Часть 1. 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:

1. 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.

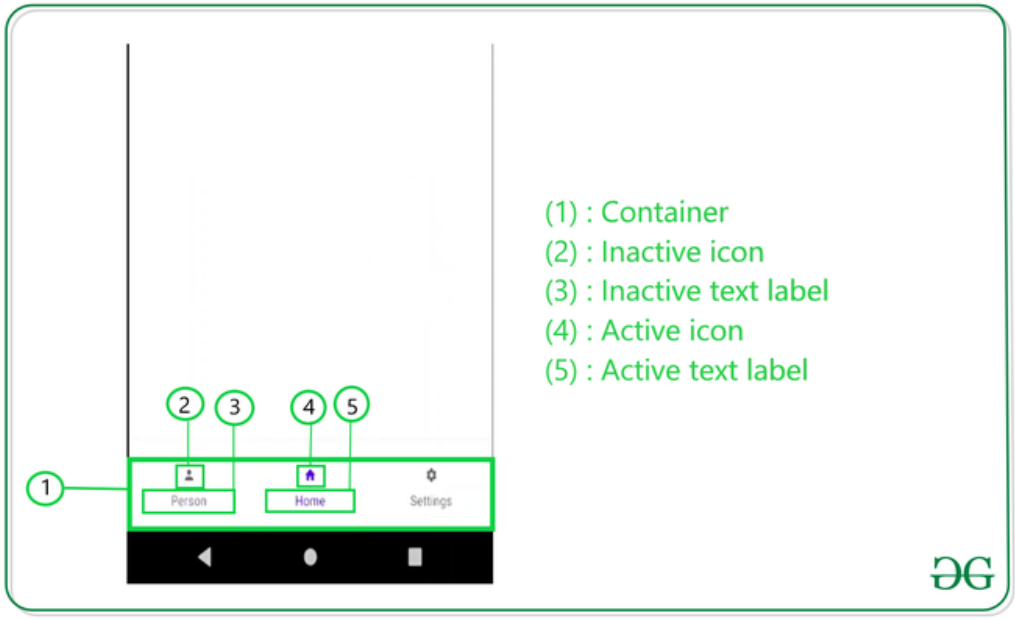
2. Повышение производительности

Использование BottomBar позволяет сократить количество нажатий, необходимых для перехода между разделами приложения, тем самым ускоряя взаимодействие пользователя с приложением и повышая его общую производительность.

3. 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.



Для использования BottomNavigationView необходимо добавить этот элемент в макет активности.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_navigation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    app:menu="@menu/bottom_nav_menu" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

CoordinatorLayout – это контейнер, который расширяет FrameLayout.

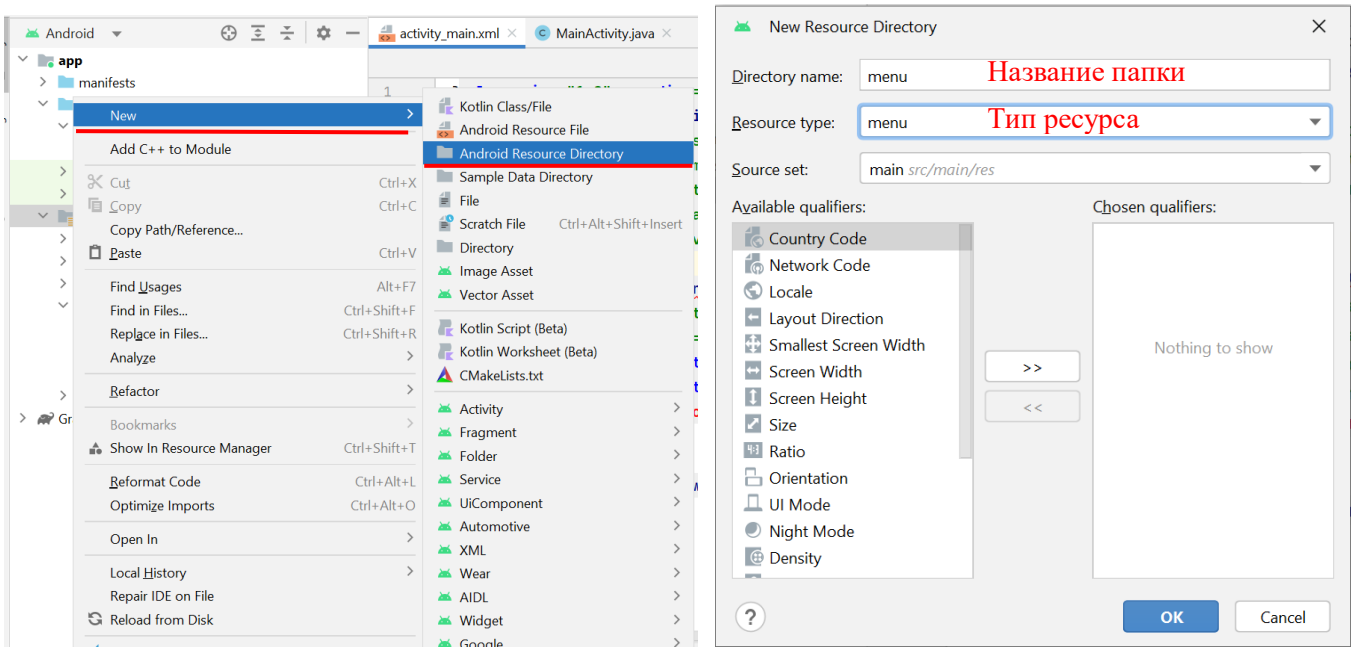
CoordinatorLayout предназначен для двух основных вариантов использования:

1. В качестве декора приложения верхнего уровня или макета Chrome.

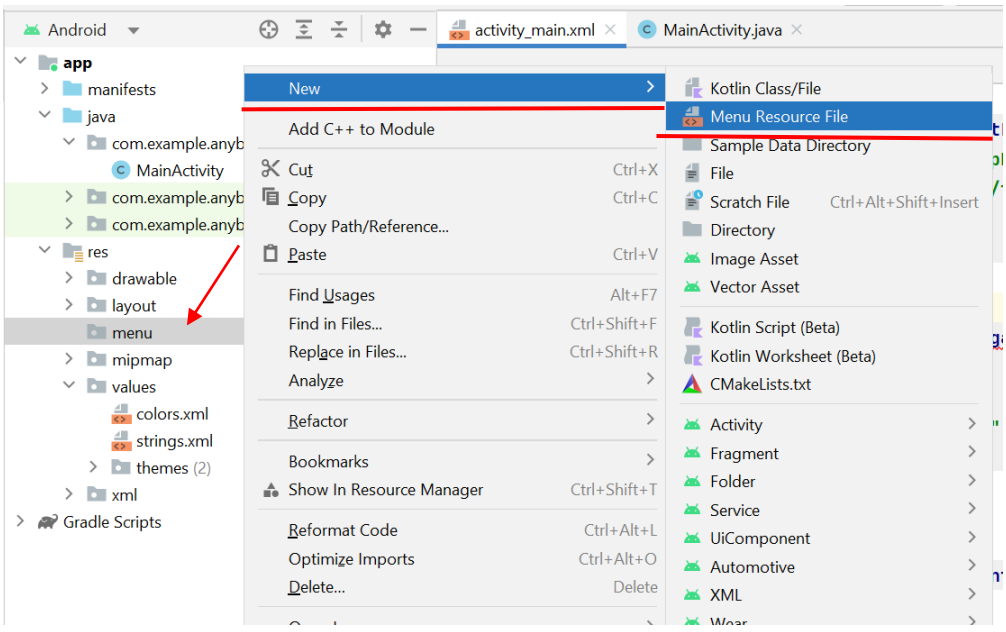
2. Как контейнер для определенного взаимодействия с одним или несколькими дочерними представлениями.

