**Специальность «Вычислительная техника**

**и программное обеспечение»**



**Лабораторная №10**

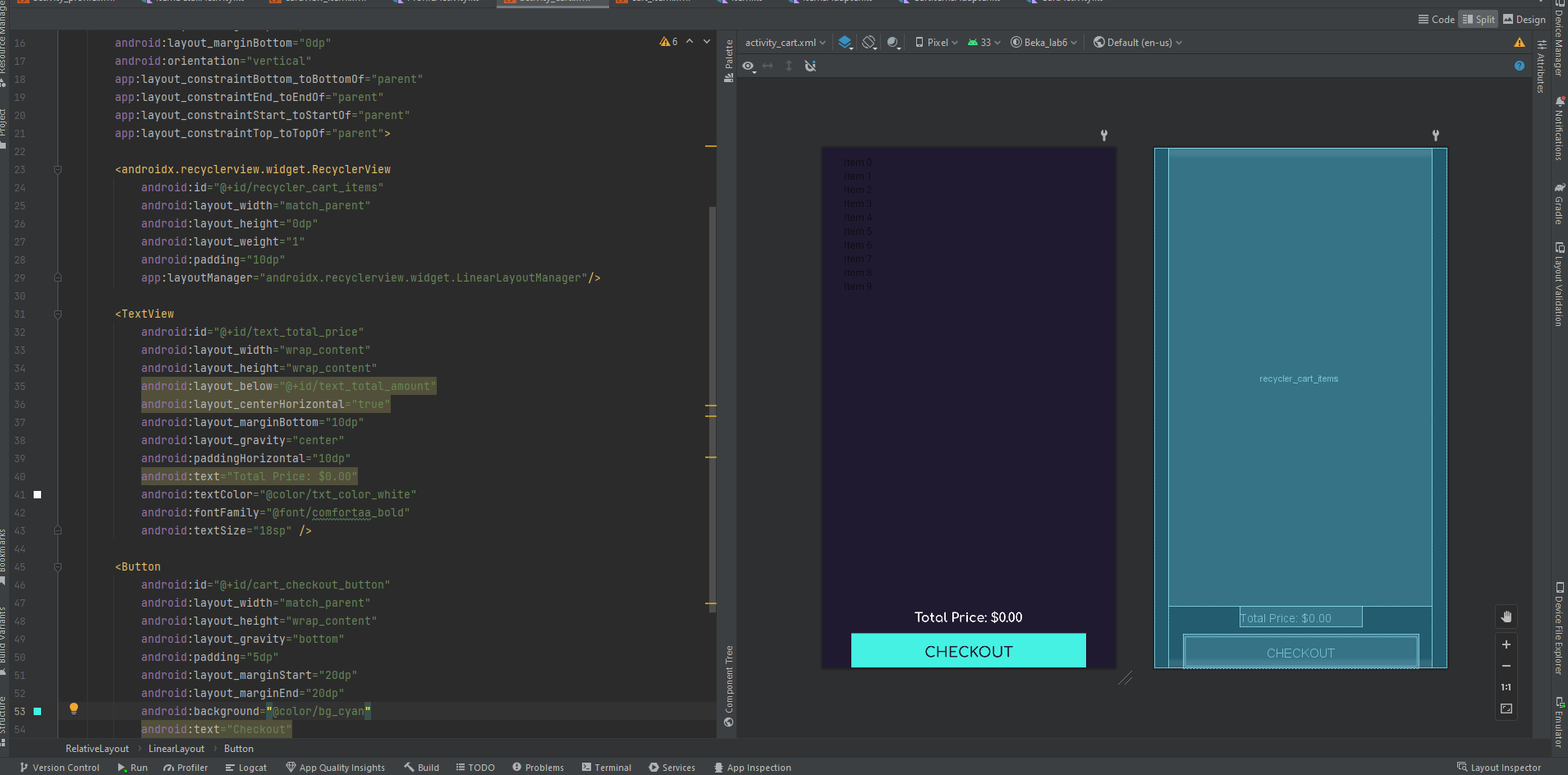
**По дисциплине:**

«Разработка мобильных приложений»

