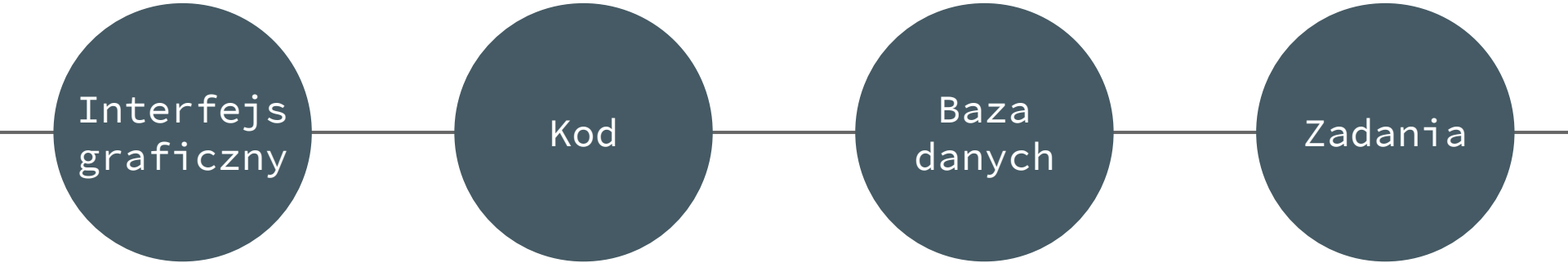
Projektowanie aplikacji mobilnych

Marek Kwak  **Piotr Węgrzyn**

Plan prezentacji



Interfejs graficzny - XML

Extensible Markup Language – XML

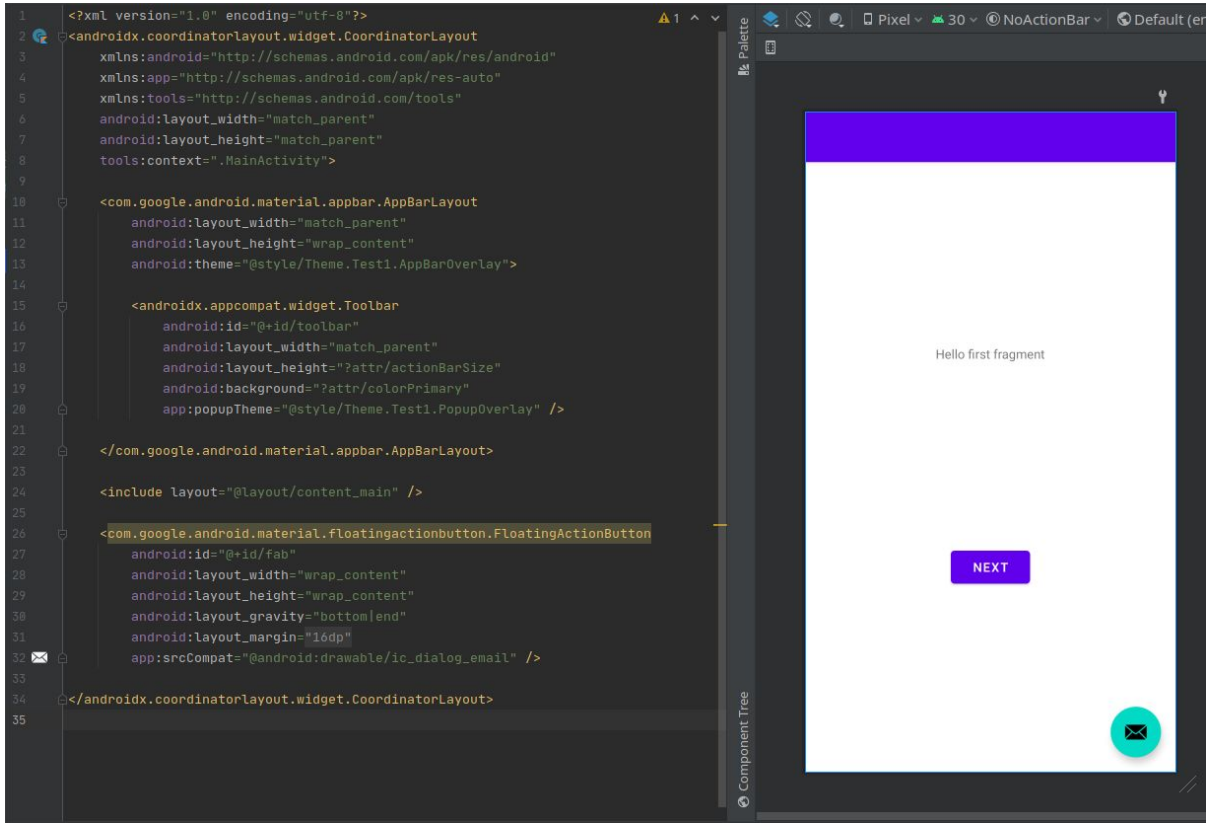
XML to uniwersalny język znaczników (podobny do HTML) szerokiego zastosowania. Służy do uporządkowanego zapisu danych.