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"**. Вводим название папки, выбираем тип ресурса из выпадающего списка.



После того, как папка создана в нее необходимо добавить XML-файлы.



После создания XML-файла его можно наполнить пунктами меню.

```
<?xml version="1.0" encoding="utf-8"?>
    <menu xmlns:android="http://schemas.android.com/apk/res/android">
        <item
            android:id="@+id/home"
            android:icon="@drawable/home"
            android:title="Главная" />

        <item
            android:id="@+id/notification"
            android:icon="@drawable/notifications"
            android:title="Уведомления" />

        <item
            android:id="@+id/settings"
            android:icon="@drawable/settings"
            android:title="Настройки" />
    </menu>
```

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

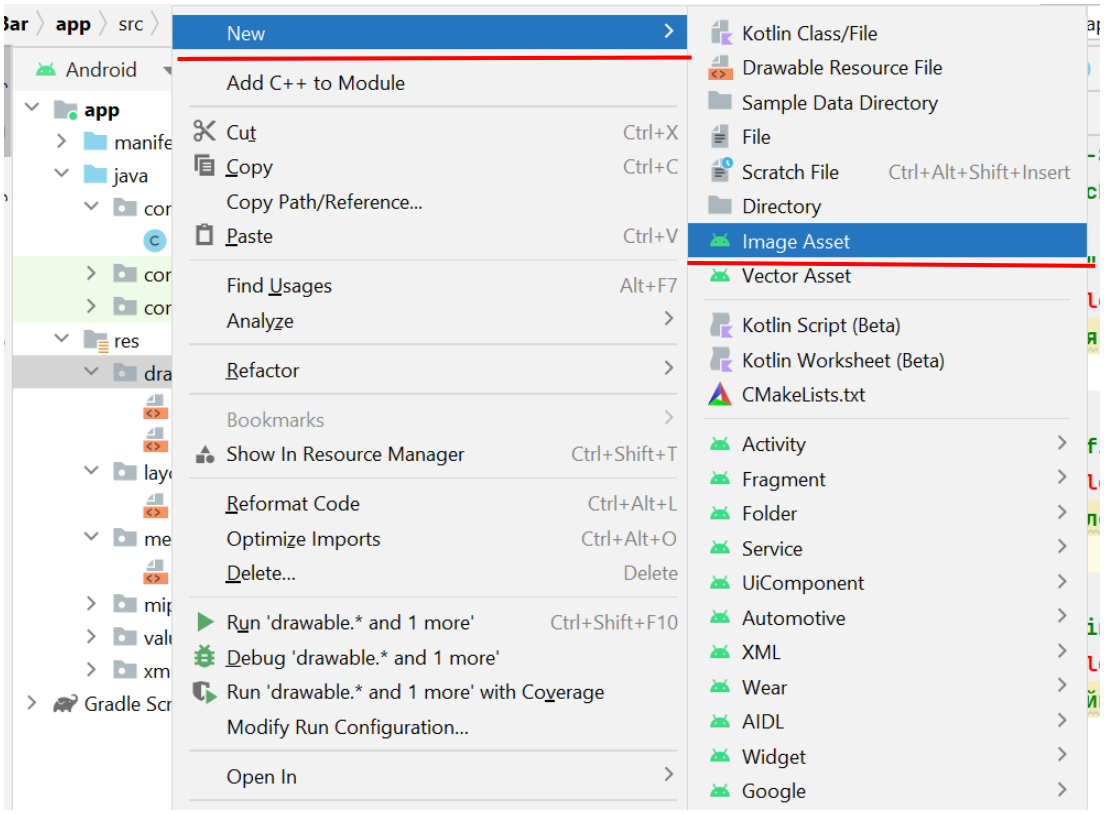
Элемент **<item>** представляет объект MenuItem, которой является одним из элементов меню. Этот элемент может содержать внутренний подэлемент <menu>, с помощью которого создается подменю.

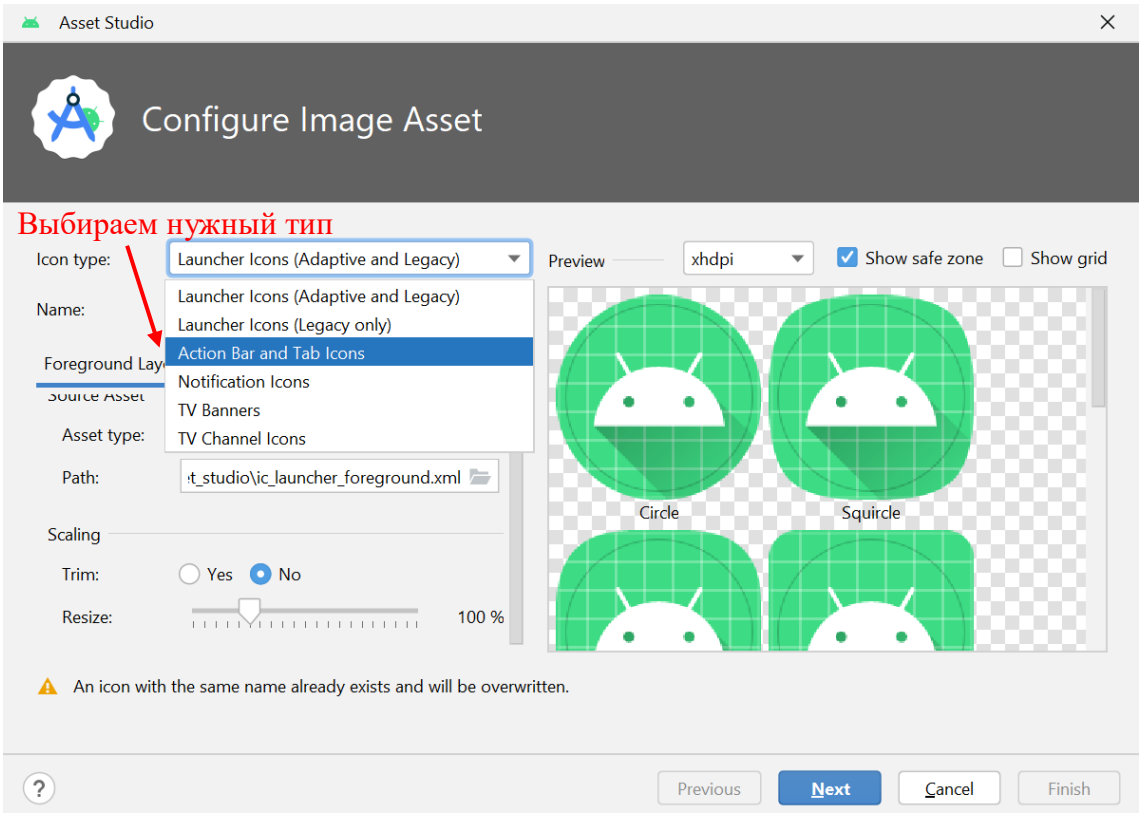
Элемент **<item>** включает следующие атрибуты, которые определяют его внешний вид и поведение:

- **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

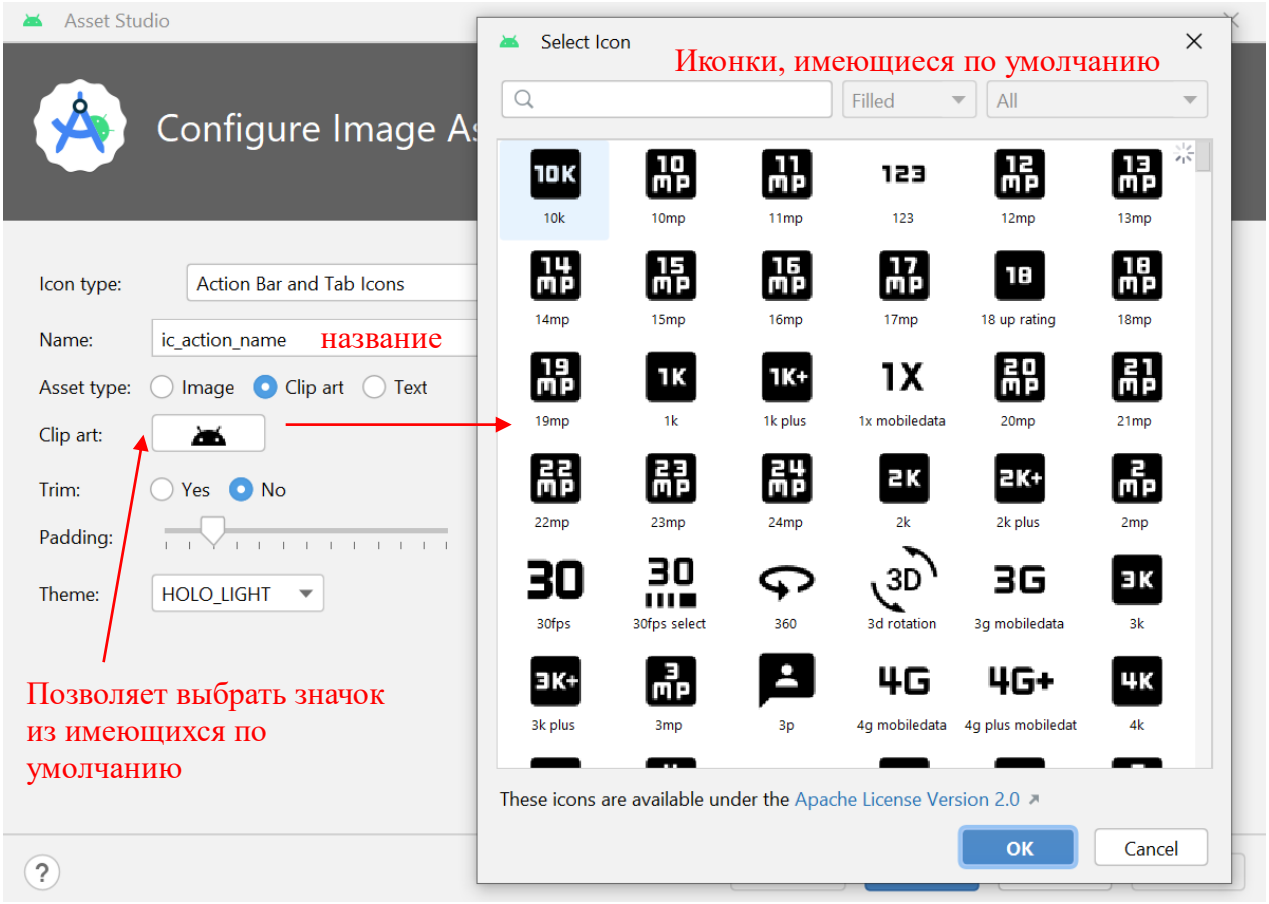
отображения каждого из этих элементов. Чтобы создать значок, нажмите на **"drawable"→ "New" → "Image Asset".**

