|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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 № 4** | |
| **по дисциплине** | |
| **«**Разработка мобильных приложений**»**  **Тема: «Списки. Создание собственного адаптера.**  **Контейнер Parcel и интерфейс Parcelable.»** | |
|  | |
| Выполнил студент группы ИКБО-33-22 | Шило Ю.С. |
| Принял преподаватель | Рысин М.Л. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2024

**ЧАСТЬ 1**

**Задание 1**

Создание View-элемента из содержимого layout-файла.

Для выполнения задания мы должны расписать методы в MainActivity.java. Данный код предоставлен на рисунке 1.1.



Рисунок 1.1 – Содержимое файла MainActivity.java

Для выполнения данной работы нам необходимо изменить содержимое файла activity\_main.xml данное изменение предоставлено на рисунке 1.2.

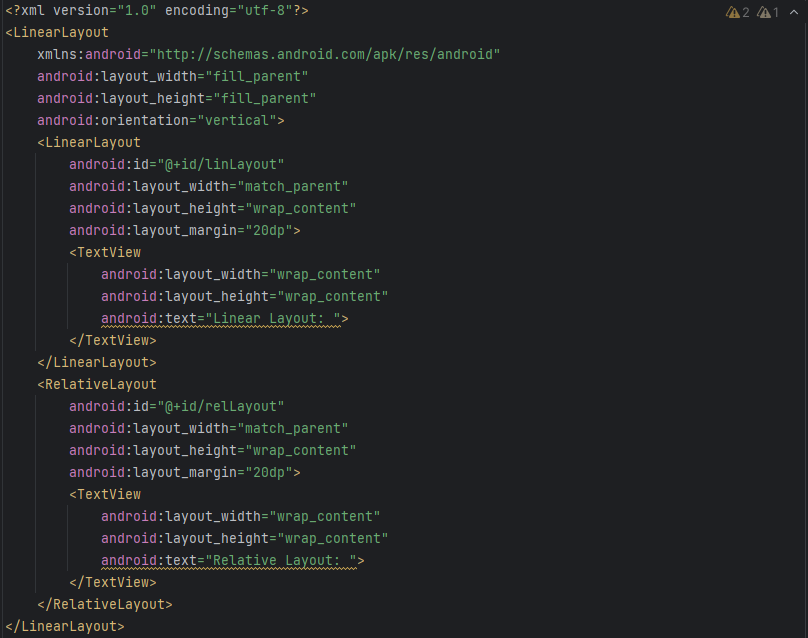


Рисунок 1.2 – Содержимое файла activity\_main.xml

Также для выполнения данного задания нам необходимо добавить еще один layout, а именно text.xml его код предоставлен на рисунке 1.3.

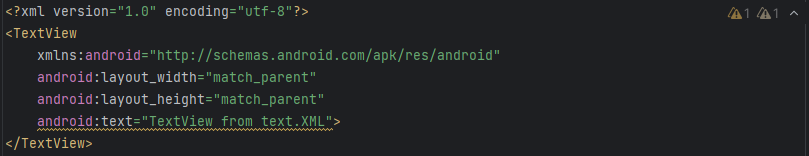


Рисунок 1.3 – Содержимое файла text.xml

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 1.4.



Рисунок 1.4 – Панель в дебаг консоли

На рисунке 1.5 показано как выглядит написанное нами приложение.



Рисунок 1.5 – Запущенное приложение на устройстве

Дополним код нашего MainActivity.java кодом с рисунка 1.6.



Рисунок 1.6 – Код, добавленный в MainActivity.java

На рисунке 1.7 показано как выглядит написанное нами приложение.



Рисунок 1.7 – Запущенное приложение на устройстве

Дополним код нашего MainActivity.java кодом с рисунка 1.8.

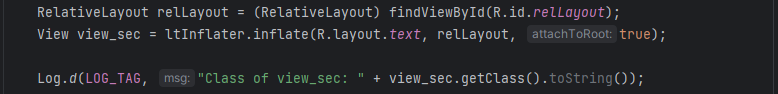


Рисунок 1.8 – Код, добавленный в MainActivity.java

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 1.9.

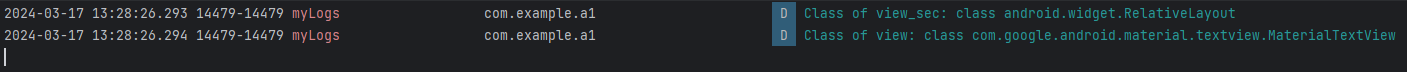


Рисунок 1.9 – Панель в дебаг консоли

На рисунке 1.10 показано как выглядит написанное нами приложение.



Рисунок 1.10 – Запущенное приложение на устройстве

**Задание 2**

Список на основе LayoutInflater.

Для выполнения данной работы нам необходимо изменить содержимое файла activity\_main.xml данное изменение предоставлено на рисунке 1.1.

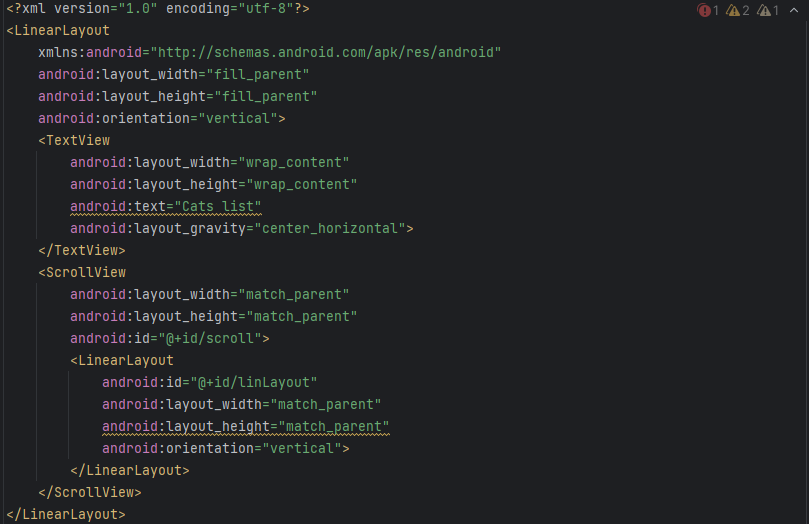


Рисунок 2.1 – Содержимое файла activity\_main.xml

Также для выполнения данного задания нам необходимо добавить еще один layout, а именно item.xml его код предоставлен на рисунке 2.2.

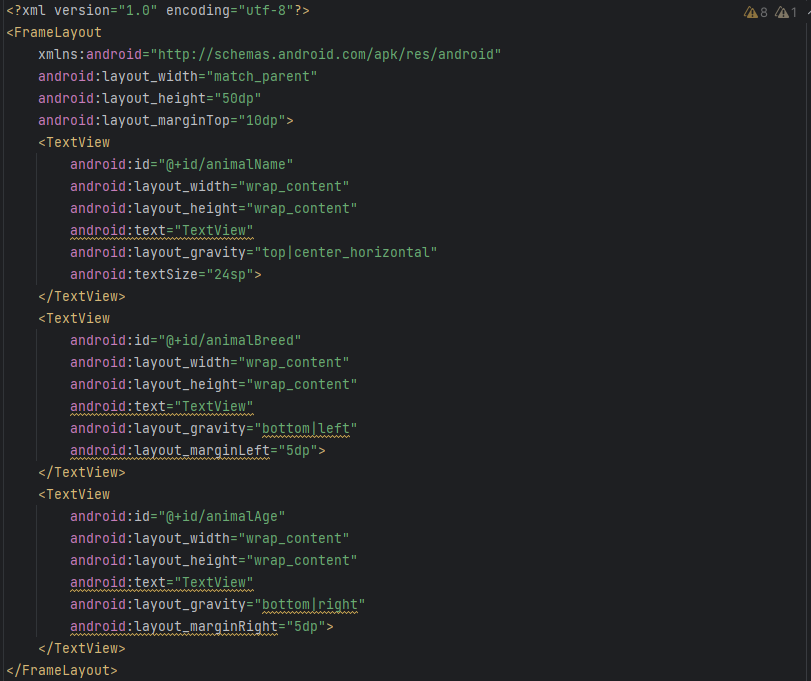


Рисунок 2.2 – Содержимое файла item.xml

Для выполнения задания мы должны расписать методы в MainActivity.java. Данный код предоставлен на рисунке 2.3.



Рисунок 2.3 – Содержимое файла MainActivity.java

На рисунке 2.4 показано как выглядит написанное нами приложение.

