|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №6**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИКБО-20-23 | Комисарик М.А. |
| **Проверил:**  Старший преподаватель кафедры МОСИТ | Шешуков Л.С. |

Москва 2025 г.СОДЕРЖАНИЕ

[**РТУ МИРЭА** 1](#_Toc195335351)

[1 ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ 3](#_Toc195335352)

[1.1 BottomBar 3](#_Toc195335353)

[1.2 ActionBar 11](#_Toc195335354)

[1.3 Navigation Drawer 16](#_Toc195335355)

[2 ПРАКТИЧЕСКОЕ ЗАДАНИЕ 23](#_Toc195335356)

[2.1 Боковая панель навигации 23](#_Toc195335357)

[2.1.1 Разметка 23](#_Toc195335358)

[2.1.2 Код 27](#_Toc195335359)

[2.1.3 Тестирование 29](#_Toc195335360)

[2.2 Нижняя панель навигации 32](#_Toc195335361)

[2.2.1 Разметка 32](#_Toc195335362)

[2.2.2 Код 34](#_Toc195335363)

[2.2.3 Тестирование 35](#_Toc195335364)

[ЗАКЛЮЧЕНИЕ 38](#_Toc195335365)

# ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

## BottomBar

В предыдущих практических работах были рассмотрены переходы между разными экранами в приложении с использованием простых элементов Button. Однако, в контексте современной Android разработки такой метод является не интуитивным и устаревшим. Поэтому, на замену такой навигации были придуманы объекты пользовательского интерфейса BottomBar и Drawer.

BottomBar, или нижняя панель навигации, является ключевым элементом пользовательского интерфейса в мобильных приложениях, работающих под управлением Android. Этот элемент предназначен для улучшения навигации и повышения удобства использования приложения за счёт предоставления быстрого доступа к основным разделам приложения. В контексте перемещения между экранами BottomBar выполняет несколько важных функций:

* улучшение пользовательского опыта: BottomBar делает навигацию по приложению интуитивно понятной и удобной. Располагаясь в нижней части экрана, он легко доступен для пользователя, что особенно важно при использовании устройства одной рукой. Это облегчает переход между ключевыми разделами приложения, не отвлекая пользователя от основного контента;
* повышение производительности: использование BottomBar позволяет сократить количество нажатий, необходимых для перехода между разделами приложения, тем самым ускоряя взаимодействие пользователя с приложением и повышая его общую производительность;
* организация контента: BottomBar помогает организовать контент в приложении, выделяя основные разделы или функции, которые должны быть всегда под рукой. Это упрощает структуру приложения и делает его более понятным для пользователя.

BottomBar представляет собой контейнер, который содержит в себе различные элементы (Рисунок 1).

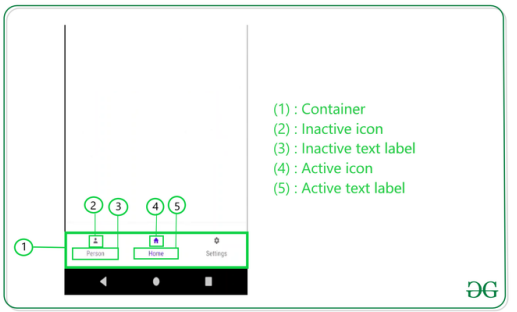


Рисунок – Наполнение BottomBar

Для использования BottomNavigationView необходимо добавить этот элемент в макет активности (Рисунок 2).

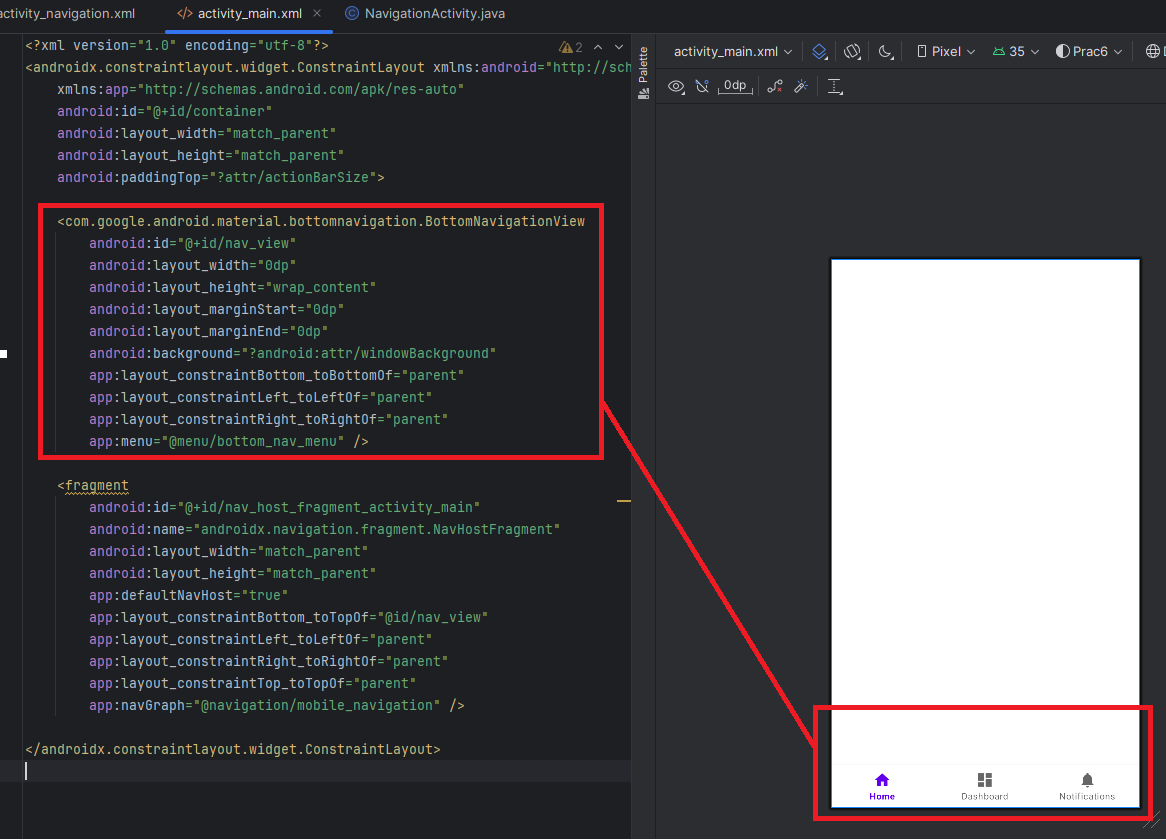


Рисунок – Описание BottomNavigationView в файле разметки

CoordinatorLayout – это контейнер, который расширяет FrameLayout.

CoordinatorLayout предназначен для двух основных вариантов использования:

* в качестве декора приложения верхнего уровня или макета Chrome;
* как контейнер для определенного взаимодействия с одним или несколькими дочерними представлениями.

После добавления BottomNavigationView в активность необходимо наполнить его элементами (экранами, между которыми будет осуществляться перемещение). Для этого создается XML-файл в папке res/menu (например, bottom\_nav\_menu.xml) и добавляются в него пункты меню. Если папка menu отсутствует в проекте, то сначала ее необходимо добавить.

Для создания новой папки нажимаем правой кнопкой мыши на "res"→ "New" → "Android Resource Directory" (Рисунок 3). Вводим название папки, выбираем тип ресурса из выпадающего списка (Рисунок 4).

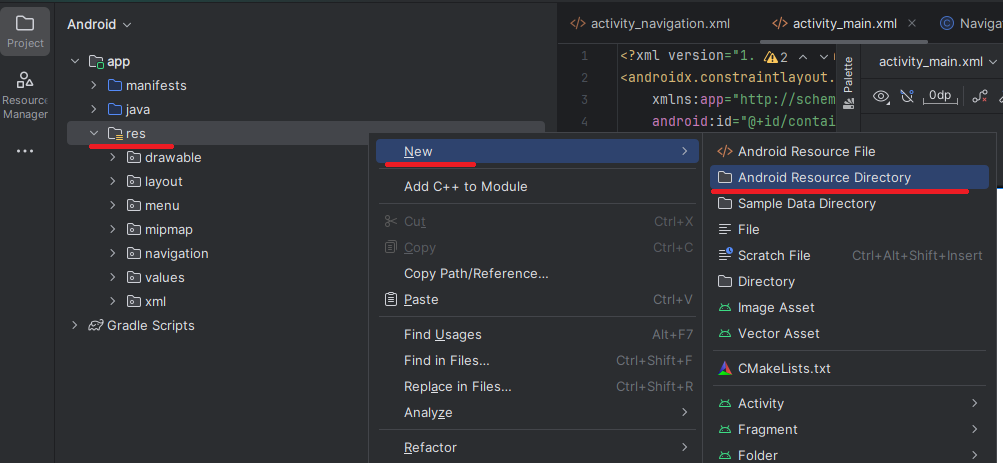


Рисунок – Создание новой папки в проекте

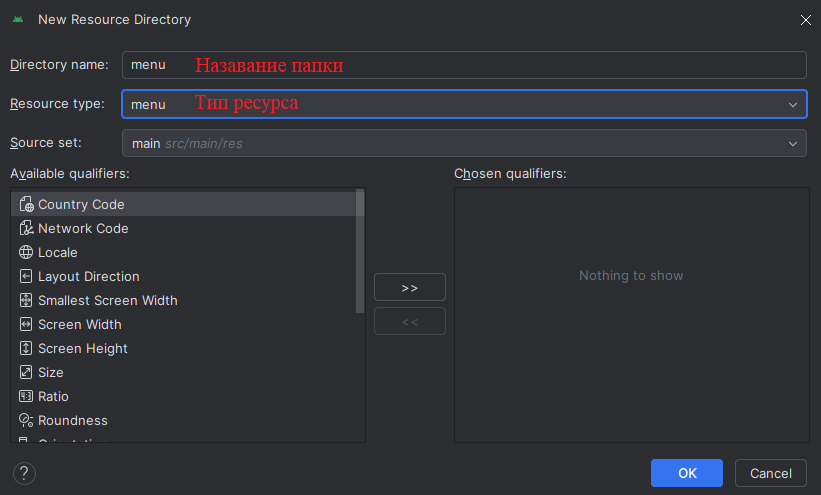


Рисунок – Настройка создаваемой папки

После создания XML-файла его можно наполнить пунктами меню (Рисунок 5).

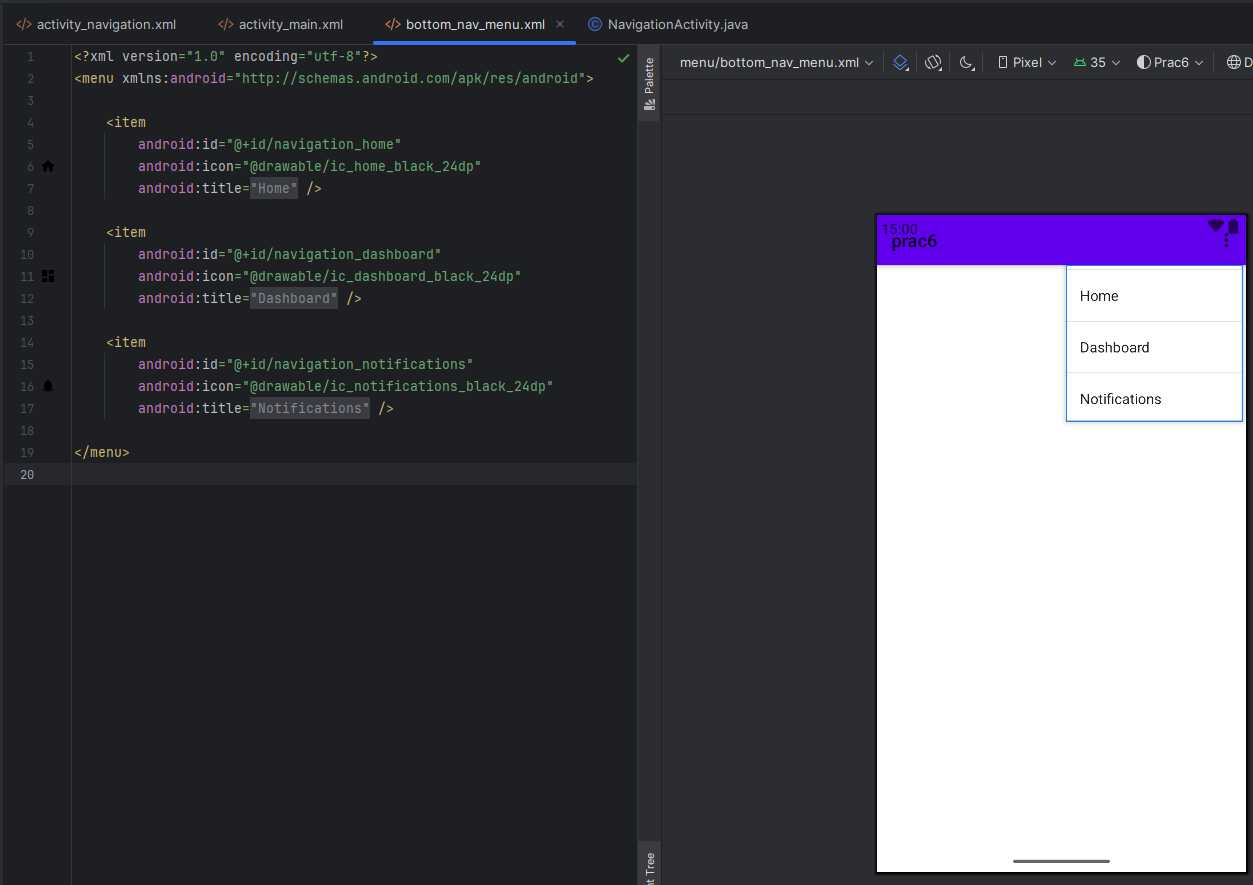


Рисунок – Графическое и кодовое представление меню навигации

Тег <menu> является корневым узлом файла и определяет меню, состоящее из одного или нескольких элементов <item> и <group>.

