Android Studio - Java: Pasek tytułu

© Copyright by 3bird Projects 2025, http://edukacja.3bird.pl

Ogólne

W wersji podstawowej, aplikacja prezentuje **dedykowany** pasek tytułu wraz z logo. Wersja rozbudowana dodatkowo zawiera rozwijane menu i odtwarzanie dźwięku.

W starych wersjach Androida za pasek tytułowy odpowiadał składnik „*ActionBar*", miał jednak ograniczone możliwości konfiguracji. Obecnie zastępuje go „*ToolBar*" (należy wyłączyć „*ActionBar*").

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii, twarde odstępy, odmienne cudzysłowy, odmienne apostrofy, odmienne myślniki. To godziny dodatkowej pracy na wykrywanie błędów i ich poprawianie. Kod należy przepisać ze zrozumieniem.

Kod - wersja podstawowa

Plik „activity_main.xml"

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolBar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:layout_gravity="center_horizontal"
        android:background="#ff0000"
        android:elevation="4dp"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

        <!-- Elementy paska umieszczamy wewnątrz LinearLayout w celu łatwego ich rozłożenia w
        jednej linii -->
```

```
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <!-- Logo umieszczamy jako <ImageView> w celu określenia jego wymiarów -->
        <ImageView
            android:id="@+id/logo"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:layout_gravity="center"
            android:layout_marginEnd="8dp"
            android:contentDescription="Biedronka"
            android:src="@drawable/ladybug" /> <!-- Odstęp między logo a tytułem -->

        <!-- Niestandardowy TextView do centrowania tytułu -->
        <TextView
            android:id="@+id/tytul"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ellipsize="end"
            android:gravity="start"
            android:singleLine="true"
            android:text="Biedronka"
            android:textColor="#FFFF00"
            android:textSize="30sp"
            android:textStyle="bold" />
    </LinearLayout>
</androidx.appcompat.widget.Toolbar>
```

```
    <TextView
        android:id="@+id/powitanie"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="100dp"
        android:lineSpacingExtra="10dp"
        android:text="Witamy w naszym sklepie!"
        android:textColor="#ffffff"
        android:textAlignment="center" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Plik „ActivityMain.java”

```
package com.example.nazwaProjektu;
import android.os.Bundle;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
```

```
        // Aby tytuł był w ogóle widoczny (traktuj Toolbar jako ActionBar):
        setSupportActionBar(findViewById(R.id.toolBar));
        if (getSupportActionBar() != null) {
            getSupportActionBar().setDisplayShowTitleEnabled(false);
        }
    }
}
```

Kod - wersja rozbudowana

Plik „themes.xml” *(res/values-night/themes.xml)*

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <style name="Base.Theme.JavaPasekTytulu"
    parent="Theme.Material3.DayNight.NoActionBar">
        <!-- Customize your dark theme here. -->
    </style>
</resources>
```

Plik „activity_main.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <!-- Jeśli chcemy, aby logo było automatycznie skalowane do wysokości paska, wpisujemy tutaj:
    -->
    <!-- app:logo="@drawable/ladybug" -->
    <!-- Można tutaj także umieścić tytuł aplikacji (ale z ograniczoną możliwością formatowania): -->
    <!-- app:title="@string/app_name" -->
    <!-- app:titleTextColor="#000000" -->
    <!-- app:titleMargins="0dp" -->
    <!-- android:actionMenuTextColor="#000000" -->
    <!-- W przypadku zastosowania LinearLayout może być konieczne wyłączenie domyślnego
tytułu: -->
```

```
<!-- app:title=" " (zwróć uwagę na spację wewnątrz cudzysłowu) -->

<androidx.appcompat.widget.Toolbar
  android:id="@+id/toolBar"
  android:layout_width="match_parent"
  android:layout_height="?attr/actionBarSize"
  android:layout_gravity="center_horizontal"
  android:background="#ff0000"
  android:elevation="4dp"
  android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

  <!-- Elementy paska umieszczamy wewnątrz LinearLayout w celu łatwego ich rozłożenia w
  jednej linii -->
  <LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="horizontal">

    <!-- Logo umieszczamy jako <ImageView> w celu określenia jego wymiarów -->
    <ImageView
      android:id="@+id/logo"
      android:layout_width="40dp"
      android:layout_height="40dp"
      android:layout_gravity="center"
      android:layout_marginEnd="8dp"
      android:contentDescription="Biedronka"
      android:src="@drawable/ladybug" /> <!-- Odstęp między logo a tytułem -->

    <!-- Niestandardowy TextView do centrowania tytułu -->
    <TextView
      android:id="@+id/tytul"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
```

```
            android:ellipsize="end"
            android:gravity="start"
            android:singleLine="true"
            android:text="Biedronka"
            android:textColor="#FFFF00"
            android:textSize="30sp"
            android:textStyle="bold" />
    </LinearLayout>
</androidx.appcompat.widget.Toolbar>

<TextView
    android:id="@+id/powitanie"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="100dp"
    android:lineSpacingExtra="10dp"
    android:text="Witamy w alejkach zawalonych paletami!\nZa każde przeciśnięcie się między
    nimi\nzdobywasz kupon rabatowy w wysokości 5zł."
    android:textColor="#ffffff"
    android:textAlignment="center" />

<ImageView
    android:id="@+id/palety"
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:layout_gravity="center"
    android:src="@drawable/palety"
    android:contentDescription="Alejka w Biedronce"
    android:clickable="true"
    android:focusable="true" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Plik „menu_toolbar.xml”

Uwaga: Należy utworzyć folder „**res** / **menu**”, a następnie w tym folderze tworzymy plik „**menu_toolbar.xml**”.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto">
  <!-- Poniżej własna ikona o nazwie "ic_menu_icon" dodana przez VectorAsset:
  <item
    android:id="@+id/action_settings"
    android:title="Settings"
    android:icon="@drawable/ic_menu_icon"
    app:showAsAction="ifRoom" />
  <item
    android:id="@+id/action_about"
    android:title="About"
    android:icon="@android:drawable/ic_menu_info_details"
    app:showAsAction="ifRoom" />
  Opcja "showAsAction=ifRoom" oznacza, że element znajdzie się bezpośrednio
  na Toolbar (jeśli będzie miejsce), zamiast w rozwijanym menu -->

  <item
    android:id="@+id/kupony"
    android:title="Kupony…"
    android:icon="@android:drawable/ic_menu_view"
    app:showAsAction="never"/>
  <item
    android:id="@+id/sklepy"
    android:title="Sklepy..."
    android:icon="@android:drawable/ic_menu_mylocation"
    app:showAsAction="never" />
  <item
    android:id="@+id/about"
    android:title="About..."
    android:icon="@android:drawable/ic_menu_info_details"
    app:showAsAction="never" />
```

