Лекционно-практическое занятие

«Разработка бокового меню навигации»

Ранее разобрали только верхнее и нижнее меню навигации, но зачастую в приложениях встречается боковая панель с пунктами, переводящими на другие экраны или с пунктами, выводящими что-то.

Пользователь может открыть панель навигации, проведя пальцем от левого края экрана. Он также может найти её на главном экране (верхнем уровне приложения), нажав на значок приложения (также известный как «гамбургер» в Android) на панели действий.

Задача приложения: Создать боковое меню похожее на известную боковую панель приложения Телеграмм:

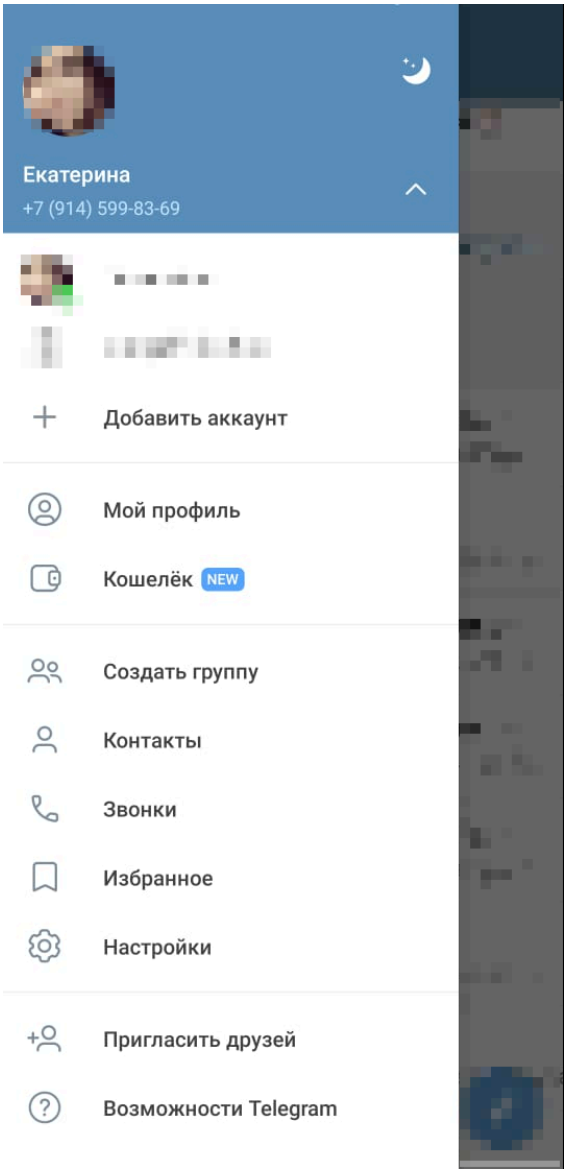
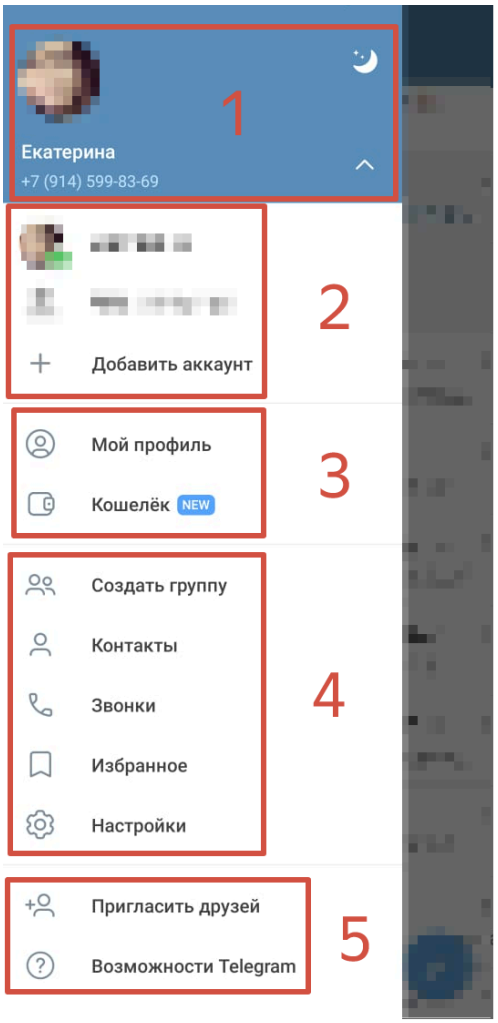


