**СОЗДАНИЕ МЕНЮ МЕТОДОМ ADD**

Задание 1

Для начала нам потребуется в папке проекта app/res/values/themes themes.xml и исправить значение атрибута заданного стиля parent на другое, без NoActionBar.

Измененный код для файла с темами предоставлен на рисунке 1.1.

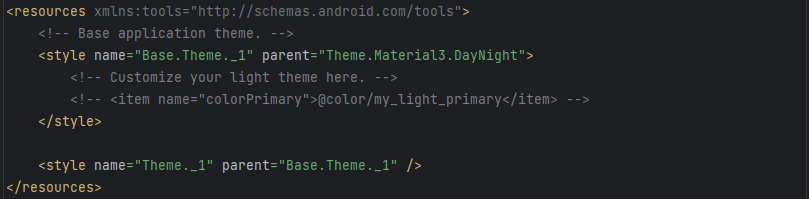


Рисунок 1.1 – Файл themes.xml

Выведем всплывающее сообщение с текстом нажатого пункта меню. Для этого создадим методы onCreateOptionsMenu и onOptionsItemSelected соответственно.

Код данных методов предоставлен на рисунке 1.1.

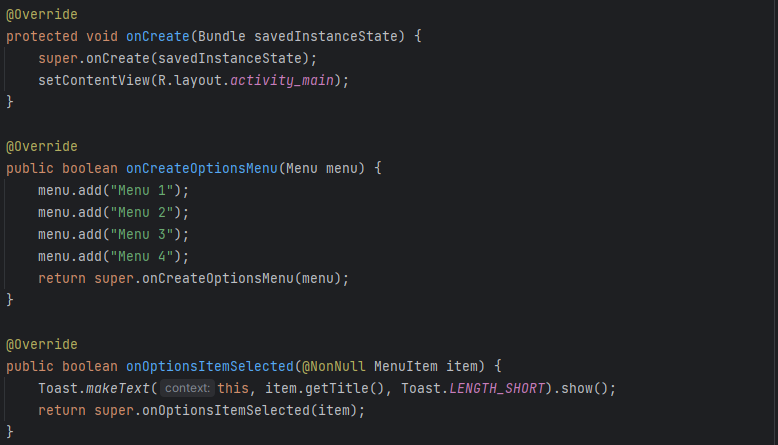


Рисунок 1.2 – Файл MainActivity.java

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 1.3.



Рисунок 1.3 – Запущенное приложение на устройстве

Задание 2

Присвоим ID существующему TextView, сотрём его текст и создадим CheckBox. Измененный код предоставлен на рисунке 2.1.



Рисунок 2.1 – Файл activity\_mai.xml

В классе главного Activity реализуем код, который использует следующие методы:

1. onCreateOptionsMenu - вызывается только при первом показе меню. Создает меню и более не используется. Здесь мы добавляем к меню пункты.
2. onPrepareOptionsMenu - вызывается каждый раз перед отображением меню. Здесь мы вносим изменения в уже созданное меню, если это необходимо
3. onOptionsItemSelected - вызывается при нажатии пункта меню. Здесь мы определяем какой пункт меню был нажат.

Реализуемый код предоставлен на рисунке 2.2.



Рисунок 2.2 – Файл MainActivity.java

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 2.3.