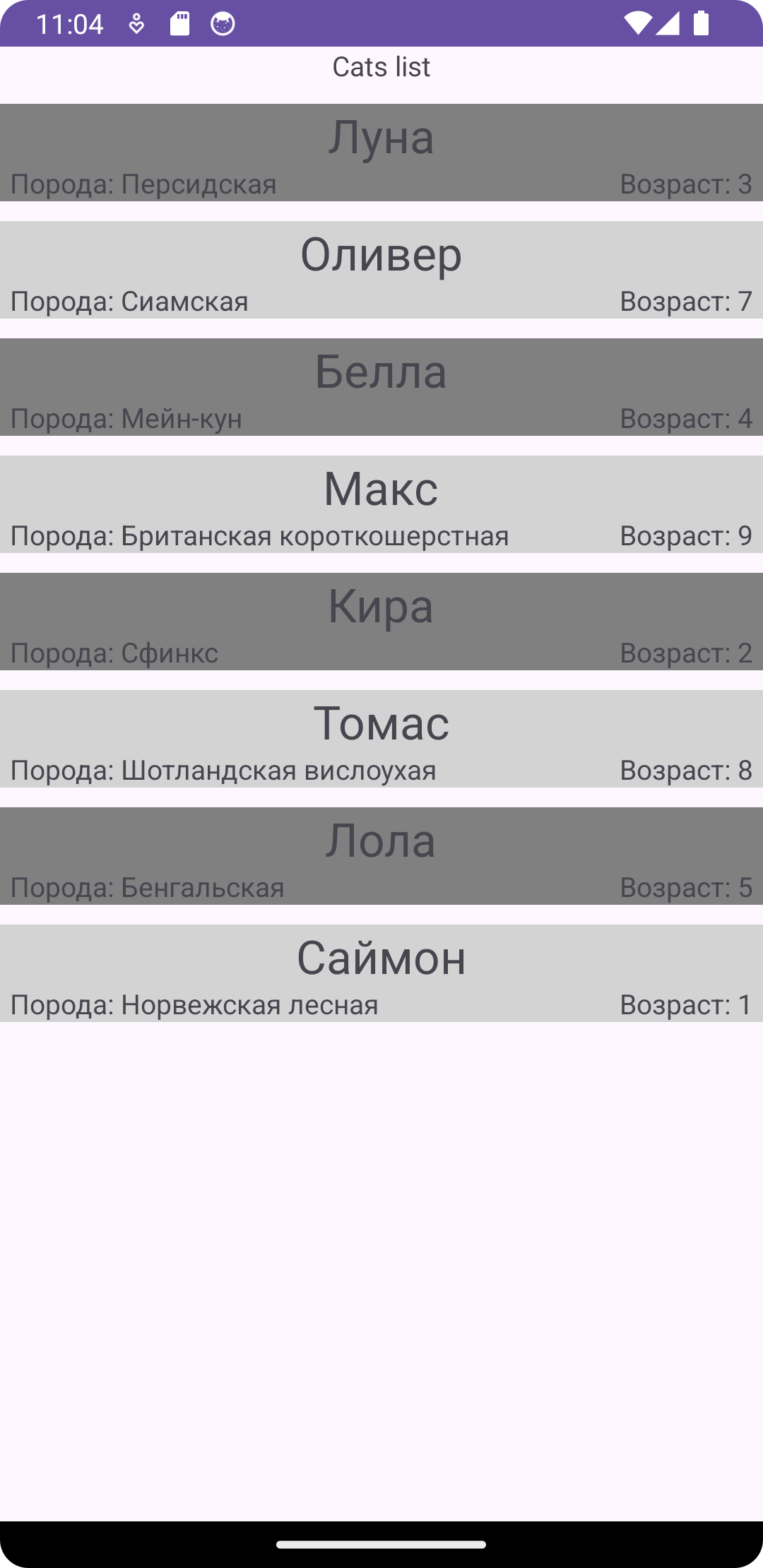


Рисунок 2.4 – Запущенное приложение на устройстве

**Задание 3**

ListView для построения списка.

Для выполнения данной работы нам необходимо изменить содержимое файла activity\_main.xml данное изменение предоставлено на рисунке 3.1.



Рисунок 3.1 – Содержимое файла activity\_main.xml

Код разметки, написанный нами в предыдущем шаге в design отображается как на рисунке 3.2

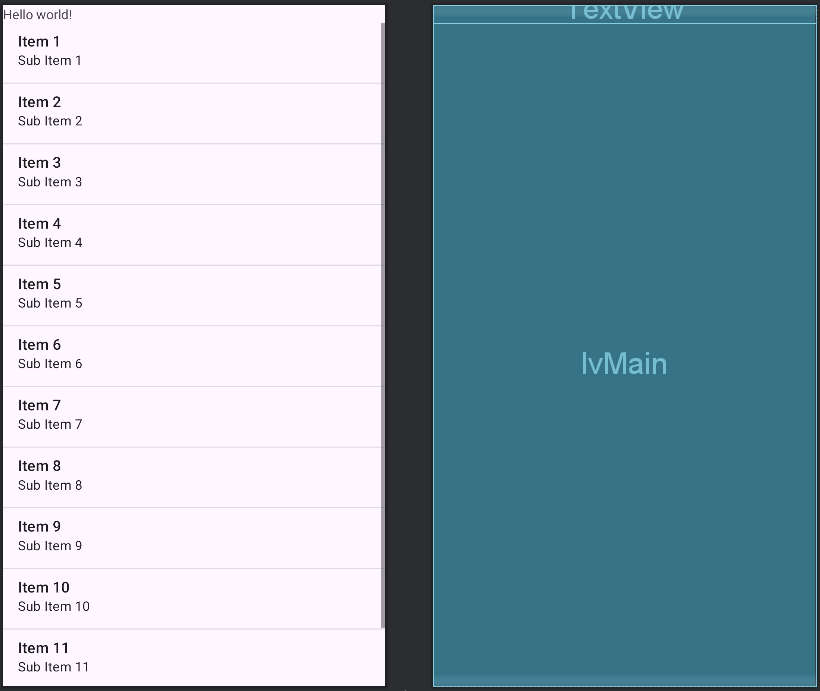


Рисунок 3.2 – Отображение написанного нами кода в design

Для выполнения задания мы должны расписать методы в MainActivity.java. Данный код предоставлен на рисунке 3.3.



Рисунок 3.3 – Содержимое файла MainActivity.java

На рисунке 3.4 показано как выглядит написанное нами приложение.

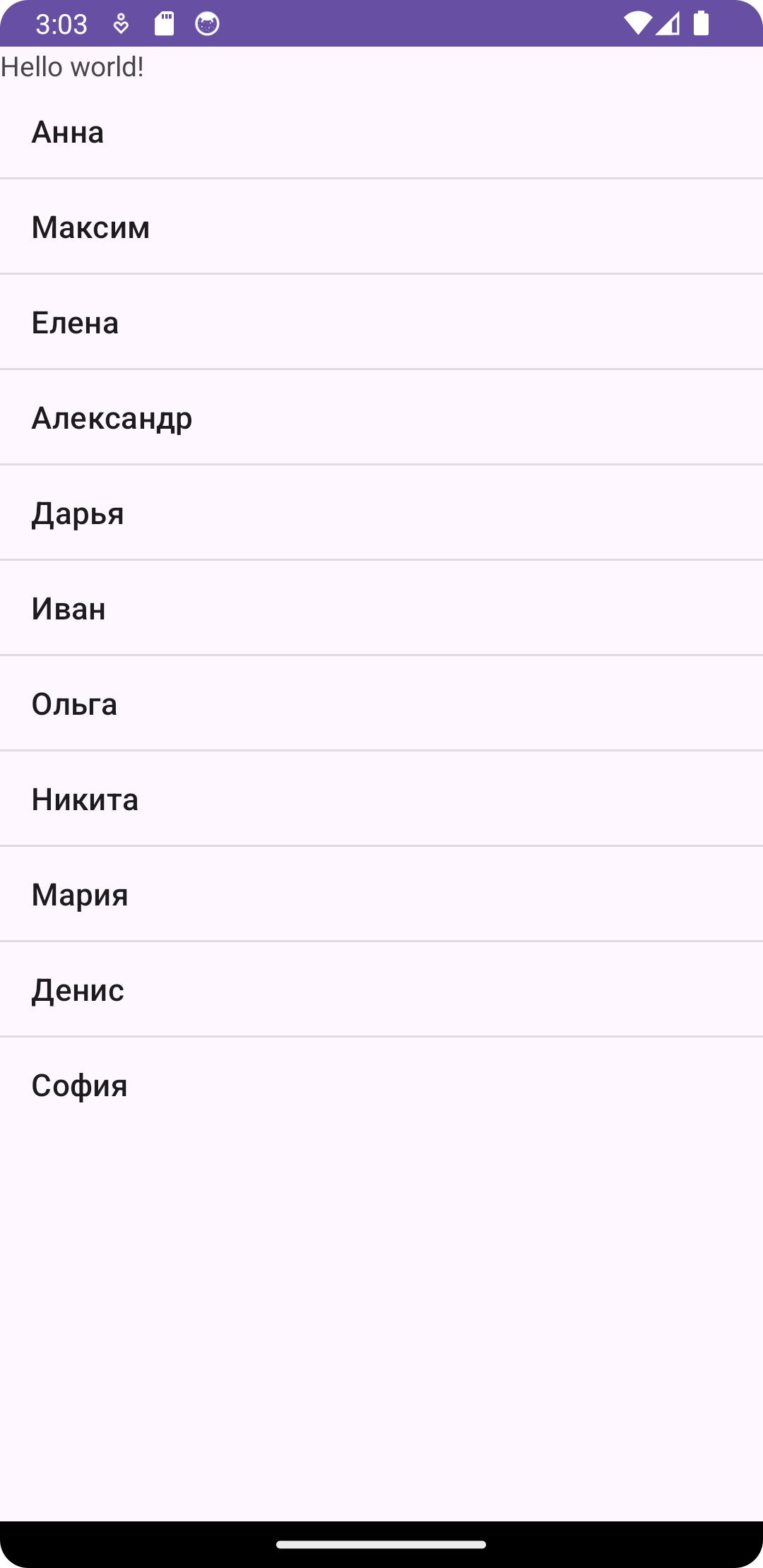


Рисунок 3.4 – Запущенное приложение на устройстве

Добавим файл item.xml который будет отвечать за элемент списка. Код данного файла предоставлен на рисунке 3.5.

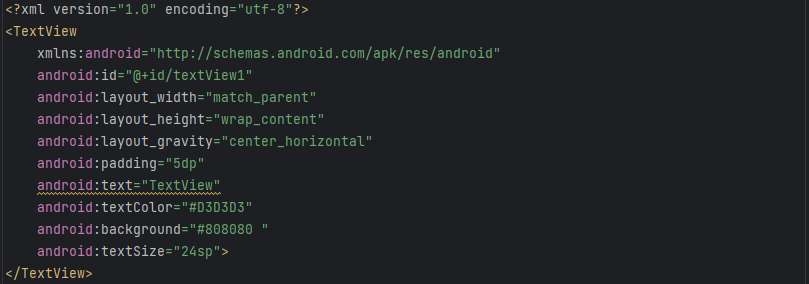


Рисунок 3.4 – Содержимое файла item.xml

На рисунке 3.6 показано как выглядит написанное нами приложение.