</menu>

Plik „MainActivity.java”

```
package com.example.nazwaProjektu;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ImageView;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import java.lang.reflect.Method;

public class MainActivity extends AppCompatActivity {
  private MediaPlayer dzwiekFanfary;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
      Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
      v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
      return insets;
    });
    // Inicjalizacja ImageView
    ImageView obrazAlejki = findViewById(R.id.palety);
```

```
    // Inicjalizacja MediaPlayer z plikiem dźwiękowym
    dzwiekFanfary = MediaPlayer.create(this, R.raw.biedronka_fanfary);

    // Aby tytuł był w ogóle widoczny (traktuj Toolbar jako ActionBar):
    setSupportActionBar(findViewById(R.id.toolBar));
    // Nadpisuje domyślny tytuł aplikacji zawarty w strings.xml
    // setTitle("Napis");
    // getSupportActionBar().setDisplayShowHomeEnabled(true);
    // getSupportActionBar().setLogo(R.drawable.ladybug); // Grafika jest automatycznie
    // skalowana
    // getSupportActionBar().setDisplayUseLogoEnabled(true);
    // Ukryj domyślny tytuł Toolbara (pobierany ze "strings.xml" lub z "AndroidManifest.xml"):
    if (getSupportActionBar() != null) {
        getSupportActionBar().setDisplayShowTitleEnabled(false);
    }

    // Ustawienie obsługi kliknięcia w zdjęcie alejki (lambda):
    obrazAlejki.setOnClickListener(widoczneZdjecie -> {
        if (dzwiekFanfary != null) {
            // Sprawdź, czy dźwięk nie jest już odtwarzany:
            if (!dzwiekFanfary.isPlaying()) {
                dzwiekFanfary.start();  // Rozpocznij odtwarzanie
            }
        }
    });
}

// Dodanie menu do Toolbara podczas startu aplikacji (predefiniowana metoda):
@Override
public boolean onCreateOptionsMenu(Menu rozwijaneMenu) {
    getMenuInflater().inflate(R.menu.menu_toolbar, rozwijaneMenu);
    // Wymuszenie wyświetlania ikon w menu za pomocą refleksji (Android od wersji Lollipop 5.0
    // domyślnie ukrywa ikony w menu rozwijanym w Toolbar).
    // Metoda "setOptionalIconsVisible" z klasy "Menu" nie jest publicznie dostępna, ale dzięki
```

```
    // refleksji można ją wywołać.
    try {
        Method aktywneIkony =
        rozwijaneMenu.getClass().getDeclaredMethod("setOptionalIconsVisible", boolean.class);
        aktywneIkony.setAccessible(true);
        aktywneIkony.invoke(rozwijaneMenu, true);
    } catch (Exception blad) {
        Log.e("Aktywacja ikon menu", "Wystąpił problem: ", blad);
    }
    return true;
}
// REFLEKSJA - można za jej pomocą pobierać informacje o strukturze danej klasy.
// - Class - reprezentuje klasy i interfejsy;
// - Method - reprezentuje metody;
// - Field - reprezentuje pola klasy;
// - Constructor - reprtezentuje konstruktory;

// Obsługa kliknięć elementów menu rozwijanego (predefiniowana metoda).
// Instrukcja "switch" obsługuje tylko stałe (final), a zasoby R.id nie są stałe, więc
// poniżej można użyć tylko instrukcji if-else.
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    int itemId = item.getItemId();
    if (itemId == R.id.kupony) {
        Toast.makeText(this, "Tu będą Twoje kupony...", Toast.LENGTH_SHORT).show();
        return true;
    } else if (itemId == R.id.sklepy) {
        Toast.makeText(this, "Tu będzie wykaz sklepów...", Toast.LENGTH_SHORT).show();
        return true;
    } else if (itemId == R.id.about) {
        // Poniżej wprowadź swoje Imię i Nazwisko
        Toast.makeText(this, "(c) Copyright by TwojeImie Nazwisko",
        Toast.LENGTH_SHORT).show();
        return true;
```

```
        } else {
            return super.onOptionsItemSelected(item);
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        // Zwolnienie zasobów MediaPlayer
        if (dzwiekFanfary != null) {
            dzwiekFanfary.release();
            dzwiekFanfary = null;
        }
    }
}
```

Info:

Na **ocenę celującą** należy dodatkowo:

- wykonać cały projekt bez pomocy nauczyciela;
- przypisać do aplikacji swoją własną ikonę;
- napisy w menu zamienić na kolor czerwony;
- zaprezentować działanie aplikacji w ten sposób, aby było słychać dźwięki (na smartfonie lub na emulatorze).

Ostatnia aktualizacja: 23 stycznia 2025.