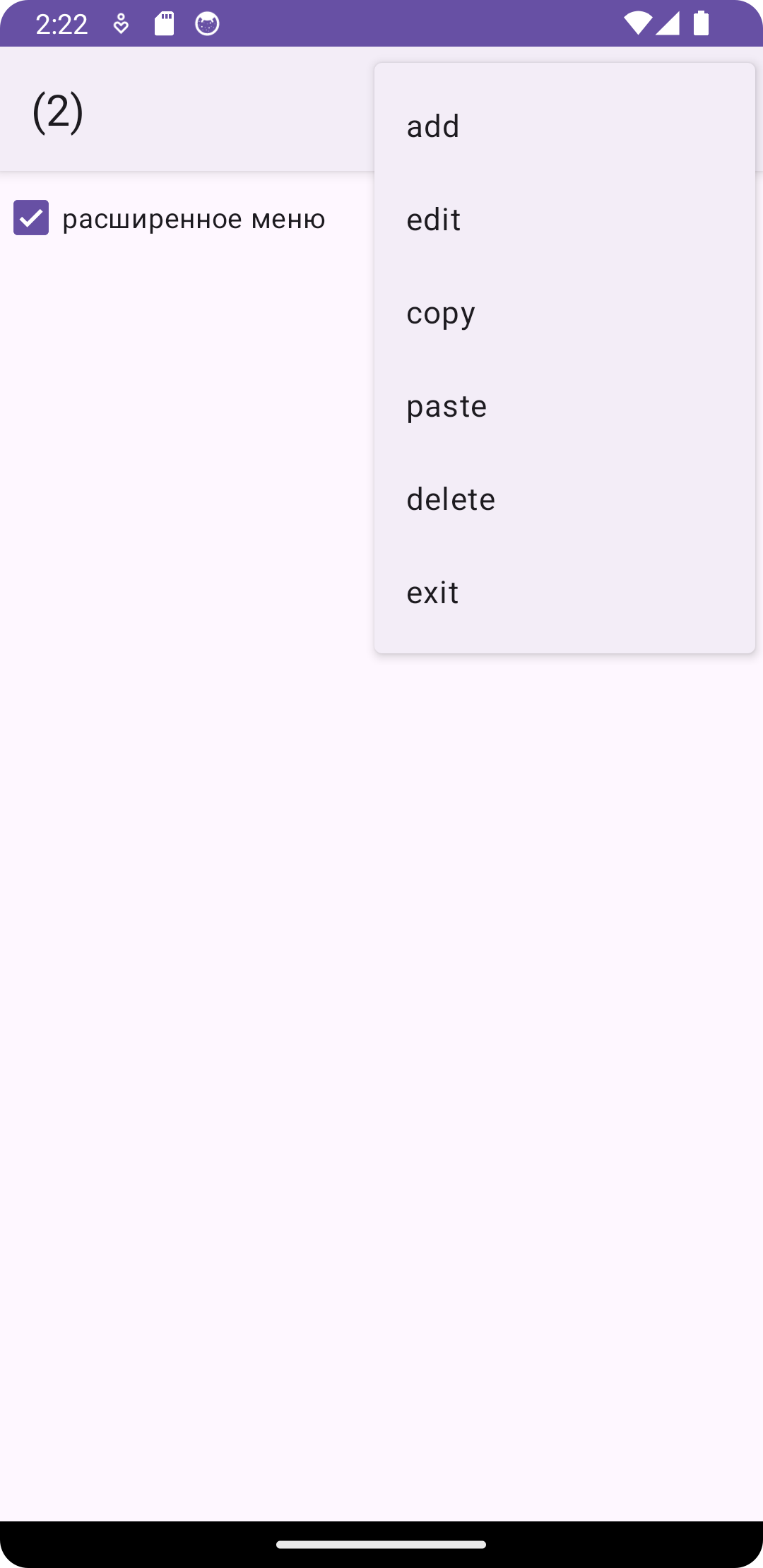


Рисунок 2.3 – Запущенное приложение на устройстве

**СОЗДАНИЕ XML-МЕНЮ**

Задание 3

Есть более удобный и предпочтительный способ создания меню – с использованием xml-файлов, аналогично layout-файлам при создании экрана.

Чтобы получить меню как в предыдущем задании, необходимо создать в папке res/menu файл mymenu.xml и изменить его как показано на рисунке 3.1.