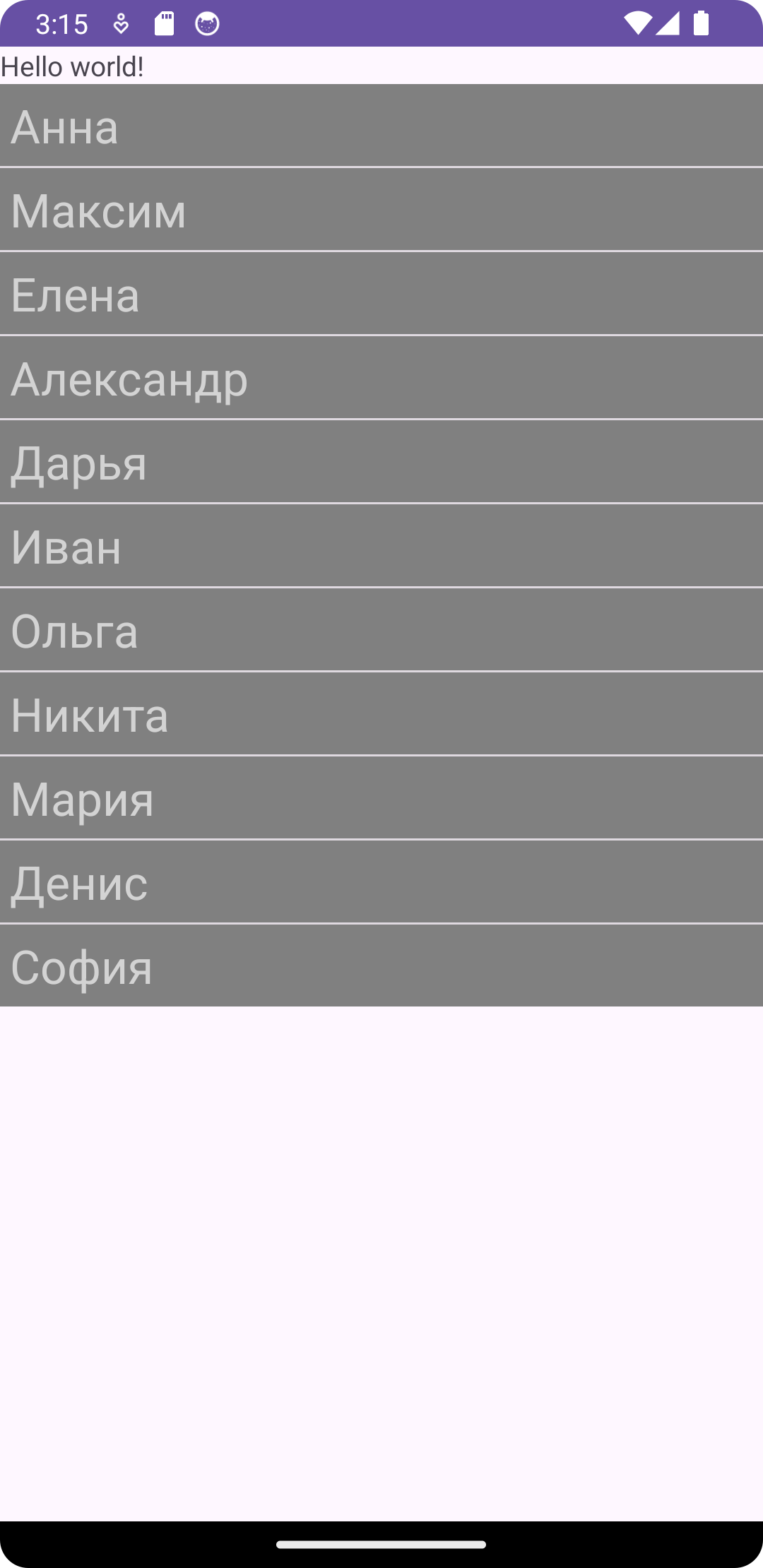


Рисунок 3.4 – Запущенное приложение на устройстве

**Задание 4**

Одиночный и множественный выбор в списке ListView.

Для выполнения данной работы нам необходимо изменить содержимое файла activity\_main.xml данное изменение предоставлено на рисунке 4.1.



Рисунок 4.1 – Содержимое файла activity\_main.xml

Для выполнения задания нам потребуется создать строковый массив в строковых ресурсах с именами людей. Данный код предоставлен на рисунке 4.2.

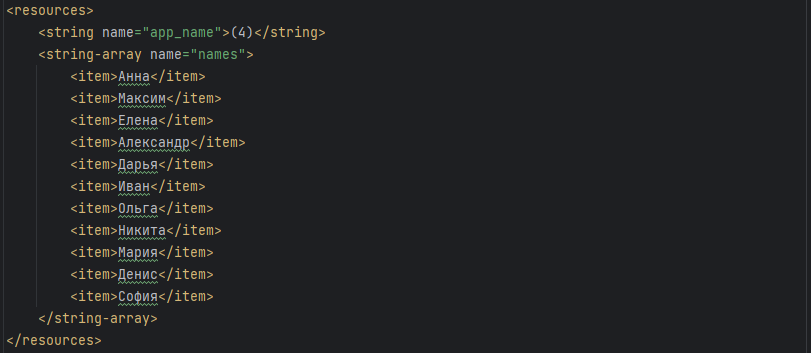


Рисунок 4.1 – Содержимое строковых ресурсов

Для выполнения задания мы должны расписать методы в MainActivity.java. Данный код предоставлен на рисунке 4.3.



Рисунок 4.3 – Содержимое файла MainActivity.java

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 4.4.

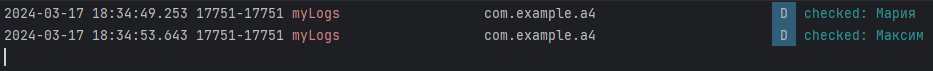


Рисунок 4.4 – Панель в дебаг консоли

На рисунке 4.5 показано как выглядит написанное нами приложение.

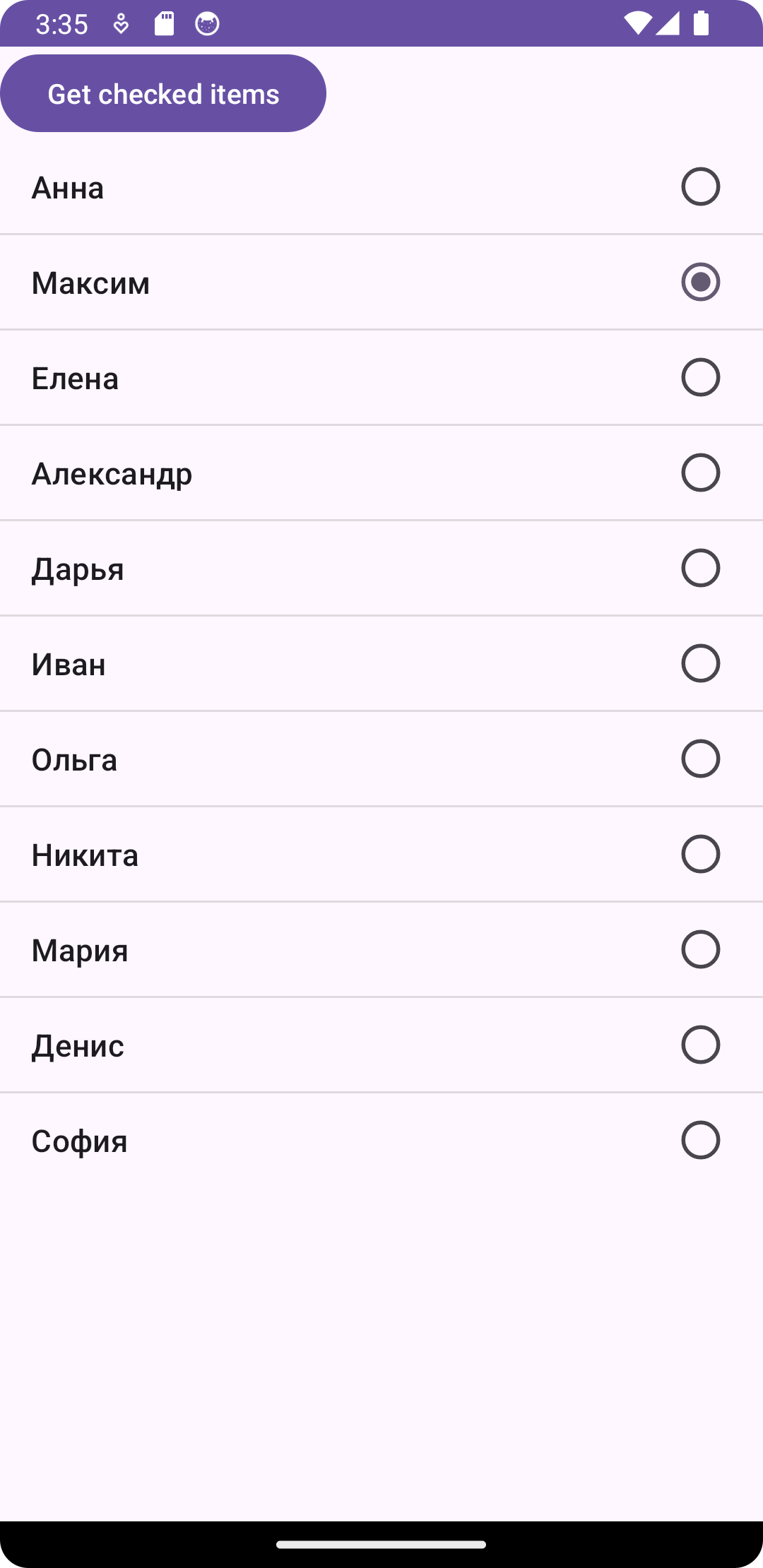


Рисунок 4.5 – Запущенное приложение на устройстве

Изменим код, написанный в MainActivity.java в предыдущем шаге, для того чтобы мы могли работать с множественным выбором. Переписанный код предоставлен на рисунке 4.6.

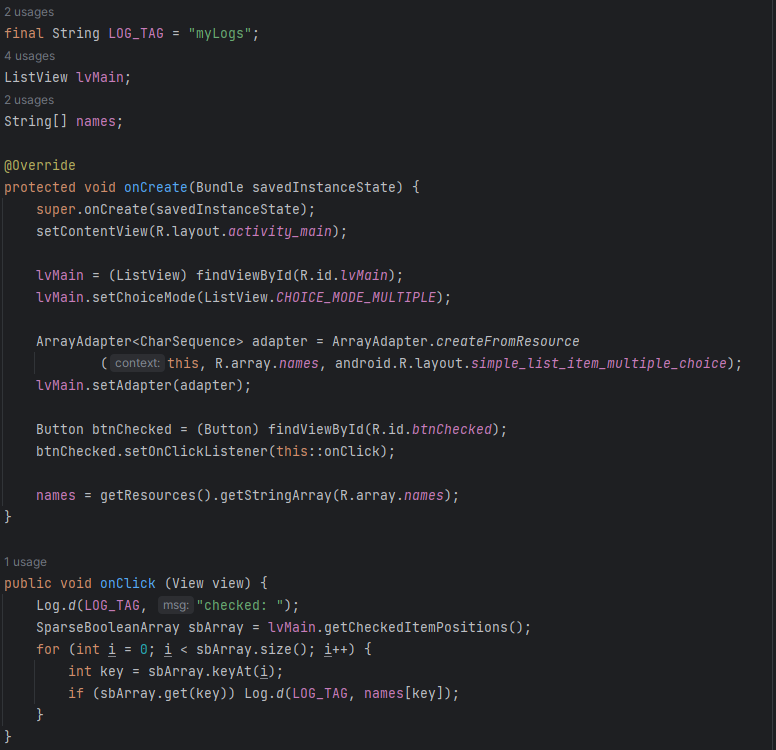


Рисунок 4.6 – Содержимое файла MainActivity.java после внесения изменений

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 4.7.