Рисунок 1 – Боковое меню мобильного приложения Telegram

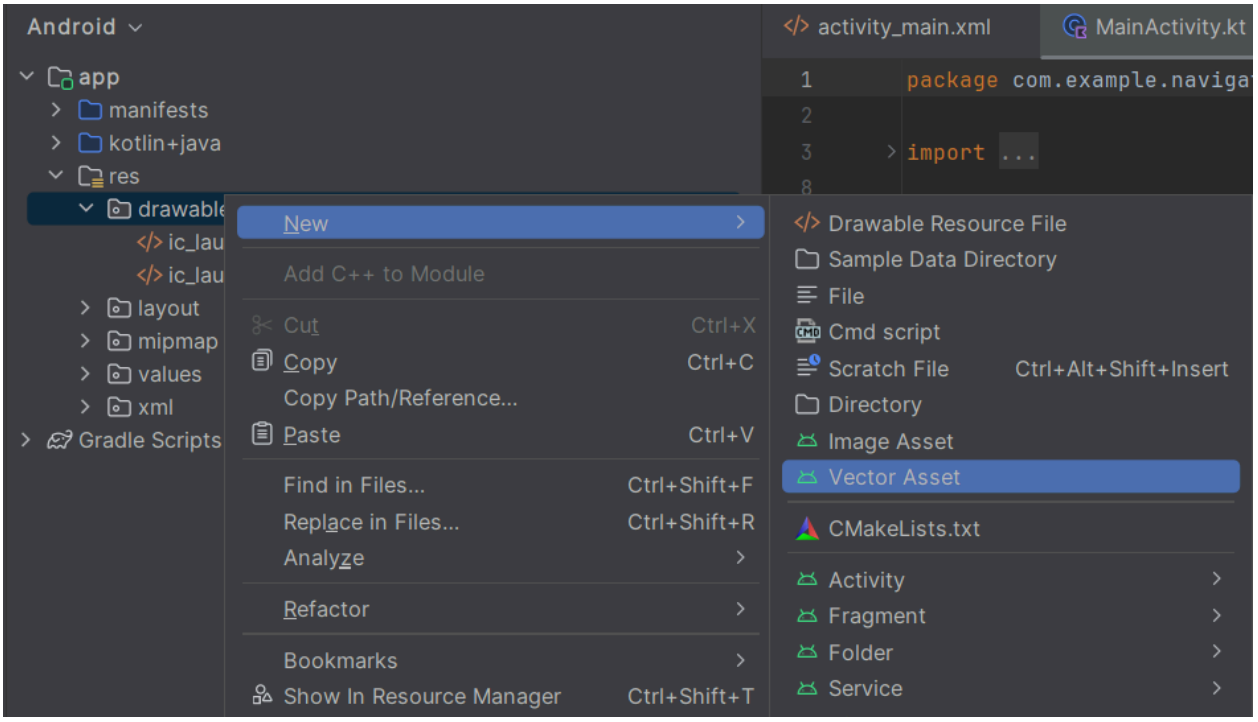
Разберем элементы, которые будем разрабатывать (выделены красным и пронумерованы на рисунке ниже):

Под цифрой 1: здесь содержится фото выбранного аккаунта, номер используемого телефона и никнэйм аккаунта, а также кнопки для смены темы и кнопка для скрытия/раскрытия имеющихся аккаунтов	
Под цифрами 2, 3, 4 и 5 указаны отдельные группы пунктов меню	
Под цифрой 2: здесь перечислены все подключенные аккаунты (в данном случае их 2) и кнопка добавления нового аккаунта	
Под цифрой 3: здесь 2 пункта Мой профиль (настройки текущего аккаунта) и Кошелек	
Под цифрой 4: здесь 4 пункта для управления приложением	
Под цифрой 5: здесь дополнительные кнопки для приглашения друзей к беседе и информационный пункт	

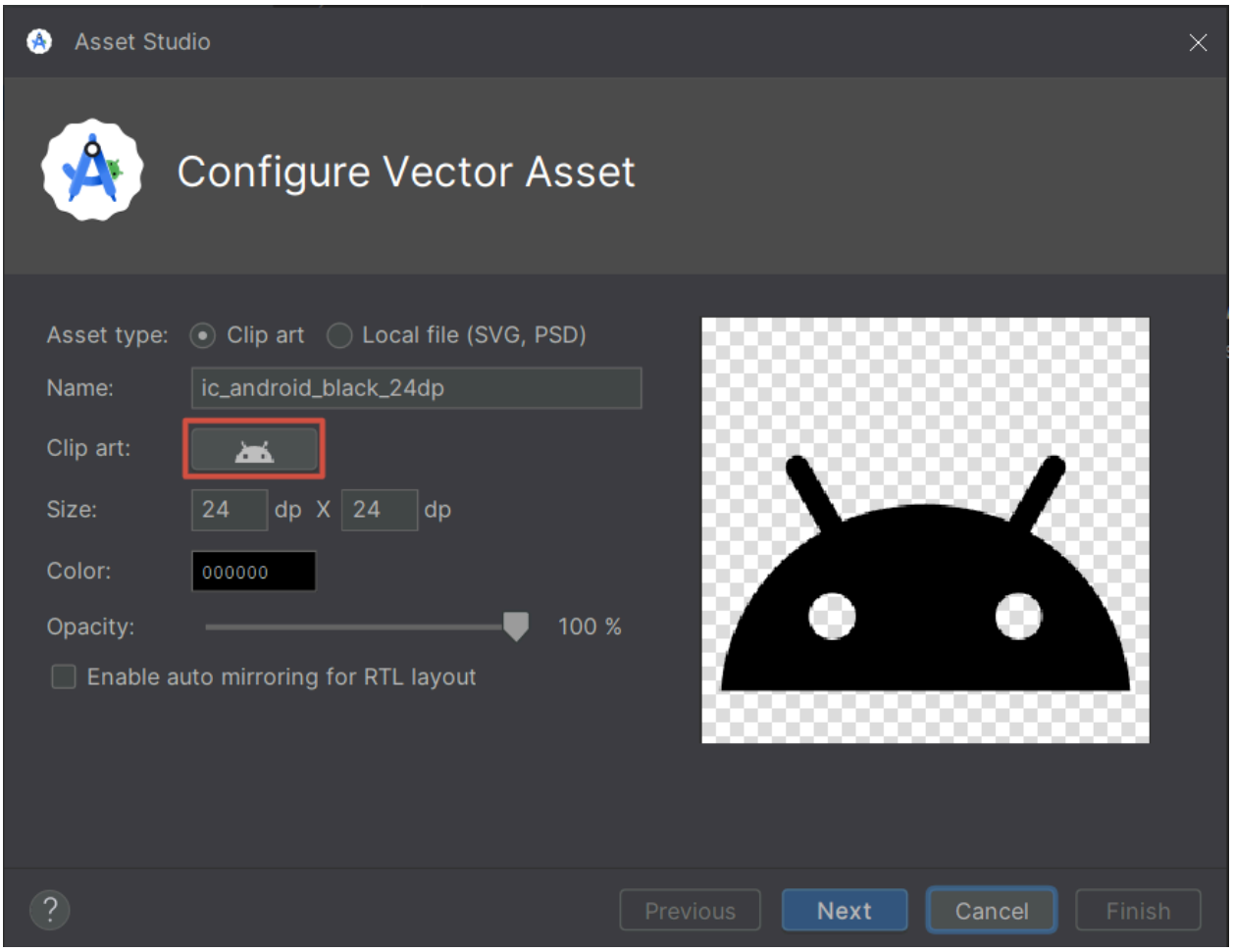
Работу самих пунктов, кнопки смены темы, свертывания/развертывания и подключение аккаутов осуществлять не будем, сделаем лишь как на картинке.