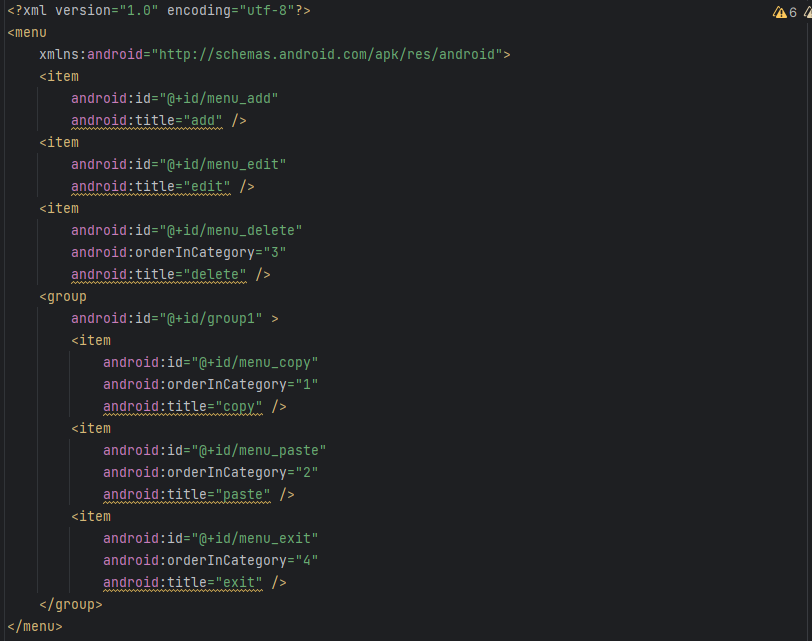


Рисунок 3.1 – Файл mymenu.xml

В методе onCreateOptionsMenu теперь не нужно в коде вручную создавать каждый пункт, достаточно связать menu, который даётся на вход, с xml-файлом. Код, который изменен по сравнению с предыдущем заданием показан на рисунке 3.2.

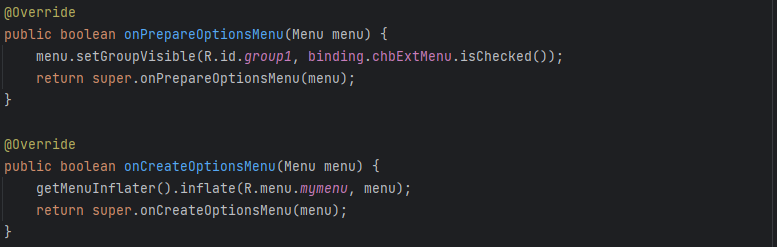


Рисунок 3.2 – Изменения в методе onCreateOptionsMenu и onPrepareOptionsMenu

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 3.3.