Элемент <item> представляет объект MenuItem, которой является одним из элементов меню. Этот элемент может содержать внутренний подэлемент <menu>, с помощью которого создается подменю.

Элемент <item> включает следующие атрибуты, которые определяют его внешний вид и поведение:

* android:id: уникальный id элемента меню, который позволяет его опознать при выборе пользователем и найти через поиск ресурса по id;
* android:icon: ссылка на ресурс drawable, который задает изображение для элемента (android:icon="@drawable/ic\_help");
* android:title: ссылка на ресурс строки, содержащий заголовок элемента. По умолчанию имеет значение "Settings";

Теперь пользователь может создавать столько элементов, сколько захочет, в bottom\_nav\_menu.xml файле. Пользователю также необходимо создать значок для отображения каждого из этих элементов. Чтобы создать значок, нужно нажать на "drawable"→ "New" → "Image Asset" (Рисунок 6).

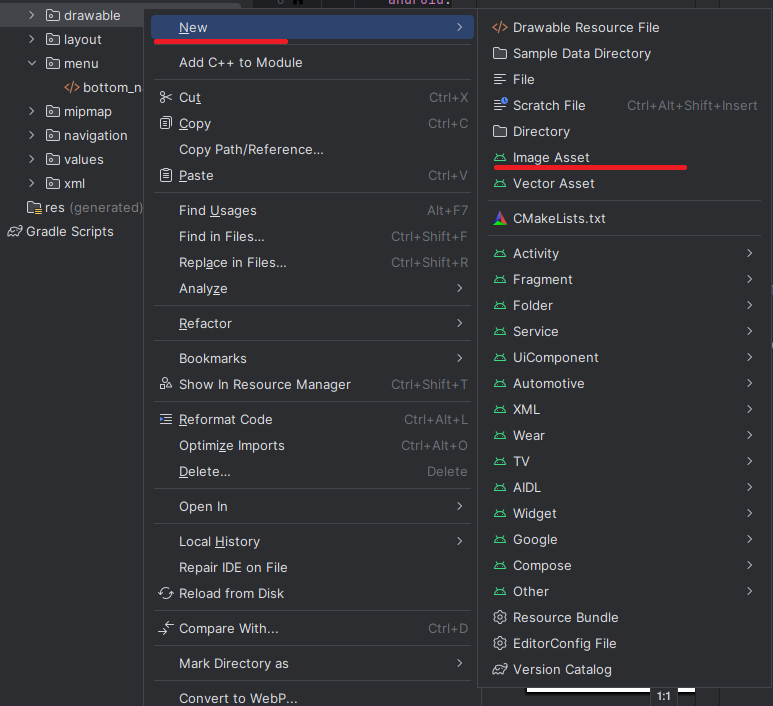


Рисунок – Создание новой иконки

Далее идёт выбор нужного типа иконки (Рисунок 7).

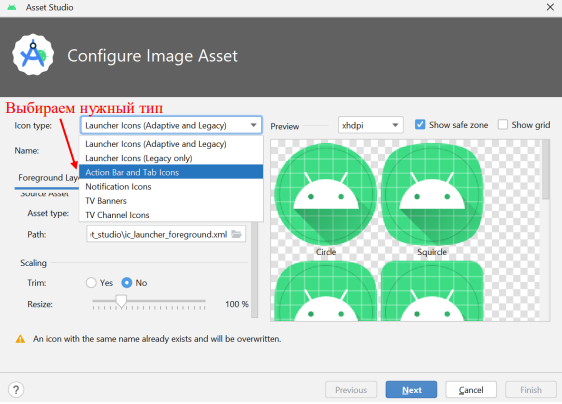


Рисунок – Выбор необходимого типа иконки

В открывшемся окне пользователь может назвать значок как угодно, но он не должен содержать ни одной заглавной буквы. Пользователь может выбрать нужный значок, выполнив поиск по нему, а когда пользователь закончит, то необходимо нажать "Next"→ "Finish" (Рисунки Рисунок 8-Рисунок 10).

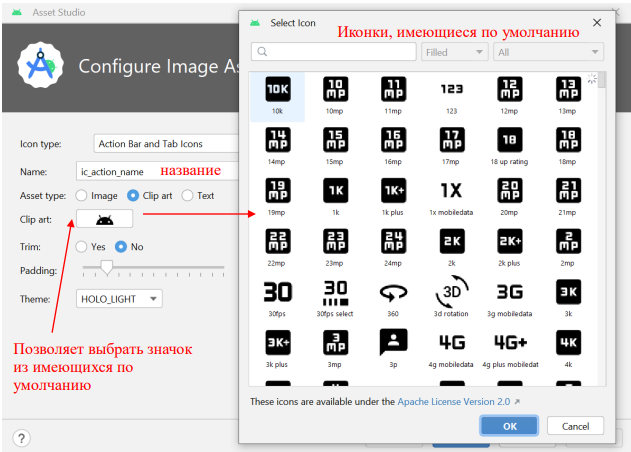


Рисунок – Настройка иконки, часть 1



Рисунок – Настройка иконки, часть 2

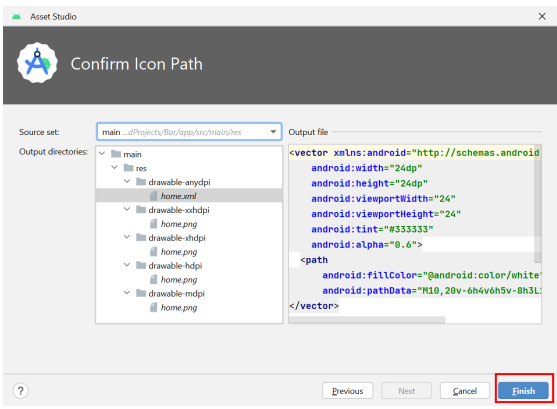


Рисунок – Настройка иконки, часть 3

Image Asset поможет создать следующие типы значков:

* иконки лаунчера;
* значки панели действий и вкладок;
* значки уведомлений.

Если запустить сборку проекта, то можно получится отображение меню в нижней части экрана (Рисунок 11).

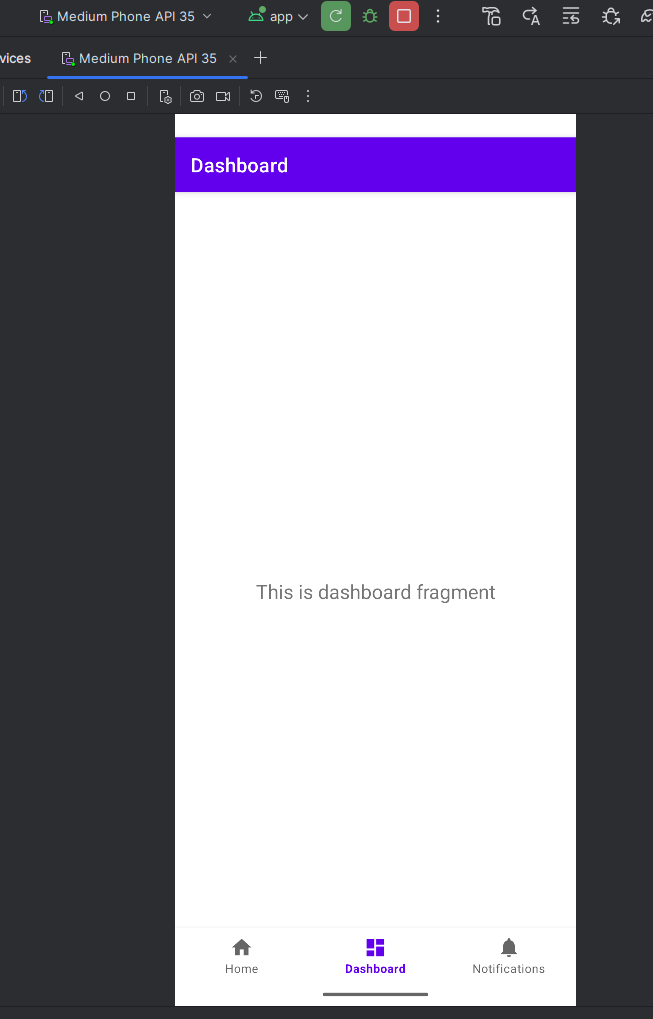


Рисунок – Отображение нижнего меню навигации

Однако только отображение не подходит для работы с BottomBar. Необходимо, чтобы при нажатии на элемент, происходили некоторые действия. Для этого необходимо установить слушатель нажатий в классе активности (Рисунок 12).

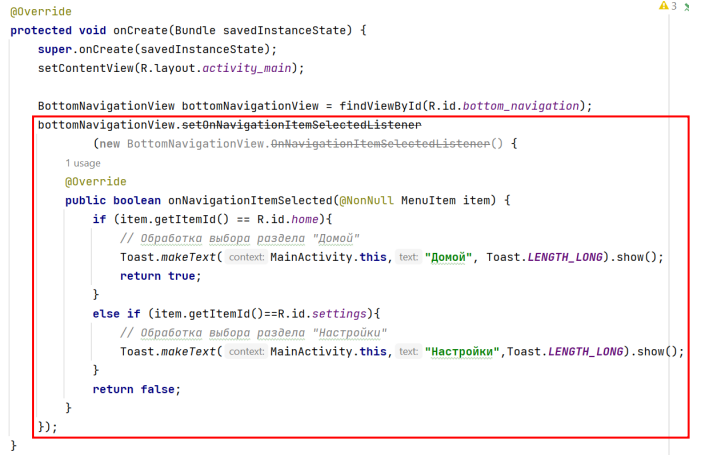


Рисунок – Реализация логики обработчика событий меню навигации

Чтобы понять, какой пункт меню выбран, вначале получаем его идентификатор через item.getItemId(). Затем пробегаемся в конструкции if…else и выбираем нужный вариант и в зависимости от выбора производим определенные действия (Рисунки Рисунок 13-Рисунок 14).

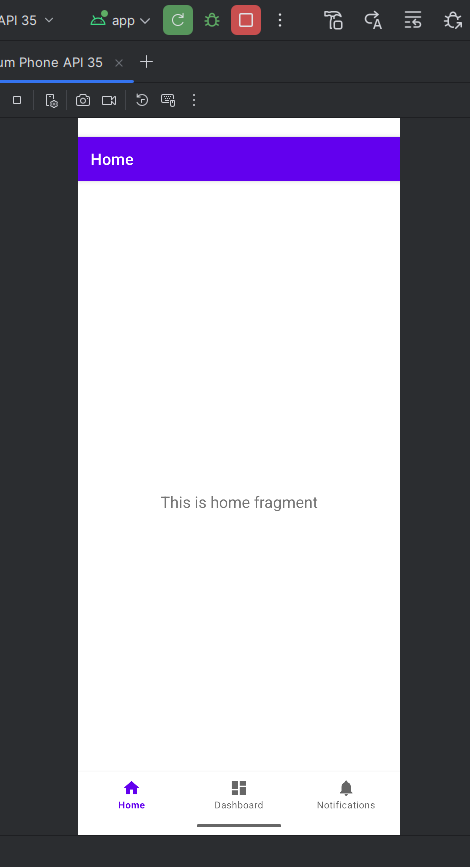


Рисунок – Отображение результата возможных событий, часть 1

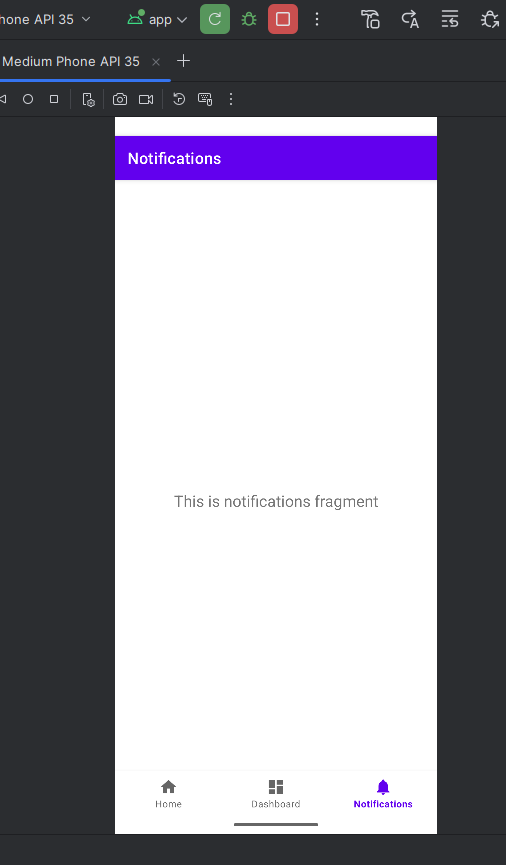


Рисунок – Отображение результата возможных событий, часть 2

## ActionBar

ActionBar в Android представляет собой верхнюю панель приложения, которая обеспечивает удобный доступ к наиболее важным функциям приложения, а также поддерживает навигацию. Создание и использование ActionBar в связке с нижним и боковым меню может значительно улучшить пользовательский интерфейс и удобство навигации в приложении, например показывая текущий экран, на котором находится пользователь.

