

Нижняя панель навигации. Знакомство с BottomNavigationView. Как добавить фрагменты в панель Bottom Navigation.

Познакомимся с нижней панелью навигации BottomNavigationView, которая позволяет переходить между экранами – пунктами назначения навигации, а также наглядно информирует пользователя о том, на каком экране он находится. Разберемся, как добавить Bottom Navigation в андроид приложение и как добавить в BottomNavigationView новые фрагменты.

1. Создаем проект
2. Необходимые библиотеки
3. Граф навигации
4. Добавление экранов
5. Хост навигации
6. BottomNavigationView
7. Добавление пунктов меню
8. Добавление идентификаторов в контроллер навигации
9. Рефакторинг

Создаем проект

С одной стороны, создать простое приложение с панелью навигации можно, не написав ни одной строчки кода. Достаточно воспользоваться готовым макетом Bottom Navigation Activity на этапе создания проекта в Android Studio. При этом создается проект приложения с нижней панелью навигации BottomNavigationView на главном экране, в которой отображается три пункта. При нажатии каждого из них меняются экраны приложения. Это все хорошо,

скажете вы, но если нужно добавить или убрать экраны и пункты для них? Или добавить нижнюю навигацию в существующее приложение?

Ок, давайте изучим структуру проекта, созданного по шаблону, и по ходу попробуем добавить сюда пару новых экранов и пунктов для них, а также рассмотрим шаги для добавления нижней панели навигации в уже существующее приложение.

Необходимые библиотеки

Если открыть файл сборки build.gradle, в секции dependencies можно увидеть подключенные библиотеки из пакета androidx.navigation (с ktx для проектов на kotlin) которые нам уже знакомы по предыдущим урокам на тему навигации в приложении. Если вы добавляете нижнюю навигацию в существующее приложение, то начинать нужно с добавления этих библиотек в ваш проект.

```
implementation 'androidx.navigation:navigation-fragment-ktx:2.3.3'  
implementation 'androidx.navigation:navigation-ui-ktx:2.3.3'
```

Граф навигации

В папке res есть папка navigation, в которой размещен все тот же граф навигации, его вы помните по предыдущим урокам. Без него, как вы понимаете, не обходится здесь и нижняя панель навигации.

```
<fragment  
    android:id="@+id/navigation_home"  
    android:name="info.fandroid.bottomnavapplication17.ui.home.HomeFragment"  
    android:label="@string/title_home"  
    tools:layout="@layout/fragment_home" />  
  
<fragment  
    android:id="@+id/navigation_dashboard"  
    android:name="info.fandroid.bottomnavapplication17.ui.dashboard.DashboardFragment"  
    android:label="@string/title_dashboard"  
    tools:layout="@layout/fragment_dashboard" />  
  
<fragment  
    android:id="@+id/navigation_notifications"  
    android:name="info.fandroid.bottomnavapplication17.ui.notifications.NotificationsF  
ragment"  
    android:label="@string/title_notifications"  
    tools:layout="@layout/fragment_notifications" />
```

Как видите, в графе навигации есть три фрагмента, которые являются пунктами назначения, или экранами, на которые мы попадаем, переходя по пунктам панели Bottom Navigation. Но обратите внимание: фрагменты в графе не содержат никаких `<action>` и не связаны стрелками переходов. Это логично, поскольку в данном случае они являются только пунктами назначения, и не ведут из одного в другой. Переход осуществляется из общего компонента над фрагментами, которым является активити-хост. Мы рассмотрим это позже. А сейчас, чтобы добавить новые экраны, прежде всего нужно добавить сюда, в граф навигации, новые пункты назначения.

Добавление экранов

Нажмите кнопку New Destination сверху и выберите Create New Destination. На экране добавления фрагментов выберите шаблон Fragment (with ViewModel). Добавьте один за другим два фрагмента – четвертый и пятый экран. назовем их: FourthFragment и FifthFragment.

Вы можете выбрать при добавлении просто Fragment blank. Но связка Fragment + ViewModel предоставляет преимущества сохранения состояния экрана при изменении конфигурации – например, при повороте устройства.