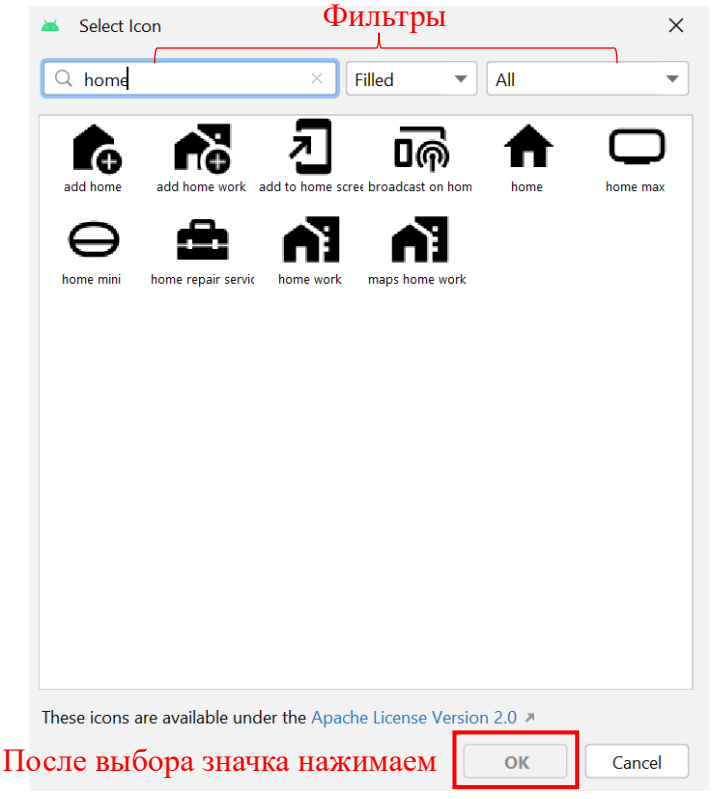


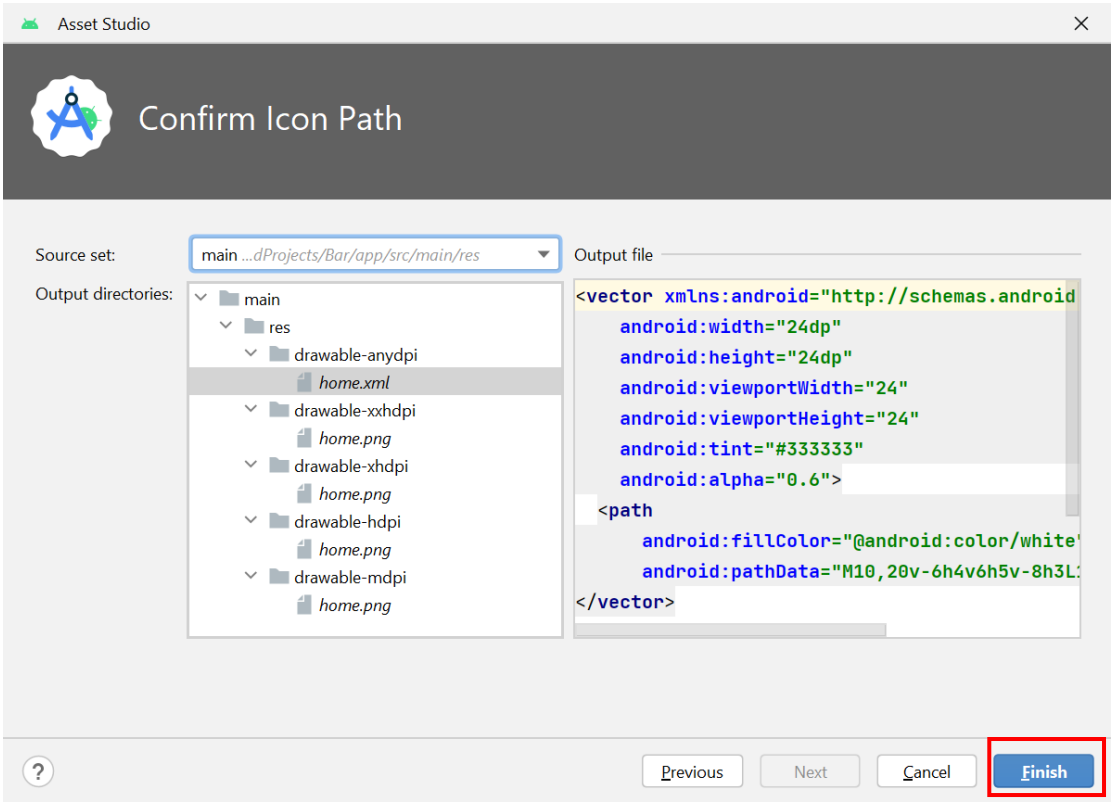


В открывшемся окне пользователь может назвать значок как угодно, но **он не должен содержать ни одной заглавной буквы**. Пользователь может выбрать

нужный значок, выполнив поиск по нему, а когда пользователь закончит, то необходимо нажать **"Next"→ "Finish"**







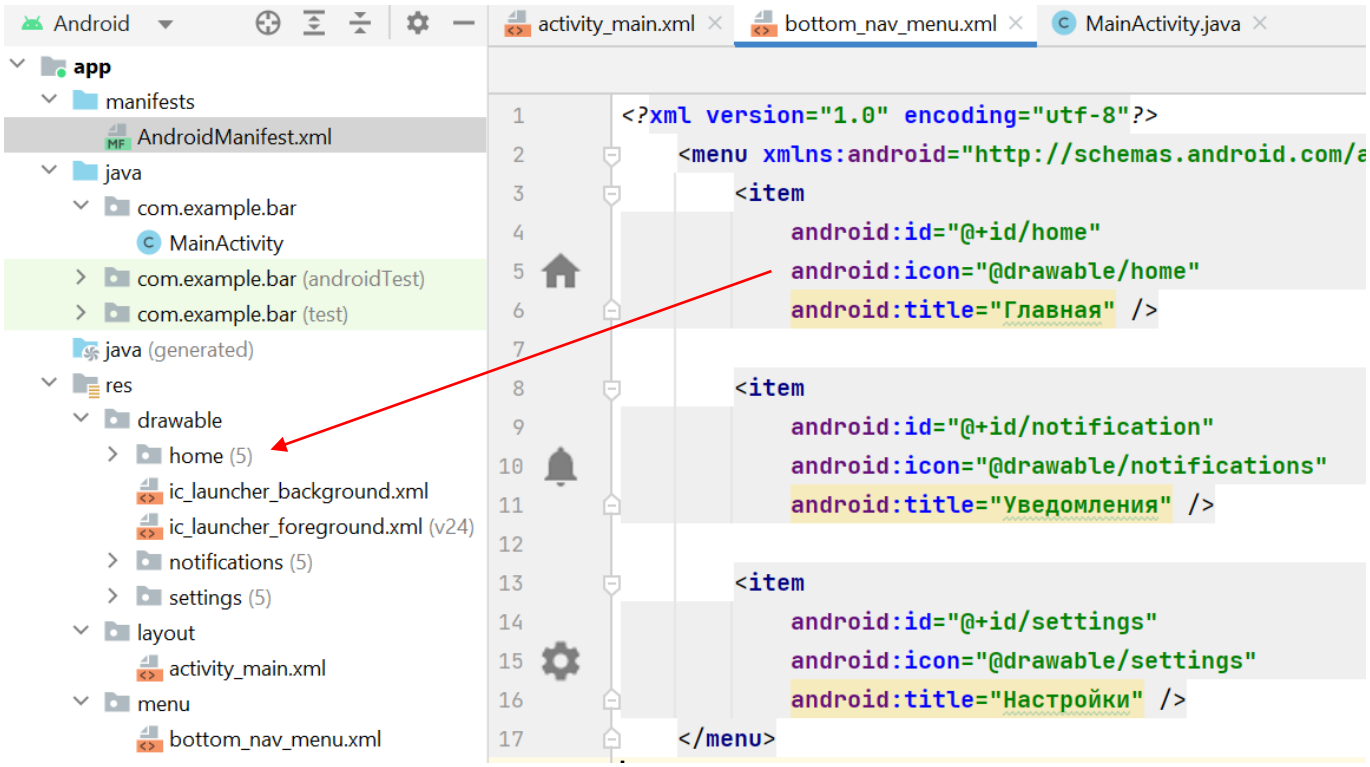
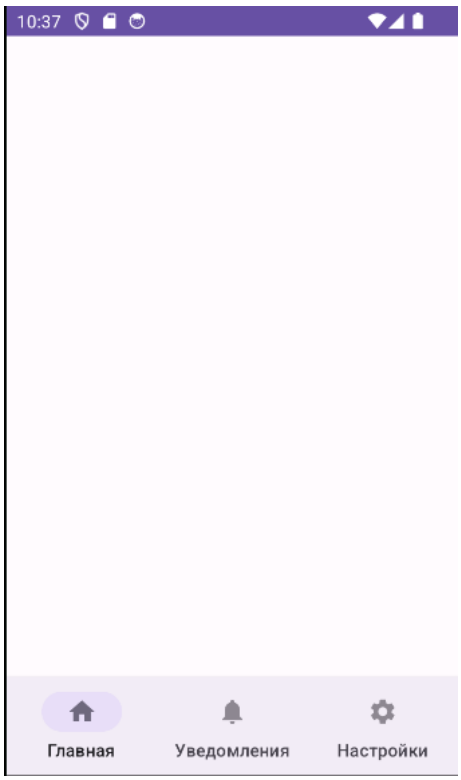


Image Asset поможет создать следующие типы значков:

- Иконки лаунчера
- Значки панели действий и вкладок
- Значки уведомлений

Если запустить сборку проекта, то можно получить следующие отображение:



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.

