|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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

[1 ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ 3](#_Toc194097368)

[1.1 ScrollView 3](#_Toc194097369)

[1.2 ListView 4](#_Toc194097370)

[1.3 Spinner 13](#_Toc194097371)

[1.4 Создание адаптера 16](#_Toc194097372)

[1.5 RecyclerView 22](#_Toc194097373)

[2 ПРАКТИЧЕСКОЕ ЗАДАНИЕ 28](#_Toc194097374)

[2.1 ListView 28](#_Toc194097375)

[2.1.1 Создание проекта 28](#_Toc194097376)

[2.1.2 Разметка 28](#_Toc194097377)

[2.1.3 Реализация логики списка 31](#_Toc194097378)

[2.1.4 Тестирование 33](#_Toc194097379)

[2.2 RecyclerView 37](#_Toc194097380)

[2.2.1 Разметка 37](#_Toc194097381)

[2.2.2 Реализация адаптера 39](#_Toc194097382)

[2.2.3 Использование в Activity 41](#_Toc194097383)

[2.2.4 Тестирование 41](#_Toc194097384)

[2.3 ScrollView 43](#_Toc194097385)

[2.4 Spinner 45](#_Toc194097386)

[ЗАКЛЮЧЕНИЕ 46](#_Toc194097387)

# 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).

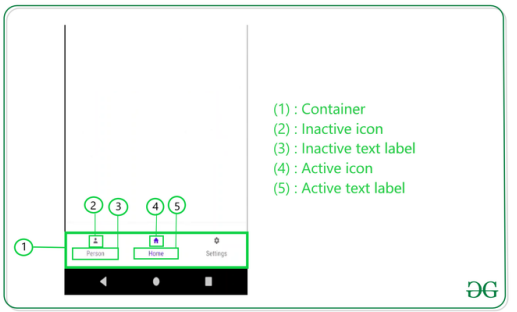


Рисунок 1 – Наполнение BottomBar

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

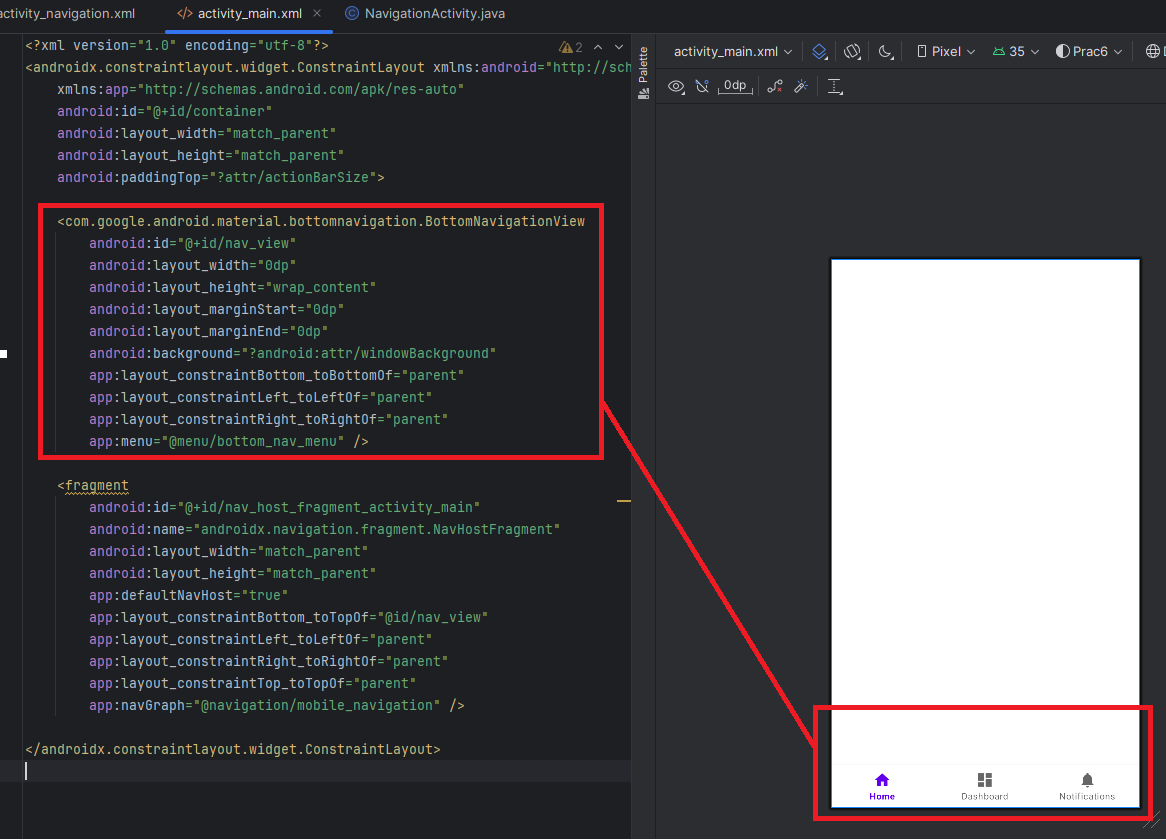


Рисунок 2 – Описание BottomNavigationView в файле разметки

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).