W projektowaniu **interfejsu** dla **aplikacji** mobilnych na system Android XML spełnia funkcję opisu wyglądu aplikacji. Za jego pomocą możemy łatwo dodawać i formatować położenie elementów na ekranie.

Interfejs w XML

Elementy w interfejsie XML znajdują się w “**Layout**’ach” (które też są elementami), czyli agregatach wyglądu – określają one tryb rozkładu elementów na ekranie.

Elementy posiadają **atrybuty**, takie jak wysokość, szerokość, marginesy, paddingi, kolory, zawartości graficzne i tekstowe etc.




```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    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=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/Theme.Test1.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/Theme.Test1.PopupOverlay" />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_main" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        app:srcCompat="@android:drawable/ic_dialog_email" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

XML – przykładowe parametry elementów interfejsu

android:layout_width

android:layout_height

android:gravity

android:text

android:textSize

android:textStyle

android:background

android:foreground

android:padding

android:margin

android:orientation

i wiele innych

```xml
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    android:textStyle="bold"
    android:padding="5dp"
    android:gravity="center"
    android:text="TEXT" />
```

```css
textview {
    display: block;
    height: fit-content;
    font-size: 1.2em;
    font-weight: bold;
    padding: 5px;
    align-content: center;
}
```

Cechy elementu w XML

Cechy elementu w CSS

Kod – język Kotlin

Język programowania Kotlin

Kotlin jest multiplatformowym językiem programowania wysokiego poziomu. Jest w pełni kompilowalny do Javy, ale ma przyjemniejszą składnię - ponadto można stosować te dwa języki równocześnie.

Ze względu na swoją strukturę, Kotlin wymusza obsługę ewentualnych wyjątków NullPointerException oraz ułatwia zachowanie porządku w kodzie.

Więcej informacji

```kotlin
class MapsActivity : FragmentActivity(), OnMapReadyCallback {
    private val mMap: GoogleMap? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_maps)
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        val mapFragment = supportFragmentManager
                .findFragmentById(R.id.map) as SupportMapFragment?
        mapFragment!!.getMapAsync( callback: this)
    }

    override fun onMapReady(googleMap: GoogleMap) {
        val dane = Data.getInstance()
        val polylineOptions = PolylineOptions()
        for (location in dane.points) {
            val point = LatLng(location.latitude, location.longitude)
            polylineOptions.add(point)
        }
        polylineOptions.color(Color.rgb( red: 0,  green: 120,  blue: 200))
        googleMap.addPolyline(polylineOptions)
```

Kod napisany w Kotlinie

```java
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync( callback: this);
    }

    @Override
    public void onMapReady(@NotNull GoogleMap googleMap) {

        Data dane = Data.getInstance();
        PolylineOptions polylineOptions = new PolylineOptions();
        for (Location location : dane.getPoints()) {
            LatLng point = new LatLng(location.getLatitude(), location.getLongitude());
            polylineOptions.add(point);
        }
        polylineOptions.color(Color.rgb( red: 0, green: 120, blue: 200));
        googleMap.addPolyline(polylineOptions);
```

Kod napisany w Javie

O co chodzi z tym pytajnikiem ?