Добавление иконок

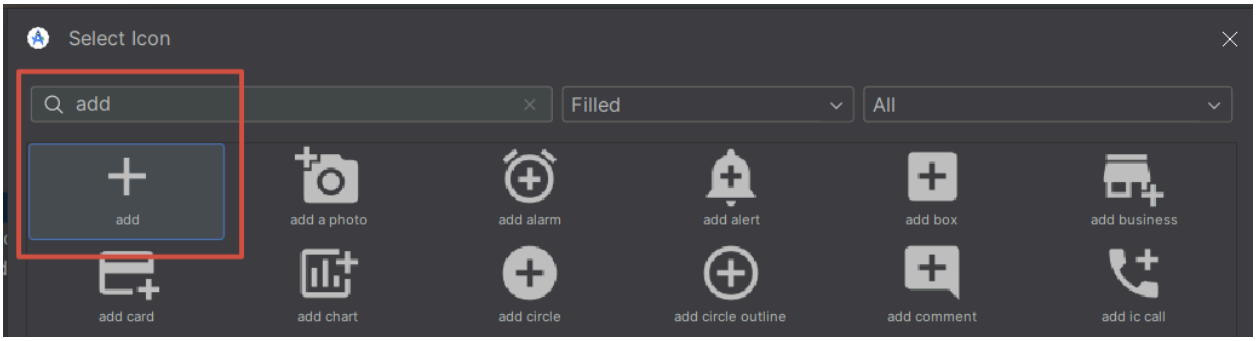
Для начала добавим все иконки, используемые в боковом меню. Для этого воспользуемся встроенными иконками. Чтобы добавить иконки нажмите ПКМ по папке **drawable**, далее выберите **New** и **Vector Asset**:



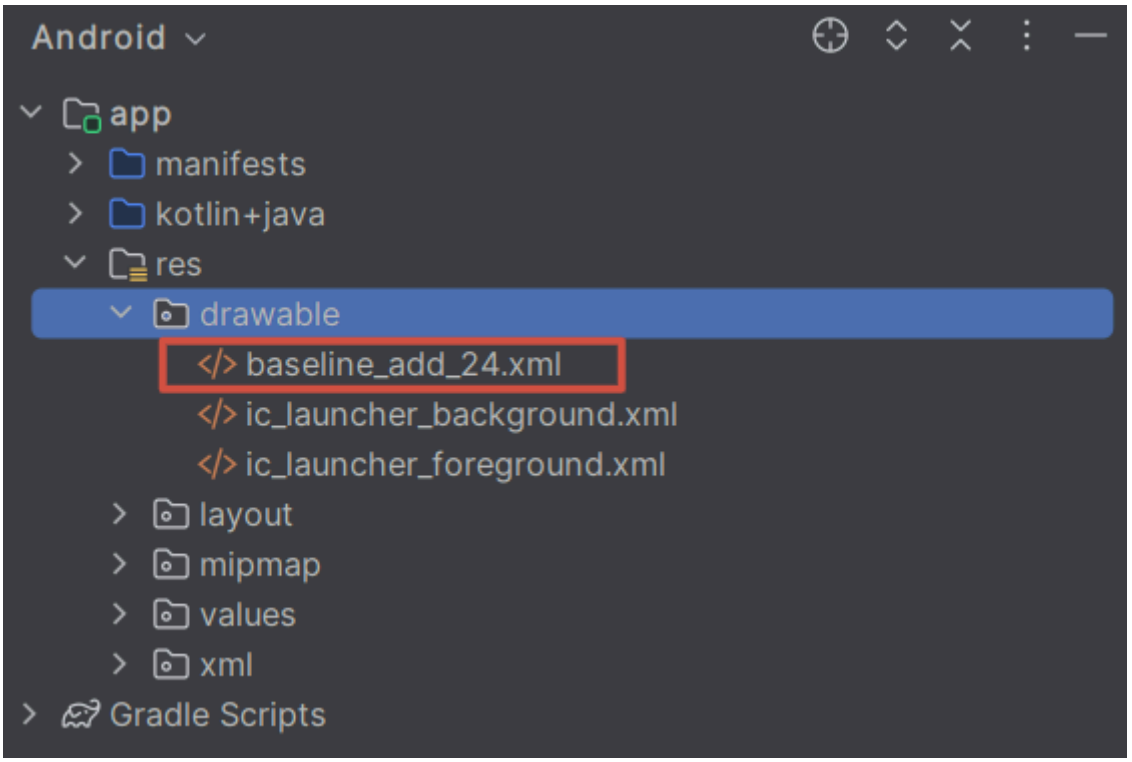
Далее нажмите на иконку в пункте Clip art (выделена красным):



В окне Select Icon введите **add**, выберите иконку с изображением плюс и нажмите OK, затем Next и Finish:



Ресурс иконки будет добавлен в папку drawable:

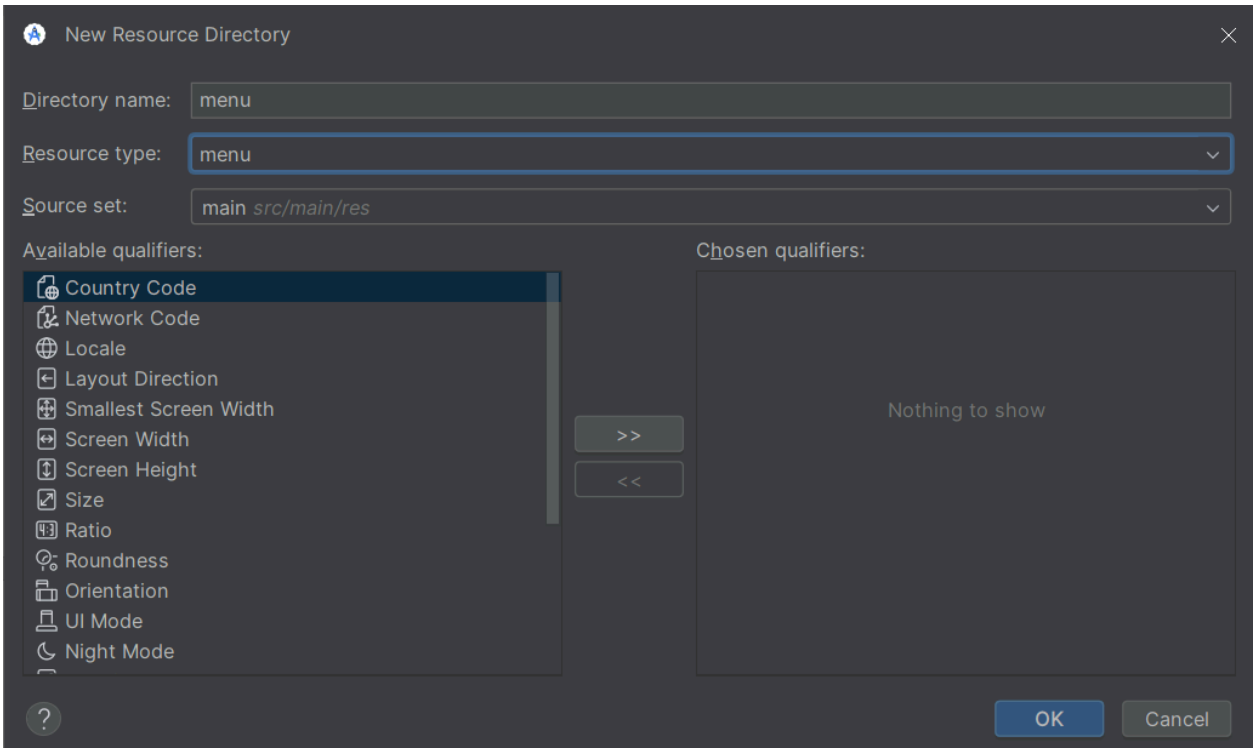


Задание 1: Подобным образом добавьте остальные иконки:

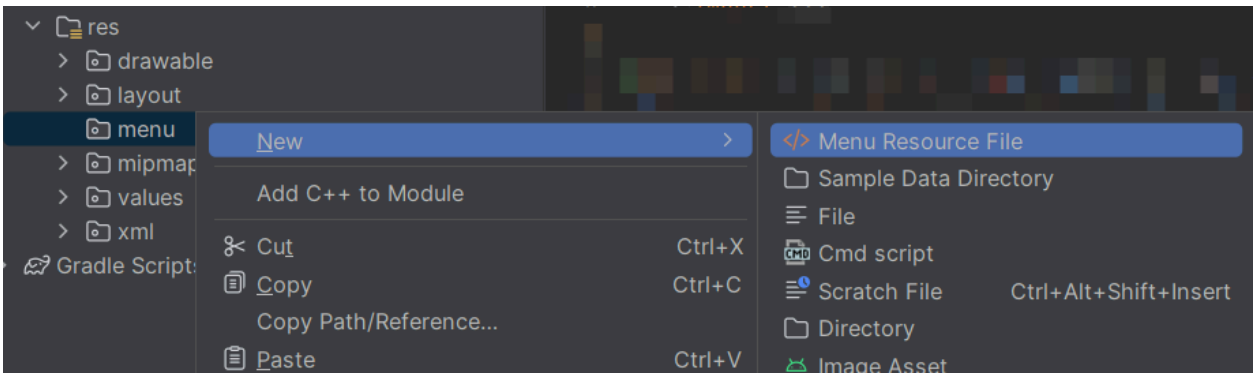
- account_circle
- credit_card
- group
- person
- call
- bookmark_border
- settings
- person_add
- question_mark

Добавление папки menu

В папку res добавьте новую ресурс директорию типа menu:



В папку menu добавьте новый меню ресурс файл и назовите его «**nav_menu**»:



Создание бокового меню

Вернемся к рисунку 1. Чтобы сделать пункты в боковом меню нужно их разделить на несколько разделов, а точнее на 4:

1) раздел с аккаунтами;
2) раздел с профилем и кошельком;
3) раздел с настройками приложения;