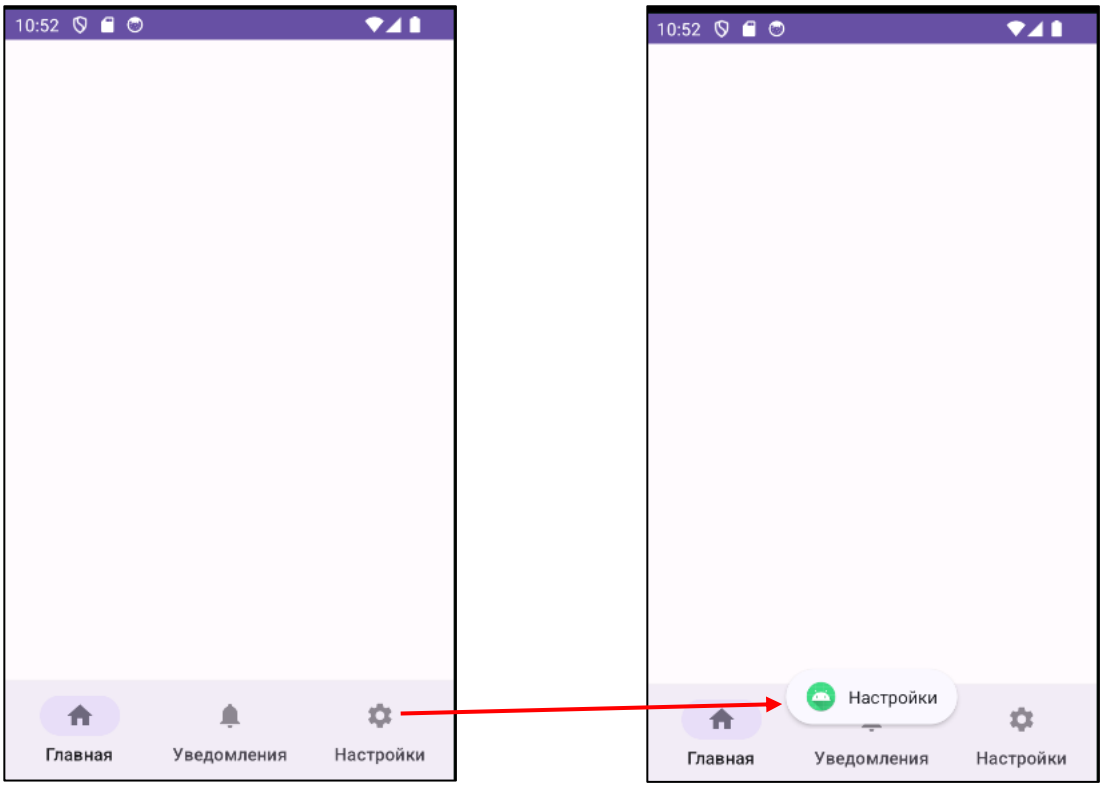
```
BottomNavigationView bottomNavigationView =
findViewById(R.id.bottom_navigation);
bottomNavigationView.setOnNavigationItemSelectedListener
        (new BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        if (item.getItemId() == R.id.home){
            // Обработка выбора раздела "Домой"
            Toast.makeText(MainActivity.this,"Домой",
Toast.LENGTH_LONG).show();
            return true;
        }
        else if (item.getItemId()==R.id.settings){
            // Обработка выбора раздела "Настройки"
            Toast.makeText(MainActivity.this,"Настройки",
Toast.LENGTH_LONG).show();
        }
        return false;
```

```
    }
});
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
    bottomNavigationView.setOnNavigationItemSelectedListener
            (new BottomNavigationView.OnNavigationItemSelectedListener() {
        1 usage
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            if (item.getItemId() == R.id.home){
                // Обработка выбора раздела "Домой"
                Toast.makeText( context: MainActivity.this, text: "Домой", Toast.LENGTH_LONG).show();
                return true;
            }
            else if (item.getItemId()==R.id.settings){
                // Обработка выбора раздела "Настройки"
                Toast.makeText( context: MainActivity.this, text: "Настройки",Toast.LENGTH_LONG).show();
            }
            return false;
        }
    });
}
```

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 – в данном случае выводим всплывающее сообщение.



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.

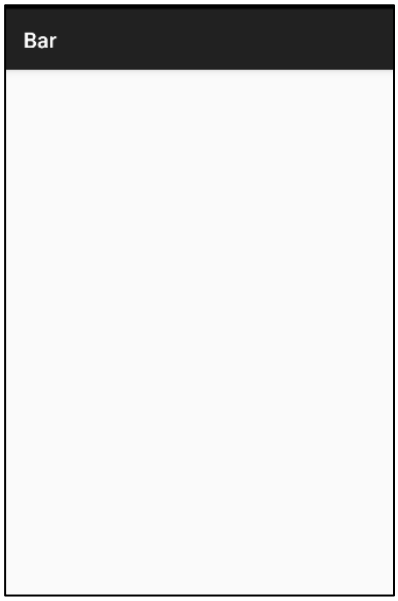
ActionBar уже встроен в стандартные темы Activity начиная с API Level 11 (Honeycomb). Для его использования убедитесь, что ваша Activity наследуется от AppCompatActivity, и используйте одну из тем Theme.AppCompat.

```
<activity
    android:name=".MainActivity"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
    <!-- Интент-фильтры и другие настройки -->
</activity>
```

```
<activity
    android:name=".MainActivity"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Теперь ActionBar будет отображаться вверху страницы.

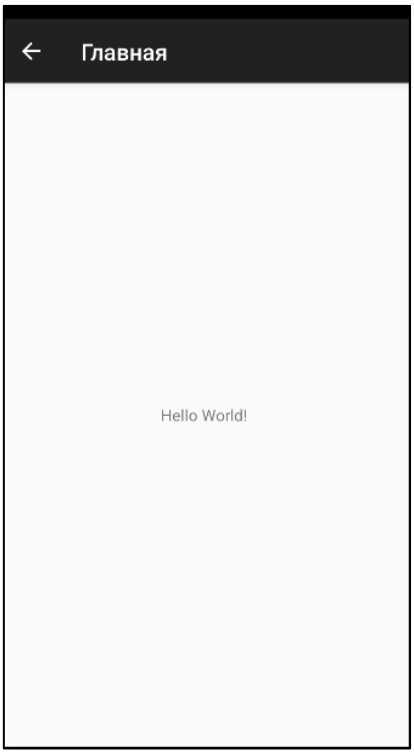


Однако такой ActionBar не несет никакой пользы. Для дальнейшей работы его необходимо настроить. В вашей активности вы можете настроить ActionBar, используя следующий код:

```
    // Получаем ActionBar
        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            // Настройки ActionBar
            actionBar.setDisplayHomeAsUpEnabled(true);   //   Показать
кнопку назад
            actionBar.setTitle("Главная"); // Установить заголовок
            // Другие настройки ActionBar
        }
```

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Получаем ActionBar
    ActionBar actionBar = getSupportActionBar();
    if (actionBar != null) {
        // Настройки ActionBar
        actionBar.setDisplayHomeAsUpEnabled(true); // Показать кнопку назад
        actionBar.setTitle("Главная"); // Установить заголовок
        // Другие настройки ActionBar
    }
}
```



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.