ActionBar уже встроен в стандартные темы Activity начиная с API Level 11 (Honeycomb). Для его использования необходимо убедиться, что выбранная Activity наследуется от AppCompatActivity, и используется одну из тем Theme.AppCompat. ActionBar будет отображаться вверху страницы (Рисунок 15).

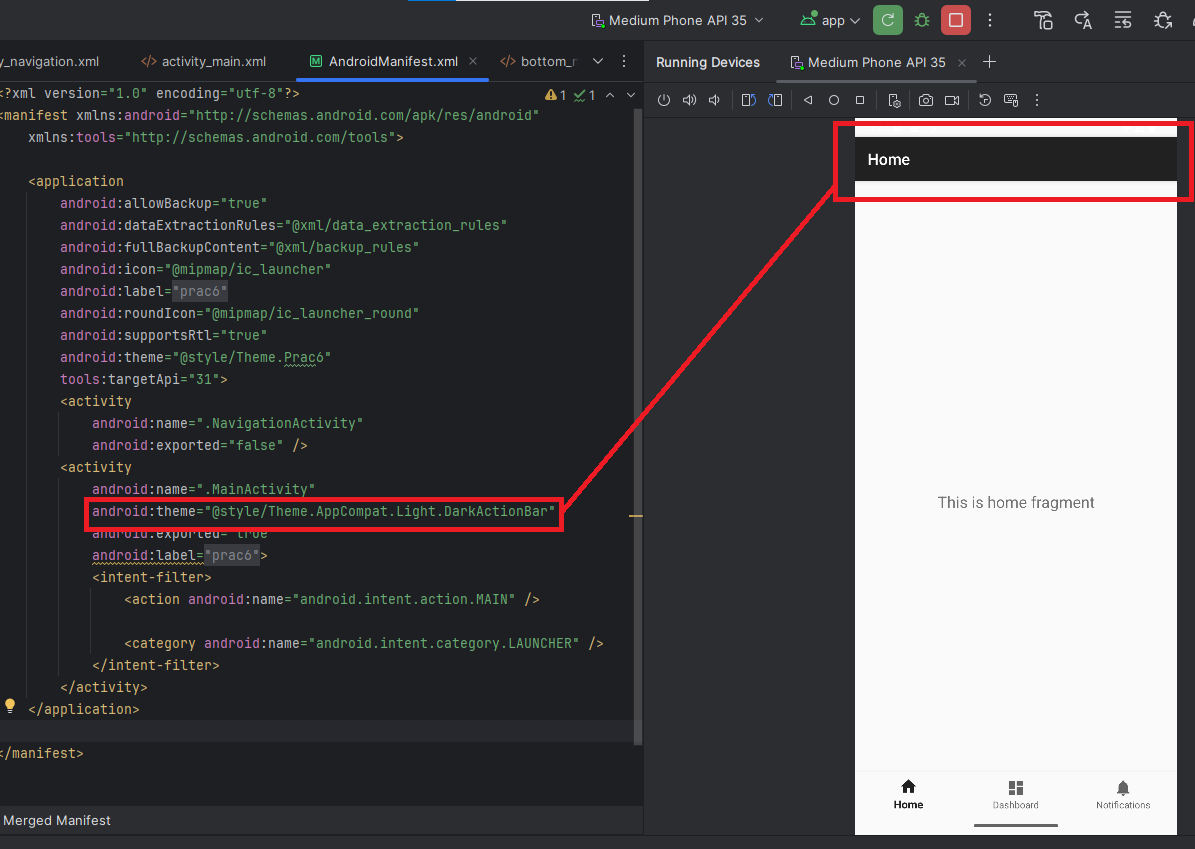


Рисунок – Графическое и кодовое представление ActionBar

Однако такой ActionBar не несет никакой пользы. Для дальнейшей работы его необходимо настроить. В активности можно настроить ActionBar на отображение какого-либо строкового значения (Рисунок 16).

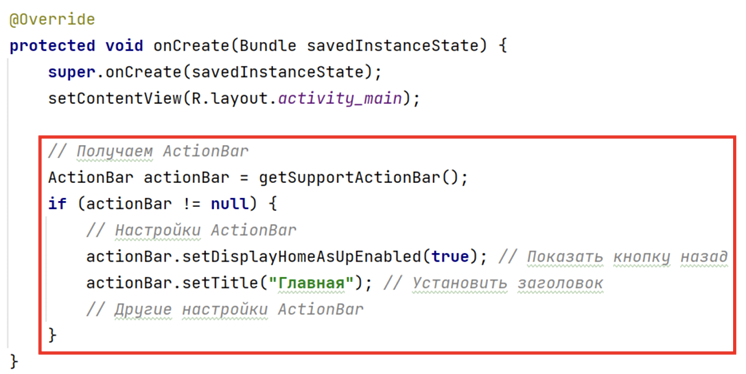


Рисунок – Настройка ActionBar в коде

Однако этого тоже может быть недостаточно, так как нам может понадобиться быстрой доступ к некоторым элементам.

Используем ранее созданный файл bottom\_nav\_menu.xml, только немного его изменим, добавив элементам меню некоторые параметры для наглядности (Рисунок 17).

* android:orderInCategory: значение этого атрибута определяет положение элемента в ActionBar. Есть два способа определить положение различных пунктов меню. Первый - предоставить одинаковое значение этого атрибута для всех элементов, и позиция будет определена в том же порядке, в каком они объявлены в коде. Второй способ - предоставить разные числовые значения для всех элементов, и тогда элементы будут располагаться в соответствии с порядком возрастания значения этого атрибута;
* app:showAsAction: этот атрибут определяет, как элемент будет присутствовать на панели действий.

На выбор для значения поля app:showAsAction предлагается четыре возможных флага:

* always: постоянно отображать элемент на панели действий;
* ifRoom: сохранить элемент, если есть свободное место;
* never: с этим флагом элемент не будет отображаться в виде значка в ActionBar, но будет присутствовать в меню переполнения;
* withText: чтобы представить элемент одновременно в виде значка и заголовка, можно дополнить этот флаг флагом always или ifRoom (always / withText or ifRoom / withText).

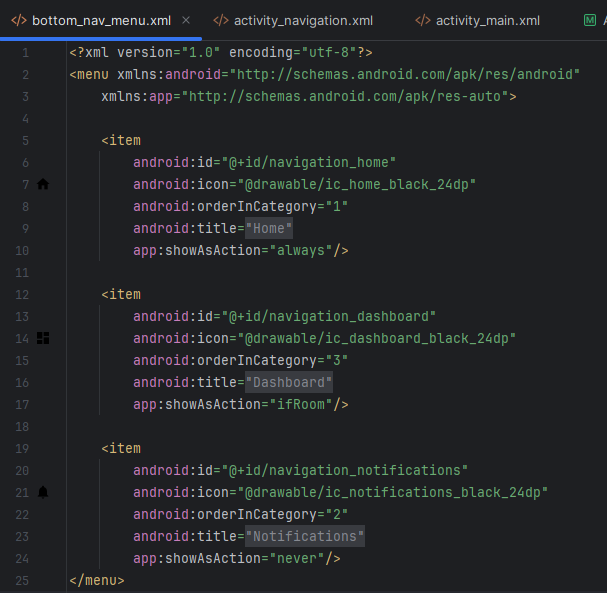


Рисунок – Измененный файл разметки bottom\_nav\_menu.xml

На рисунке Рисунок 17 определено меню с тремя элементами, но само определение элементов в файле еще не создает меню. Это всего лишь описание. Чтобы вывести его на экран, надо использовать его в классе Activity. Для этого надо переопределить метод onCreateOptionsMenu.

Метод getMenuInflater возвращает объект MenuInflater, у которого вызывается метод inflate(). Этот метод в качестве первого параметра принимает ресурс, представляющий наше описание меню в xml, и наполняет им объект menu, переданный в качестве второго параметра.

Теперь, если мы нажмем на любой из пунктов меню, то ничего не произойдет. Чтобы привязать к меню действия, нам надо переопределить в классе activity метод onOptionsItemSelected (Рисунок 18).

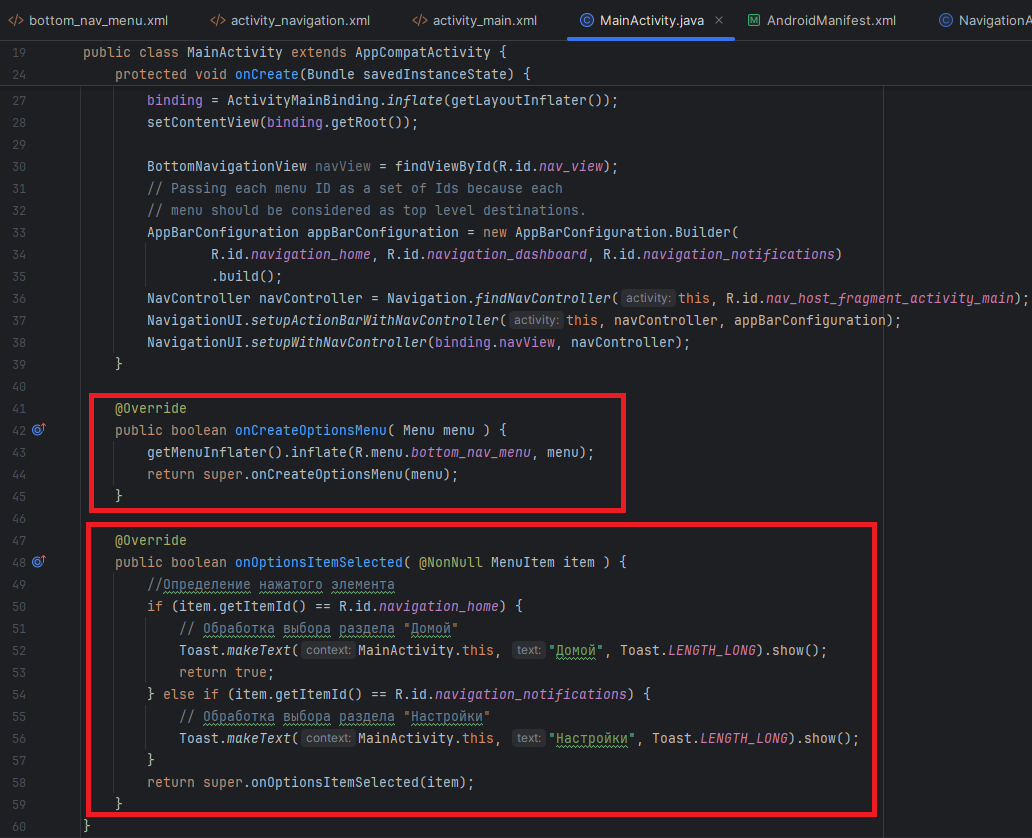


Рисунок – Переопределение необходимых методов класса ActionBar

И при нажатии на троеточие в правом верхнем углу экрана, а после нажатия на “Notifications” появится сообщение “Notifications” (Рисунок 19).