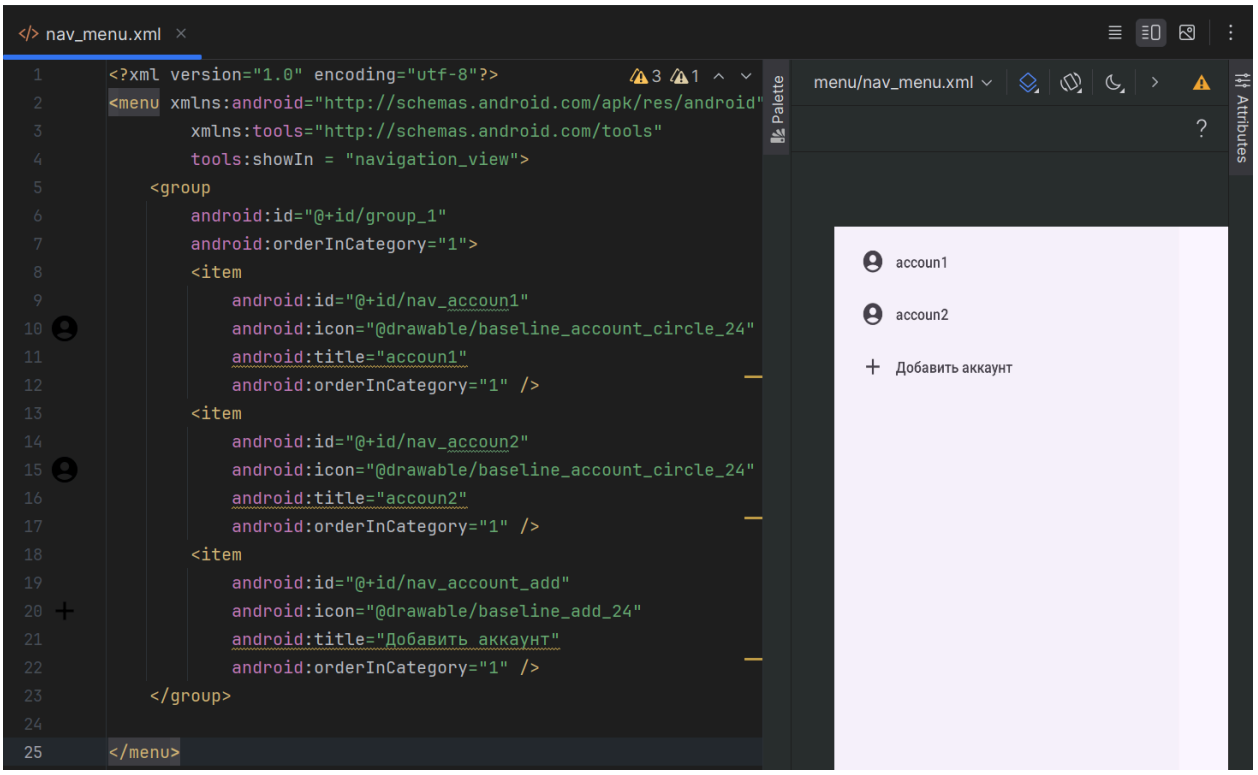
4) раздел с пунктами для приглашения друзей и информацией о возможностях телеграмм.

Проанализировав можно заметить, что у данных разделов нет никаких заголовков и все они отделены черной тонкой линией.

Чтобы создать несколько разделов в навигационном меню Android Studio без заголовков в файле XML, можно использовать MenuItem с нулевым заголовком, а затем группировать их внутри menu с помощью атрибута android:group и android:orderInCategory.

В файле «**nav_menu**» создайте 4 тэга **group** для логической группировки элементов меню. При помощи атрибута **android:orderInCategory** внутри group будем определять порядок отображения элементов внутри группы. Элементы в группах с меньшим **orderInCategory** будут отображаться раньше.

Итак, код первой группы будет следующим:



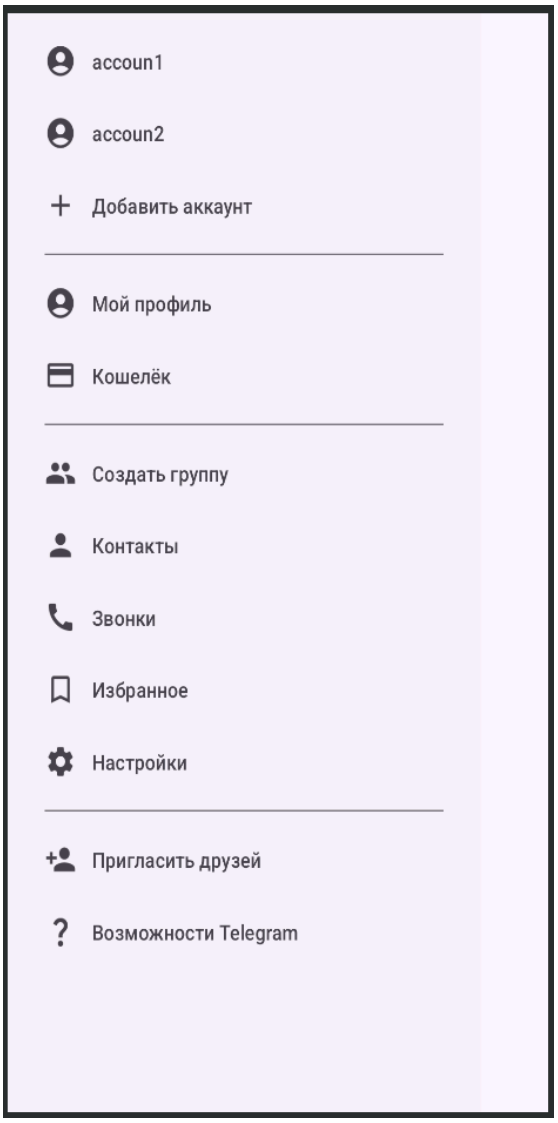
Разберем код:

- внутри тэга group имеется идентификатор, который будет определять конкретную группу и атрибут orderInCategory с порядком, так сказать очереди, равный 1.