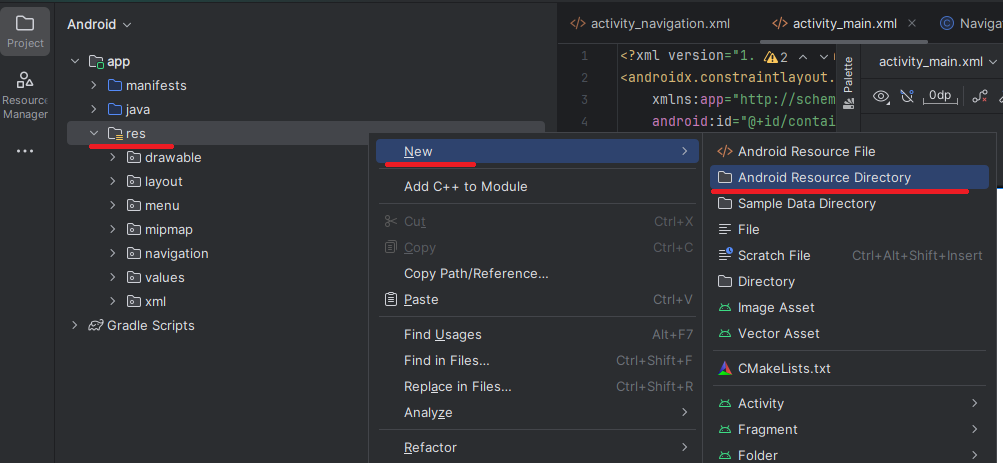


Рисунок 3 – Создание новой папки в проекте

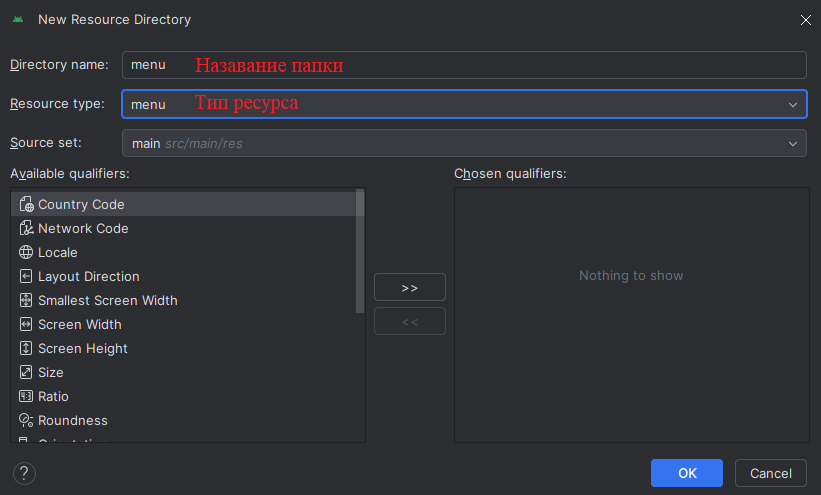


Рисунок 4 – Настройка создаваемой папки

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

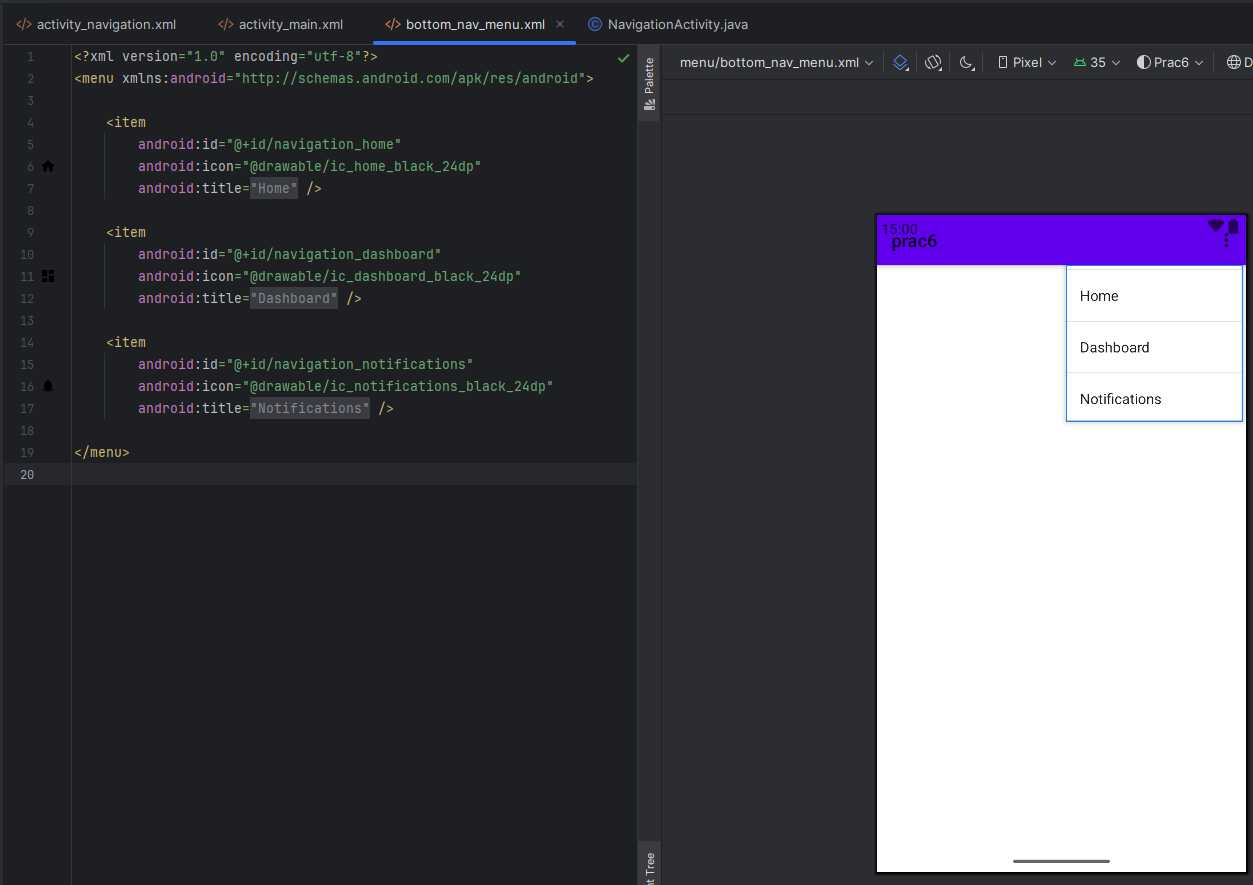


Рисунок 5 – Графическое и кодовое представление меню навигации

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).