- **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 предлагается четыре возможных флага:

a) **always:** Постоянно отображать элемент на панели действий.

b) **ifRoom:** Сохранить элемент, если есть свободное место.

c) **never:** С этим флагом элемент не будет отображаться в виде значка в ActionBar, но будет присутствовать в меню переполнения.

d) **withText:** Чтобы представить элемент одновременно в виде значка и заголовка, можно дополнить этот флаг флагом always или ifRoom (always|withText or ifRoom|withText).

```
    <item
        android:id="@+id/home"
        android:icon="@drawable/home"
        android:orderInCategory="1"
        android:title="Главная"
        app:showAsAction="always"/>

    <item
        android:id="@+id/notification"
        android:icon="@drawable/notifications"
        android:orderInCategory="3"
        android:title="Уведомления"
        app:showAsAction="ifRoom"/>

    <item
        android:id="@+id/settings"
        android:icon="@drawable/settings"
        android:orderInCategory="2"
        android:title="Настройки"
        app:showAsAction="never"/>
</menu>
```

Мы определили меню с тремя элементами, но само определение элементов в файле еще не создает меню. Это всего лишь описание. Чтобы вывести его на экран, нам надо использовать его в классе Activity. Для этого надо переопределить метод **onCreateOptionsMenu**.

```
@Override
public boolean onCreateOptionsMenu( Menu menu ) {

    getMenuInflater().inflate(R.menu.bottom_nav_menu, menu);
    return super.onCreateOptionsMenu(menu);
}
```

Метод getMenuInflater возвращает объект MenuInflater, у которого вызывается метод inflate(). Этот метод в качестве первого параметра принимает ресурс,

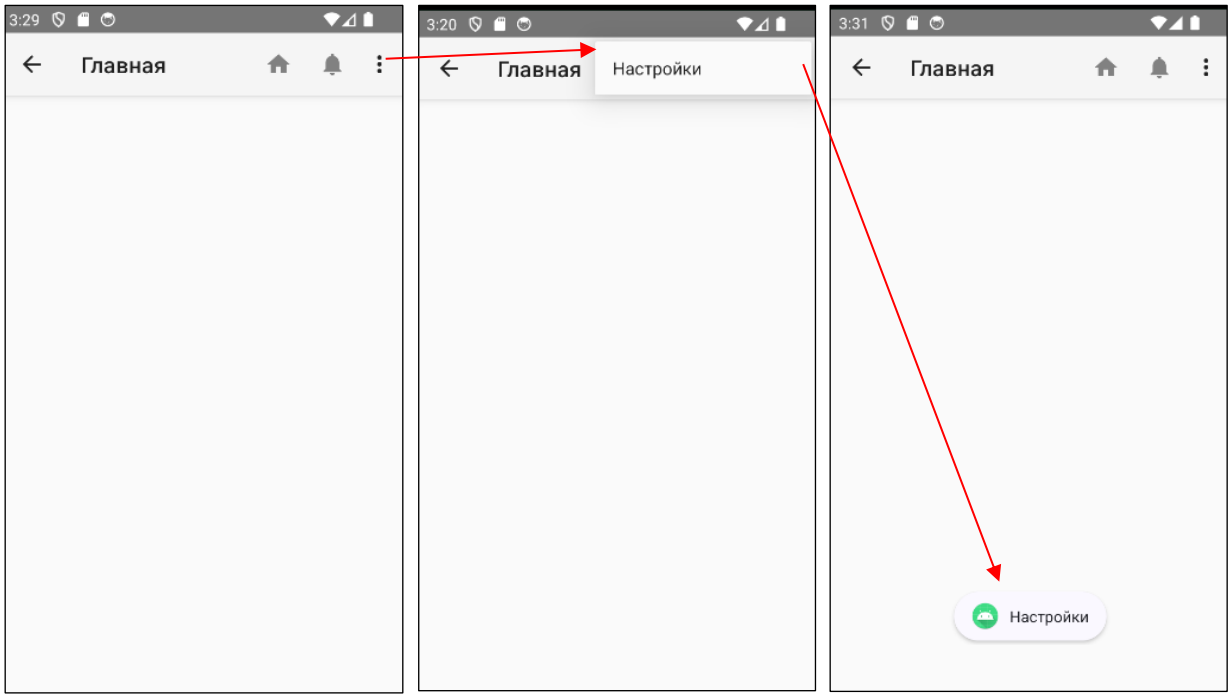
представляющий наше описание меню в xml, и наполняет им объект menu, переданный в качестве второго параметра.

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**.

```
@Override
public boolean onOptionsItemSelected( @NonNull MenuItem item ) {
   //Определение нажатого элемента
   if (item.getItemId() == R.id.home){
      // Обработка выбора раздела "Домой"
      Toast.makeText(MainActivity.this,"Домой", Toast.LENGTH_LONG).show();
      return true;
   }
   else if (item.getItemId()==R.id.settings){
      // Обработка выбора раздела "Настройки"
      Toast.makeText(MainActivity.this,"Настройки",Toast.LENGTH_LONG).show();
   }
   return super.onOptionsItemSelected(item);
```

activity_main.xml | MainActivity.java | bottom_nav_menu.xml | AndroidManifest.xml

```
34      // the user opens the menu for the first time
        6 usages
35      @Override
36      public boolean onCreateOptionsMenu( Menu menu ) {
37
38          getMenuInflater().inflate(R.menu.bottom_nav_menu, menu);
39          return super.onCreateOptionsMenu(menu);
40      }
        4 usages
41      @Override
42      public boolean onOptionsItemSelected( @NonNull MenuItem item ) {
43          //Обработка нажатия кнопок меню
44          if (item.getItemId() == R.id.home){
45              // Обработка выбора раздела "Домой"
46              Toast.makeText( context: MainActivity.this, text: "Домой", Toast.LENGTH_LONG).show();
47              return true;
48          }
49          else if (item.getItemId()==R.id.settings){
50              // Обработка выбора раздела "Настройки"
51              Toast.makeText( context: MainActivity.this, text: "Настройки",Toast.LENGTH_LONG).show();
52          }
53          return super.onOptionsItemSelected(item);
54      }
55  }
```



Часть 3. 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 вашей активности добавьте DrawerLayout как корневой элемент. Внутри DrawerLayout разместите ваш основной контент и NavigationView для выдвижной панели:

```
<androidx.drawerlayout.widget.DrawerLayout
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/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!-- Основной контент -->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <!-- Добавьте сюда свой контент -->
    </FrameLayout>

    <!-- Выдвижная панель -->
    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        app:menu="@menu/drawer_menu" />

</androidx.drawerlayout.widget.DrawerLayout>
```

Создайте XML-файл в папке res/menu (например, drawer_menu.xml) и добавьте пункты меню для вашей выдвижной панели.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
```

```
        android:id="@+id/nav_home"
        android:icon="@drawable/home"
        android:title="Главная" />
    <item
        android:id="@+id/nav_notifications"
        android:icon="@drawable/notifications"
        android:title="Уведомления" />
    <item
        android:id="@+id/nav_setting"
        android:icon="@drawable/settings"
        android:title="Настрйоки" />
    <!-- Добавьте дополнительные пункты здесь -->
</menu>
```

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

Для интеграции ActionBar с DrawerLayout используйте **ActionBarDrawerToggle**, который добавит иконку меню для открытия и закрытия Drawer и обеспечит анимацию иконки.