В граф навигации добавилось два фрагмента. Переименуйте их идентификаторы в соответствии с именами других пунктов назначения: navigation_fourth и navigation_fifth. Переименование всегда лучше делать через рефакторинг, нажатием комбинации Shift + F6 – это гарантирует изменение всех возможных связей.

```
<fragment
    android:id="@+id/navigation_home"
    android:name="info.fandroid.bottomnavapplication17.ui.home.HomeFragment"
    android:label="@string/title_home"
    tools:layout="@layout/fragment_home" />

<fragment
    android:id="@+id/navigation_dashboard"
    android:name="info.fandroid.bottomnavapplication17.ui.dashboard.DashboardFragment"
    android:label="@string/title_dashboard"
    tools:layout="@layout/fragment_dashboard" />

<fragment
    android:id="@+id/navigation_notifications"
    android:name="info.fandroid.bottomnavapplication17.ui.notifications.NotificationsF
ragment"
    android:label="@string/title_notifications"
    tools:layout="@layout/fragment_notifications" />
<fragment
```

```

        android:id="@+id/navigation_fourth"
        android:name="info.fandroid.bottomnavapplication17.ui.fourth.FourthFragment"
        android:label="@string/title_fourth"
        tools:layout="@layout/fourth_fragment" />
<fragment
    android:id="@+id/navigation_fifth"
    android:name="info.fandroid.bottomnavapplication17.ui.fifth.FifthFragment"
    android:label="@string/title_fifth"
    tools:layout="@layout/fifth_fragment" />

```

Можно было оставить идентификаторы как было, но более правильно с точки зрения Code Convention, чтобы названия однотипных элементов были логически похожими. Да и находить их будет легче потом.

Также измените значение параметра `android:label` – оно используется для заголовка экрана. Впишите сюда текст, который вы хотите видеть в заголовке соответствующего экрана, желательно сохранить его затем в строковых ресурсах. Для сохранения нажмите желтую лампочку и выберите пункт «Extract string resource», а затем придумайте название строки, которая сохранится в файле `res/values/strings.xml`.

На этом добавление экранов закончено. Но, если запустить приложение сейчас, они не будут отображаться, и нижняя панель навигации никак не изменится.

Хост навигации

Если мы откроем папку `layout`, то увидим макеты всех фрагментов – пунктов назначения, и макет главного активити.

Макет `activity_main.xml` содержит компонент `fragment` – это хост навигации, в котором отображаются фрагменты – пункты назначения. Он связан с графом навигации через параметр `app:navGraph`.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"

```

```

        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu" />

<fragment
    android:id="@+id/nav_host_fragment"
    android:name="androidx.navigation.fragment.NavHostFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:defaultNavHost="true"
    app:layout_constraintBottom_toTopOf="@id/nav_view"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:navGraph="@navigation/mobile_navigation" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

BottomNavigationView

Также здесь присутствует и компонент BottomNavigationView. Это и есть нижняя панель навигации. Компонент BottomNavigationView является частью библиотеки компонентов материального дизайна. Эта библиотека должна быть импортирована в файле сборки модуля:

```

dependencies {
    // ...
    implementation 'com.google.android.material:material:<version>'
    // ...
}

```

Обычно она уже есть в проекте по умолчанию.

Рассмотрим компонент BottomNavigationView более подробно. Он содержит идентификатор `nav_view`, и имеет значение ширины – `0dp`. Это рекомендуемое значение для всех элементов, ширина которых регулируется ограничениями, заданными корневым элементом разметки `ConstraintLayout`. Мы видим эти ограничения ниже, они привязывают панель к нижней части родителя, который, в свою очередь, занимает весь экран.

Высота BottomNavigationView задана по контенту, но можно установить фиксированную нужного вам размера.

Далее идут два параметра `layout_marginStart` и `layout_marginEnd`. Они регламентируют отступ слева и справа, но поскольку значения здесь равны нулю, они ни на что не влияют, и их можно удалить.

Параметр `android:background` содержит ссылку на атрибут `windowBackground` и делает фон панели такой же, как фон экрана. Если вы хотите получить цвет панели, такой как на гайдлайнах сайта material.io, то замените параметр `android:background` `android:attr/windowBackground` таким:

```
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/nav_view"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    style="@style/Widget.MaterialComponents.BottomNavigationView.Colored"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:menu="@menu/bottom_nav_menu" />
```

Кстати, на сайте material.io можно найти много полезных рекомендаций как по дизайну, так и по реализации компонентов визуального интерфейса. На странице [Bottom Navigation](#), например, указано, что рекомендованное число элементов панели нижней навигации должно быть от трех до пяти. Подписи должны быть максимально короткими и состоять, по возможности, из одного слова.

На вкладке `implementation` дается пример реализации нижней панели без графа навигации, а также примеры кастомизации и оформления фона и цвета пунктов нижней панели, добавления пунктам бейджей и т.д.