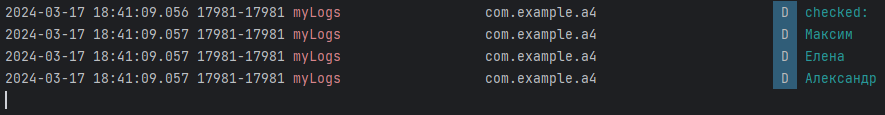


Рисунок 4.7 – Панель в дебаг консоли

На рисунке 4.8 показано как выглядит написанное нами приложение.

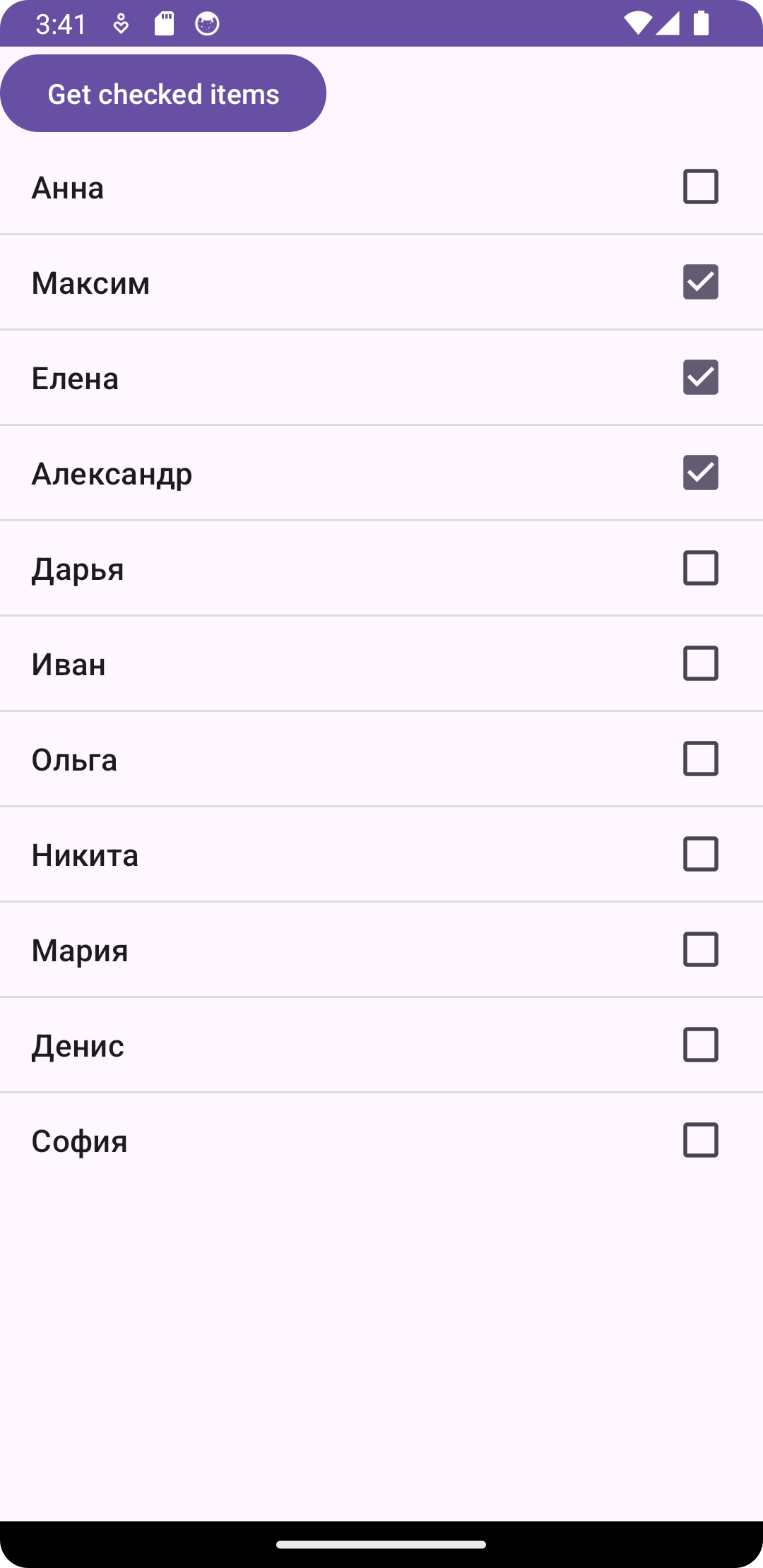


Рисунок 4.8 – Запущенное приложение на устройстве

**Задание 5**

События в ListView.

Для выполнения данной работы нам необходимо изменить содержимое файла activity\_main.xml данное изменение предоставлено на рисунке 5.1.

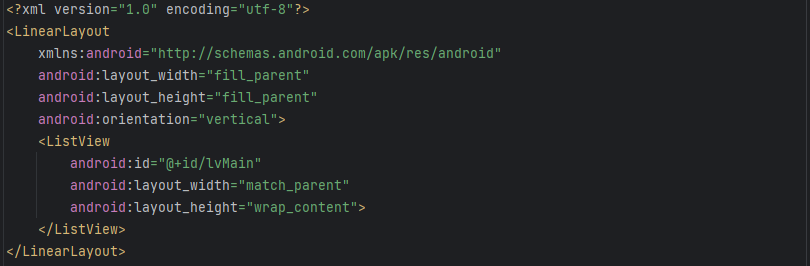


Рисунок 5.1 – Содержимое файла activity\_main.xml

Для выполнения задания нам потребуется создать строковый массив в строковых ресурсах с именами людей. Данный код предоставлен на рисунке 5.2.



Рисунок 5.2 – Содержимое строковых ресурсов

Для выполнения задания мы должны расписать методы в MainActivity.java. Данный код предоставлен на рисунке 5.3.

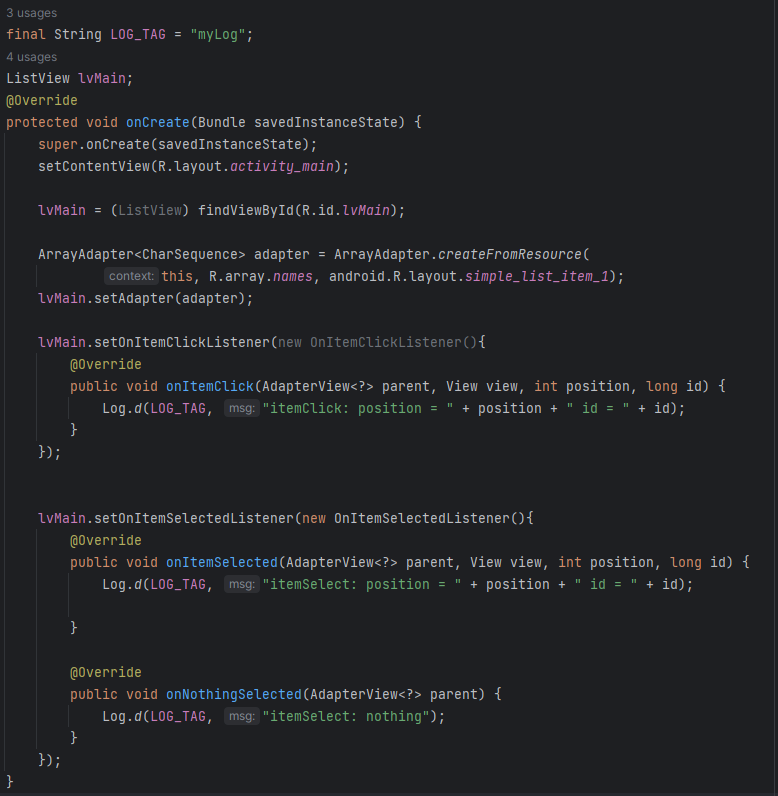


Рисунок 5.3 – Содержимое файла MainActivity.java

На рисунке 5.4 показано как выглядит написанное нами приложение.