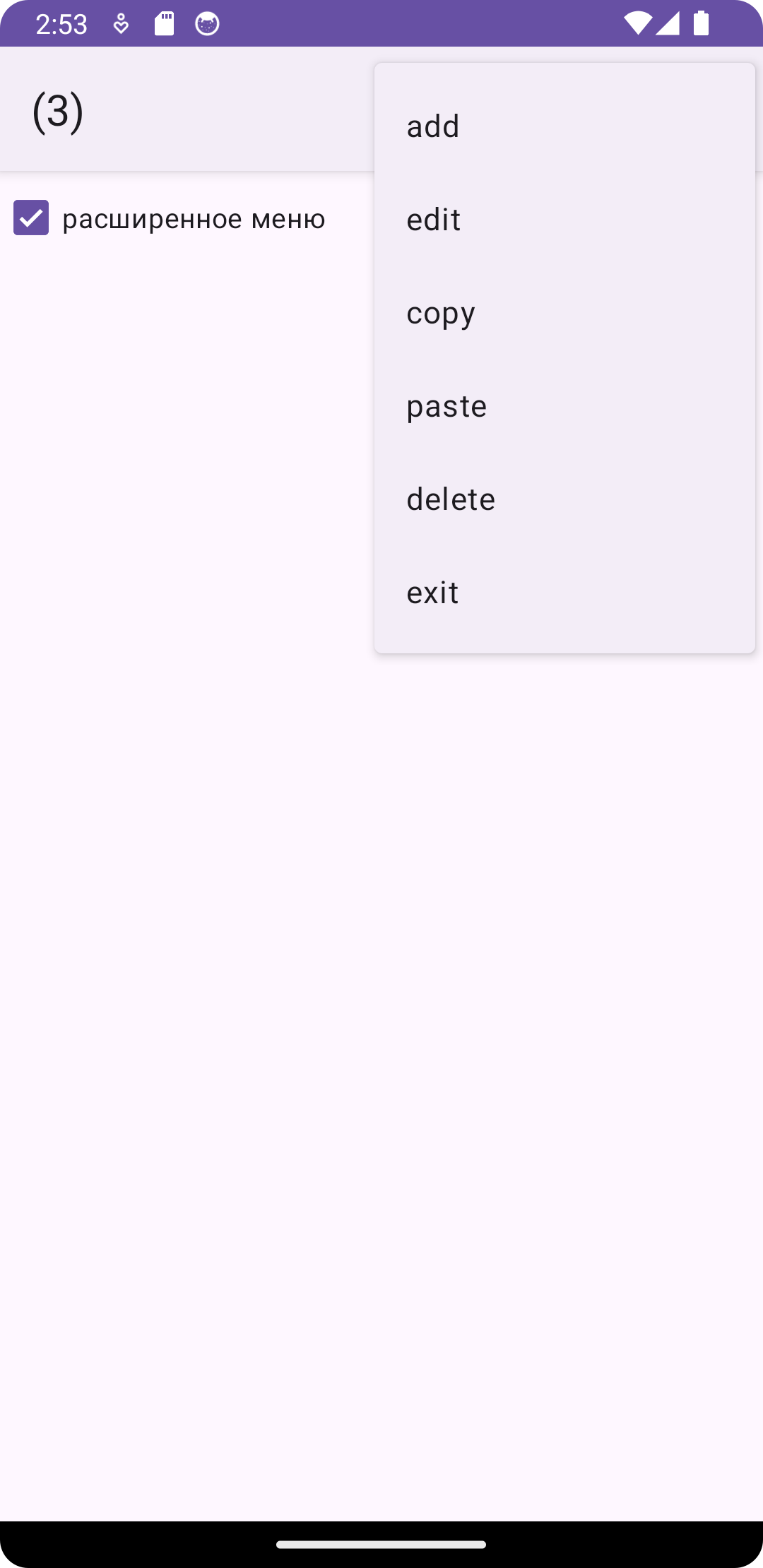


Рисунок 3.3 – Запущенное приложение на устройстве

**РАСШИРЕННЫЕ СРЕДСТВА ПАНЕЛИ ACTIONBAR**

Задание 4

В проекте создан файл ресурса меню res\menu\ mymenu.xml код данного файла предоставлен на рисунке 4.1.

id, title и icon – это идентификатор, текст и иконка элемента (иконки системные).

Атрибут showAsAction – это режим показа элемента, может принимать значения:

1. never – не показывать элемент;
2. ifRoom – показывать, если есть место;
3. always – всегда показывать.



Рисунок 4.1 – Файл mymenu.xml

Код для работы с меню в MainActivity.java предоставлен на рисунке 4.2.

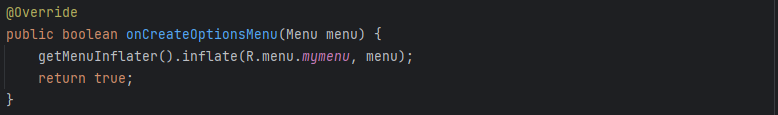


Рисунок 4.2 – Метод onCreateOptionsMenu в файле MainActivity.java

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 3.3.