Добавление пунктов меню

Но вернемся к нашему проекту и компоненту `BottomNavigationView` в макете `activity_main.xml`.

Нас интересует последний параметр:

```
app:menu="@menu/bottom_nav_menu"
```

Он ссылается на файл `bottom_nav_menu.xml` в папке `res/menu`. Этот файл содержит описание пунктов меню нижней панели навигации. Следовательно, добавлять новые пункты следует здесь.

```

<?xml version="1.0" encoding="utf-8"?>
<navigation 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/mobile_navigation"
    app:startDestination="@+id/navigation_home">

    <fragment
        android:id="@+id/navigation_home"
        android:name="info.fandroid.bottomnavapplication17.ui.home.HomeFragment"
        android:label="@string/title_home"
        tools:layout="@layout/fragment_home" />

    <fragment
        android:id="@+id/navigation_dashboard"
        android:name="info.fandroid.bottomnavapplication17.ui.dashboard.DashboardFragn
ent"
        android:label="@string/title_dashboard"
        tools:layout="@layout/fragment_dashboard" />

    <fragment
        android:id="@+id/navigation_notifications"
        android:name="info.fandroid.bottomnavapplication17.ui.notifications.Notificati
onsFragment"
        android:label="@string/title_notifications"
        tools:layout="@layout/fragment_notifications" />
    <fragment
        android:id="@+id/navigation_fourth"
        android:name="info.fandroid.bottomnavapplication17.ui.fourth.FourthFragment"
        android:label="@string/title_fourth"
        tools:layout="@layout/fourth_fragment" />
    <fragment
        android:id="@+id/navigation_fifth"
        android:name="info.fandroid.bottomnavapplication17.ui.fifth.FifthFragment"
        android:label="@string/title_fifth"
        tools:layout="@layout/fifth_fragment" />
</navigation>

```

Добавим пару элементов `<item>` для четвертого и пятого фрагментов. Их идентификаторы должны совпадать с идентификаторами пунктов назначения в графе навигации. Также укажите соответствующие строковые ресурсы в качестве названия пунктов в `android:title`.

Добавление идентификаторов в контроллер навигации

Осталось добавить идентификаторы пунктов навигации в конфигурацию контроллера навигации в классе `MainActivity`.

```

package info.fandroid.bottomnavapplication17

import android.os.Bundle
import com.google.android.material.bottomnavigation.BottomNavigationView
import androidx.appcompat.app.AppCompatActivity
import androidx.navigation.findNavController
import androidx.navigation.ui.AppBarConfiguration
import androidx.navigation.ui.setupActionBarWithNavController
import androidx.navigation.ui.setupWithNavController

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val navView: BottomNavigationView = findViewById(R.id.nav_view)

        val navController = findNavController(R.id.nav_host_fragment)
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        val appBarConfiguration = AppBarConfiguration(setOf(
            R.id.navigation_home,
            R.id.navigation_dashboard,
            R.id.navigation_notifications,
            R.id.navigation_fourth,
            R.id.navigation_fifth))
        setupActionBarWithNavController(navController, appBarConfiguration)
        navView.setupWithNavController(navController)
    }
}

```

Переменная класса `AppBarConfiguration` содержит набор идентификаторов пунктов навигации и передается вместе с контроллером – хостом навигации – в функцию `setupActionBarWithNavController`. Это нужно для того, чтобы система считала каждый из пунктов назначения пунктом верхнего уровня. В противном случае каждый фрагмент будет отображать стрелку возврата назад в тулбаре слева, как это отображается на дочерних активити и фрагментах.

Попробуйте убрать какой-либо из идентификаторов пунктов назначения из набора, переданного в переменную `appBarConfiguration`, запустите приложение и посмотрите на экран пропущенного пункта назначения, и вы поймете, о чем речь.

Теперь, когда добавление закончено, запустите приложение. Все добавленные пункты меню должны отображаться в нижней панели навигации, и при нажатии вести на соответствующие экраны.

Рефакторинг

Теперь можно навести порядок в проекте. Дело в том, что изначально в проекте фрагменты каждого экрана со своими `ViewModel` расположены в отдельных папках по имени каждого пункта меню нижней панели навигации, и все это лежит в папке `ui`. Новые же добавленные фрагменты попали просто в главный пакет. Нужно создать в папке `ui` новые пакеты по имени добавленных экранов. Затем нужно перенести туда добавленные фрагменты вместе с их привязанными `ViewModel`. Делается это простым перетаскиванием в дереве проекта, с открытием окна рефакторинга, в котором нужно подтвердить операцию.