```
var b: String? = "abc" // can be set null
b = null // ok
print(b)
var a: String = "abc" // Regular initialization means non-null by default
a = null // compilation error
```

```
if ( null != nullableVariable) {
 nullableVariable.someMethodCall()
} else {
 // fallback flow
}
```

```
nullableVariable?.someMethodCall()
```

Pętle

```
for (i in 1..3) {
    println(i)
}
```

```
for (i in array.indices) {
    println(array[i])
}
```

```
val max = if (a > b) a else b
```

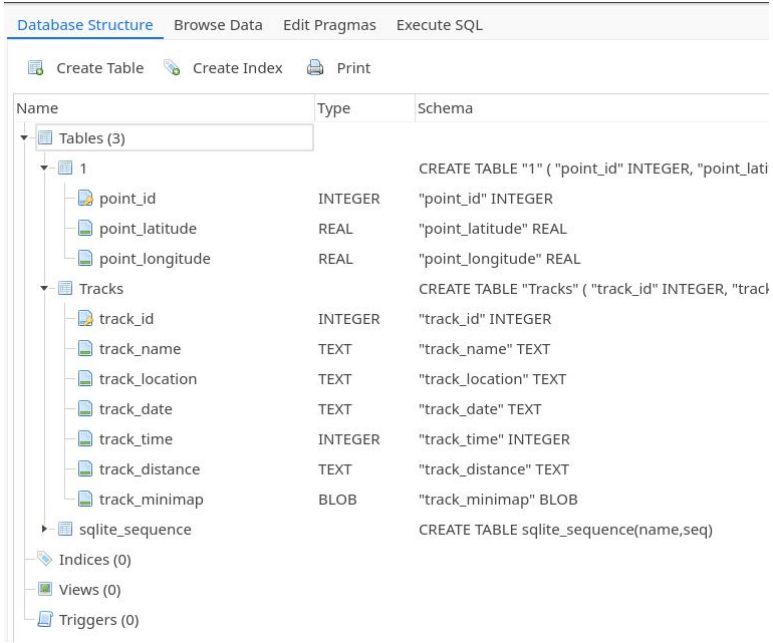
Baza danych – SQLite

SQLite – baza relacyjna

SQLite to wariacja na temat **PostgreSQL**'a. Ma w wielu środowiskach swoje implementacje i zastosowanie ze względu na szybkość działania i niewielki narzut.

W bibliotekach Androida jest implementowany przez dwie biblioteki - **android.database.sqlite** oraz androidx.room.

Plik z bazą danych znajduje się w pamięci wewnętrznej aplikacji, a dzięki użyciu **Singletonu** można otrzymać do niej dostęp z dowolnego miejsca w aplikacji.




```
20  val databaseHelper = dane.databaseHelper
21
22  val cursor: Cursor = databaseHelper.getCursorToTracksTable()
23
24  if (cursor.count != 0) {
25      while (cursor.moveToNext()) {
26          val temp = MainItem(
27              cursor.getInt(0),
28              cursor.getString(1),
29              cursor.getString(2),
30              cursor.getString(3),
31              cursor.getInt(4),
32              cursor.getString(5)
33          )
34          itemsList.add(temp)
35      }
36  }
```

Struktura bazy SQLite

Dane umieszczane są w tabelach. Każdy rekord to unikalny zestaw danych.

Wykorzystanie w kodzie

Odpowiedź na zapytanie otrzymujemy za pomocą klasy Cursor, z której następnie wyciągamy potrzebne nam dane.

Zadania 

Streszczenie zadań – całość dostępna na GitHubie

Zadanie 1

Proste projektowanie elementów interfejsu graficznego w XML

Zadanie 2

Podstawy Kotlina na przykładzie implementacji algorytmu

Zadanie 3

Praktyczne wykorzystanie bazy danych w projekcie

Dziękujemy za uwagę 📌

Powodzenia z zadaniami!