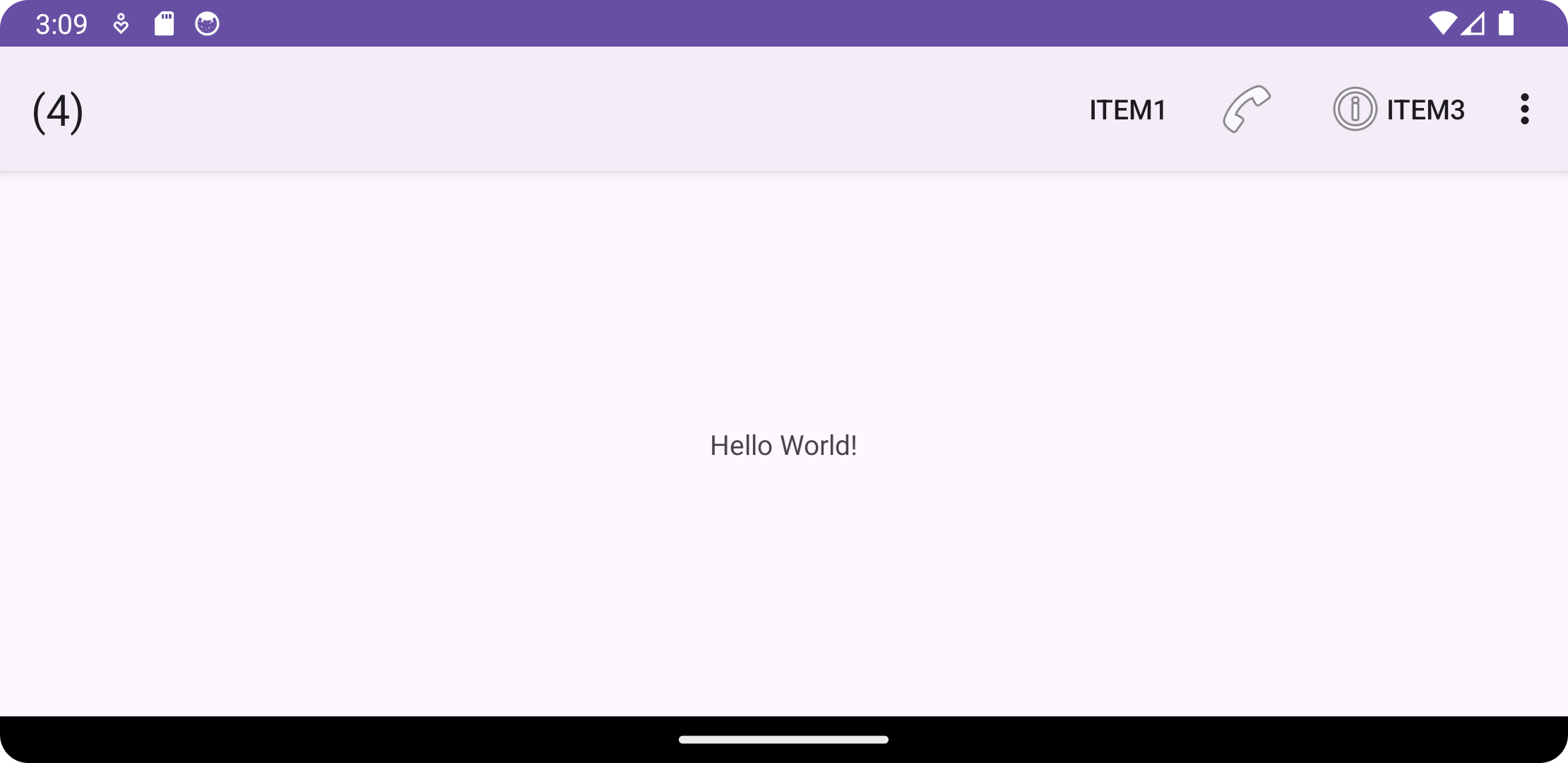


Рисунок 3.3 – Запущенное приложение на устройстве

Для того чтобы установить для панели ActionBar в качестве заголовка мою фамилию и имя, а подзаголовка – мою учебную группу. Нам потребуется расширить метод onCreate. Данное дополнение предоставлено на рисунке 3.4.



Рисунок 3.4 – Измененный метод onCreate

Также реализуем кнопку для скрытия\отображения панели xml код данной кнопки предоставлен на рисунке 3.5.

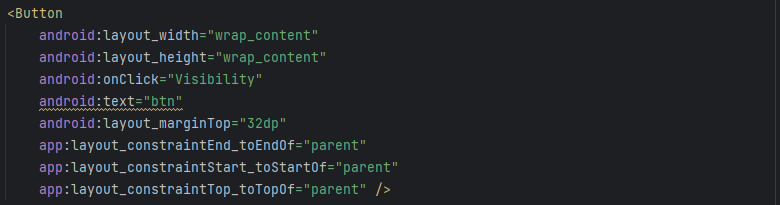


Рисунок 3.5 – xml код отвечающий за кнопку

Также нам необходимо добавить метод, который будет отвечать за скрытие\отображения панели. Данный код предоставлен на рисунке 3.6.