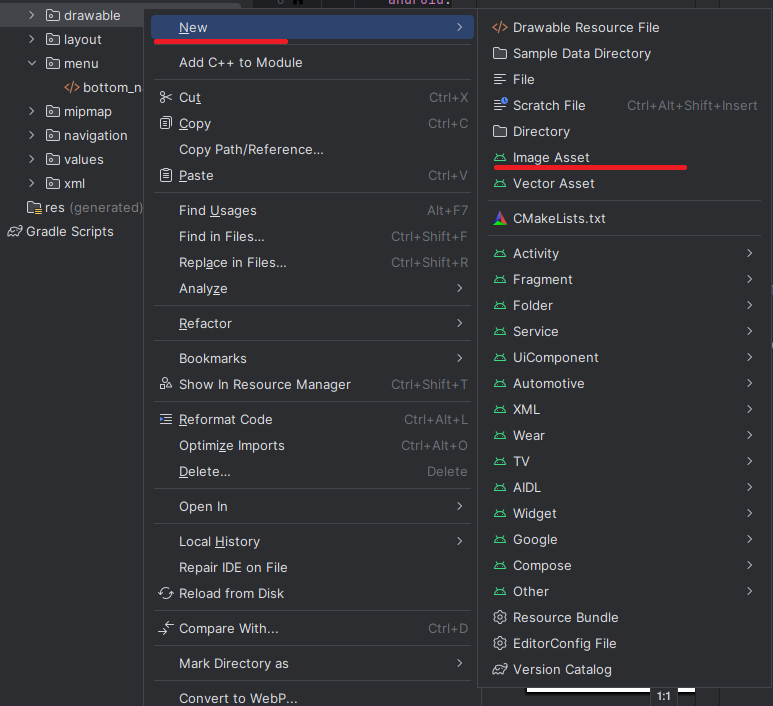


Рисунок 6 – Создание новой иконки

Далее идёт выбор нужного типа иконки (Рисунок 7).

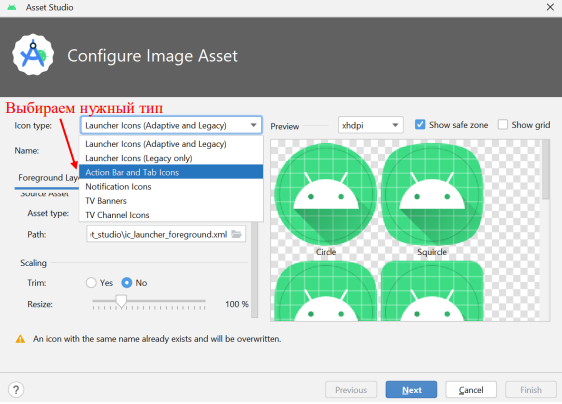


Рисунок 7 – Выбор необходимого типа иконки

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).

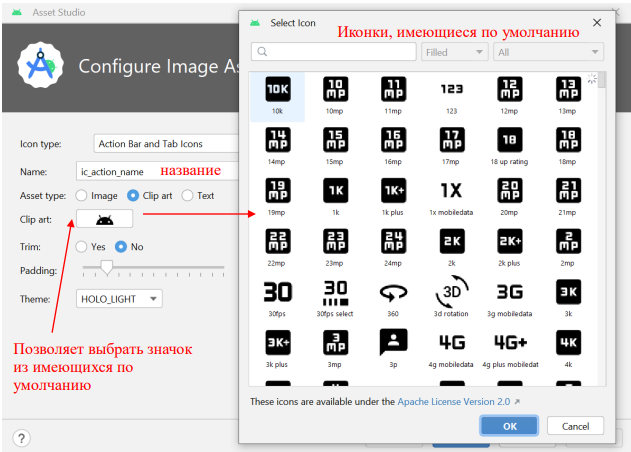


Рисунок 8 – Настройка иконки, часть 1



Рисунок 9 – Настройка иконки, часть 2

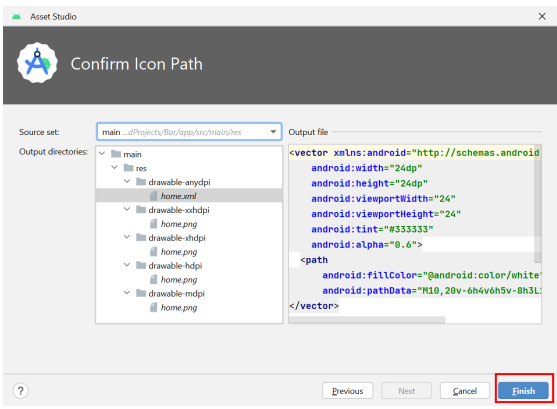


Рисунок 10 – Настройка иконки, часть 3

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).

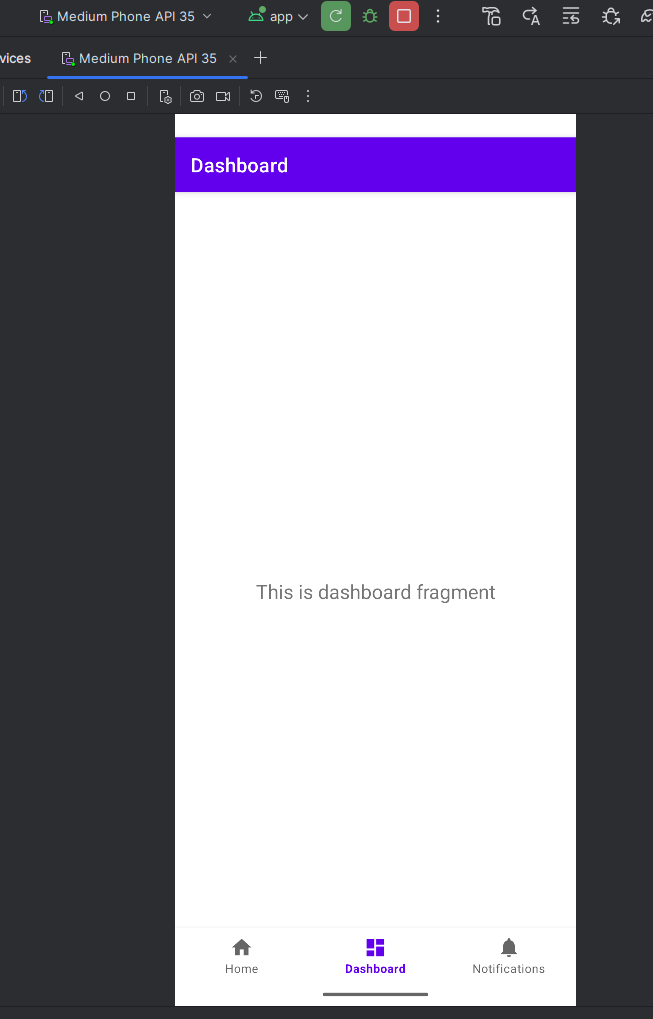


Рисунок 11 – Отображение нижнего меню навигации

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).