```
DrawerLayout drawer = findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        MainActivity.this, drawer, R.string.drawer_open,
R.string.drawer_close);
if (drawer != null) {
    drawer.addDrawerListener(toggle);
}
toggle.syncState();
// to make the Navigation drawer icon always appear on the action bar
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

Теперь необходимо добавить слушатель для обработки нажатия на элементы.

```
    // Обработка нажатия на иконку меню в ActionBar для открытия и
закрытия Drawer
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (toggle.onOptionsItemSelected(item)) {
            return true;
```

```
        }
        return super.onOptionsItemSelected(item);
    }
```

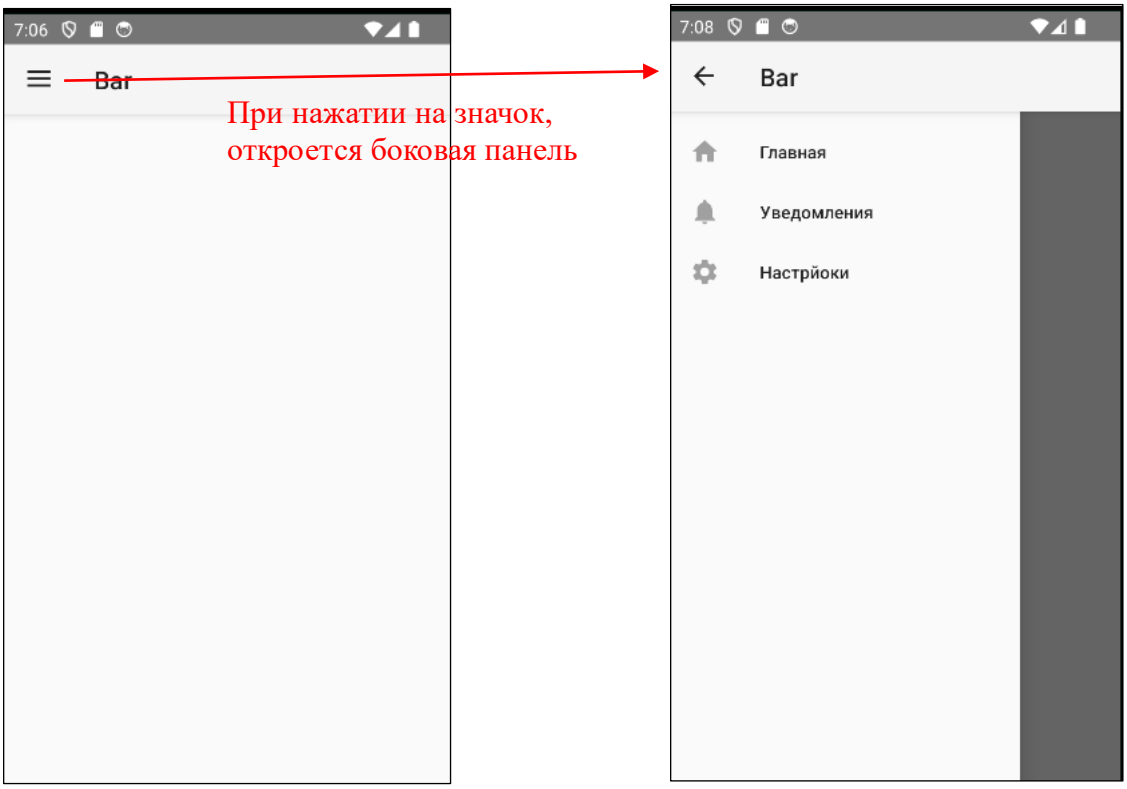
```
public DrawerLayout drawer;
4 usages
public ActionBarDrawerToggle toggle;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    drawer = findViewById(R.id.drawer_layout);
    toggle = new ActionBarDrawerToggle(
            activity: MainActivity.this, drawer, R.string.drawer_open, R.string.drawer_close);
    if (drawer != null) {
        drawer.addDrawerListener(toggle);
    }
    toggle.syncState();
    // to make the Navigation drawer icon always appear on the action bar
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
}
```

DrawerLayout

ActionBar

```
4 usages
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (toggle.onOptionsItemSelected(item)) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```



Чтобы использовать **ActionBarDrawerToggle**, нужно создать экземпляр при помощи конструктора с такими 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.

```
// Получаем ActionBar
ActionBar actionBar = getSupportActionBar();
```

```
if (actionBar != null) {
    actionBar.setDisplayHomeAsUpEnabled(true); // Показать кнопку
назад
    actionBar.setTitle("Главная"); // Установить заголовок