- далее идут несколько тэгов item, которые задают сами пункты в этом разделе. Для каждого пункта создается свой идентификатор, иконка, отображаемый текст (атрибут title) и порядок, принадлежащий разделу, т.е. 1.

Задание 2: Создайте остальные разделы при помощи тэгов group. Для каждого раздела атрибут orderInCategory будет свой, т.е. для второго раздела orderInCategory = «2», для третьего 3 и т.д.

В итоге на примерном дизайне должно получиться следующее:

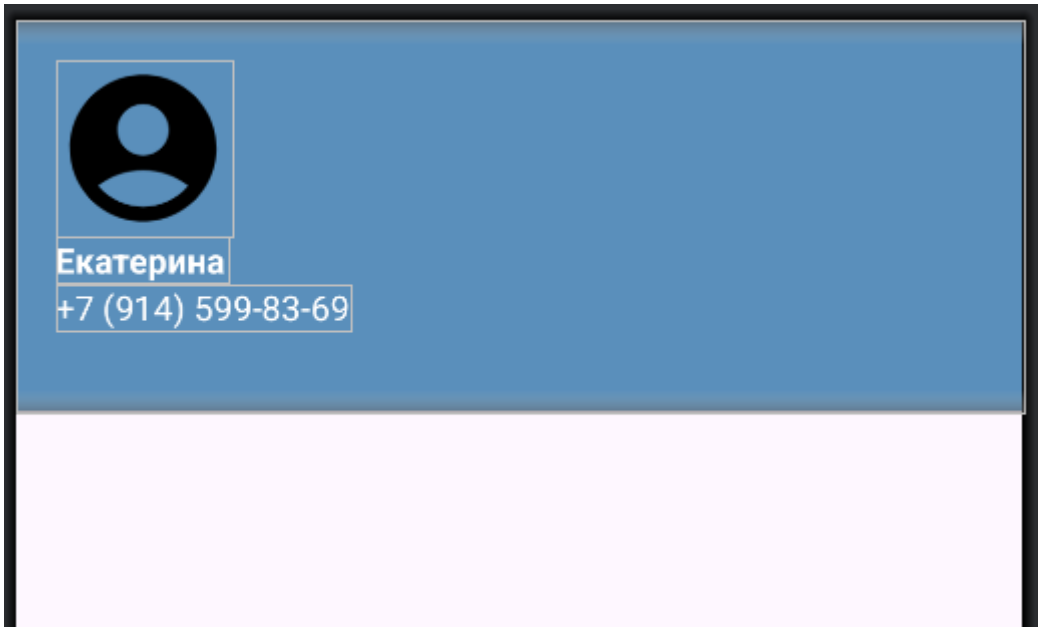


На рисунке выше видно, что для совпадения с рисунком 1 не хватает верхнего раздела с выбранным аккаунтом, который содержит фото аккаунта, номер телефона и никнэйм. Чтобы его добавить нужно создать новый ресурс файл в папке **layout**. Назовем его «**nav_header**».

Задание 3: В этой разметке создайте простой **линейный** контейнер, в котором будет изображение и 2 текста. Необходимые настройки:

- Для отступов от краев задайте атрибуты gravity="center_vertical" и paddingTop="16dp" и paddingStart="16dp" в тэге контейнера, а также высоту контейнера 160dp и фон #5A8FBB.
- Размер изображения с фото 72x72 и для изменения цвета иконки атрибут app:tint="@android:color/white" (не забудьте про импорт атрибута, начинающийся с app).
- Цвет текста из @color/white. Текст с никнэймом полужирный.

Должно получиться следующее:



Остальные кнопки (кнопка для смены темы и кнопку свертывания/развертывания) делать не будем. Нас пока интересуют только пункты меню.

Добавление бокового меню в разметку главной активности

Откройте разметку «activity_main.xml».

Чтобы отобразить значок панели (три горизонтальные черточки) на всех экранах нашего приложения, будем использовать компонент DrawerLayout.

Добавьте <androidx.drawerlayout.widget.DrawerLayout в качестве корневого представления. DrawerLayout - это ключевой компонент, который обеспечивает возможность выдвижения панели навигации. Внутри этого тэга добавьте:

- xmlns:android=http://schemas.android.com/apk/res/android стандартная строка;
- ширина и высота match_parent;
- идентификатор «drawer_layout», который позволит ссылаться на него в коде;
- атрибут android:fitsSystemWindows="true", который гарантирует, что панель учитывает системные вставки (например, для строки состояния);
- атрибут tools:openDrawer="start", который предназначен для предварительного просмотра во время разработки в Android Studio - он открывает панель по умолчанию в предварительном просмотре макета (не забудьте про импорт tools);
- tools:context=".MainActivity", который указывает, что этот макет используется классом MainActivity. На этом тэг DrawerLayout закрывается, но не совсем.

После корневого представления добавьте линейный контейнер с вертикальной ориентацией. В нем будет содержаться еще 2 тэга:

1) <androidx.appcompat.widget.Toolbar - это панель приложения (или панель действий) в верхней части экрана. У него атрибуты следующие:
   a. ширина ширина match_parent, а высота 56dp;
   b. идентификатор toolbar;
   c. атрибут android:elevation="4dp", который создает небольшую тень, чтобы придать ей трехмерный вид.

