**Android Navigation. Знакомство с BottomNavigationView. Как добавить фрагменты в панель Bottom Navigation.**

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

**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.

**Неоходимые 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.



|  |  |
| --- | --- |
| 1  2 | implementation 'androidx.navigation:navigation-fragment-ktx:2.3.3'  implementation 'androidx.navigation:navigation-ui-ktx:2.3.3' |

1. добавляем в файл сборки build.gradle

buildFeatures**{** viewBinding=true  
**}**

1. прописываем в Main Activity переменную для обращения к BottomNavigationView через binding чтобы не находить ее по id

**lateinit var binding: ActivityMainBinding**

override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 **binding= ActivityMainBinding.inflate(*layoutInflater*)** setContentView(**binding.*root***)

а затем прописываем нажатия на кнопки

binding.bt.setOnNavigationItemReselectedListener **{** when (**it**.*itemId*) {  
 android.R.id.*home* -> finish()  
 R.id.*navigation\_home* -> {  
 Toast.makeText(this, "Sync", Toast.*LENGTH\_SHORT*).show()  
 }  
  
 R.id.*navigation\_dashboard* -> {  
  
 Toast.makeText(this, "Save", Toast.*LENGTH\_SHORT*).show()  
 }  
  
 }  
 true  
  
**}**

**часть 2**

**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.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <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.NotificationsFragment"      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 вверху и выберите Create New Destination. На экране добавления фрагментов выберите шаблон Fragment (with ViewModel).  Добавьте один за другим два фрагмента – четвертый и пятый экран. Я их так и назову: 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.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | <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.NotificationsFragment"      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 resourse», 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. Что ж, идем дальше.

**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.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | <?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:



|  |  |
| --- | --- |
| 1  2  3  4  5 | dependencies {  // ...  implementation 'com.google.android.material:material:<version>'  // ...  } |

Обычно она уже есть в проекте по умолчанию.

Рассмотрим компонент BottomNavigationView боле 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:



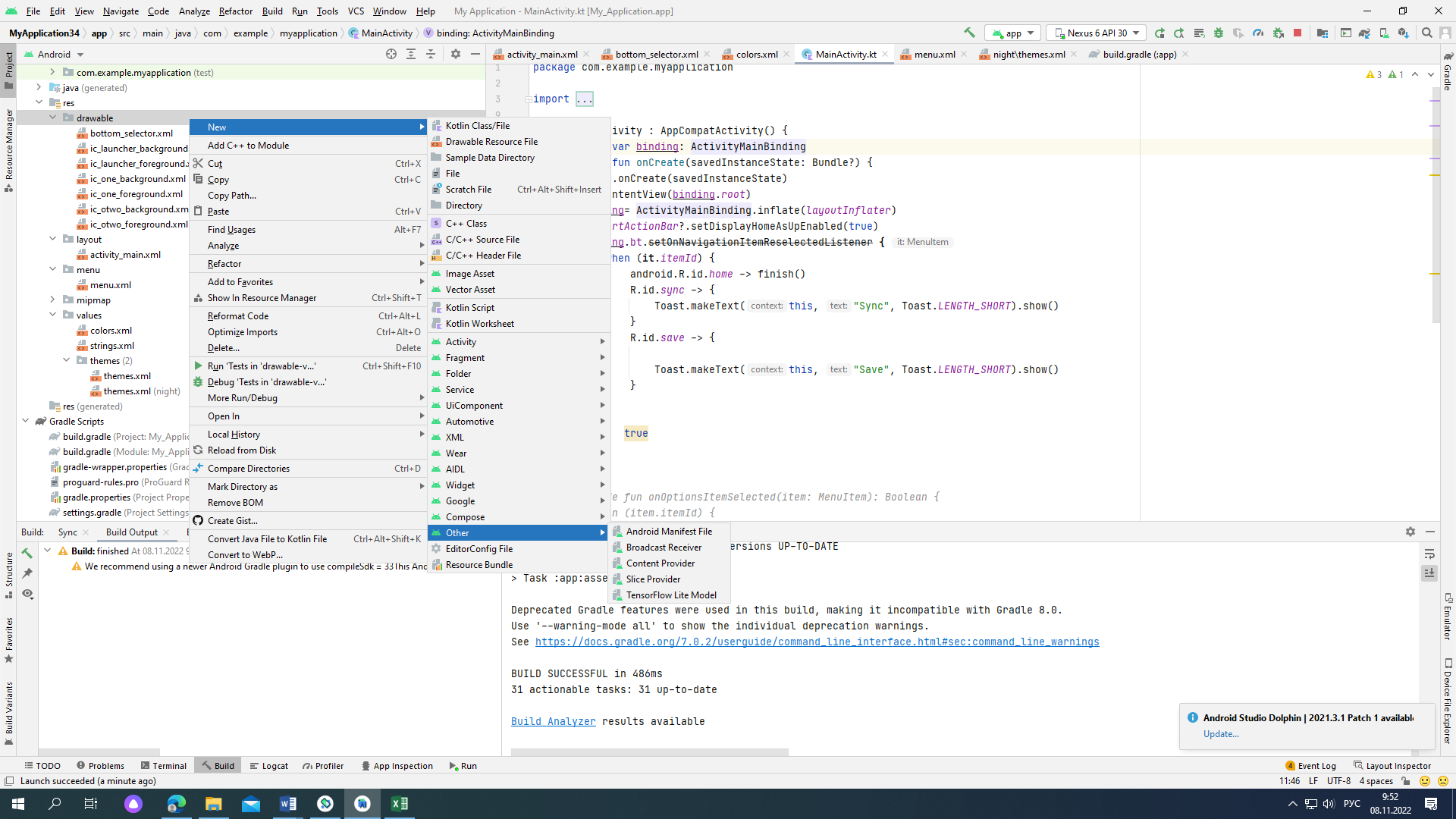
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <com.google.android.material.bottomnavigation.BottomNavigationView      android:id="@+id/nav\_view"      android:layout\_width="0dp"      android:layout\_height="wrap\_content"     <mark> style="@style/Widget.MaterialComponents.BottomNavigationView.Colored"</mark>      app:layout\_constraintBottom\_toBottomOf="parent"      app:layout\_constraintLeft\_toLeftOf="parent"      app:layout\_constraintRight\_toRightOf="parent"      app:menu="@menu/bottom\_nav\_menu" /> |