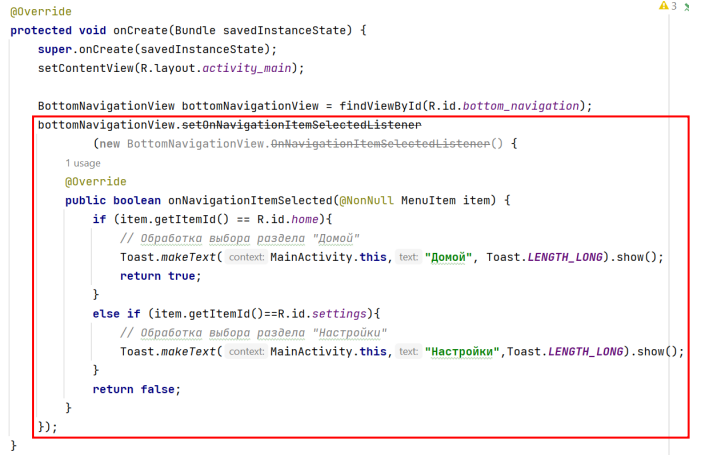


Рисунок 12 – Реализация логики обработчика событий меню навигации

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).

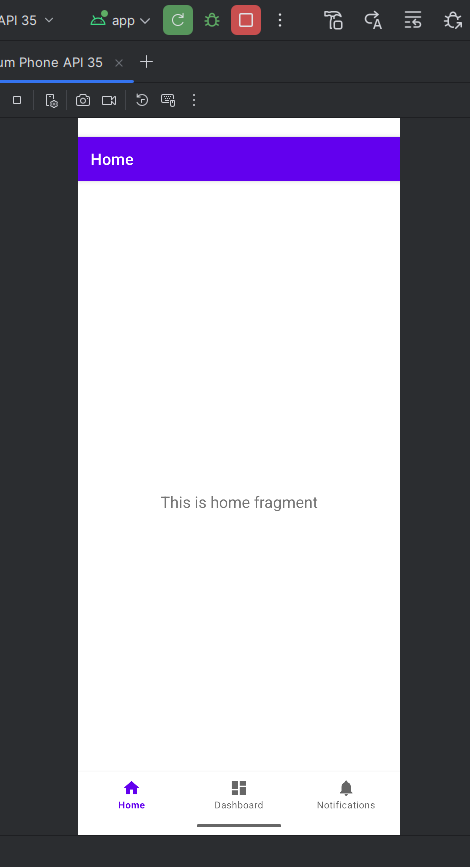


Рисунок 13 – Отображение результата возможных событий, часть 1

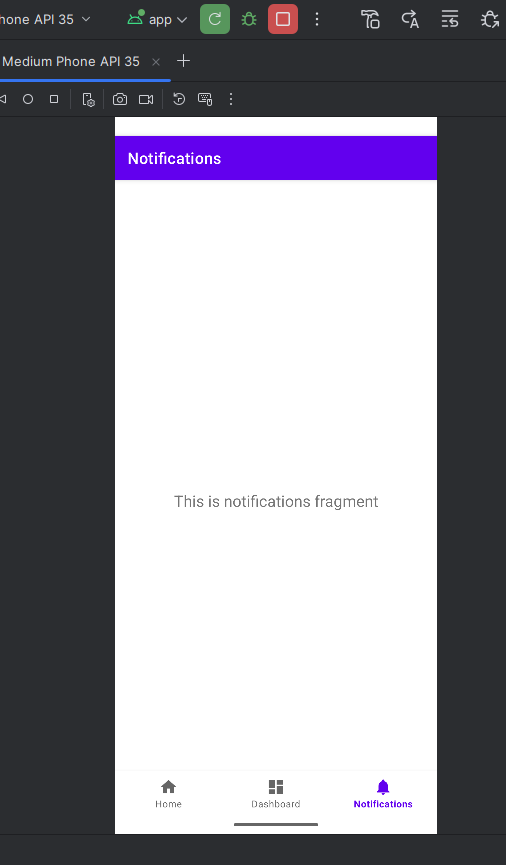


Рисунок 14 – Отображение результата возможных событий, часть 2