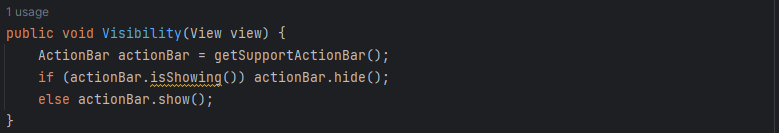


Рисунок 3.6 – Добавленный метод Visibility

Запустим наше приложение на виртуальном устройстве. Отображение на устройстве предоставлено на рисунке 3.7.



Рисунок 3.7 – Запущенное приложение на устройстве

**ПЕРЕЛИСТЫВАНИЕ СТРАНИЦ НА ОСНОВЕ ФРАГМЕНТОВ**

Задание 5

В проект добавим файл разметки фрагмента fragment\_blank.xml и сформируем его содержимое в соответствии с заданием рисунок 4.1.

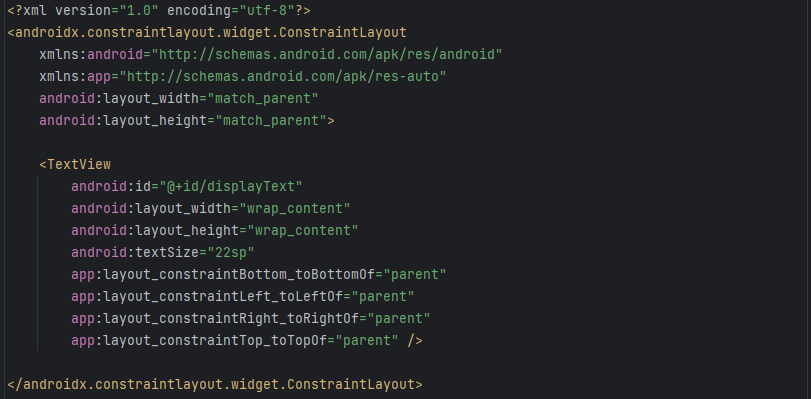


Рисунок 4.1 – Содержимое файла fragment\_blank.xml

Добавим в проект класс фрагмента BlankFragment, сформируем его код в соответствии с заданием рисунок 4.2.

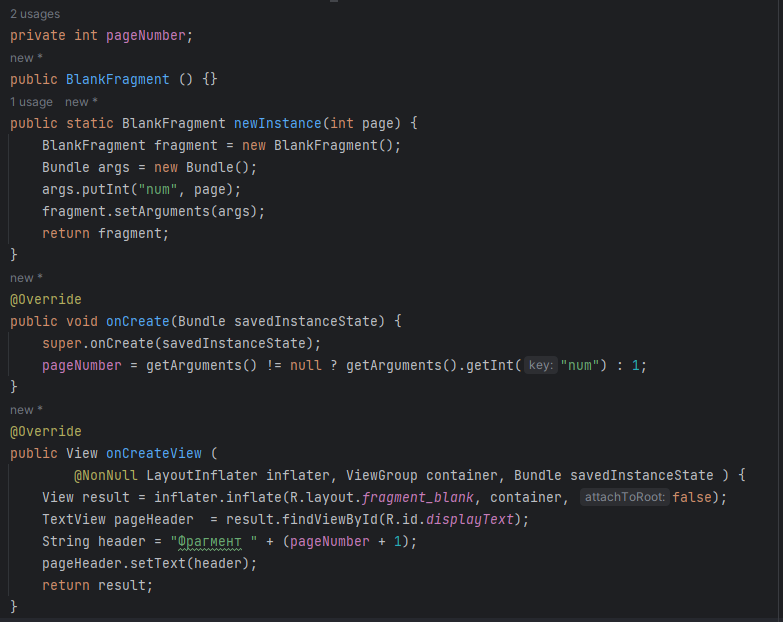


Рисунок 4.2 – Содержимое файла BlankFragment.java

Создадим новый класс MyAdapter, реализуйте в нём код в соответствии с заданием рисунок 4.3.



Рисунок 4.3 – Содержимое файла MyAdapter.java

Установим в файле activity\_main.xml элемент ViewPager2 в соответствии с заданием рисунок 4.4.



Рисунок 4.4 – Содержимое файла activity\_main.xml

Сформируйте код MainActivity.java в соответствии с заданием рисунок 4.5.