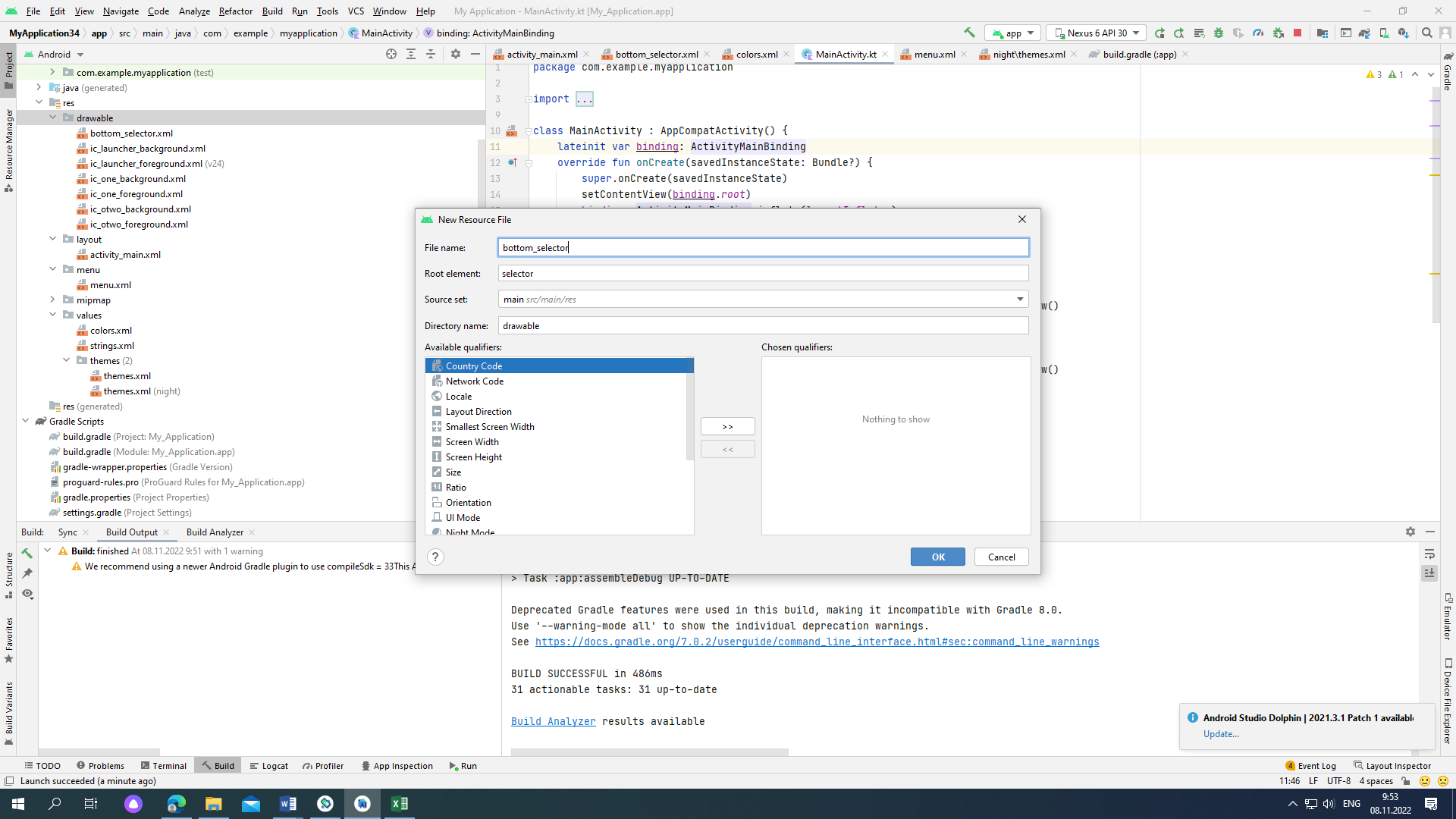
Кстати, на сайте material.io можно найти много полезных рекомендаций как по дизайну, так и по реализации компонентов визуального интерфейса. На странице Bottom Navigation, например, указано, что рекомендованное число элементов панели нижней навигации должно быть от трех до пяти. Подписи должны быть максимально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

На вкладе 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.