## 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).

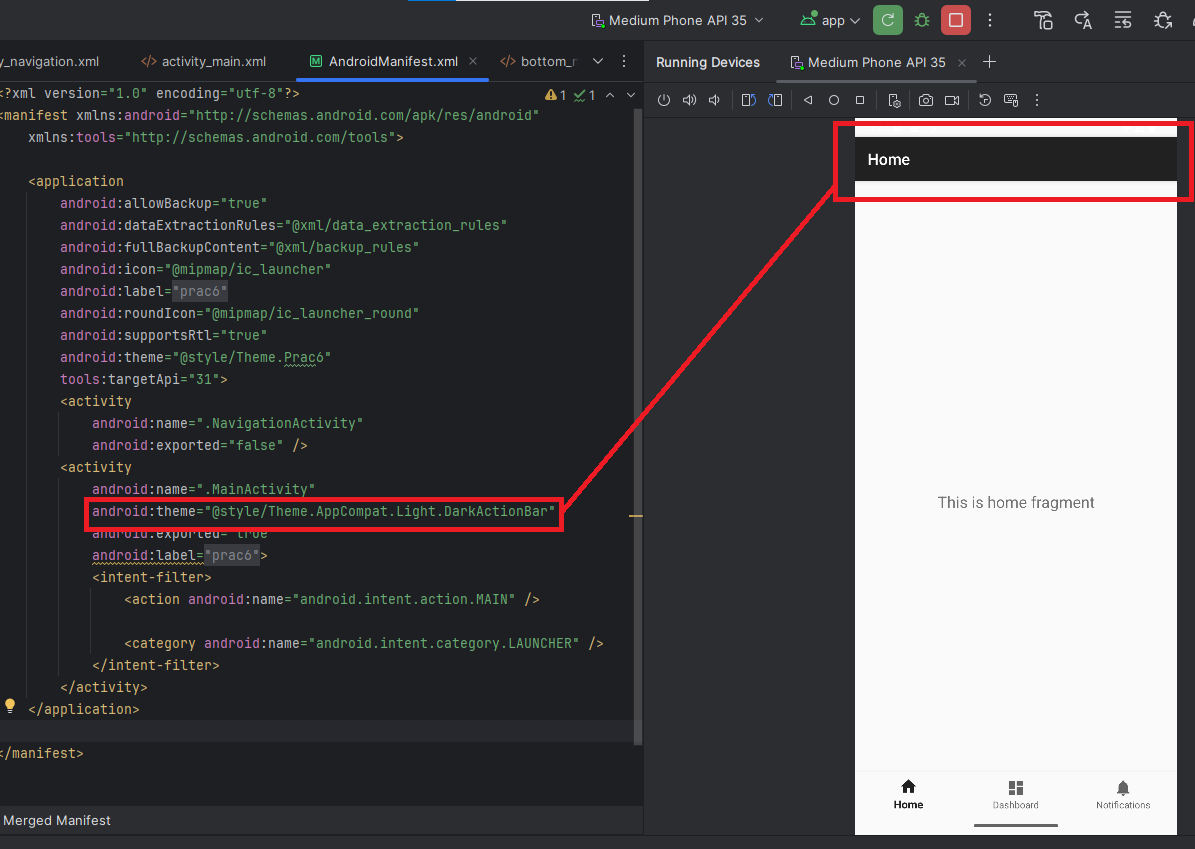


Рисунок 15 – Графическое и кодовое представление ActionBar

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).

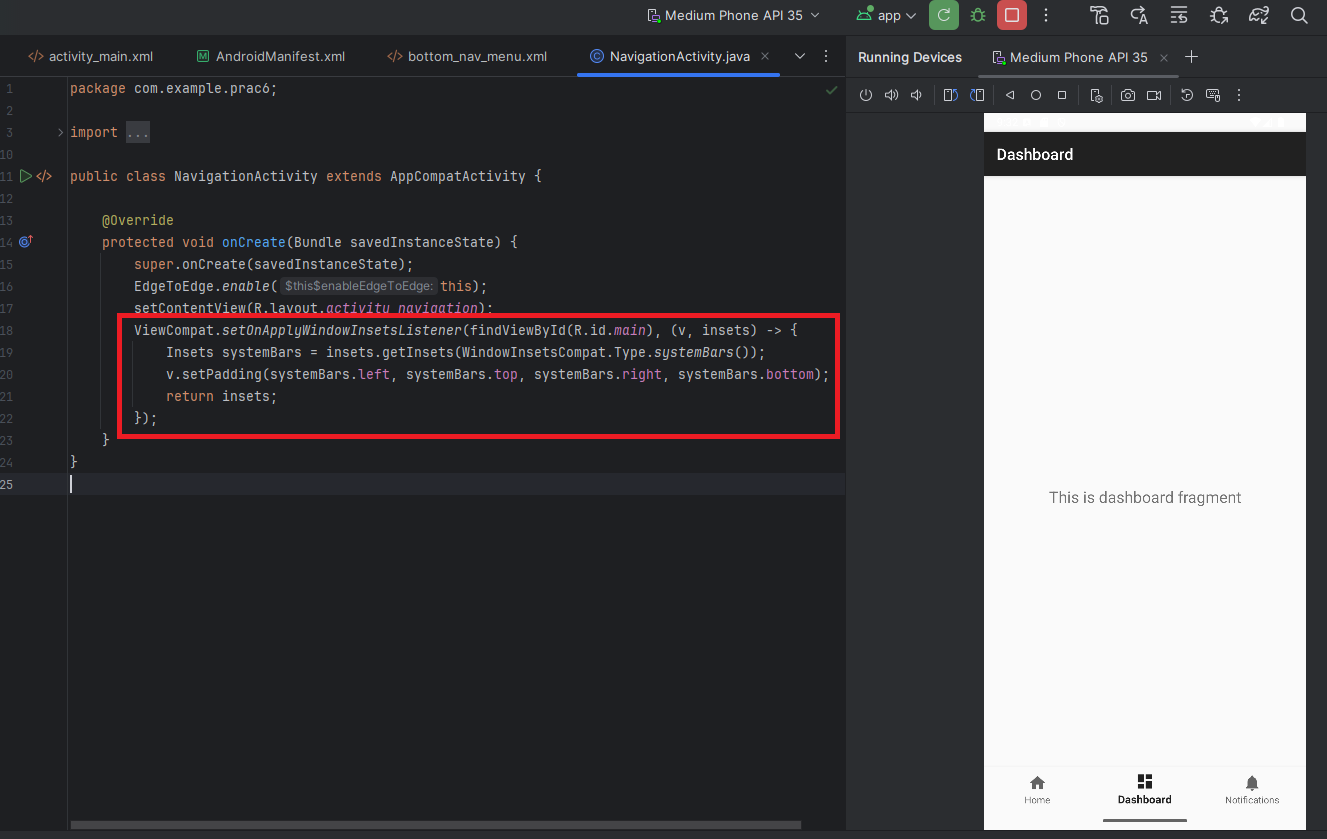


Рисунок 16 – Реализация логики обработки событий на ActionBoard

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).