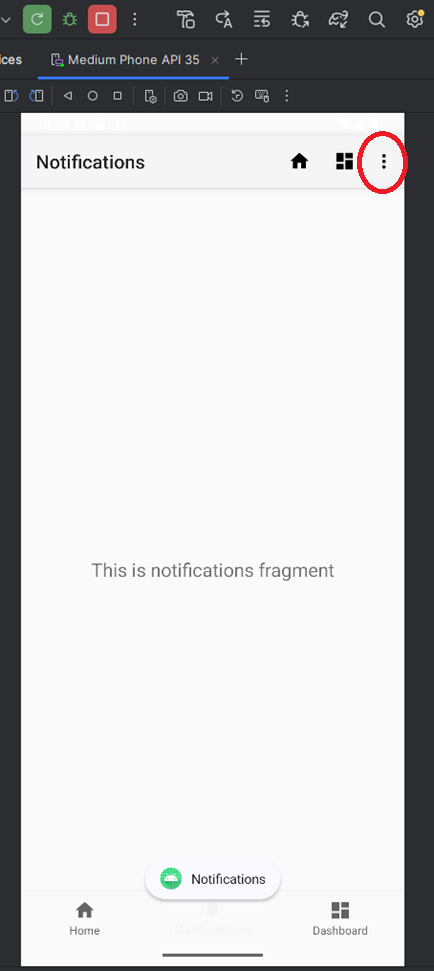


Рисунок – Отображение ActionBar в приложении

## Navigation Drawer

Создание выдвижной панели (Drawer) в Android обычно реализуется с помощью DrawerLayout и NavigationView. DrawerLayout используется как корневой контейнер интерфейса, который позволяет разместить основное содержимое приложения и выдвижную панель. NavigationView представляет собой панель навигации, которая отображается в DrawerLayout и содержит элементы меню для навигации по различным разделам приложения.

В файле макета активности необходимо добавить DrawerLayout как корневой элемент. Внутри DrawerLayout разместим основной контент и NavigationView для выдвижной панели (Рисунок 20).

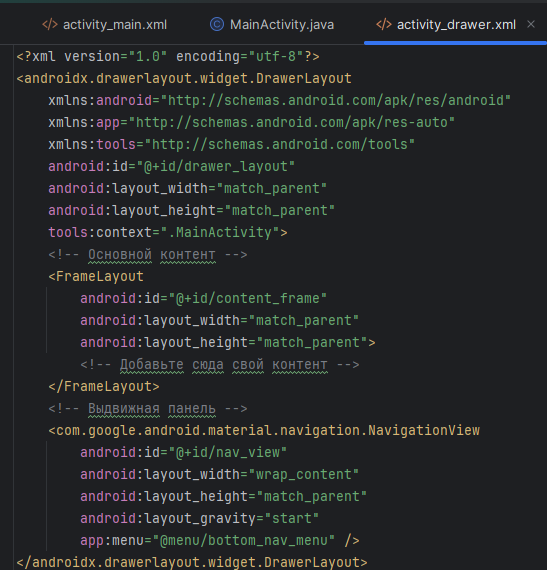


Рисунок – Код файла разметки activity\_drawer.xml

Создаётся XML-файл в папке res/menu (например, drawer\_menu.xml) и добавьте пункты меню для вашей выдвижной панели (Рисунок 21).

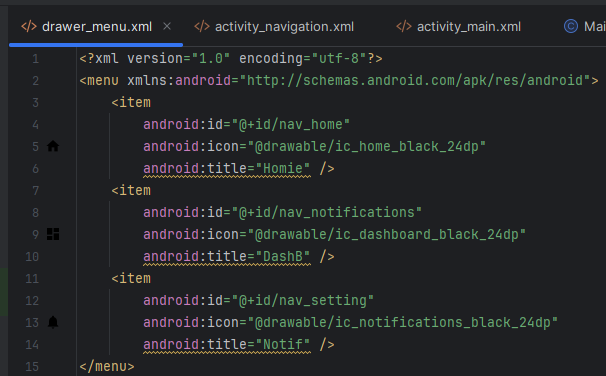


Рисунок – Описание элементов в файле drawer\_menu.xml

Затем по аналогии с BottomBar создать слушатель нажатий для каждого элемента меню.

Для интеграции ActionBar с DrawerLayout используется ActionBarDrawerToggle, который добавит иконку меню для открытия и закрытия Drawer и обеспечит анимацию иконки (Рисунок 22).

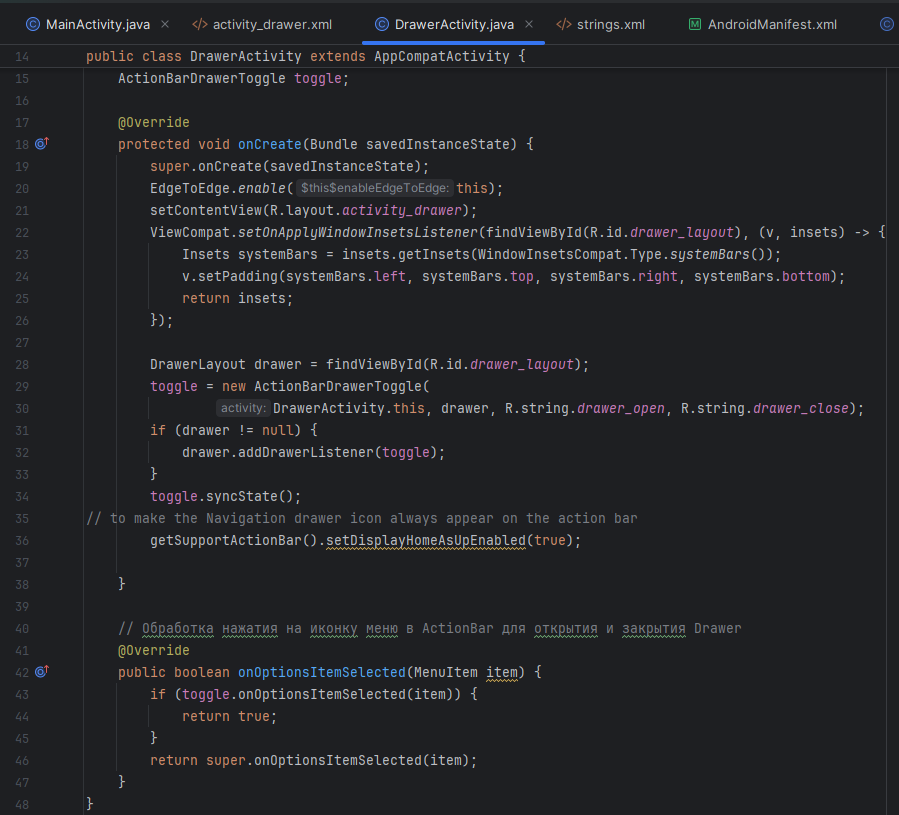


Рисунок – Реализация логики NavigationDrawer

Также на рисунке Рисунок 22 был добавлен слушатель для обработки нажатия на элементы (onOptionsItemSelected).

В итоге будет отображаться всплывающее меню с выбранными строковыми значениями (Рисунки Рисунок 23-Рисунок 24).



Рисунок – Отображение название всплывающего меню, часть 1

После запуска приложения пользователь нажмёт на иконку всплывающего меню, находящуюся в левом верхнем углу, и увидит содержимое всплывающего меню (Рисунок 24).

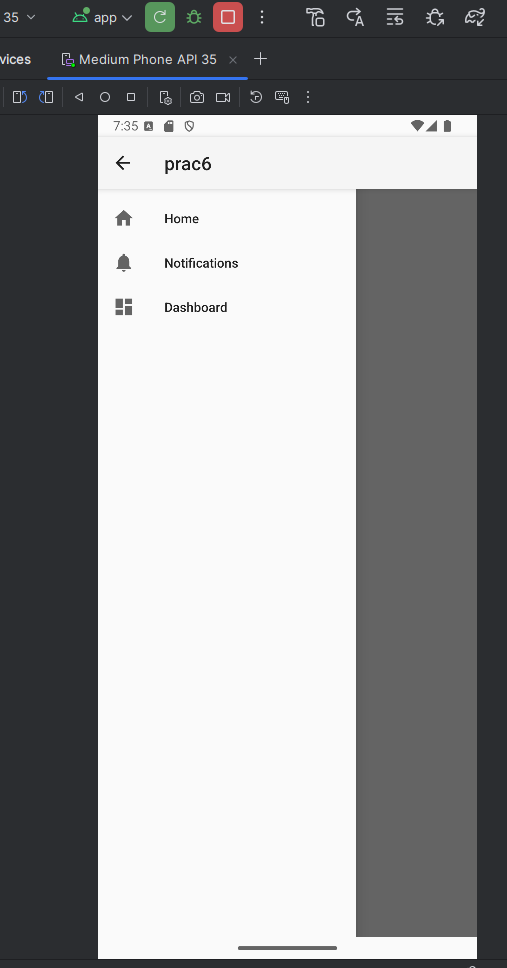


Рисунок – Отображение названия всплывающего меню, часть 2

Чтобы использовать ActionBarDrawerToggle, нужно создать экземпляр при помощи конструктора со следующими аргументами:

* Activity, в котором размещается боковая панель навигации;
* DrawerLayout – drawable ресурс, используемый в качестве индикатора панели. Стандартный навигационный значок панели доступен в Download the Action Bar Icon Pack;
* Строковый ресурс для обозначения открытой панели (для специальных возможностей);
* Строковый ресурс для обозначения закрытой панели (для специальных возможностей).

Теперь, в зависимости от использования класса ActionBarDrawerToggle в списке панели навигации, необходимо вызвать ActionBarDrawerToggle в нескольких местах жизненного цикла activity.

Кроме этого, ActionBar можно использовать вместе с BottomNavigationView для создания единой системы навигации. В этом случае ActionBar обычно служит для отображения контекстной информации (например, заголовка текущей страницы), а BottomNavigationView для навигации между основными разделами приложения (Рисунки Рисунок 25-Рисунок 26).

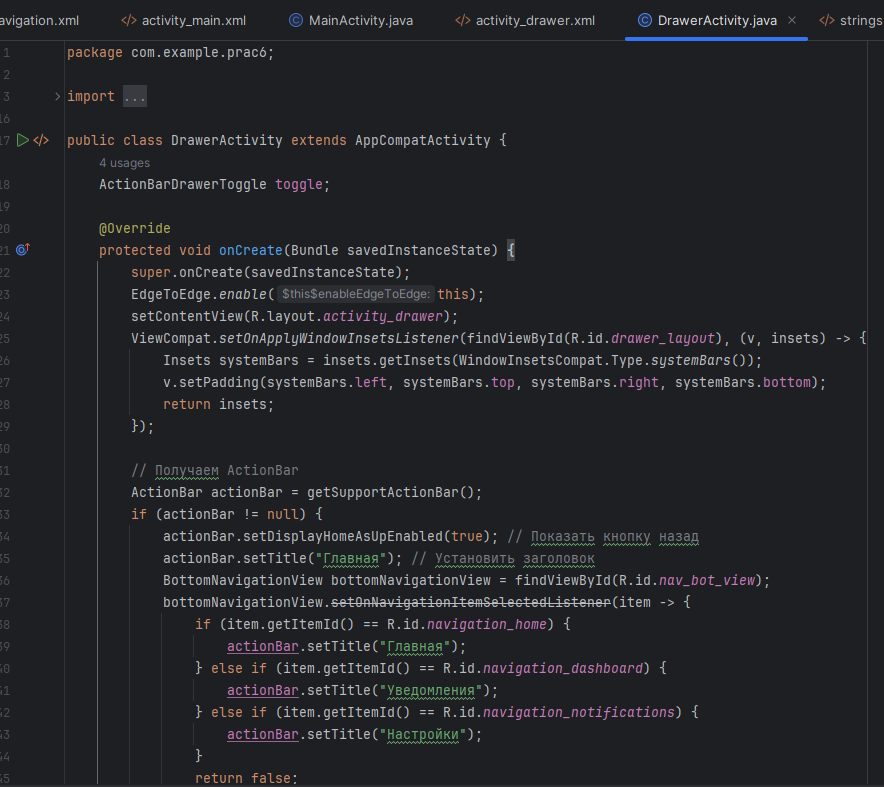


Рисунок – Реализация логики BottomNavigationView, часть 1

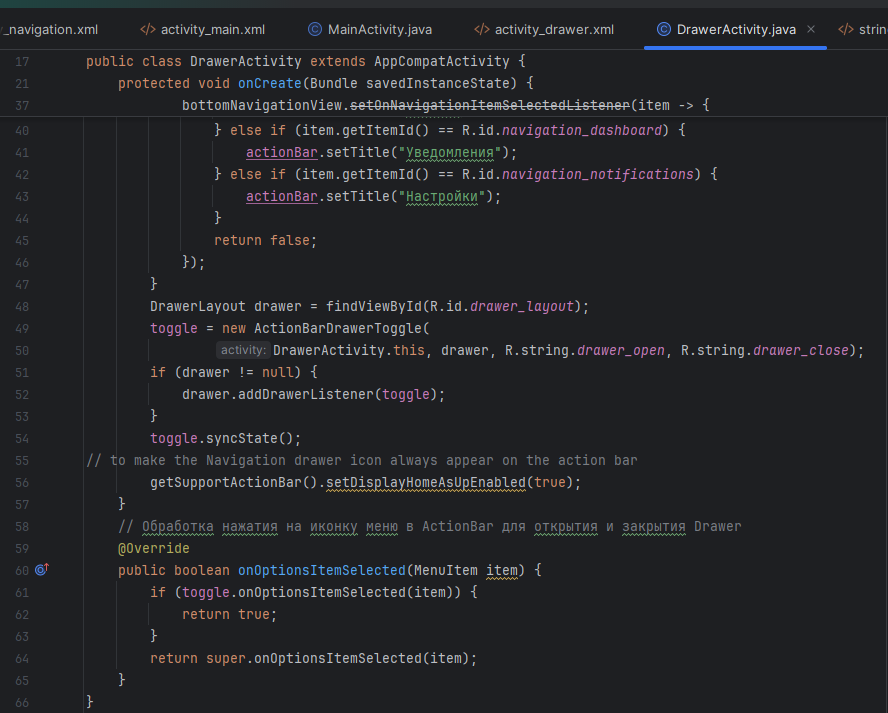


Рисунок – Реализация логики BottomNavigationView, часть 2

У пользователя появится возможность переходить по нижней панели на различные фрагменты (Рисунки Рисунок 27-Рисунок 28).