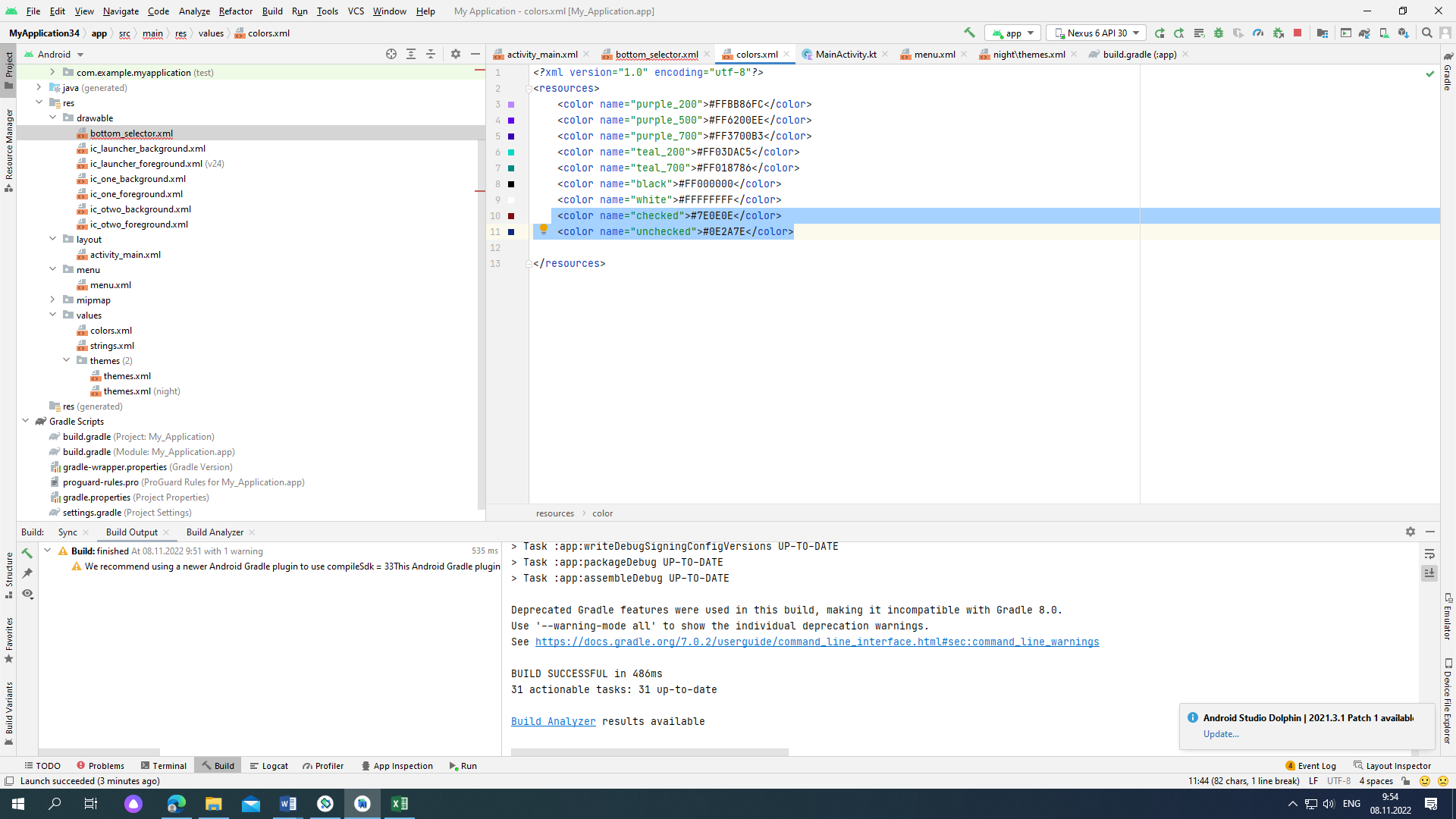
**Добавление пунктов меню**

1. компонент BottomNavigationView, меняем у него backgroundcolor
2. Создаем меню, как в прошлой практической из трех кнопок
3. у компонент BottomNavigationView в свойстве menu делаем ссылку на созданное меню
4. создаем файл для изменения цвета кнопок