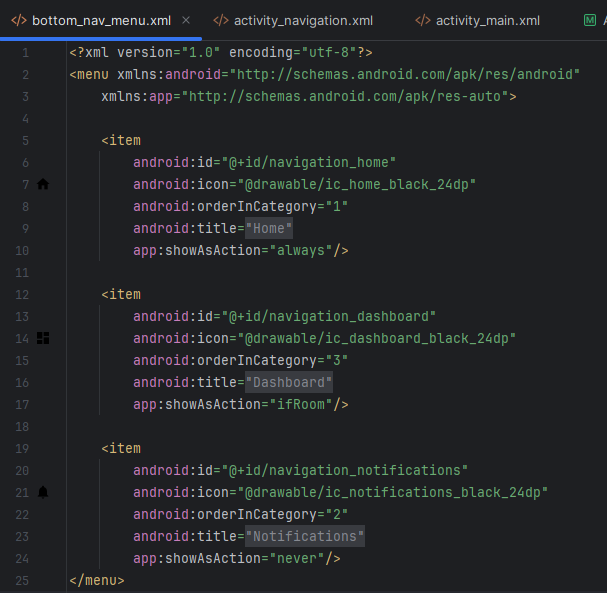


Рисунок 17 – Измененный файл разметки bottom\_nav\_menu.xml

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).

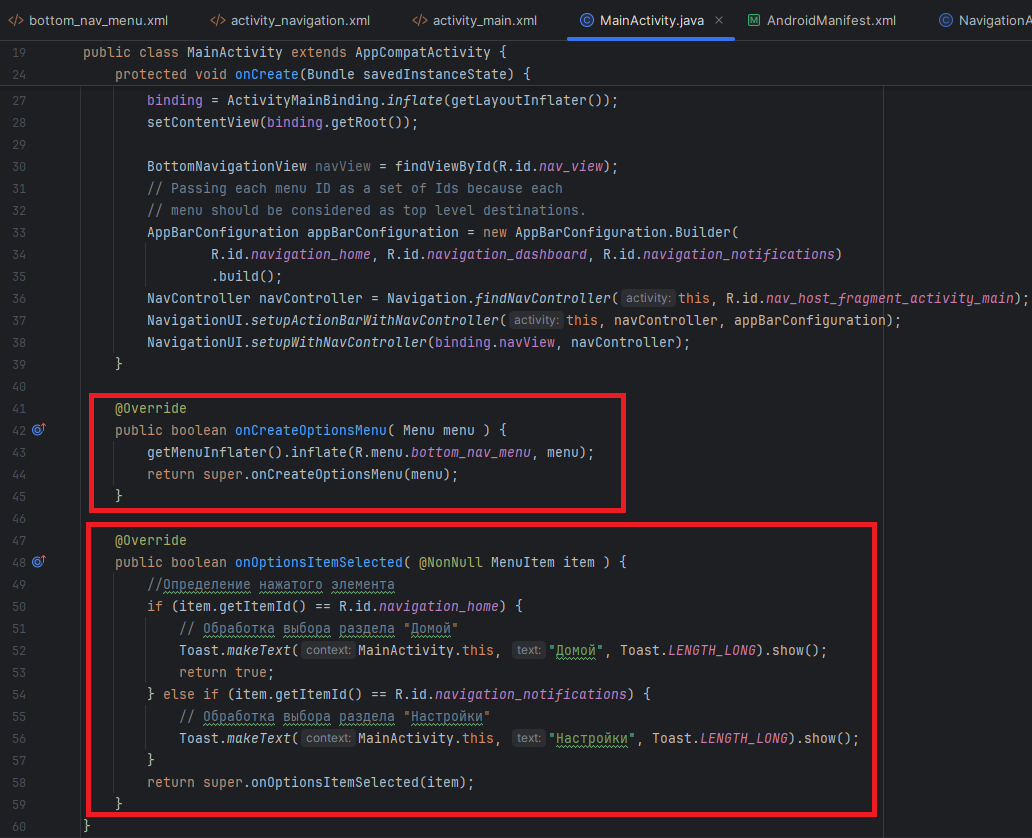


Рисунок 18 – Переопределение необходимых методов класса ActionBar

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

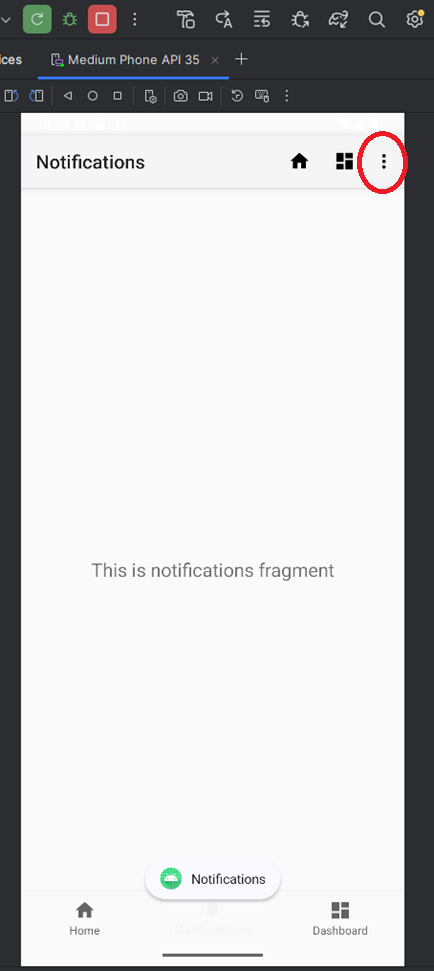


Рисунок 19 – Отображение ActionBar в приложении

## 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).

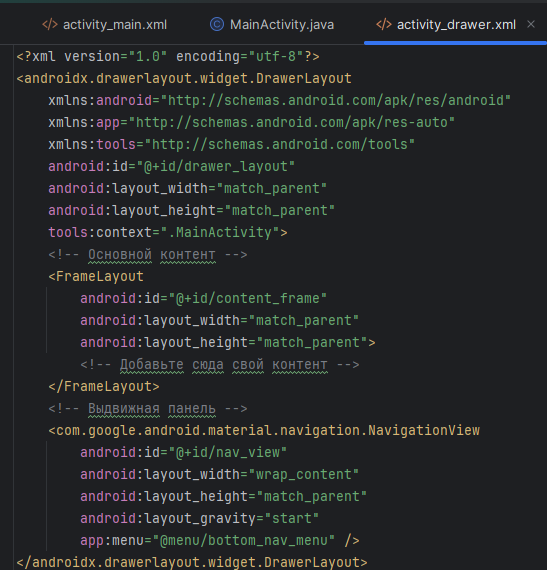


Рисунок 20 – Код файла разметки activity\_drawer.xml

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

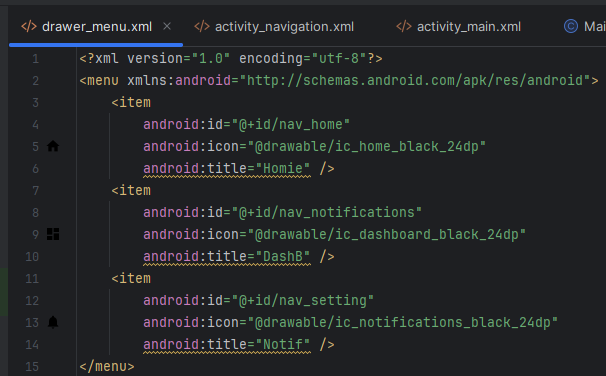


Рисунок 21 – Описание элементов в файле drawer\_menu.xml

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).