    BottomNavigationView bottomNavigationView =
findViewById(R.id.bottom_navigation);
    bottomNavigationView.setOnNavigationItemSelectedListener(item -> {
        if (item.getItemId() == R.id.home) {
            actionBar.setTitle("Главная");
        } else if (item.getItemId() == R.id.notification) {
            actionBar.setTitle("Уведомления");
        } else if (item.getItemId() == R.id.settings) {
            actionBar.setTitle("Настройки");
        }
        return false;
    });
}
```

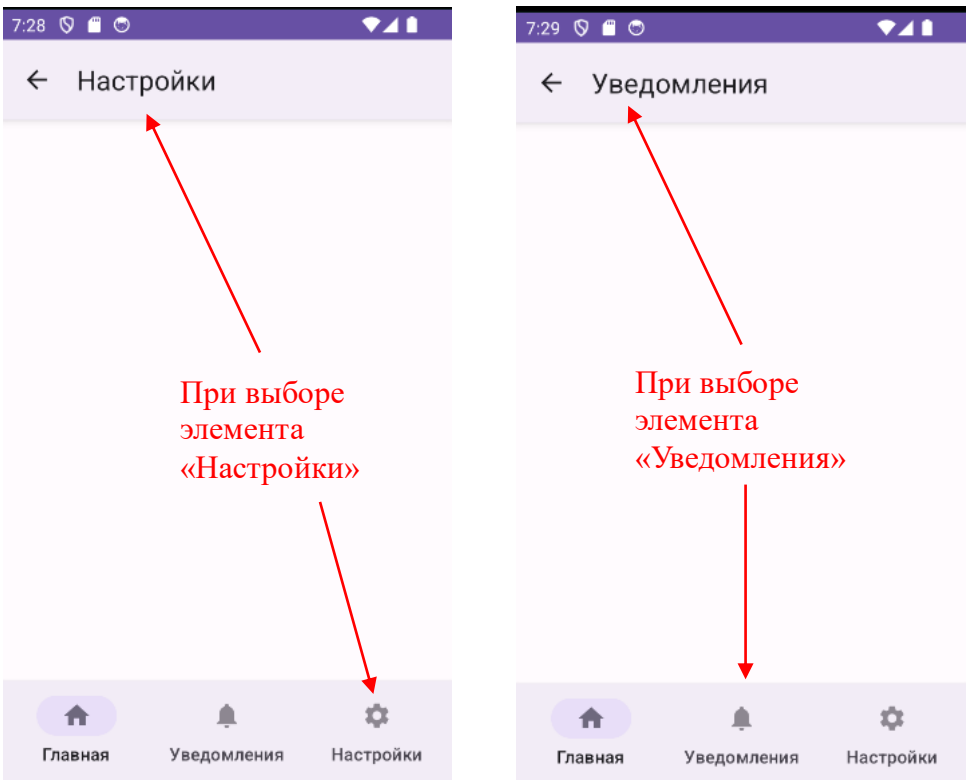
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Получаем ActionBar
    ActionBar actionBar = getSupportActionBar();
    if (actionBar != null) {
        // Настройки ActionBar
        actionBar.setDisplayHomeAsUpEnabled(true); // Показать кнопку назад
        actionBar.setTitle("Главная"); // Установить заголовок

        BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
        bottomNavigationView.setOnNavigationItemSelectedListener(item -> {
            if (item.getItemId() == R.id.home) {
                actionBar.setTitle("Главная");
            } else if (item.getItemId() == R.id.notification) {
                actionBar.setTitle("Уведомления");
            } else if (item.getItemId() == R.id.settings) {
                actionBar.setTitle("Настройки");
            }
            return false;
        });
    }
}
```

ActionBar

BottomBar



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.

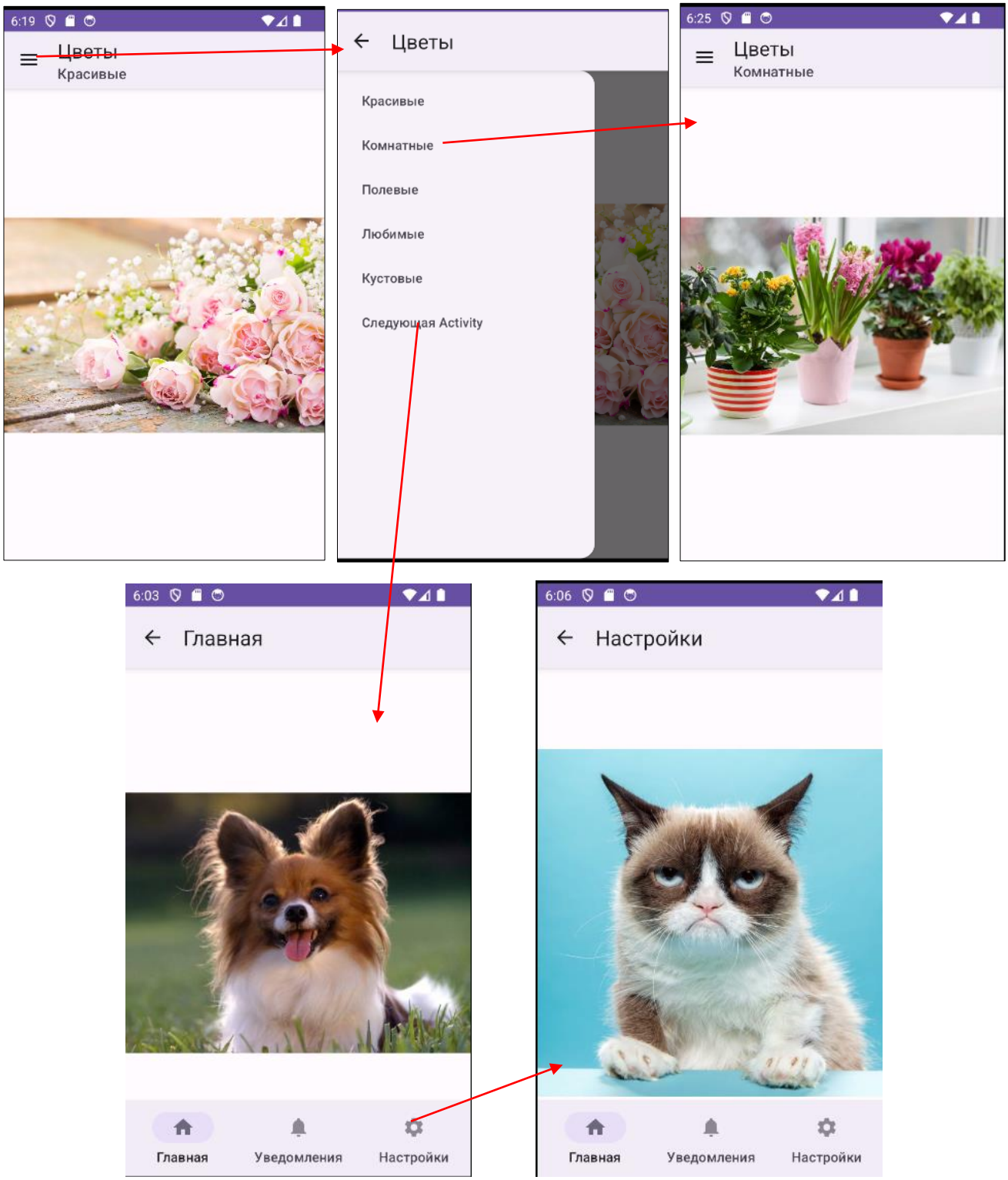
Задание

1. Реализовать несколько Activity. На первом Activity должна отображаться боковая панель навигации. Наполнить ее элементами и реализовать изменение фрагментов в зависимости от выбранного элемента. Реализовать переход на вторую Activity из одного из элемента панели.

2. На второй Activity реализовать интеграцию BottomBar с ActionBar. При нажатии на элемент BottomBar необходимо отображать текущий элемент в ActionBar и изменять наполнение Activity с помощью фрагментов.

Примечание: В каждой панели должно быть не менее 3 элементов! Для элементов обязательно BottomBar добавить иконки. Выбрать свою тематику!!!

ПРИМЕР



Источники

1) https://developer.android.com/reference/com/google/android/material/bottomappbar/BottomAppBar
2) https://www.geeksforgeeks.org/bottom-navigation-bar-in-android/
3) https://www.geeksforgeeks.org/navigation-drawer-in-android/

4) https://developer.alexanderklimov.ru/android/navigation_drawer_activity.php?ysclid=ltosagrigp987193251
5) https://startandroid.ru/ru/uroki/vse-uroki-spiskom/177-urok-107-android-3-actionbar-razmeschenie-elementov.html
6) https://www.geeksforgeeks.org/actionbar-in-android-with-example/
7) https://learntutorials.net/ru/android/topic/97/navigationview