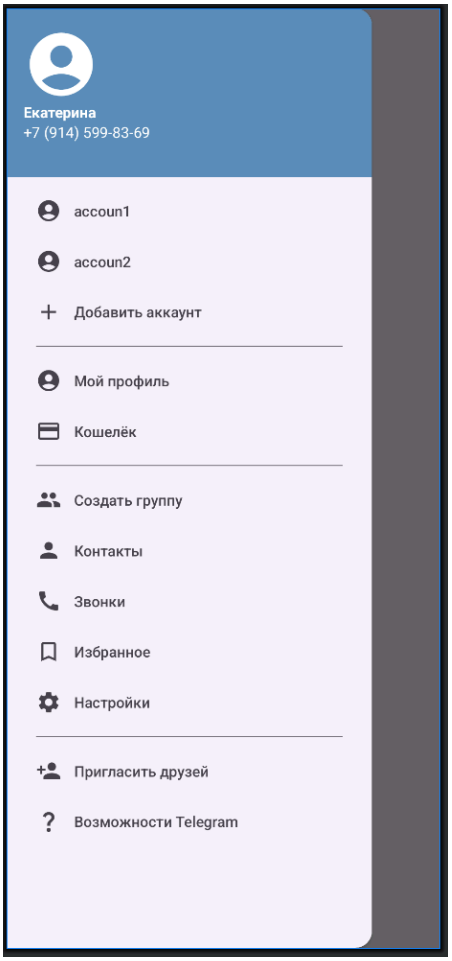
2) <FrameLayout - это контейнер для основного содержимого. Он очень важен; в нём будут динамически размещаться фрагменты с помощью FragmentManager. Добавьте к нему ширину и высоту match_parent и

идентификатор «fragment_container», который необходим для замены фрагментов в макете.

После линейного контейнера создадим сам «ящик» навигации со всем нашими пунктами и заголовком – это тэг <com.google.android.material.navigation.NavigationView. В нем нужно добавить:

- ширина и высота match_parent;
- идентификатор, например, nav_view;
- атрибут android:layout_gravity="start", чтобы расположить боковое меню на левом (начальном) краю экрана;
- атрибут app:headerLayout="@layout/nav_header", который укажет на отдельный файл макета (у нас это разметка с заголовком nav_header.xml) для определения заголовка в верхней части ящика.
- атрибут app:menu="@menu/nav_menu" со ссылкой на файл ресурсов меню (у нас это разметка nav_menu.xml), в котором определены элементы панели навигации.

Должно получиться следующее:



Добавление фрагментов

Для каждого пункта меню нужен свой фрагмент. Пунктов у нас 12, НО, чтобы все 12 фрагментов не создавать **создадим всего один**, который будет выводить строку, содержащую наименование выбранного пункта.

Создайте фрагмент «**MenuFragments**» как в работе по теме «Фрагменты». В разметке фрагмента по умолчанию уже есть строка. Добавьте к ней идентификатор, чтобы позже можно было изменять текст.

В классе фрагмента добавьте функцию **onViewCreated**, в которой из главной активити будет приходить наименование выбранного пункта по сообщению **title** (textFragment – это идентификатор в разметке фрагмента):

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)

    val title = arguments?.getString( key: "title") ?: "No title"
    binding.textFragment.text = title
}
```

Добавление строк в strings

В папке res – values откройте strings.xml и добавьте 2 строки:

```
<string name="open_nav">Open Navigation Drawer</string>
<string name="close_nav">Close Navigation Drawer</string>
```

Эти строки понадобятся при создании навигационного меню.

Изменение MainActivity

Для начала в классе MainActivity после наследования от AppCompatActivity после запятой добавьте наследование NavigationView.OnNavigationItemSelectedListener для обработки выбора элементов в навигационном ящике.

Добавьте 2 глобальных свойства binding и drawerLayout унаследованный от DrawerLayout.

Далее в функции onCreate инициализируйте биндинг свойство и дайте ему root права. Также инициализируйте свойство drawerLayout.

Дальше нужно установить панель инструментов в качестве панели действий при помощи ссылок из биндинга:

```
val toolbar = binding.toolbar
setSupportActionBar(toolbar)

val navigationView = binding.navView
```

Зарегистрируйте действие слушателя элементов меню при помощи строки **navigationView.setNavigationItemSelectedListener(this)**, который

«подтягивается» из унаследованного NavigationView.OnNavigationItemSelectedListener.

Запишите следующий фрагмент кода, который настраивает панель навигации и загружает начальный фрагмент в приложении Android:

```
val toggle = ActionBarDrawerToggle( activity: this, drawerLayout, toolbar, R.string.open_nav, R.string.close_nav)
drawerLayout.addDrawerListener(toggle)
toggle.syncState()