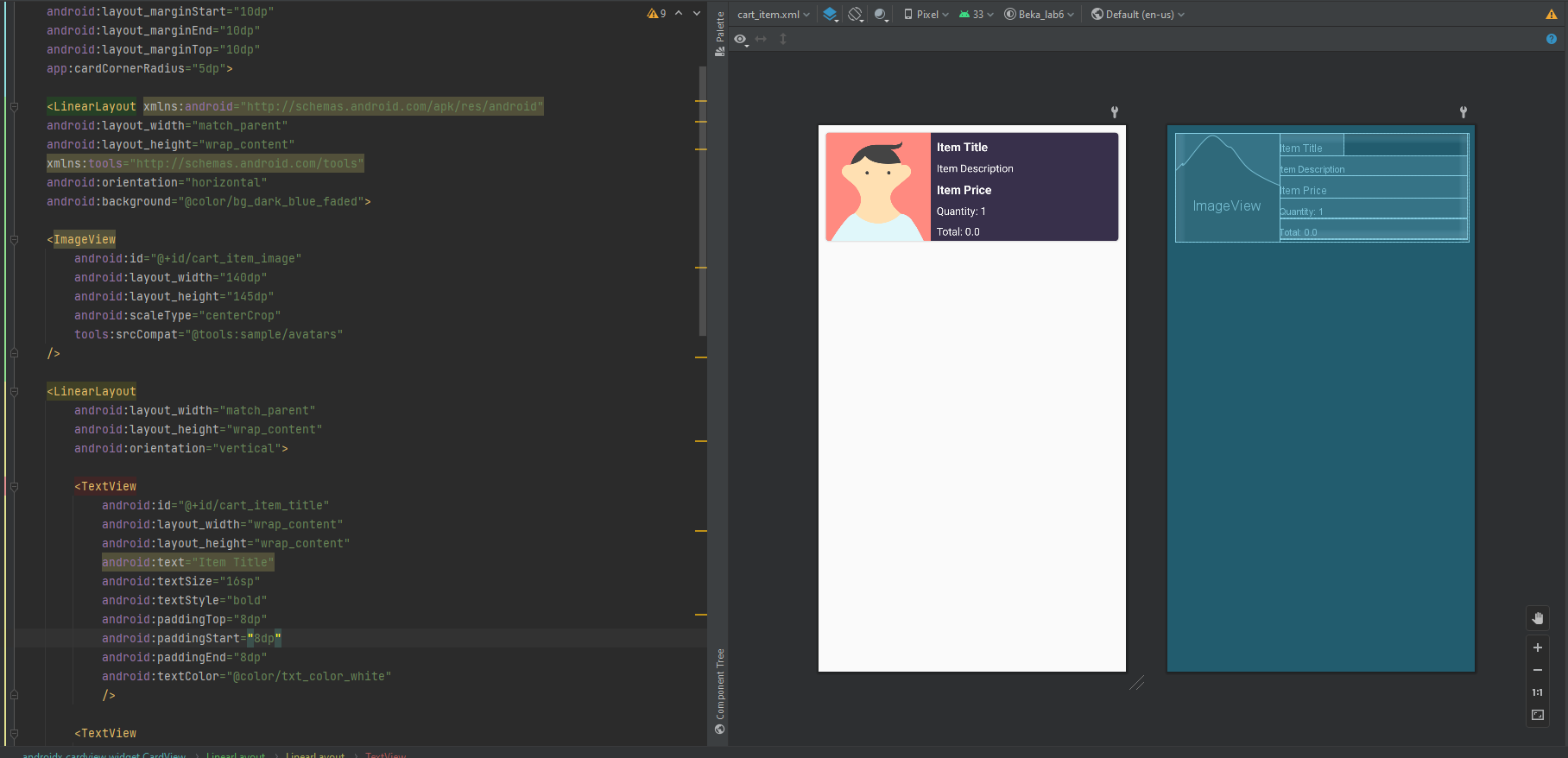
**Выполнил:** Мурат Бекежан

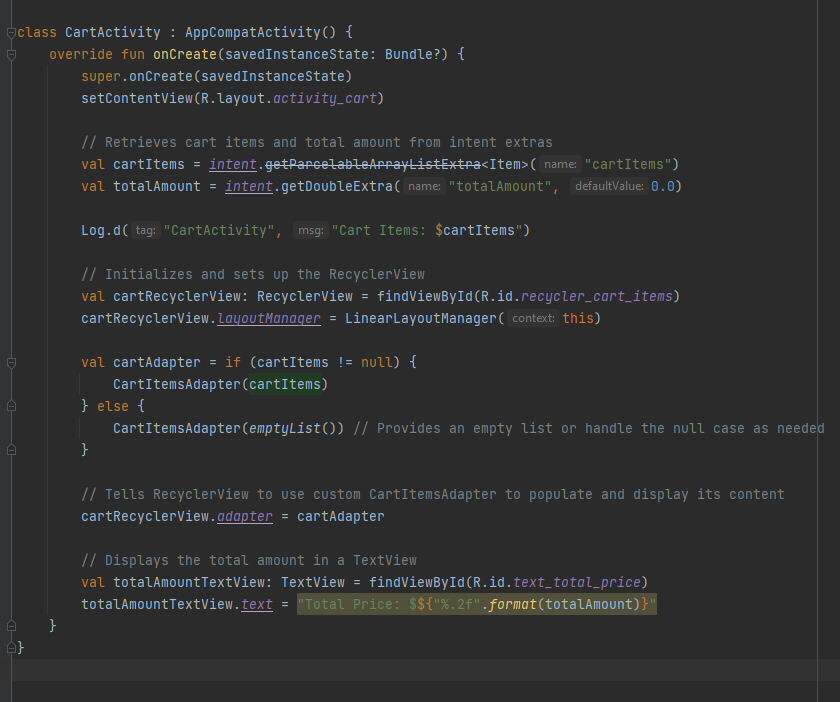
ВТиПО 21\_2у

Я решил добавить функционал корзины. Куда я просто добавляю товары и вижу итоговую сумму с отображением добавленных товаров. В самом конце я добавлю код каждого активити и класса, без drawable и других настроек проекта.

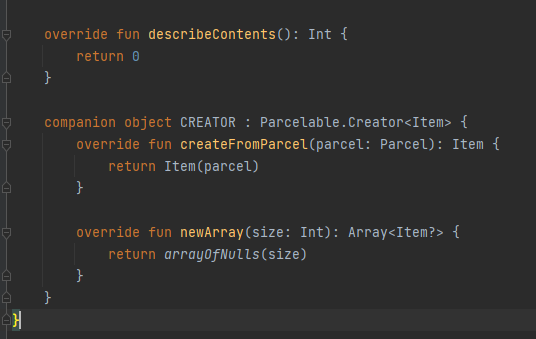
Для этого я создал activity\_cart.xml  


И для отображения товара, cart\_item.xml



Класс CartActivity, отвечает за отображение товаров в корзине. Он получает лист item с помощью Intent и интерфейса Parcel между