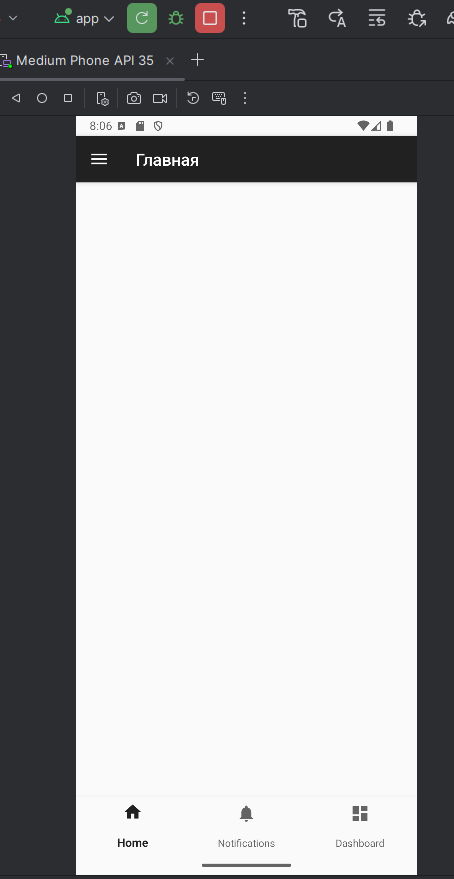


Рисунок – Работа панели навигации, часть 1

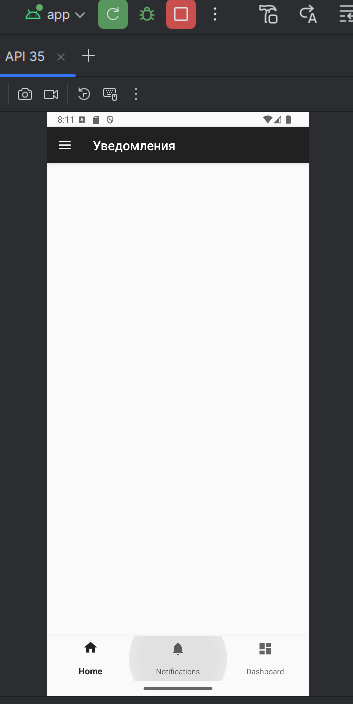


Рисунок – Работа панели навигации, часть 2

Это позволяет пользователю быстро переключаться между разделами приложения с помощью нижнего меню, в то время как верхняя панель (ActionBar) отображает контекстную информацию о выбранном разделе или предоставляет дополнительные опции действий, связанных с текущим экраном. Использование ActionBar в сочетании с боковым и нижним меню позволяет создать гибкую и интуитивно понятную систему навигации для пользователей, обеспечивая легкий доступ ко всем важным функциями в приложении.

# ПРАКТИЧЕСКОЕ ЗАДАНИЕ

## Боковая панель навигации

### Разметка

Для начала создадим проект (Рисунок 30).

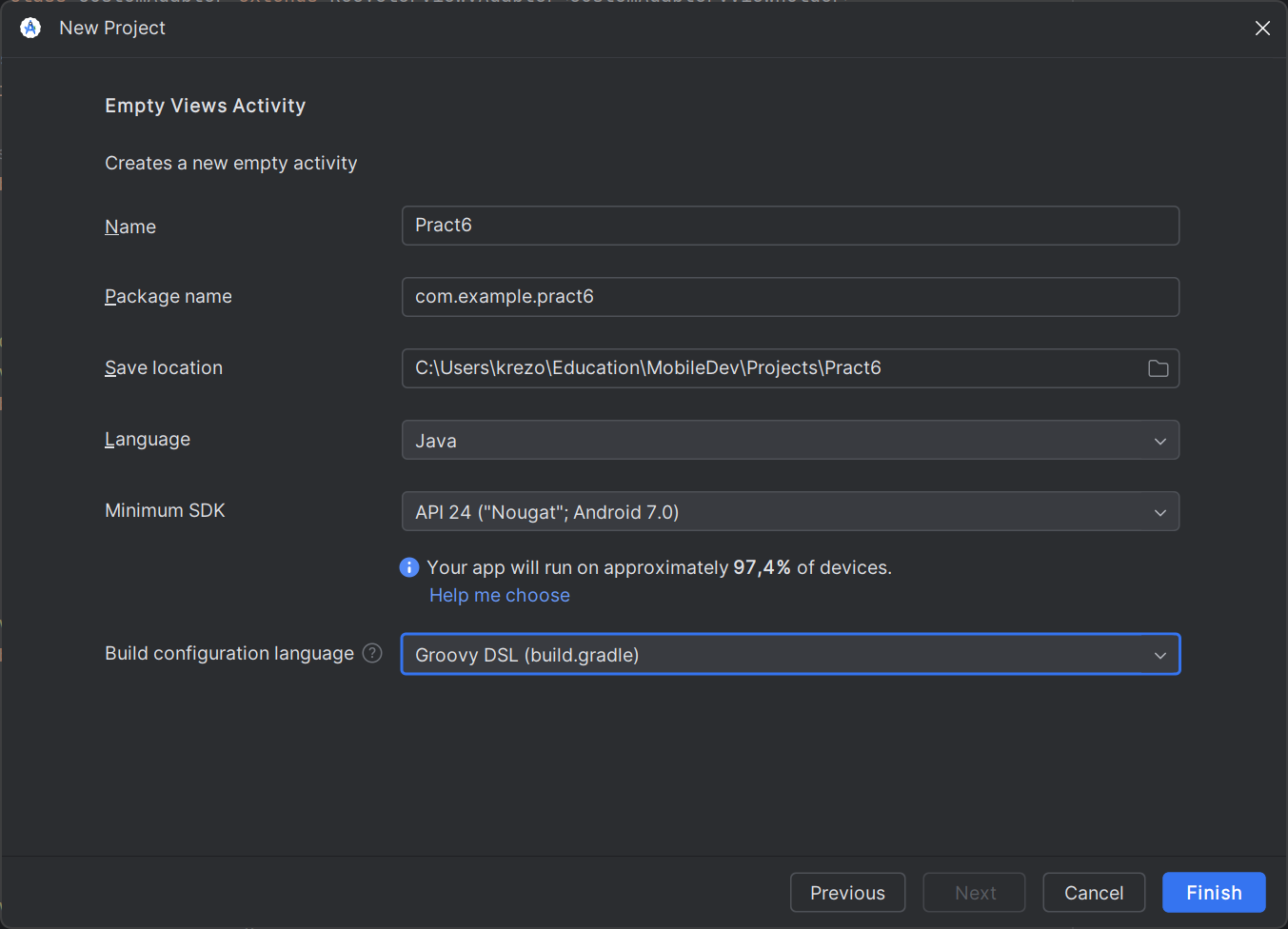


Рисунок – Создание проекта

Создадим файл разметки бокового меню activity\_main\_side.xml, перед этим создав папку ресурсов main (Рисунки Рисунок 30-Рисунок 31).

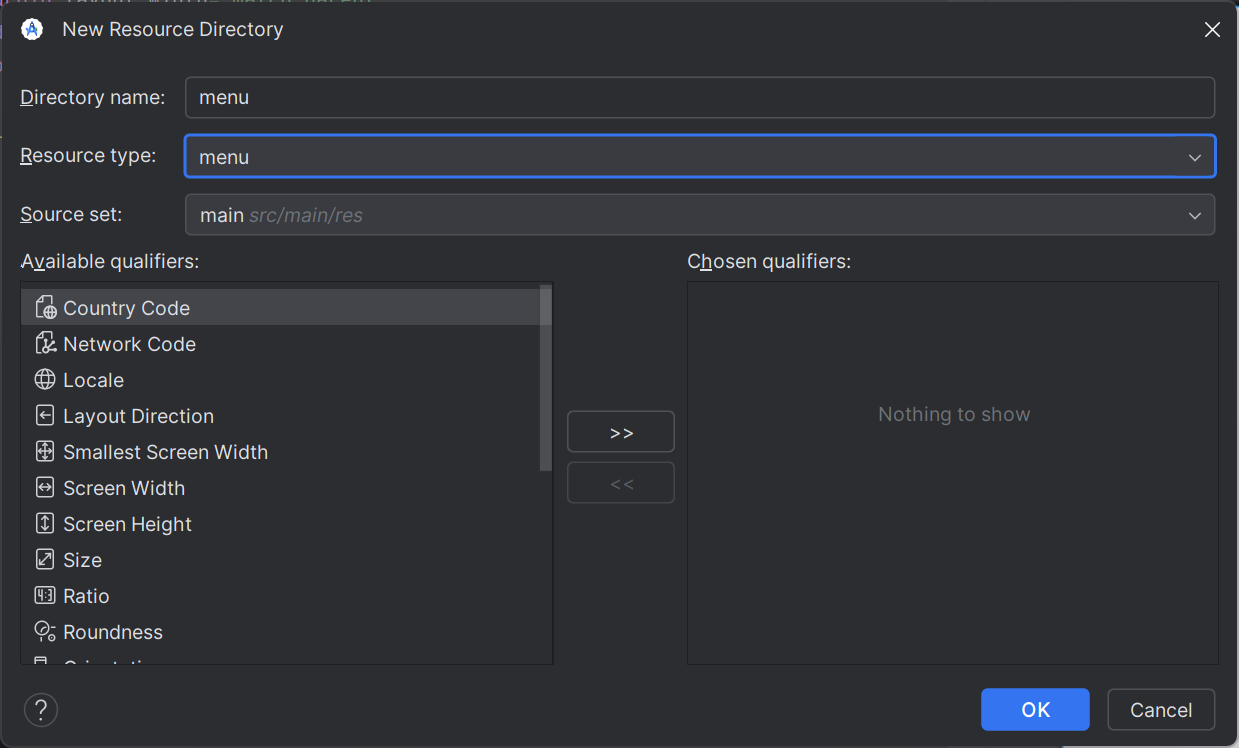


Рисунок – Создание папки ресурсов меню

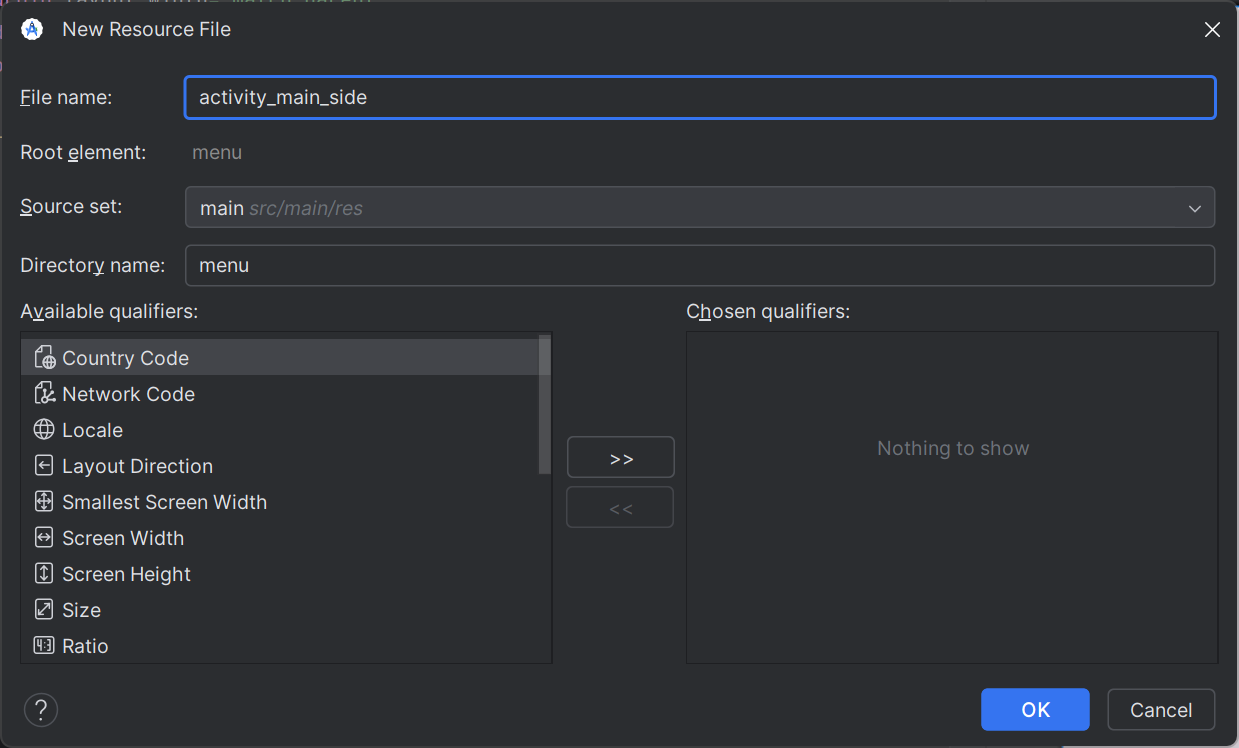


Рисунок – Создание файла разметки меню activity\_main\_side.xml

Заполним новый файл разметки меню необходимыми элементами (Рисунок 32).