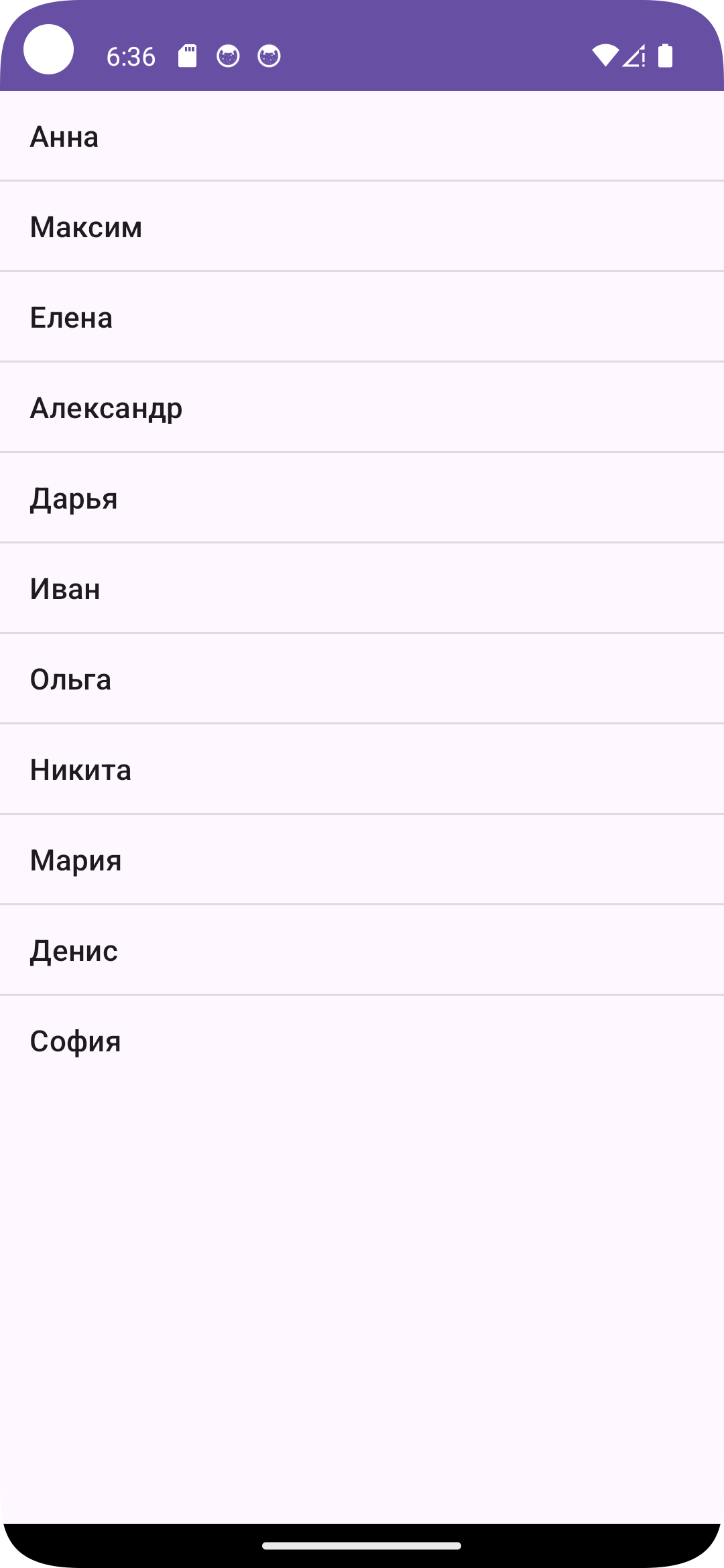


Рисунок 5.4 – Запущенное приложение на устройстве

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 5.5.

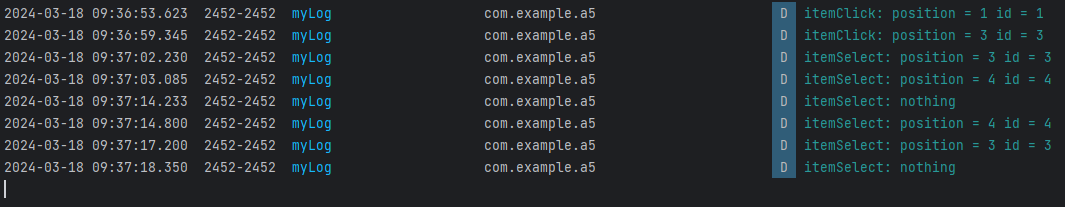


Рисунок 5.5 – Панель в дебаг консоли

Изменим код, написанный в MainActivity.java в предыдущем шаге, для того чтобы мы могли работать с множественным выбором. Переписанный код предоставлен на рисунке 5.6.

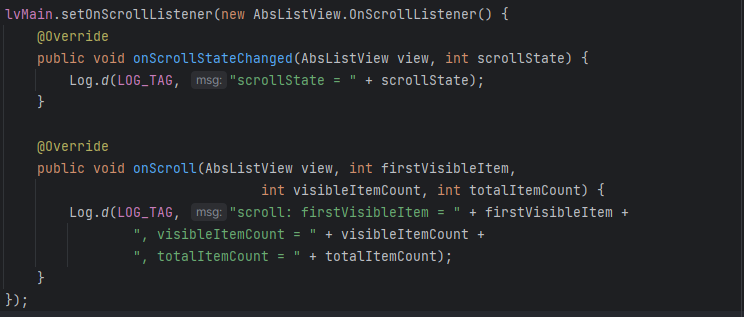


Рисунок 5.6 – Содержимое файла MainActivity.java после внесения изменений

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 5.7.

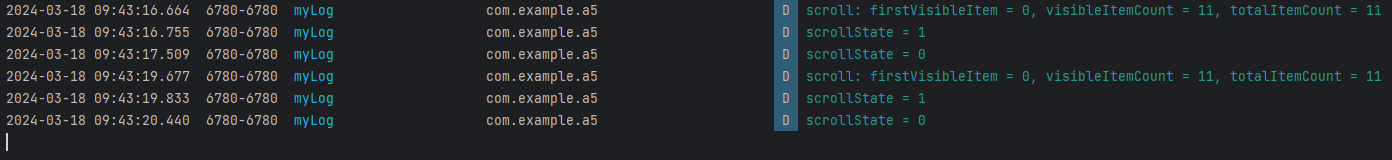


Рисунок 5.7 – Панель в дебаг консоли

**Задание 6**

Список на основе собственного адаптера.

Для выполнения данной работы нам необходимо изменить содержимое файла activity\_main.xml данное изменение предоставлено на рисунке 6.1.

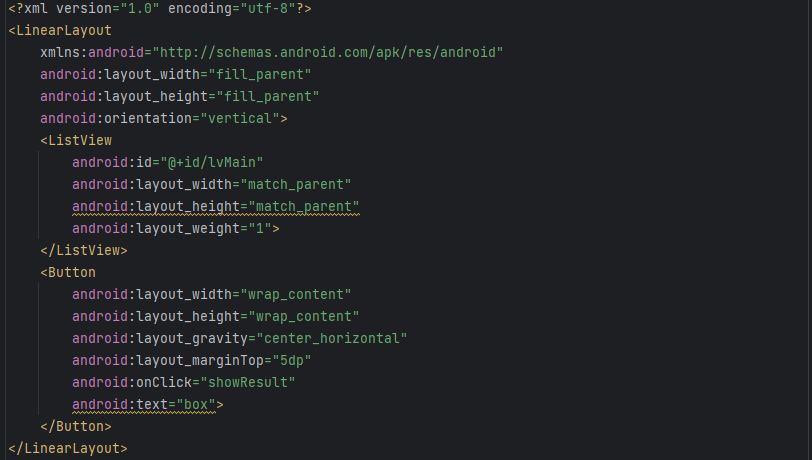


Рисунок 6.1 – Содержимое файла activity\_main.xml

Добавим файл item.xml который будет отвечать за элемент списка. Код данного файла предоставлен на рисунке 6.2.



Рисунок 6.2 – Содержимое файла item.xml

Реализуем класс которой будет отвечать за продукты нашего магазина. Код данного класса предоставлен на рисунке 6.3.

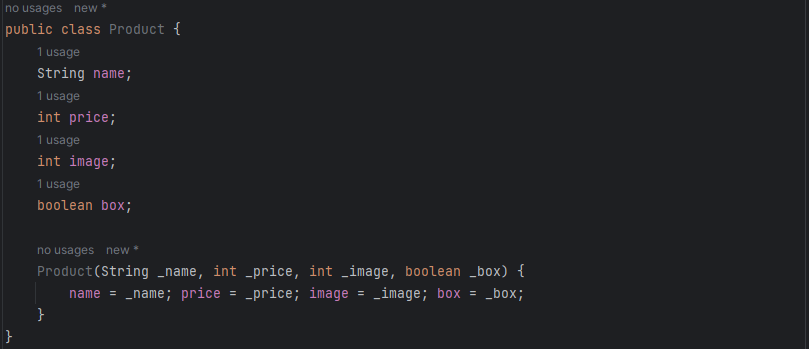


Рисунок 6.3 – Реализация класса отвечающий за продукты

Создадим класс, который будет отвечать за наш собственный адаптер. Код данного класса предоставлен на рисунке 6.4.

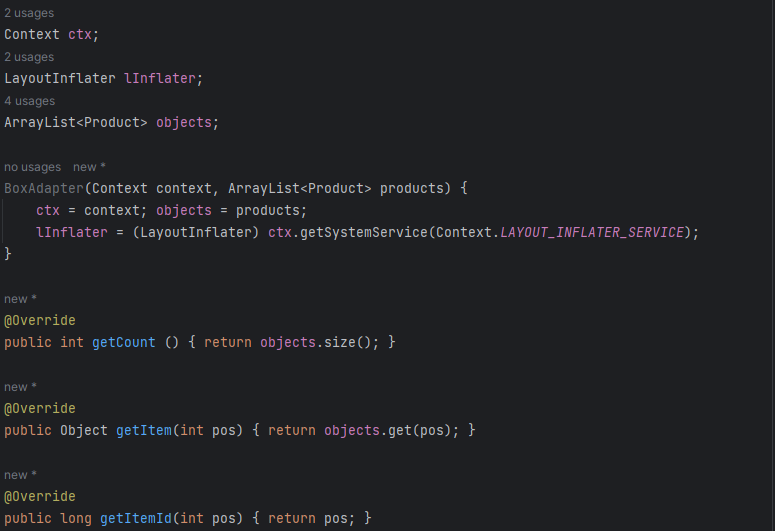


Рисунок 6.4 – Код отвечающий за наш адаптер

Для выполнения задания мы должны расписать методы в MainActivity.java. Данный код предоставлен на рисунке 6.5.

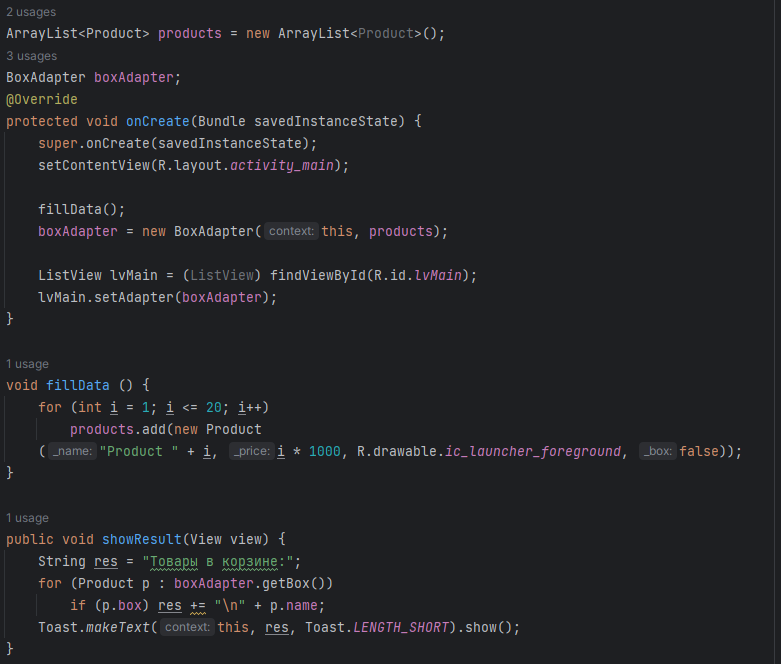


Рисунок 6.5 – Содержимое файла MainActivity.java

На рисунке 6.6 показано как выглядит написанное нами приложение.