ActivityProfile и CartActivity. Также настраивает отображение элементов в RecyclerView и переходит на cart\_activity.xml только если есть товары в корзине. Также содержит переменную для оторбражения итоговой стоимости  


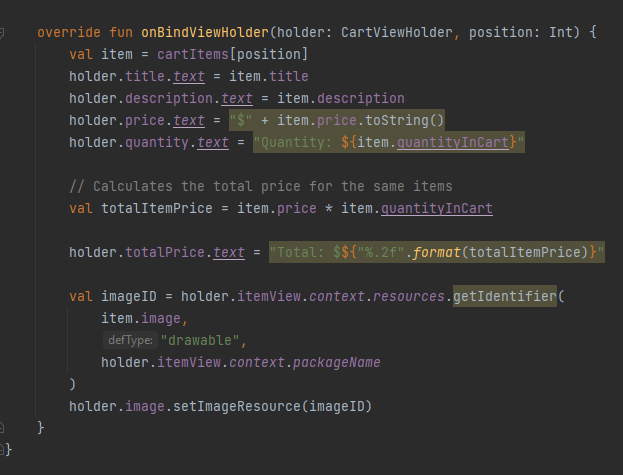
Класс Item, чтобы передавать данные атрибутов объекта Item, необходимо было использовать интерфейс Parcel, куда записывались данные из объекта Item



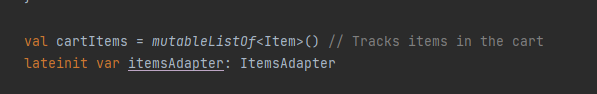
Для обработки данных создал класс CartItemsAdapter, задача этого класса такая же, как и у ItemsAdapter – связать данные Item с RecyclerView.