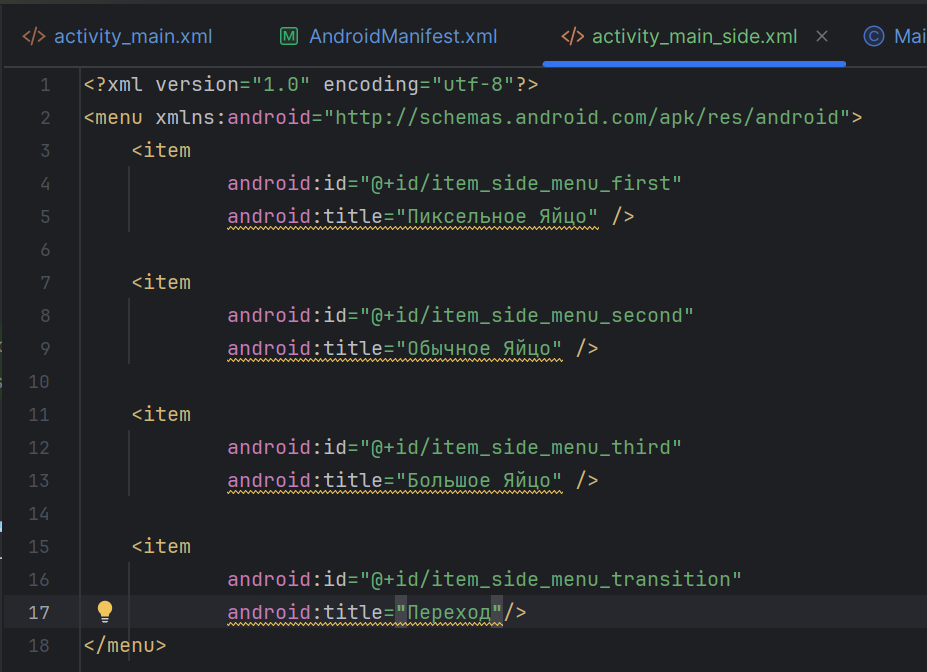


Рисунок 32 – Файл разметки activity\_main\_side.xml

Для того, чтобы появился ActionBar, необходимо установить совместимую тему в манифесте (Рисунок 33).

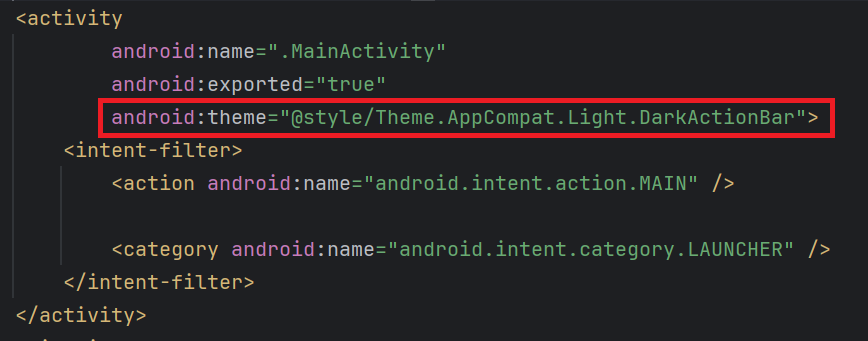


Рисунок – Настройки MainActivity в AndroidManifest.xml

Создадим три фрагмента Fragment1, Fragment2 и Fragment3 и заполним их аналогично фрагменту Fragment1 (Рисунок 34).

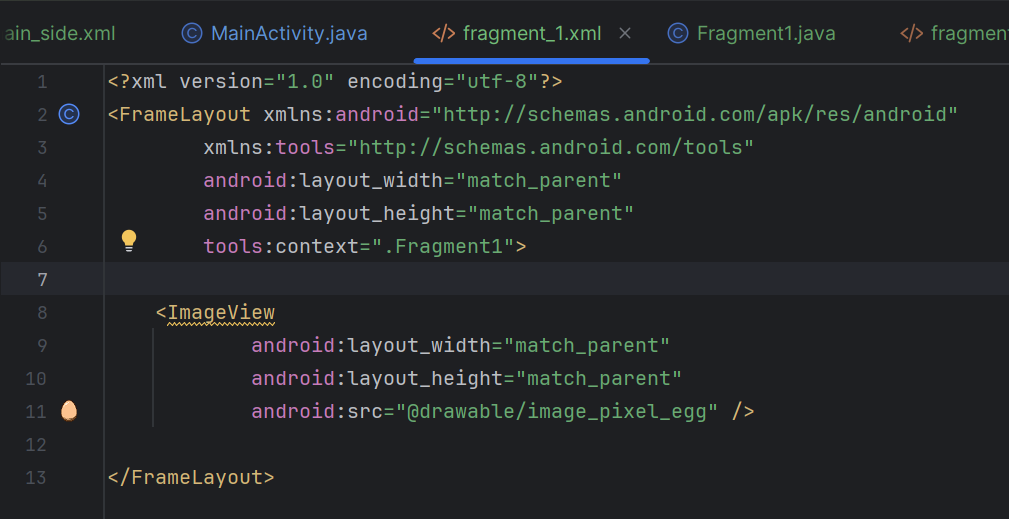


Рисунок – Файл разметки fragment\_1.xml

Для отображения опций выпадающего меню в файл разметки activity\_main.xml добавим элемент NavigationView, где установим ссылку на разметку меню. Также расположим картинку на весь экран (Рисунок 35).

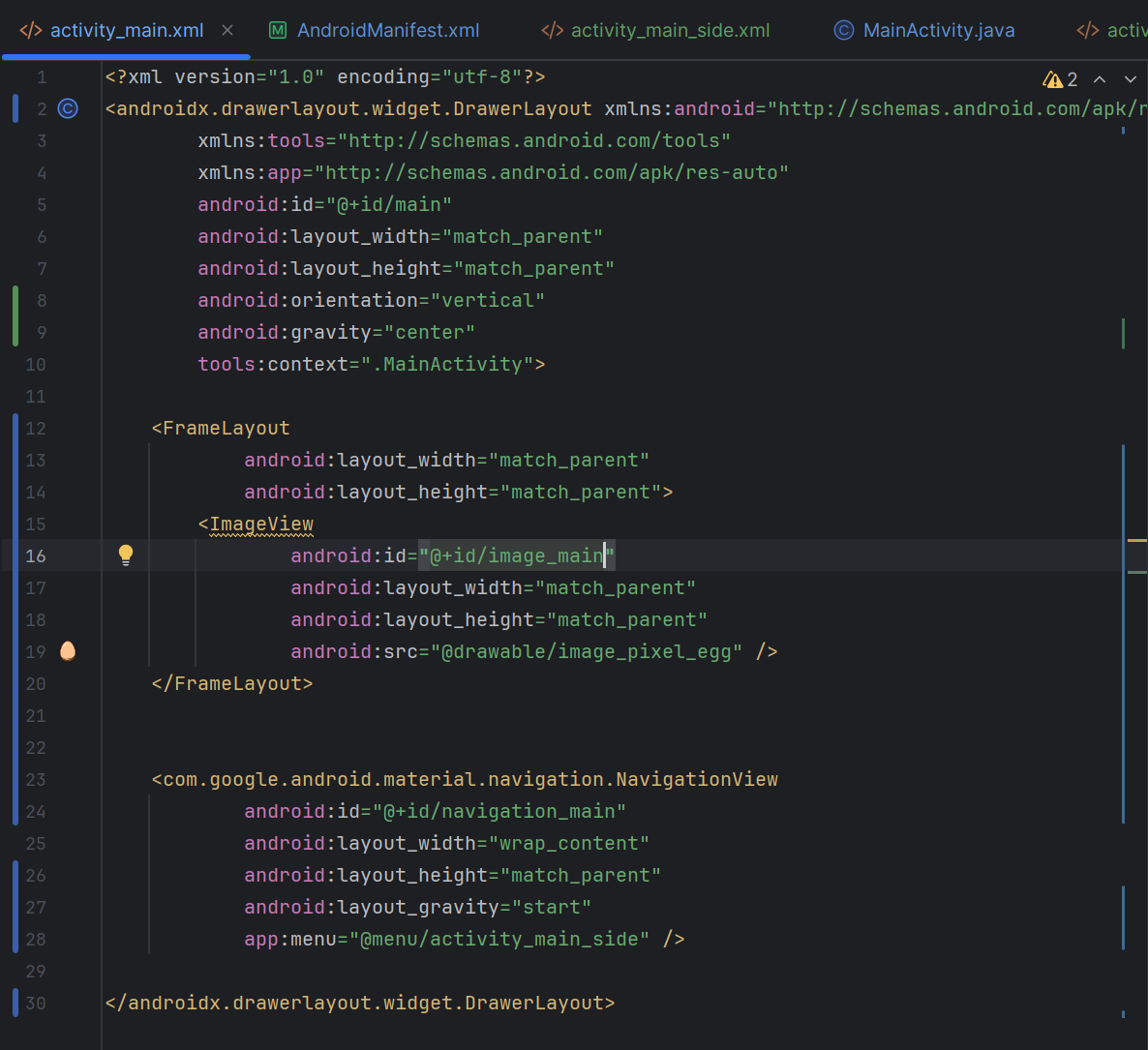


Рисунок – Файл разметки activity\_main.xml

Создадим класс Activity и назовем его SecondActivity (Рисунок 36).

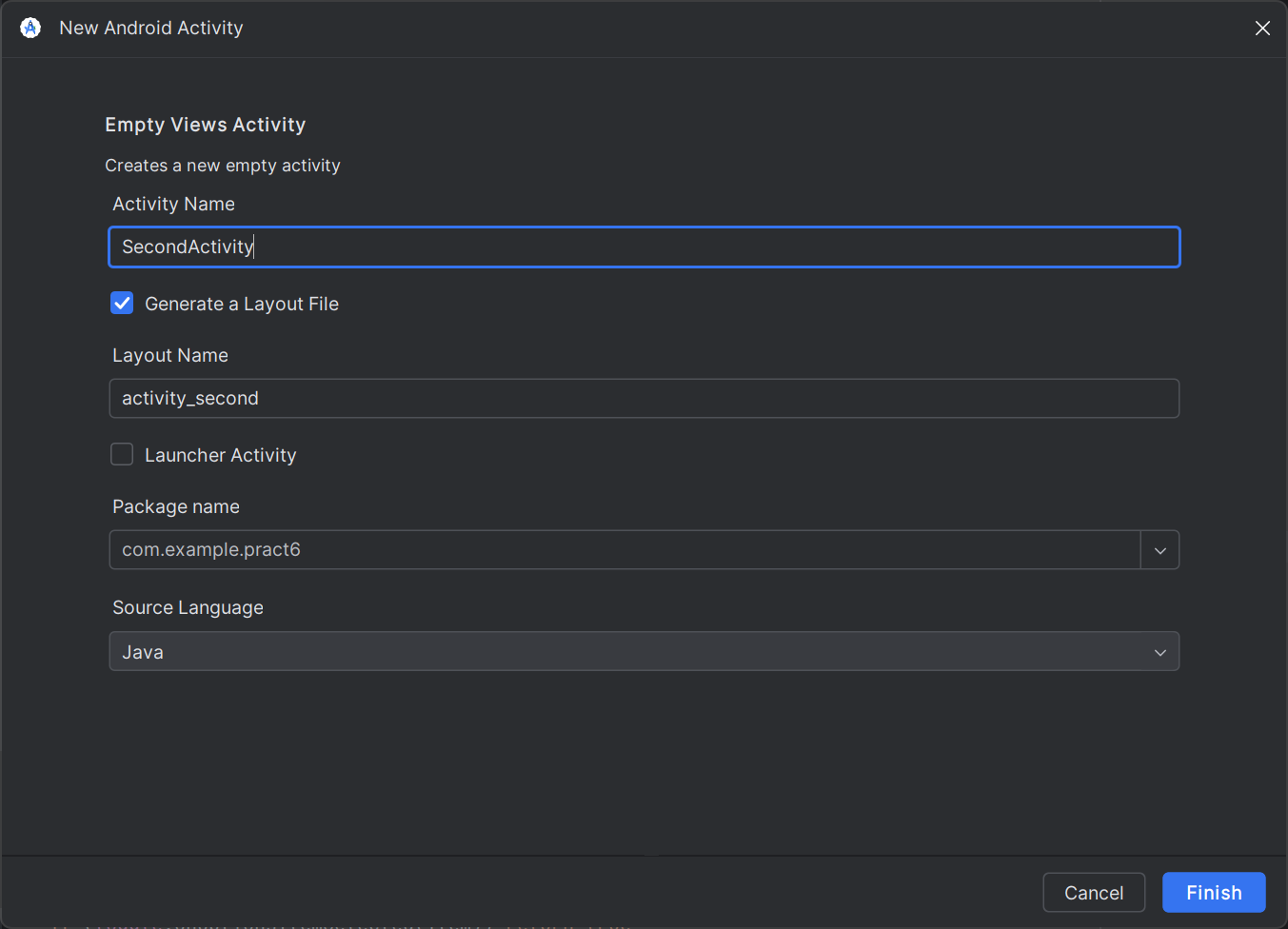


Рисунок – Создание SecondActivity

### Код

В методе onCreate() класса ActivityMain создадим кнопку для открытия выпадающего бокового меню и укажем название (Рисунок 37).