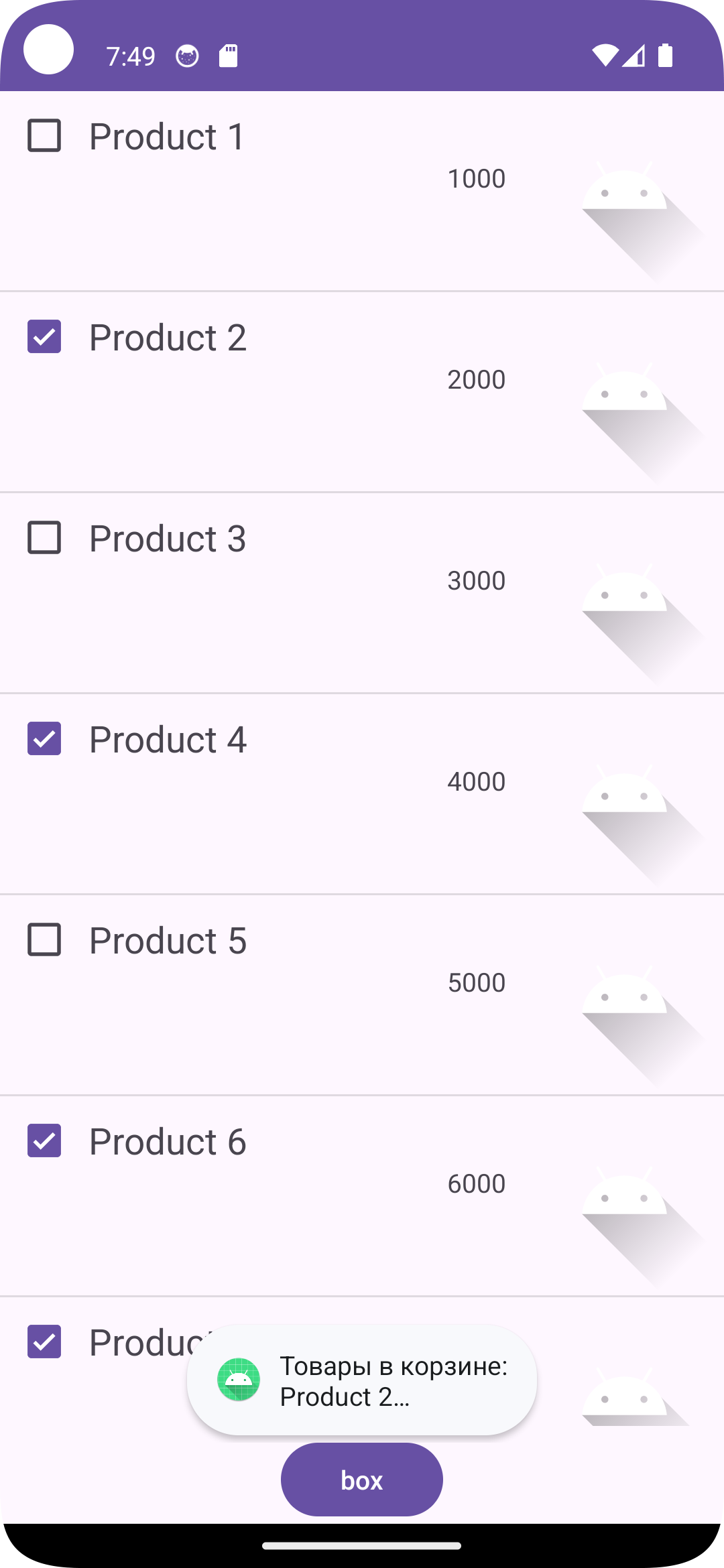


Рисунок 6.6 – Запущенное приложение на устройстве

**Задание 7**

Header и Footer в списках.

Для выполнения данного задания первоначально нам необходимо добавить два поля в строковых ресурсах. Данное добавление продемонстрированно на рисунке 7.1.

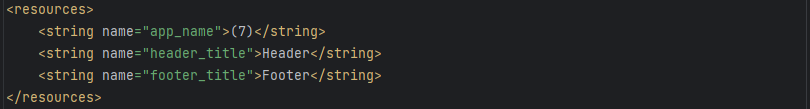


Рисунок 7.1 – Содержимое файла strings.xml

Также нам потребуется реализовать xml разметку для нашего header. Код реализации предоставлен на рисунке 7.2.



Рисунок 7.2 – Содержимое файла header.xml

Также нам потребуется реализовать xml разметку для нашего footer. Код реализации предоставлен на рисунке 7.3.

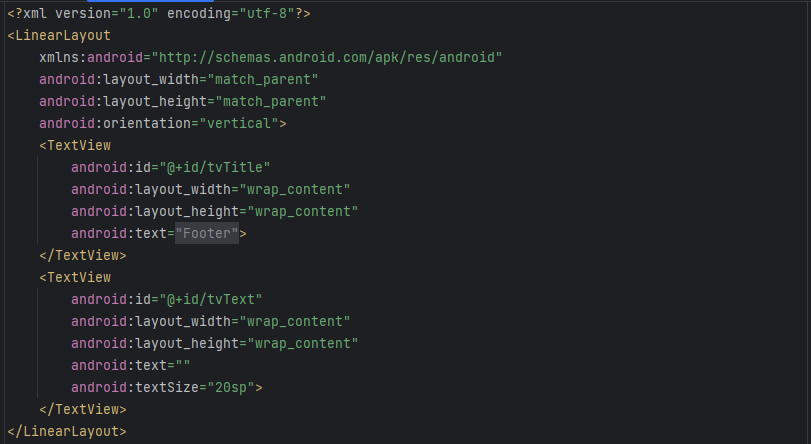


Рисунок 7.3 – Содержимое файла header.xml

Для выполнения данного задания мы должны дополнить код нашей программы реализованного в задании 6. Добавленный код предоставлен на рисунке 7.4.

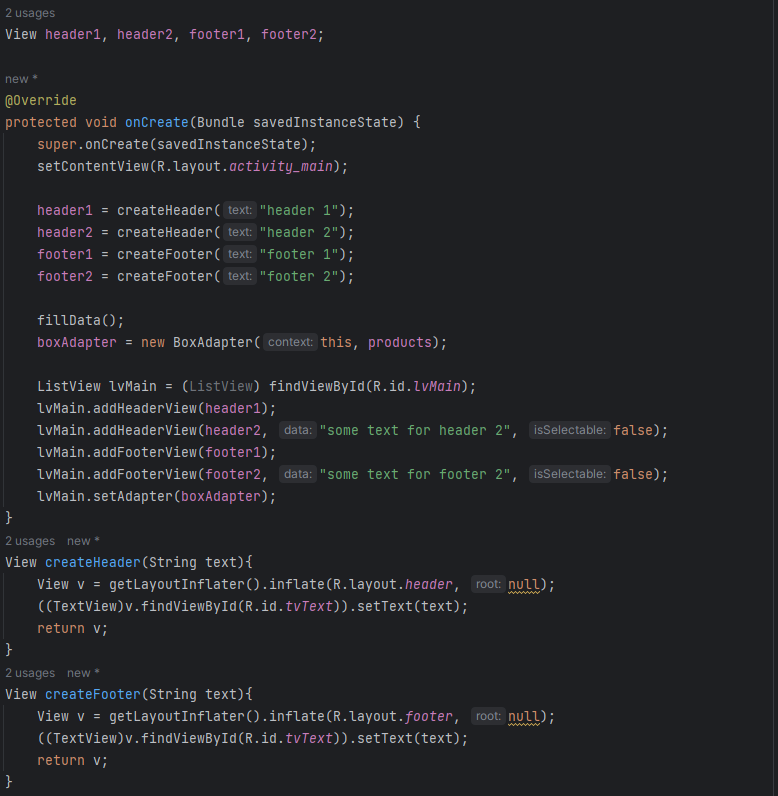


Рисунок 7.4 – Часть кода, которая была изменена в файле MainActivity.java

На рисунке 5.4 показано как выглядит написанное нами приложение.