if (savedInstanceState == null) {
    supportFragmentManager.beginTransaction()
        .replace(R.id.fragment_container, MenuFragments()).commit()
}
```

Пояснения кода:

- первая строка строка создает объект ActionBarDrawerToggle. Этот объект отвечает за синхронизацию состояния панели навигации (открыта или закрыта) с внешним видом значка toggle (или значка меню «три полосы») на панели приложения (панели инструментов);
- вторая строка добавляет ActionBarDrawerToggle в качестве прослушивателя в DrawerLayout. Это очень важно: DrawerLayout должен знать о toggle и получать обновления о его статусе открытия/закрытия. Это позволяет toggle реагировать на изменения состояния панели и соответствующим образом обновлять панель приложения;
- третья строка синхронизирует состояние ActionBarDrawerToggle с текущим состоянием DrawerLayout. Это важно, особенно при повторном создании действия (например, после поворота экрана). Это гарантирует, что значок toggle на панели приложения правильно отражает, открыт или закрыт ящик. Без этого значок может находиться в непоследовательном состоянии;
- далее условный блок, который выполняется только в том случае, если действие создается впервые (а не воссоздается из предыдущего состояния после изменения конфигурации, например, поворота экрана), в этом случае savedInstanceState Bundle имеет значение null. Внутри условного блока запускается транзакция фрагмента и его отображение.

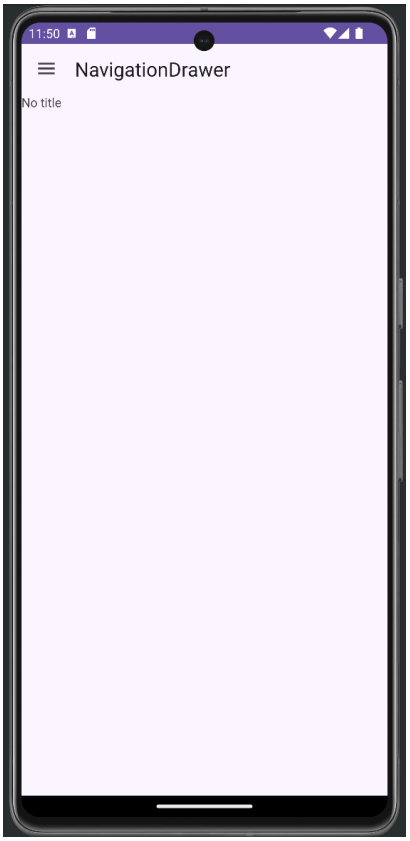
При наследовании от NavigationView.OnNavigationItemSelectedListener необходимо унаследовать функцию onNavigationItemSelected:

```
override fun onNavigationItemSelected(item: MenuItem): Boolean {
    val fragment = MenuFragments()
    val bundle = Bundle()
    bundle.putString("title", item.title.toString())
    fragment.arguments = bundle
    supportFragmentManager.beginTransaction()
        .replace(R.id.fragment_container, fragment)
        .commit()
    drawerLayout.closeDrawer(GravityCompat.START)
    return true
}
```

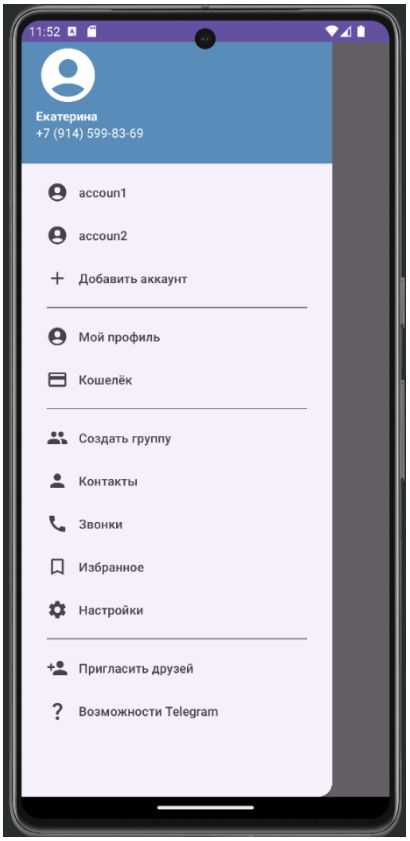
Пояснения кода:

- **val fragment = MenuFragments()**: Создает новый экземпляр MenuFragments;
- **val bundle = Bundle()**: Создает Bundle для передачи данных во фрагмент;
- **bundle.putString("title", item.title.toString())**: Добавляет заголовок выбранного пункта меню в Bundle с ключом «title»;
- **fragment.arguments = bundle**: Присваивает Bundle свойству arguments фрагмента. Это позволяет фрагменту получать данные, в нашем случае наименование пункта в меню;
- **supportFragmentManager.beginTransaction().replace(R.id.fragment_container, fragment) .commit()**: Это выполняет транзакцию фрагмента. Она заменяет фрагмент, который в данный момент находится в контейнере с идентификатором R.id.fragment_container, на новый экземпляр MenuFragments;
- **drawerLayout.closeDrawer(GravityCompat.START)**: Закрывает панель навигации после выбора элемента;
- **return true**: Указывает, что выбор элемента был обработан.

Протестируйте проект. При первоначальном запуске на экране следующее:



После нажатия на три полоски появляется меню:



При выборе пункта в меню закрывается меню и отображается текст пункта. Например, выбор пал на пункт меню «Создать группу»:


11:53
NavigationDrawer
Создать группу

САМОСТОЯТЕЛЬНАЯ ЧАСТЬ

Задание 1: Добавьте 3 фрагмента: ProfileFragment, SettingsFragment, AboutTelegramFragment.

ProfileFragment должен открываться когда в боковом меню выбран пункт «Мой профиль» и выводить короткое всплывающее сообщение с названием пункта.

SettingsFragment должен открываться когда в боковом меню выбран пункт «Настройки» и выводить длительное всплывающее сообщение с названием пункта.

AboutTelegramFragment должен открываться когда в боковом меню выбран пункт «Возможности Telegram» и выводить название пункта в качестве строки.

Замечание: Для понимания того, что открылся один из фрагментов выше, измените задний фон в разметке фрагментов. Лучше если они будут разных цветов (см. короткое видео «Задание 1 Боковое меню.webm» на сайте bpkinfo.ru).

Задание 2*: Добавьте кнопку для смены режима (Кнопка с изображением луны на рисунке 1 - ). При нажатии на эту кнопку меняйте тему приложения на «Темный режим» и меняйте кнопку на изображение солнца. При нажатии на солнце происходит обратная смена темы и изображение кнопки.

Задание 3*: Добавьте кнопку стрелочка напротив логина и номера телефона пользователя (Кнопка с изображением стрелки на рисунке 1 - ). При нажатии на эту кнопку данные о всех аккаунтах и кнопка «Добавить аккаунт» должны скрываться. При повторном нажатии раскрываться.