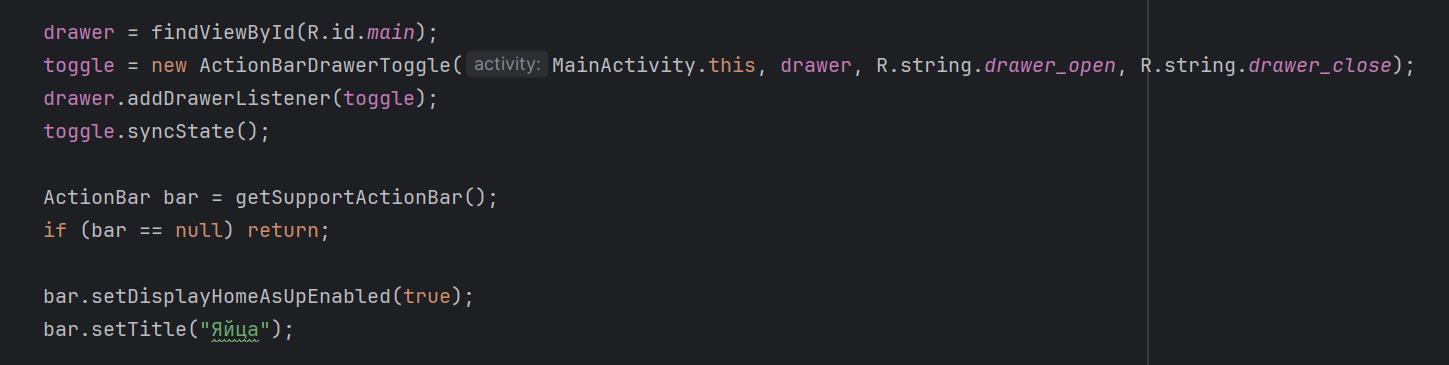


Рисунок – Описание метода onCreate() класса MainActivity, часть 1

В остальной части метода реализуем обработку нажатия на разные кнопки меню (Рисунок 38).

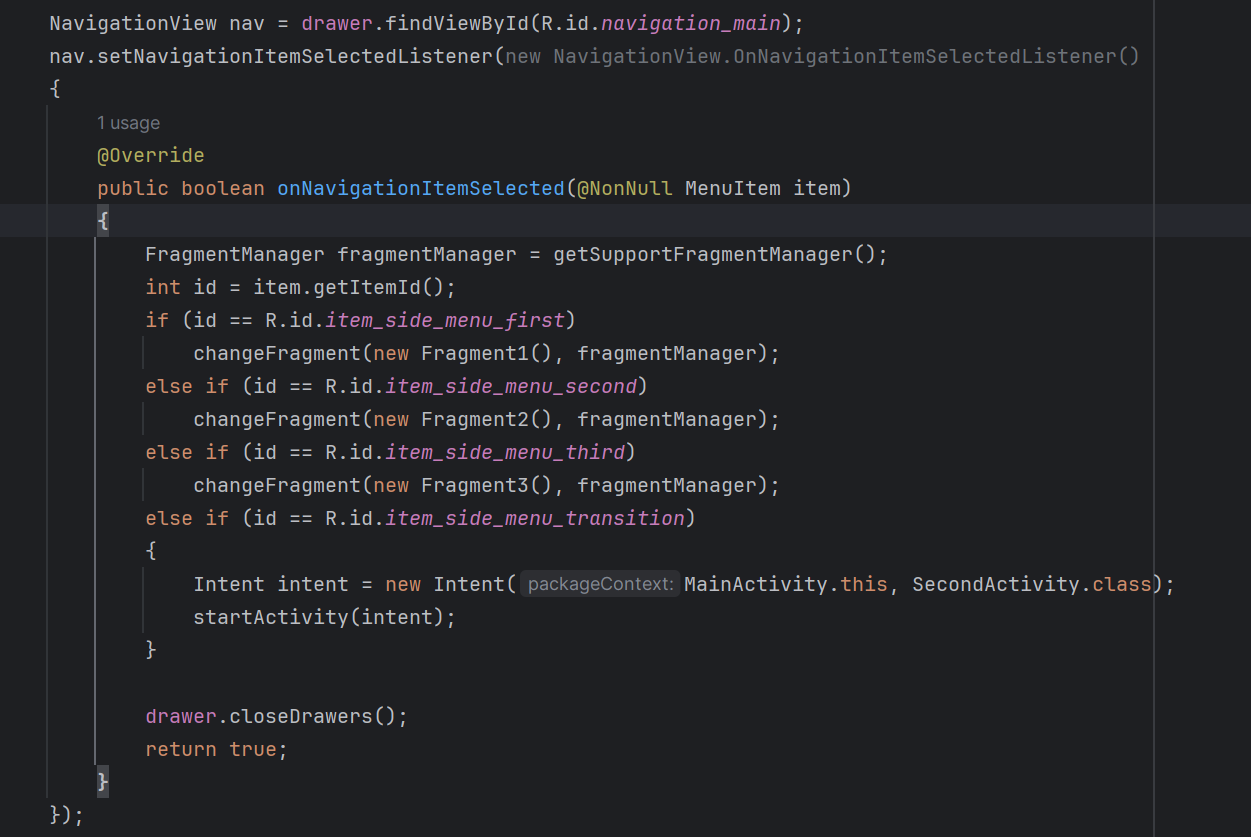


Рисунок – Описание метода onCreate() класса MainActivity, часть 2

Опишем метод changeFragment(), для смены отображаемого фрагмента. Также для работы ActionBarDrawer перегрузим метод onOptionsItemSelected(), в котором требуется обработать нажатие на кнопку, открывающую меню (Рисунок 39).

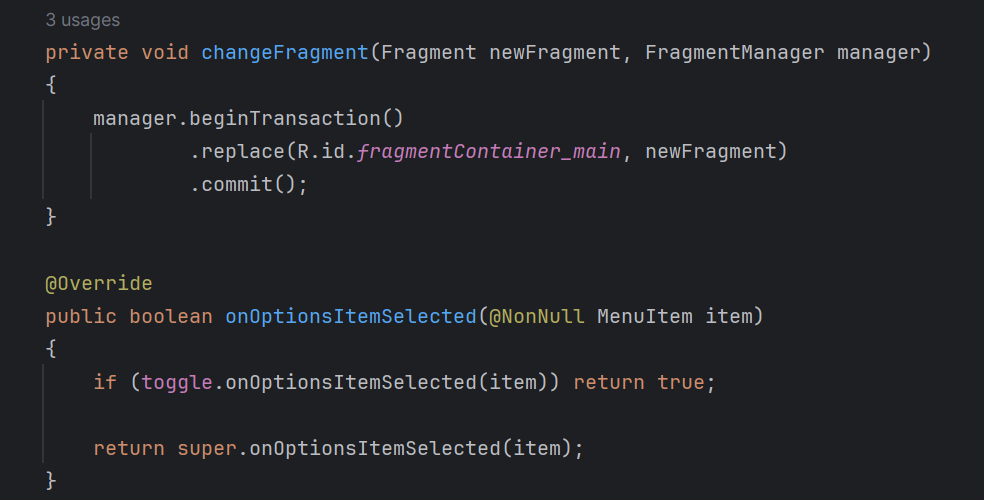


Рисунок 39 – Описание методов changeFragment() и onOptionsItemSelected() класса MainActivity

### Тестирование

Зайдем в приложение. Изначально на экране картинка не выбрана (Рисунок 40).



Рисунок – Начальный экран приложения

При нажатии на иконку в левом верхнем углу открывается левая панель навигации (Рисунок 41).

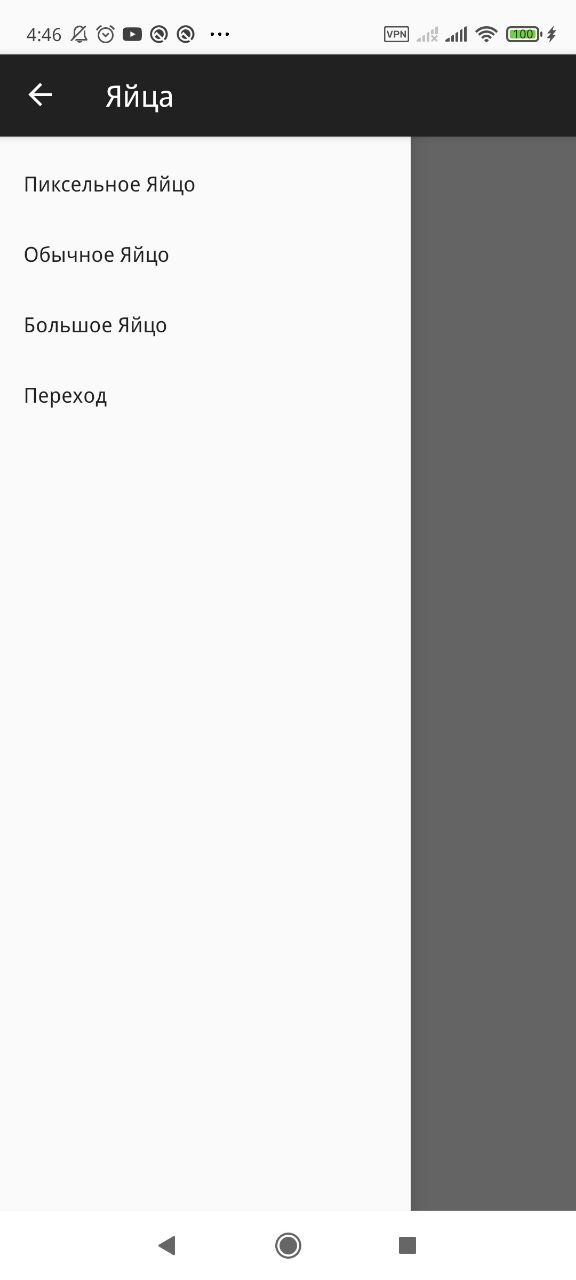


Рисунок – Левая панель навигации

На рисунках Рисунок 42-Рисунок 45 представлен результат нажатия на опции с 1 по 4 соответственно.



Рисунок – Нажатие на опцию «Пиксельное Яйцо»



Рисунок 43 – Нажатие на опцию «Обычное Яйцо»



Рисунок 44 – Нажатие на опцию «Большое Яйцо»



Рисунок 45 – Нажатие на опцию «Переход»

При нажатии на опцию «Переход» происходит переход на вторую Activity.

## Нижняя панель навигации

### Разметка

Для начала создадим три новых фрагмента Fragment4, Fragment5 и Fragment6 по аналогии с Fragment1. Потом создадим файл разметки нижнего меню activity\_second\_bottom.xml (Рисунок 46).