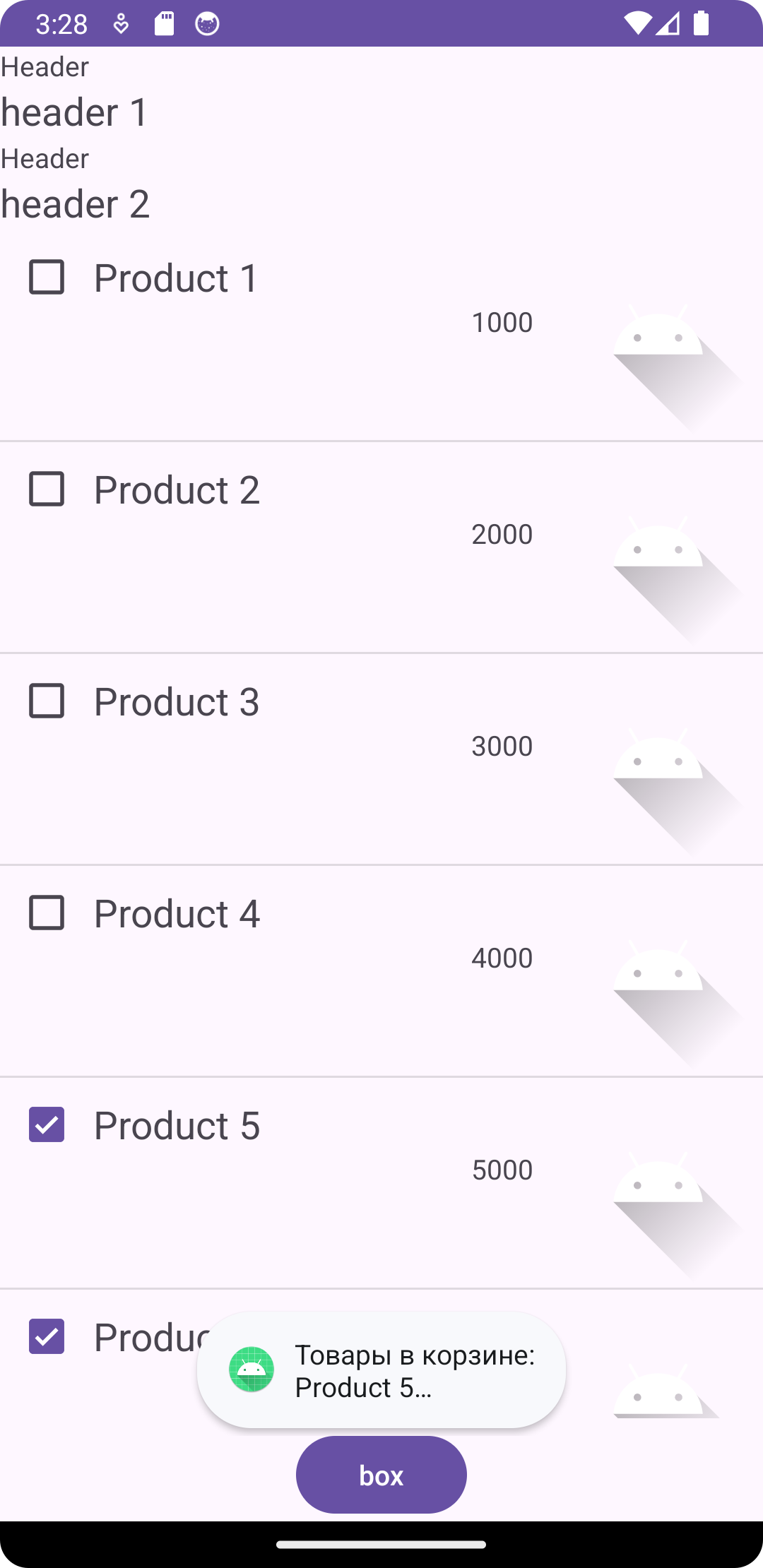


Рисунок 5.4 – Запущенное приложение на устройстве

**Задание 8**

Контейнер Parcel и интерфейс Parcelable для передачи данных между Activity.

Для выполнения задания мы должны расписать методы в MainActivity.java. Данный код предоставлен на рисунке 8.1.



Рисунок 8.1 – Содержимое файла MainActivity.java

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 8.2.

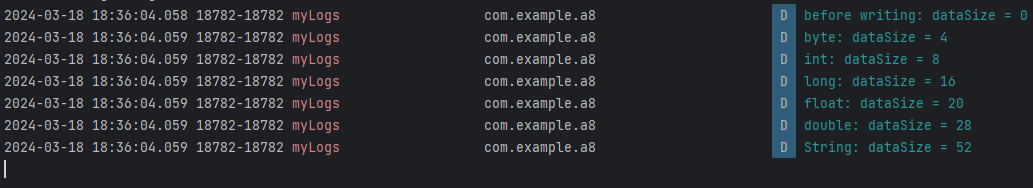


Рисунок 8.3 – Панель в дебаг консоли

Изменим код, написанный в MainActivity.java в предыдущем шаге, для того чтобы мы могли работать с множественным выбором. Переписанный код предоставлен на рисунке 8.4.



Рисунок 5.6 – Содержимое файла MainActivity.java после внесения изменений

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 8.5.



Рисунок 8.5 – Панель в дебаг консоли

**Задание 9**

Для решения данного задания мы должны добавить строковые ресурсы. Демонстрация их добавления предоставлена на рисунке 9.1.

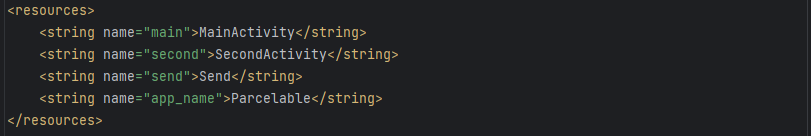


Рисунок 9.1 – Содержимое файла строковых ресурсов

Нам необходимо изменить содержимое файла activity\_main.xml данное изменение предоставлено на рисунке 9.2.

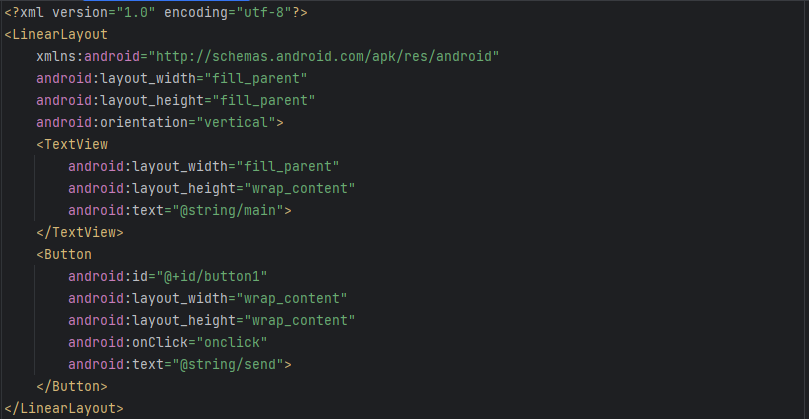


Рисунок 9.2 – Содержимое файла activity\_main.xml

Добавим новый класс MyObject, описывающий объект для передачи (файл MyObject.java). Реализация данного кода предоставлена на рисунке 9.3.



Рисунок 9.3 – Код отвечающий за объект для передачи

Также нам необходимо создать второй activity. Его реализация предоставлена на рисунке 9.4.



Рисунок 9.4 – Код, отвечающий за второй Activity

Реализуем класс, отвечающий за второе Activity. Он предоставлен на рисунке 9.5.

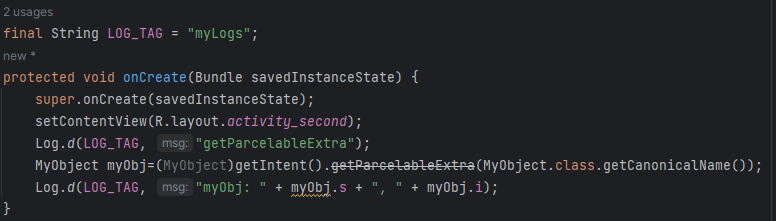


Рисунок 9.5 – Класс, отвечающий за второй Activity

Реализуем класс, отвечающий за первое Activity. Он предоставлен на рисунке 9.6.



Рисунок 9.5 – Класс, отвечающий за перавое Activity

В консоли LogCat у нас вывелся текст, предоставленный на рисунке 9.6.

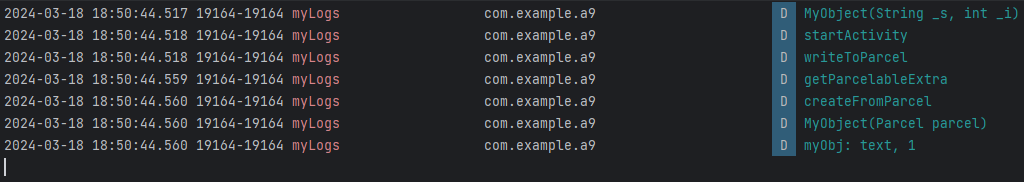


Рисунок 9.6 – Панель в дебаг консоли

**ЧАСТЬ 2**

Для выполнения данной работы нам потребуется реализовать первое Activity в котором будут находиться товары, которые пользователь может приобрести. Разметка данного Activity предоставлена на рисунке 10.1.

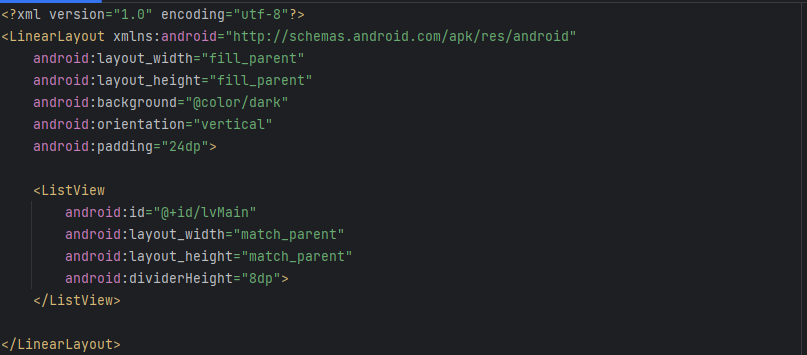


Рисунок 10.1 – Разметка первого Activity

Нам необходимо реализовать разметки для нашего header. Ее реализация предоставлена на рисунках 10.2.

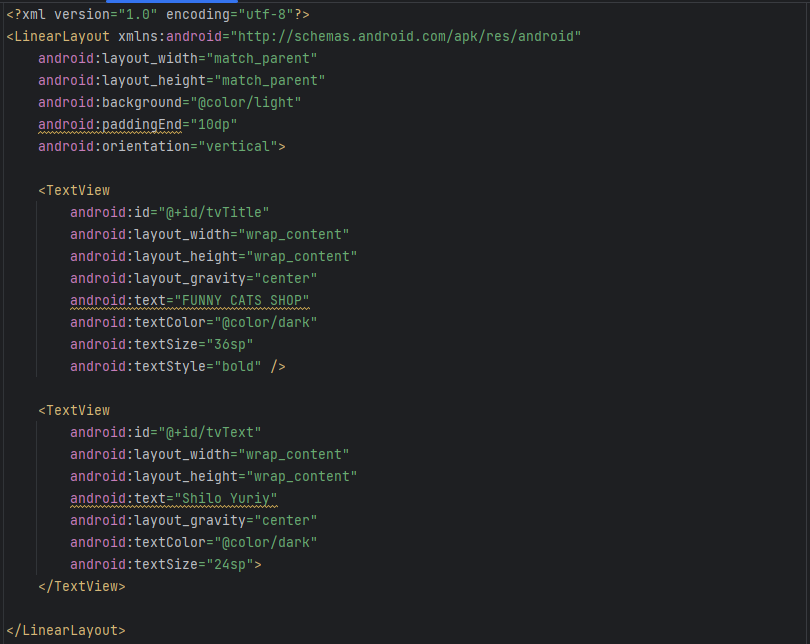


Рисунок 10.1 – Разметка header

Реализуем класс, который будет отвечать за взаимодействие с первым Activity. Код данного класса предоставлен на рисунке 10.3

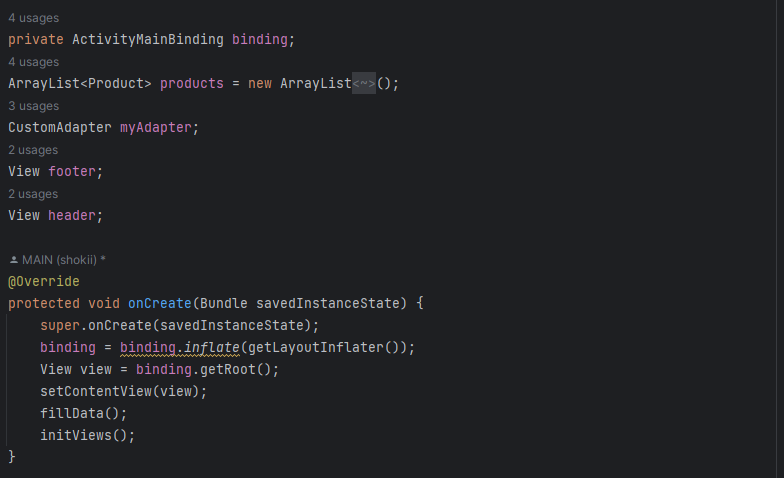


Рисунок 10.3 – Часть кода, отвечающего за первое Activity

Создадим класс, который будет отвечать за продукты нашего магазина рисунок 10.4.