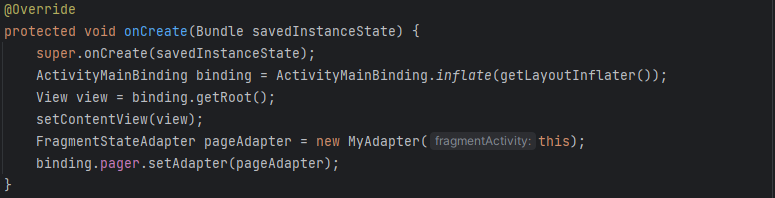


Рисунок 4.6 – Содержимое файла MainActivity.java

Запустим программу и проверим с помощью перелистывания возможность перемещаться по страницам. Запущенное приложение показано на рисунке 4.7.



Рисунок 3.7 – Запущенное приложение на устройстве

**ВКЛАДКИ НА ОСНОВЕ TABBED LAYOUT**

Задание 6

Создадим копию проекта из предыдущего задания.

Измените файл activity\_main.xml в соответствии с заданием рисунок 6.1.

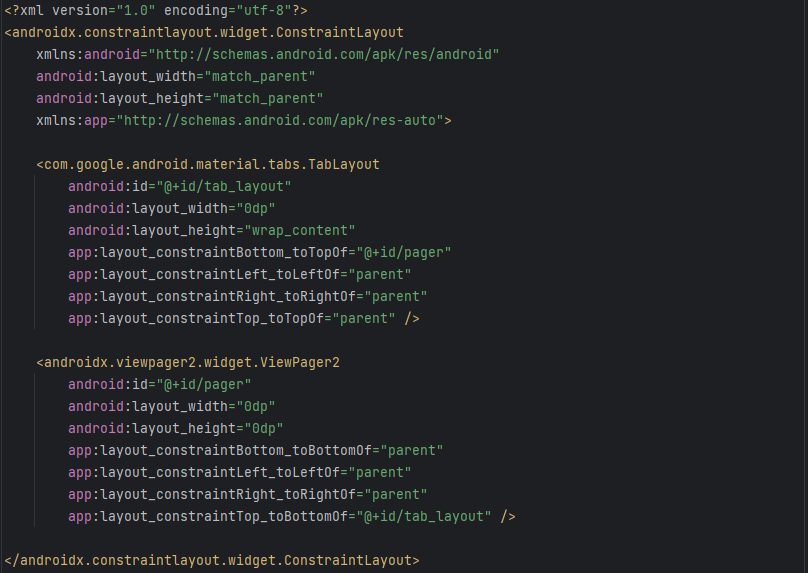


Рисунок 6.1 – Содержимое файла activity\_main.xml

Измените код MainActivity в соответствии с заданием рисунок 6.2.

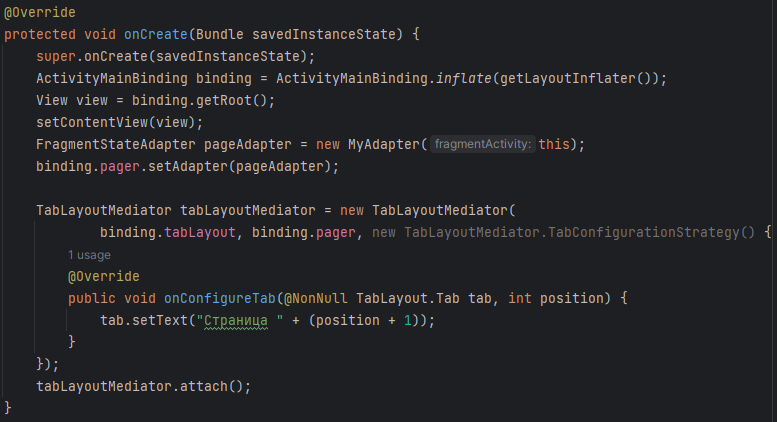


Рисунок 6.2 – Содержимое файла MainActivity.java

Запустим программу и проверим работоспособность интерактивных вкладок-заголовков страниц. Запущенное приложение показано на рисунке 6.3.



Рисунок 6.3 – Запущенное приложение на устройстве

**ПРАКТИЧЕСКОЕ ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАЗРАБОТКИ**

Задание 7