**Нижнее меню в Android**

**Пример работы с BottomNavigationView.**

Нижняя панель навигации BottomNavigationView представляет собой новый компонент пользовательского интерфейса в стиле Material Design. Этот виджет был создан для замены верхнего меню приложения, чтобы пользователям не пришлось тянуться пальцем через все 5.5″ дисплея для нажатия одной кнопки. Это обеспечивает более простую навигацию и может освежить дизайн вашего приложения

**Пример работы с BottomNavigationView в Android**

**Виджет BottomNavigationView** — это наследник FrameLayout с внутренними контейнерами ViewGroup для элементов меню. Это достаточно легковесный и просто в использовании виджет в стилях материального дизайна.

Переходим к работе. Создайте [новый проект в Android Studio](http://javadevblog.com/pishem-hello-world-na-android-rabotaem-v-android-studio.html) или добавьте виджет BottomNavigationView из библиотеки поддержки (Design Support library) в существующий проект.

Сама библиотека поддержки должна быть добавлена в зависимостях файла **gradle.build** на уровне проекта:

Java

|  |
| --- |
| // gradle.build  ...  compile **'com.android.support:design:25.3.1'**  ... |

*Обратите внимание, что версия библиотеки у вас в проект может быть другая.*

Посмотрим на макет **activity\_main.xml**:

Java

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout 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:id="@+id/container"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="vertical"      tools:context="com.javadevblog.bottomnavigationviewapp.MainActivity">        <FrameLayout          android:id="@+id/fl\_content"          android:layout\_width="match\_parent"          android:layout\_height="0dp"          android:layout\_weight="1" />        <android.support.design.widget.BottomNavigationView          android:id="@+id/navigation"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_gravity="bottom"          android:background="@android:color/darker\_gray"          app:menu="@menu/navigation" />    </LinearLayout> |

Из листинга выше видно, что FrameLayout (место для фрагментов) занимает все пространство, вытесняя BottomNavigationView вниз экрана. Это довольно частая практика и позволяет обойтись без ненужных дополнительных вложеностей в макете.

Само меню выглядит так:

Java

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <menu xmlns:android="http://schemas.android.com/apk/res/android">        <item          android:id="@+id/navigation\_home"          android:icon="@drawable/ic\_home\_black\_24dp"          android:title="@string/title\_home" />        <item          android:id="@+id/navigation\_dashboard"          android:icon="@drawable/ic\_dashboard\_black\_24dp"          android:title="@string/title\_dashboard" />        <item          android:id="@+id/navigation\_notifications"          android:icon="@drawable/ic\_notifications\_black\_24dp"          android:title="@string/title\_notifications" />    </menu> |

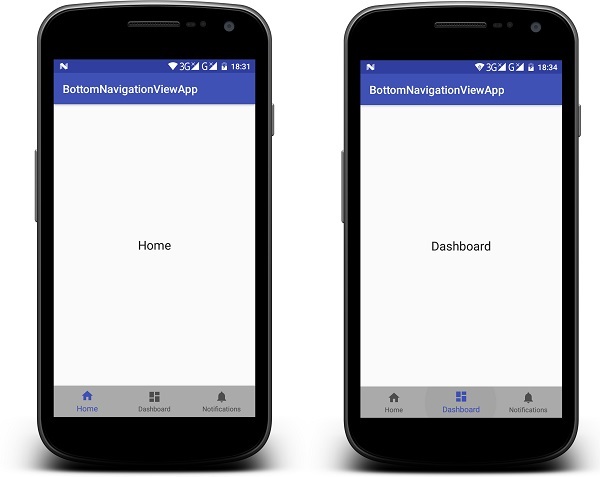
Где android:icon — векторная иконка действия на пункте меню. Эти иконки, а также весь проект вы найдете по ссылке в конце статьи.

Содержимое класса **MainActivity**:

**import** android.os.Bundle;  
**import** android.support.annotation.NonNull;  
**import** android.support.design.widget.BottomNavigationView;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.MenuItem;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** BottomNavigationView.OnNavigationItemSelectedListener **mOnNavigationItemSelectedListener** = **new** BottomNavigationView.OnNavigationItemSelectedListener() {  
  
 @Override  
 **public boolean** onNavigationItemSelected(@NonNull MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** R.id.***navigation\_home***:  
 *// loadFragment(HomeFragment.newInstance());* Toast.*makeText*(getApplicationContext(),  
 **"Вы выбрали HOME"**,  
 Toast.***LENGTH\_SHORT***).show();  
 **return true**;  
 **case** R.id.***navigation\_dashboard***:  
 Toast.*makeText*(getApplicationContext(),  
 **"Вы выбрали DashBoard"**,  
 Toast.***LENGTH\_SHORT***).show();  
  
 **return true**;  
 **case** R.id.***navigation\_notifications***:  
 Toast.*makeText*(getApplicationContext(),  
 **"Вы выбрали Notification"**,  
 Toast.***LENGTH\_SHORT***).show();  
 **return true**;  
 }  
 **return false**;  
 }  
 };  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.***navigation***);  
 navigation.setOnNavigationItemSelectedListener(**mOnNavigationItemSelectedListener**);  
 }  
  
}

Добавили обработчик нажатия на пункты нижнего меню и написали простой метод loadFragment(), который на вход принимает [Fragment](http://javadevblog.com/chto-takoe-android-fragment-zhiznenny-j-tsikl-i-primer-ispol-zovaniya-fragmentov.html" \t "_blank).

Теперь запустим приложение и посмотрим результат:

[](http://javadevblog.com/wp-content/uploads/2017/08/home.jpg)

Как видите, фрагменты переключаются по нажатию на пункты нижнего меню виджета BottomNavigationView. Это довольно простой и удобный способ придать своему приложению новизны и оставаться в тренде разработки под Android.