Рисунок 10.4 – Часть кода, отвечающего за продукты магазина

Нам необходимо создать свой собственный адаптер. Его код предоставлен на рисунке 10.5.

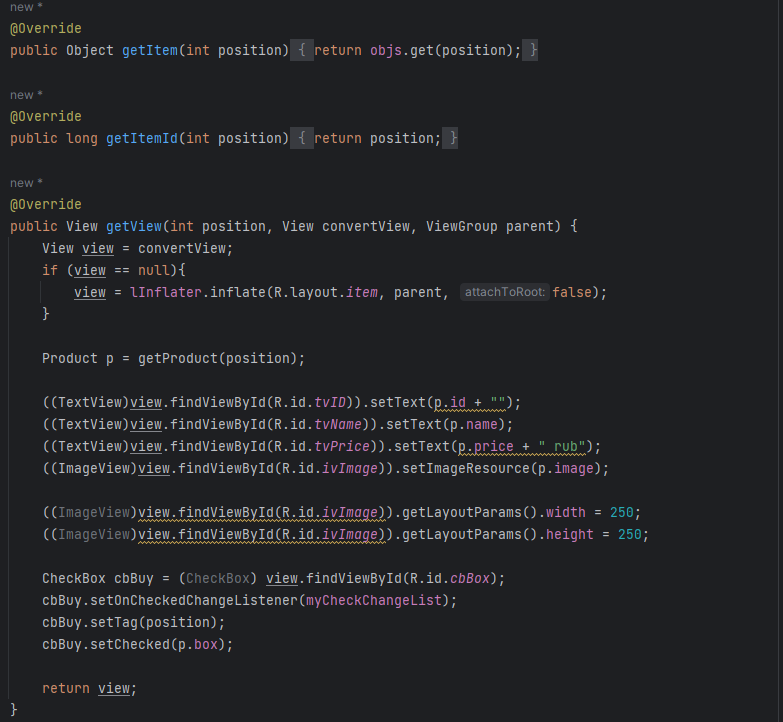


Рисунок 10.5 – Часть кода, отвечающего за наш собственный адаптер

Реализуем разметку, которая будет отвечать за элемент нашего списка. Данная разметка предоставлена на рисунке 10.6

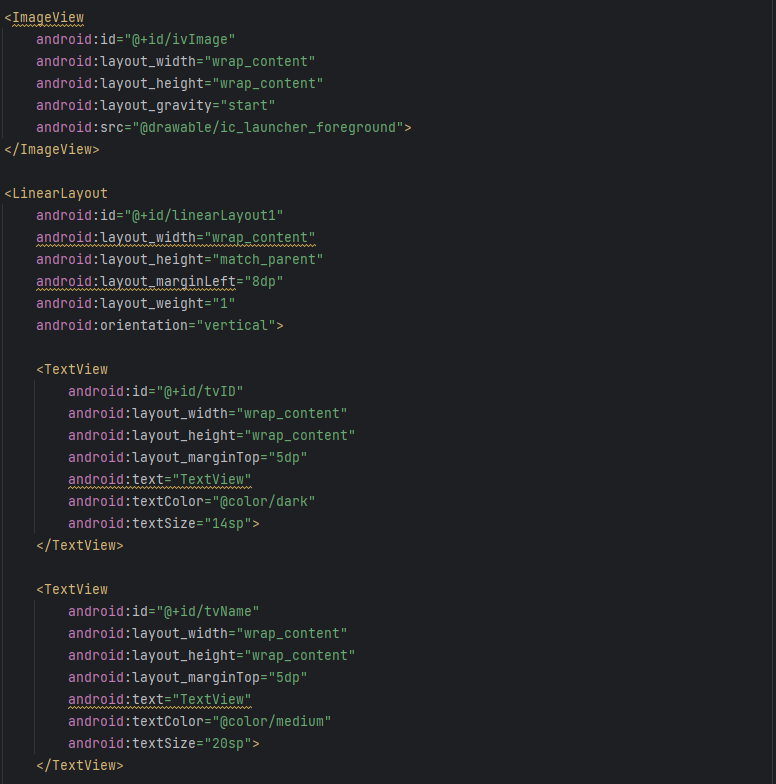


Рисунок 10.6 – Часть разметки, отвечающей за один из элементов списка

Для выполнения задания нам необходимо реализовать второе Activity. Код его разметки предоставлен на рисунке 10.7.

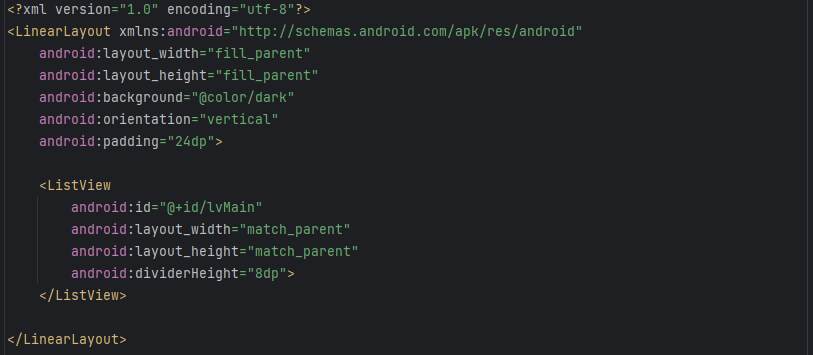


Рисунок 10.7 – Разметка второго Activity

Создадим класс, который будет отвечать за взаимодействие со вторым Activity. Код данного класса предоставлен на рисунке 10.8

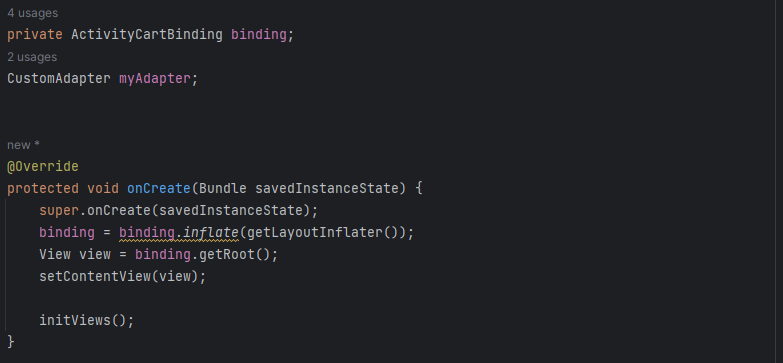


Рисунок 10.8 – Часть кода, отвечающего за первое Activity

Приложение, запущенное на физическом устройстве предоставлено на рисунке 10.9.

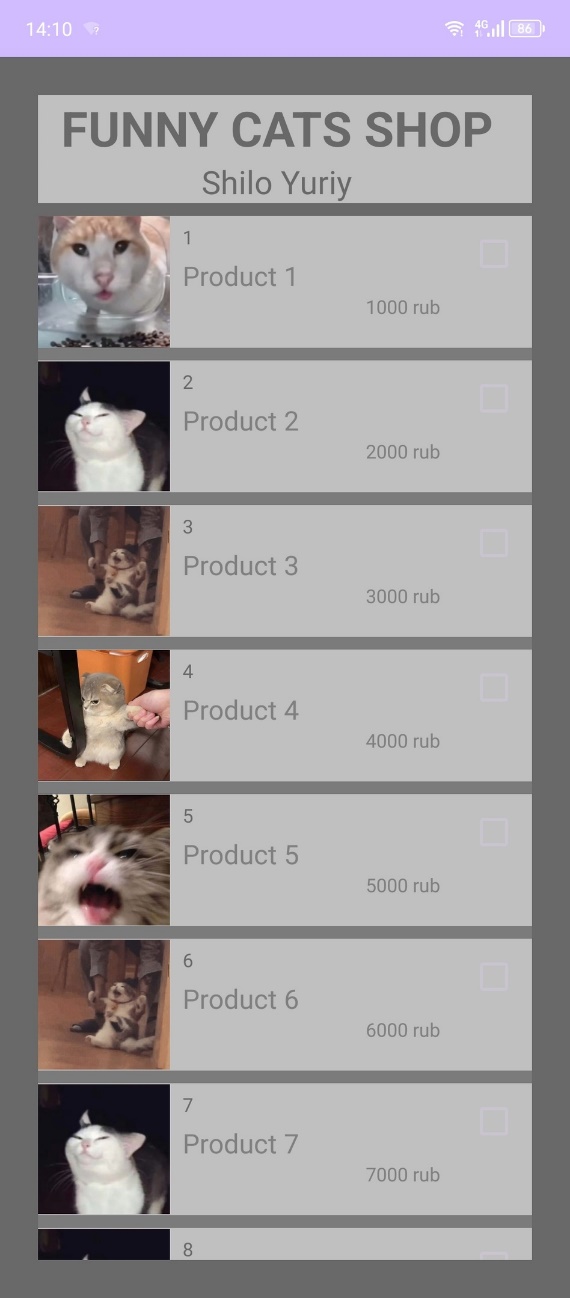


Рисунок 10.9 – Приложение запущенное на физическом устройстве

**ВЫВОД**

В результате выполнения данной практической работы мы научились создавать кастомизированные списков на основе адаптера и реализовывать механизмы передачи объектов между Activity в Android-приложениях.