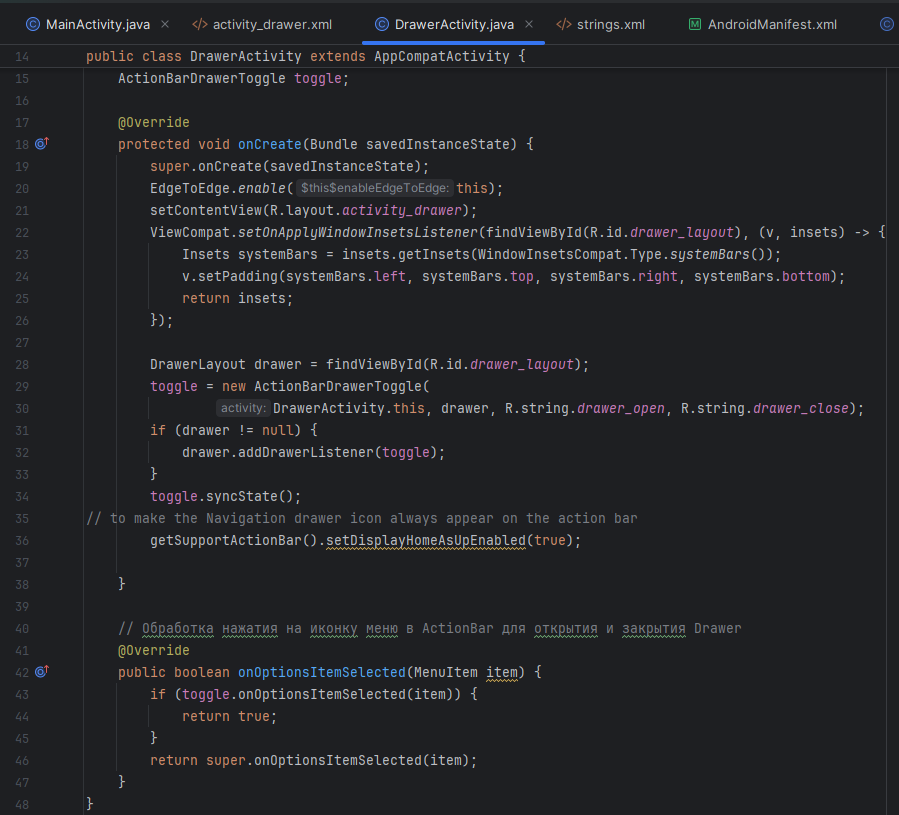


Рисунок 22 – Реализация логики NavigationDrawer

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 23 – 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).

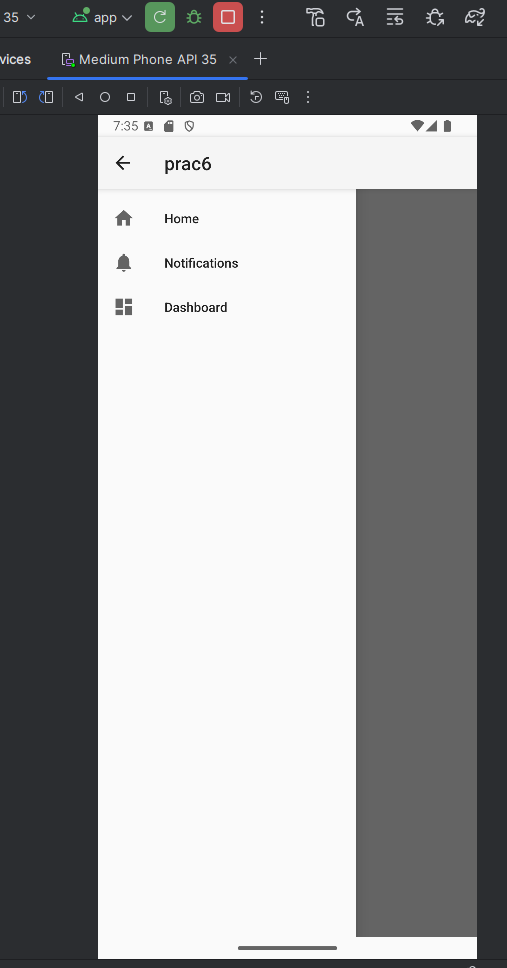


Рисунок 24 – Отображение названия всплывающего меню, часть 2

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).