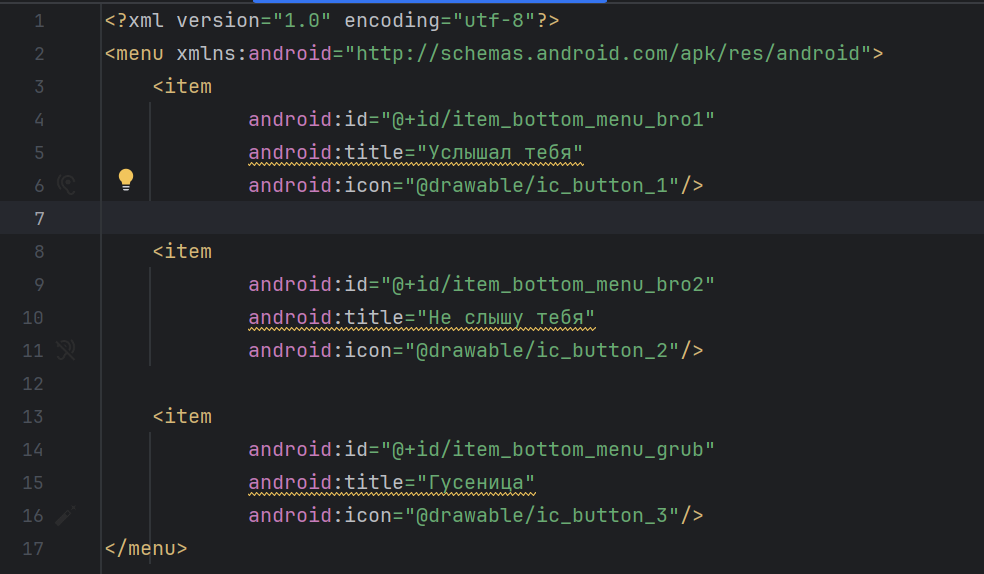


Рисунок 46 – Файл разметки activity\_second\_bottom.xml

После этого создадим необходимые иконки и заполним файл разметки activity\_second.xml (Рисунок 47).

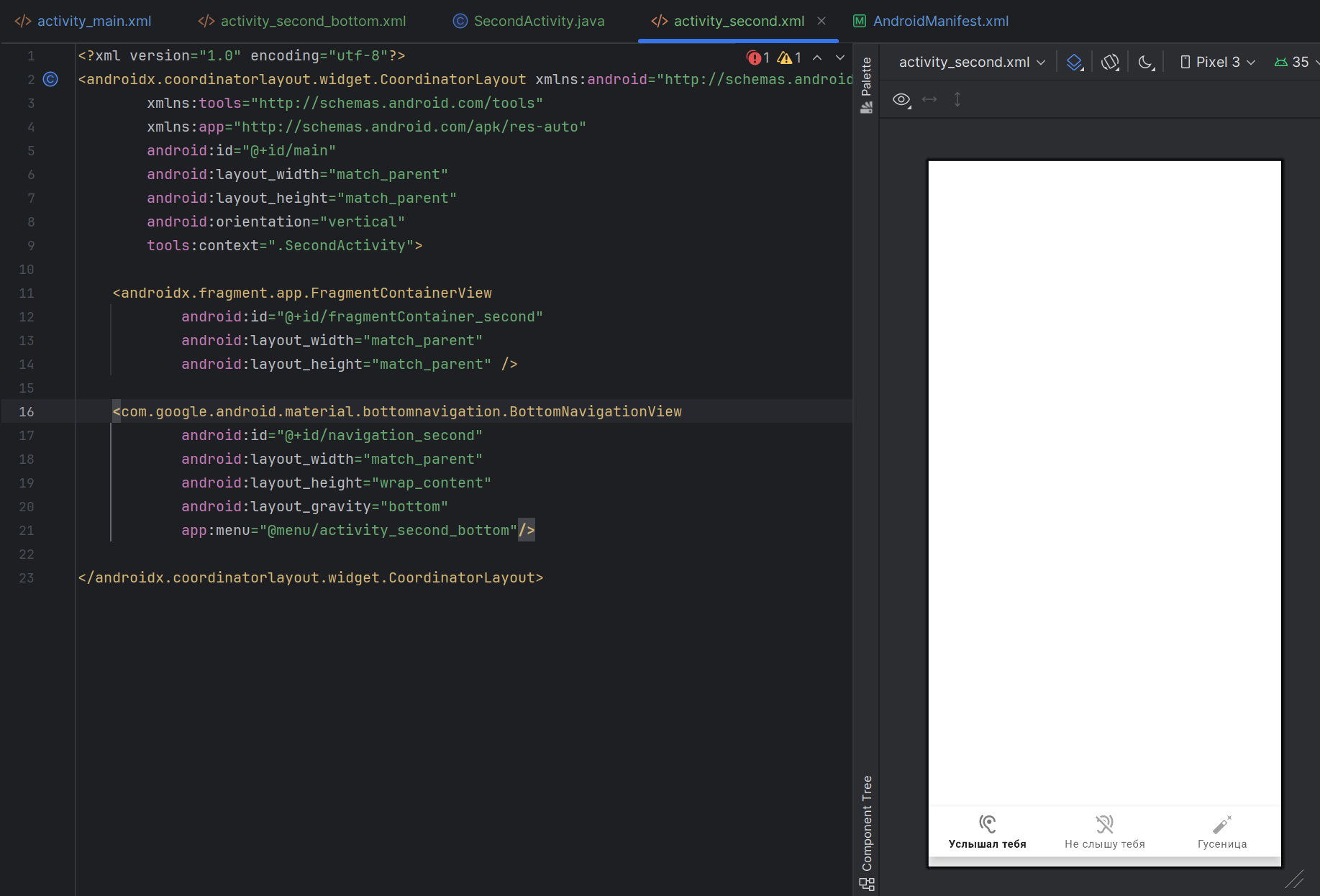


Рисунок 47 – Файл разметки activity\_second.xml

### Код

В методе onCreate() класса SecondActivity аналогично классу MainActivity реализуем логику работы нижнего меню (Рисунок 48).

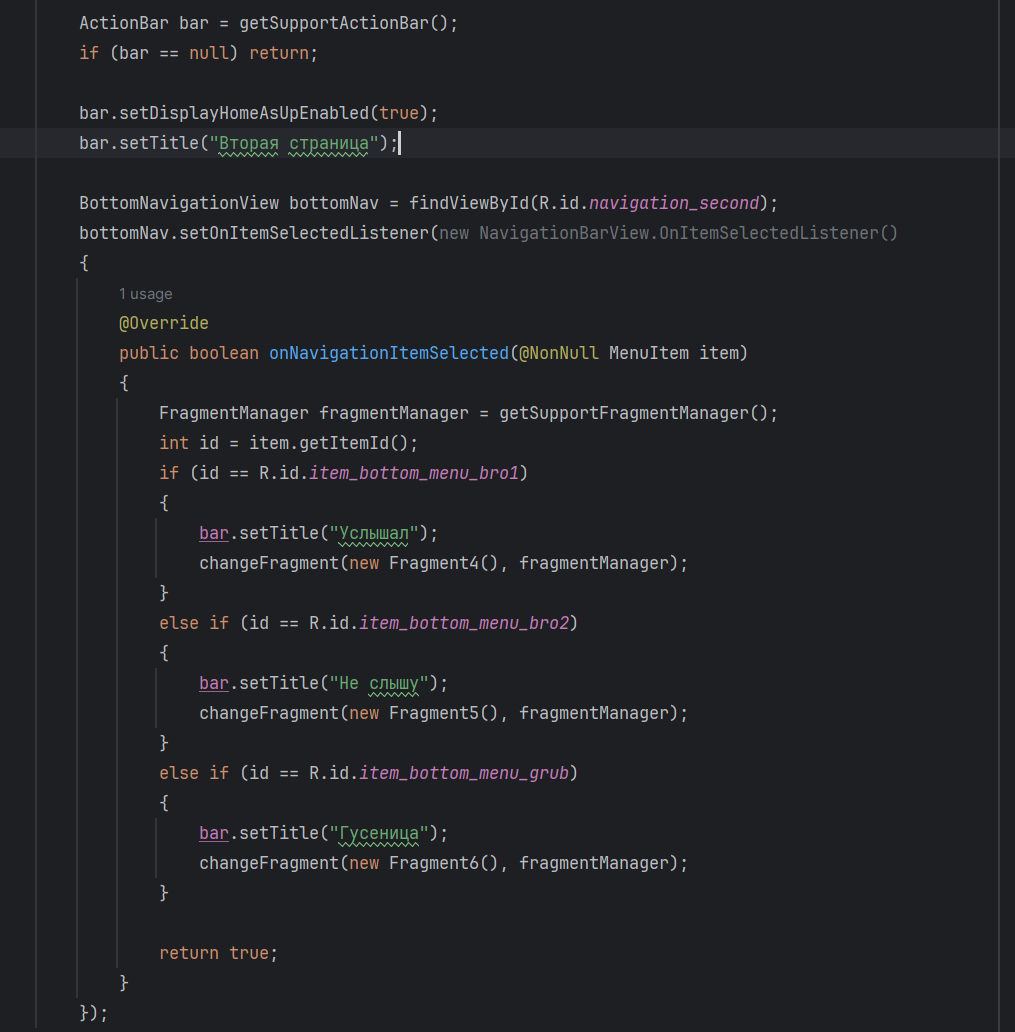


Рисунок – Описание метода onCreate() класса SecondActivity

Описание метода changeFragment() представлено на рисунке (Рисунок 49).

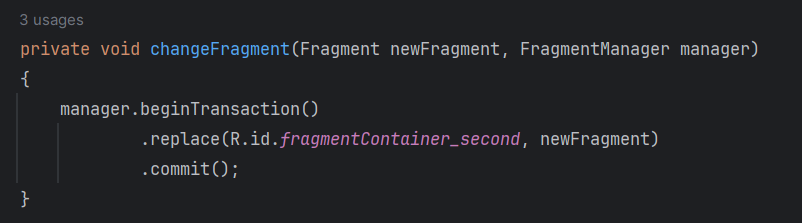


Рисунок – Описание метода changeFragment() класса SecondActivity

### Тестирование

Сначала запустим приложение и перейдем на вторую страницу с помощью последней опции бокового выпадающего меню (Рисунок 50).



Рисунок – Начальный вид второй страницы

После этого понажимаем на кнопки нижнего меню (Рисунки Рисунок 51-Рисунок 53).



Рисунок – Вторая страница после нажатия на «Услышал тебя»

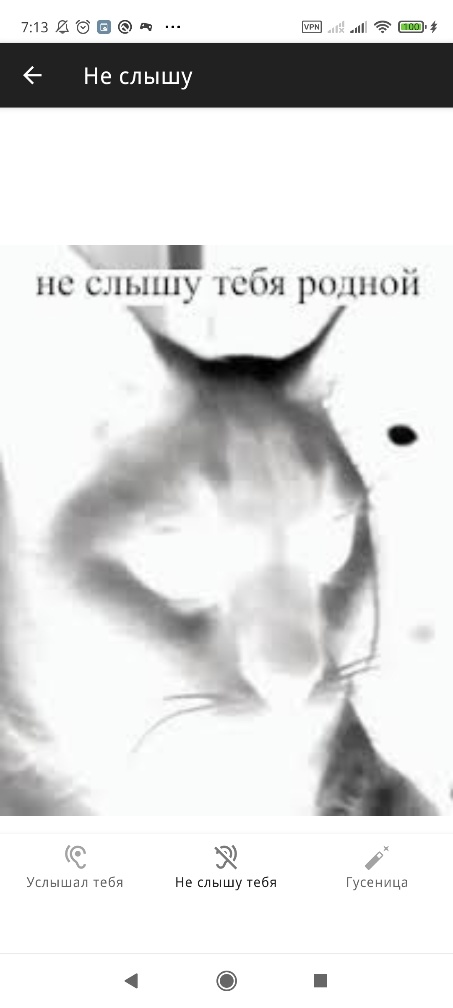


Рисунок – Вторая страница после нажатия на «Не слышу тебя»

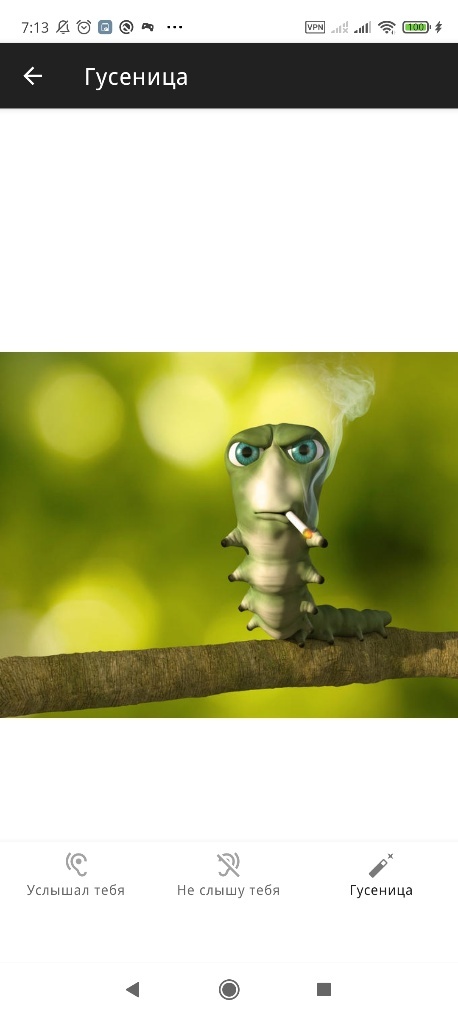


Рисунок – Вторая страница после нажатия на «Гусеница»

На панели ActionBar корректно отображается желаемое название.

ЗАКЛЮЧЕНИЕ

В ходе данной практической работы были получены знания связанные с NavigationBar, ActionBar и BottomBar. Полученные знания были закреплены путём выполнения практического задания.