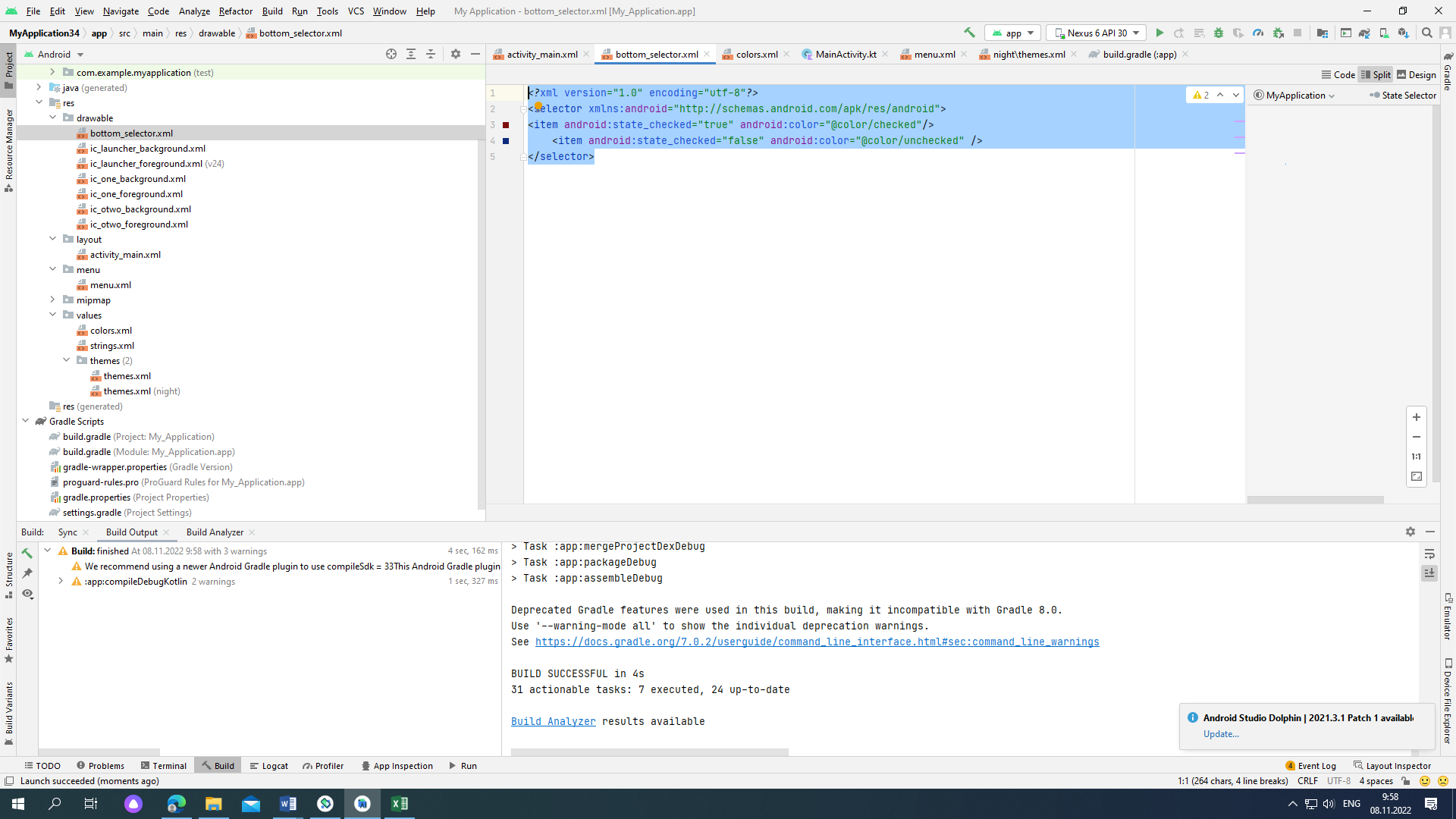


1. допишем цвета в файл color



1. добавляем изменения цвета при нажатии на кнопки в файле bottom\_selector

<item android:state\_checked="true" android:color="@color/checked"/>  
<item android:state\_checked="false" android:color="@color/unchecked" />



Но вернемся к нашему проекту и компоненту BottomNavigationView в макете activity\_main.xml.

Нас интересует последний параметр:



|  |  |
| --- | --- |
| 1 | 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.**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | <?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.DashboardFragment"          android:label="@string/title\_dashboard"          tools:layout="@layout/fragment\_dashboard" />        <fragment          android:id="@+id/navigation\_notifications"          android:name="info.fandroid.bottomnavapplication17.ui.notifications.NotificationsFragment"          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.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | 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(arraysetOf(                  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 АppBarConfiguration 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.