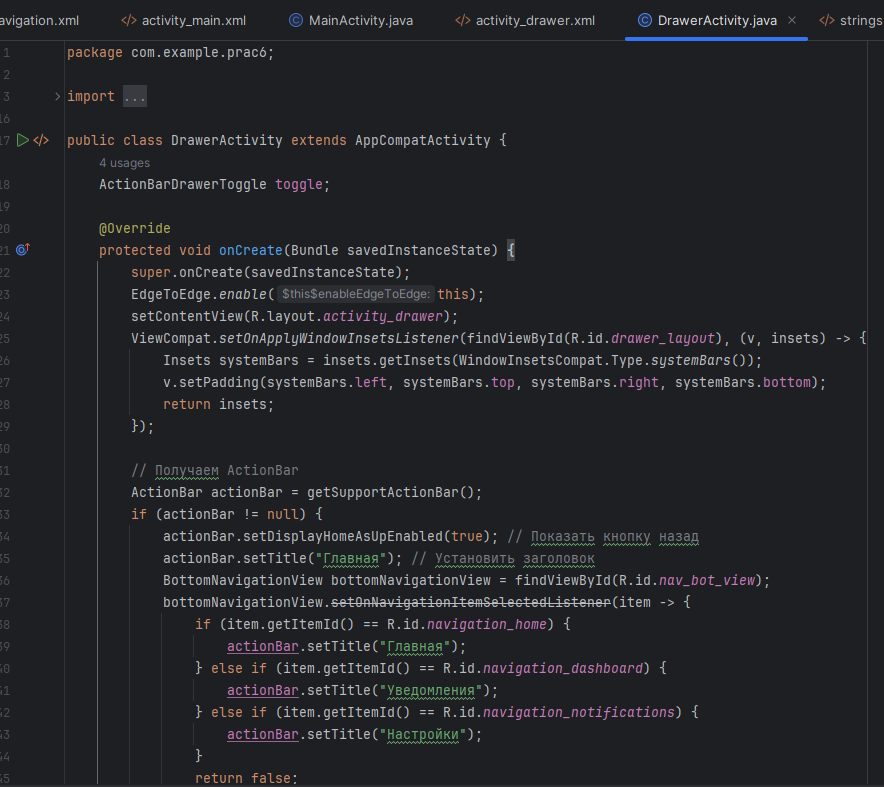


Рисунок 25 – Реализация логики BottomNavigationView, часть 1

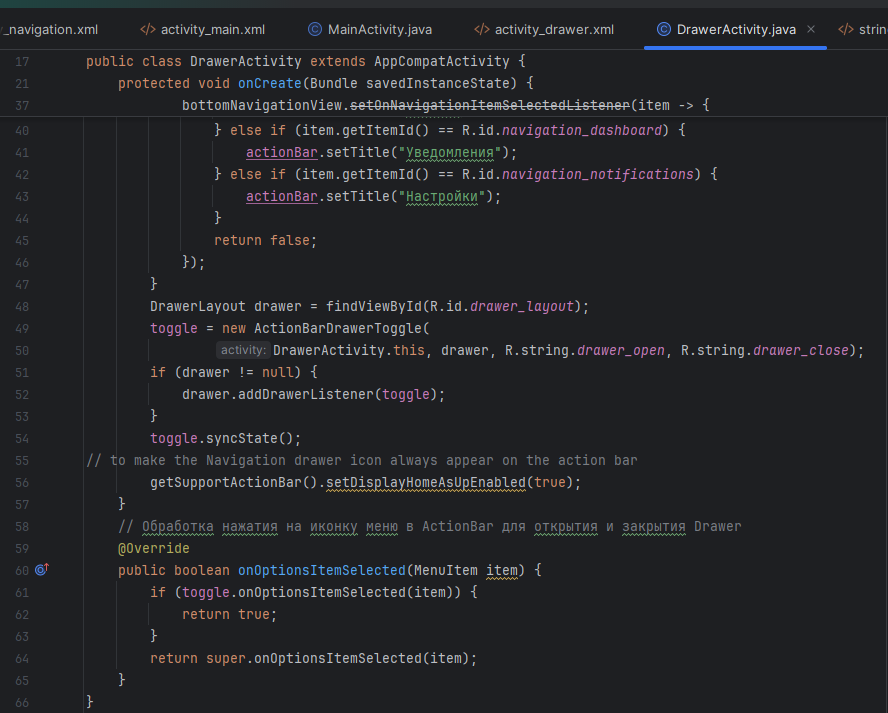


Рисунок 26 – Реализация логики BottomNavigationView, часть 2

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

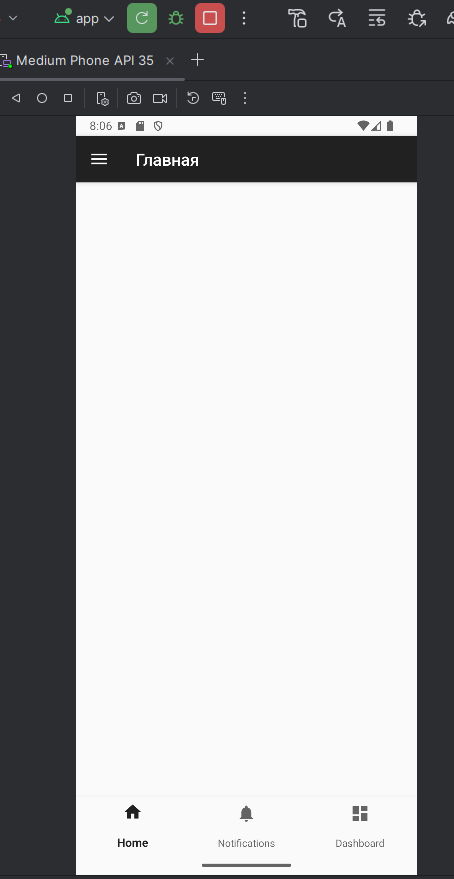


Рисунок 27 – Работа панели навигации, часть 1

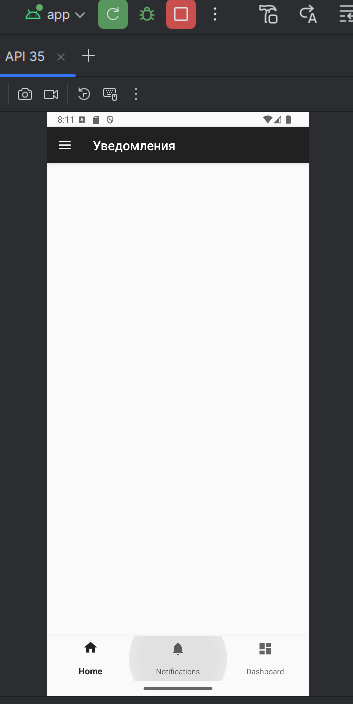


Рисунок 28 – Работа панели навигации, часть 2

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

ЗАКЛЮЧЕНИЕ

В ходе проделанной работы было проведено ознакомление с ListView и RecyclerView, а также другими инструментами отображения списков. Полученные знания были закреплены путём выполнения практического задания.