



Wstęp do Material Design (III)

Marek Wojtuszkiewicz



TabLayout

Odpowiada za wyświetlanie zakładek

Zakładki można definiować w layoucie

Tytuł zakładki: atrybut text

Ikona zakładki: atrybut icon

```
<android.support.design.widget.TabLayout
    android:id="@+id/tab_layout"
    android:layout_height="wrap_content"
    android:layout_width="match_parent">

    <android.support.design.widget.TabItem
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        android:text="@string/action_contacts"/>

    <android.support.design.widget.TabItem
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        android:text="@string/action_gallery"/>

    <android.support.design.widget.TabItem
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        android:text="@string/action_slideshow"/>

</android.support.design.widget.TabLayout>
```



TabLayout

Odpowiada za wyświetlanie zakładek

Może współpracować z niezależnym widokiem ViewPager

```
<android.support.design.widget.TabLayout  
    android:id="@+id/tab_layout"  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"/>
```

```
viewPager = (ViewPager) findViewById(R.id.content_main);  
tabLayout = (TabLayout) findViewById(R.id.tab_layout);  
tabLayout.setupWithViewPager(viewPager);
```



TabLayout

Odpowiada za wyświetlanie zakładek

Może być zawarty w widoku ViewPager

```
<android.support.v4.view.ViewPager
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.TabLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="top" />

</android.support.v4.view.ViewPager>
```



DrawerLayout

Pełni rolę nawigacji w aplikacji

Najbardziej zewnętrzny element w layoucie

```
<android.support.v4.widget.DrawerLayout

    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <FrameLayout
        android:id="@+id/page_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main"
        app:menu="@menu/activity_drawer" />

</android.support.v4.widget.DrawerLayout>
```



ActionBarDrawerToggle

„Spina” DrawerLayout z
ActionBar/Toolbar

```
drawer = (DrawerLayout) findViewById(R.id.drawer_layout);

ActionBarDrawerToggle toggle =
    new ActionBarDrawerToggle(this, drawer, toolbar, R.string.nav_open, R.string.nav_close);
drawer.addDrawerListener(toggle);
toggle.syncState();
```



NavigationView

Korzysta ze statycznych zasobów aplikacji typu @menu

Musi zawierać się w widoku DrawerLayout

```
<android.support.v4.widget.DrawerLayout

    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <FrameLayout
        android:id="@+id/page_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main"
        app:menu="@menu/activity_drawer" />

</android.support.v4.widget.DrawerLayout>
```



NavigationView - menu

Menu można dzielić na sekcje: <group>

checkableBehavior:
none/single/(all)

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_contacts"
            android:icon="@drawable/ic_contacts_black_24dp"
            android:title="@string/action_contacts" />
        <item
            android:id="@+id/nav_gallery"
            android:icon="@drawable/ic_menu_gallery"
            android:title="@string/action_gallery" />
    </group>

    <item android:title="@string/nav_category_other">
        <menu>
            <item
                android:id="@+id/nav_share"
                android:icon="@drawable/ic_menu_share"
                android:title="@string/action_share" />
            <item
                android:id="@+id/nav_settings"
                android:icon="@drawable/ic_settings_black_24dp"
                android:title="@string/action_settings" />
        </menu>
    </item>

</menu>
```




NavigationView – obsługa zdarzeń

Należy zaimplementować interfejs

`NavigationView.OnNavigationItemSelectedListener` oraz użyć metody

`NavigationView#setNavigationItemSelectedListener`

```
@Override
public boolean onNavigationItemSelectedListener(MenuItem item) {
    int id = item.getItemId();

    switch (id) {
        case R.id.nav_share:
            break;
        case R.id.nav_settings:
            break;
    }

    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```



Wstęp do Material Design (III) - zadanie

Rozszerz aplikację utworzoną w poprzednim zadaniu:

- Dodaj nową aktywność która uruchamiana jest w momencie udanego logowania
- Główną zawartością ekranu powinien być ViewPager wyświetlający 3 różne strony (zawartość stron dowolna, ale różna od siebie)
- Każda strona powinna mieć swoją zakładkę
- Poszczególne strony można wyświetlić na dwa sposoby:
 - klikając na odpowiednią zakładkę w nagłówku aktywności
 - Klikając odpowiednią pozycję w widoku NavigationView
- Nagłówek NavigationView powinien zawierać imię/nazwisko obecnie zalogowanego użytkownika, jego adres e-mail oraz zdjęcie