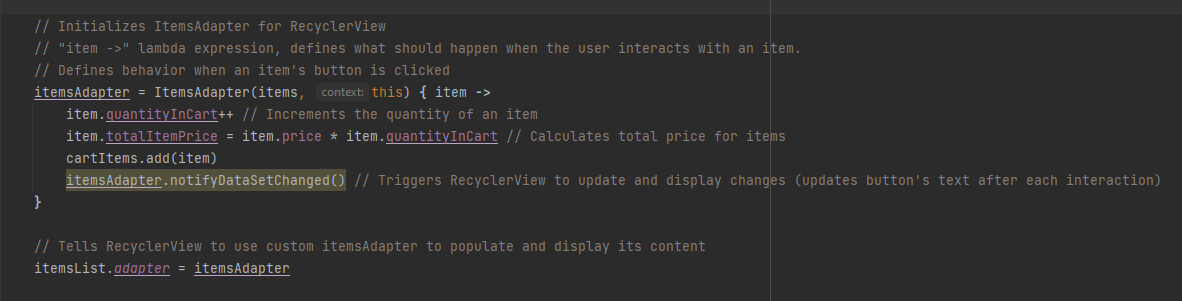
Здесь я указываю, что необходимо отобразить для cartItem, а также подсчитываю сумму для товаров, в том числе и дубликатов.



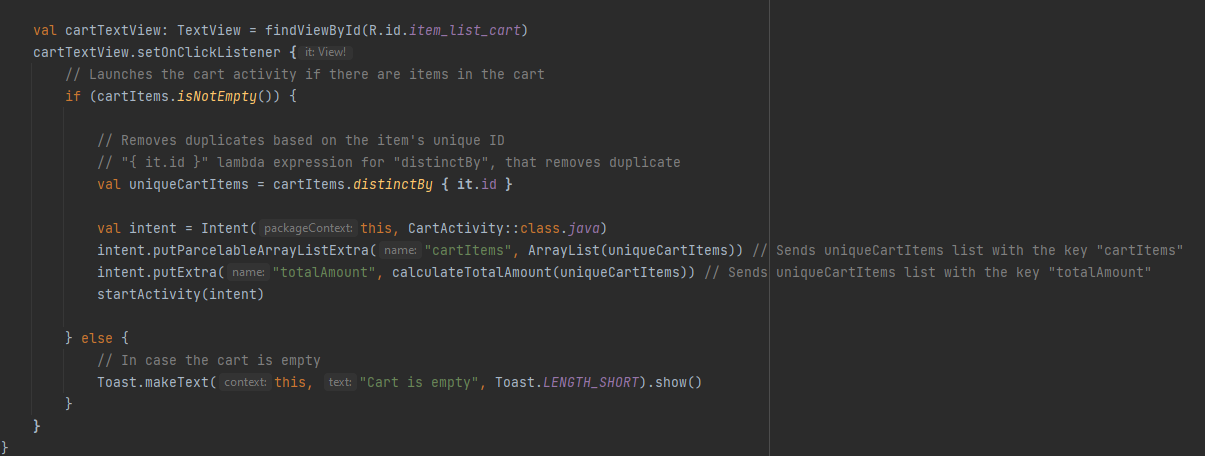
Возвращаюсь в класс ProfileActivity, где у меня отображается список товаров для добавлению в корзину. Здесь я инициализировал переменные для отслеживания товаров в корзине, т.е. храню их в списке. ItemsAdapter необходимо инициализировать после onCreate(), потому как я сначала создал список элементов и настроил RecyclerView и чтобы получить данные, а не значения null или различные вылеты, как и было изначально.



Ниже в коде я инициализирую itemsAdapter, где я провожу необходимые подсчеты итоговой суммы и изменения в item



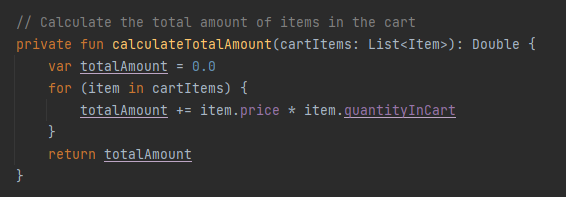
Здесь я запускаю cart\_activity.xml при нажатии на иконку корзины. Также удаляю дубликаты товаров, чтобы в случае, если один и тот же товар был добавлен более 1го раза, он не показывался в корзине как 2 отдельных товара, вместо этого, товар будет один, однако будет показано сколько было добавлено этого товара и его общая сумма. Аналогия стаку предметов в играх.  
После этого, эти uniqueCartItems запаковываются в Parcel и отправятся в CartActivity



Вот пример для понимания (3 одинаковых товара, добавленные в корзину)

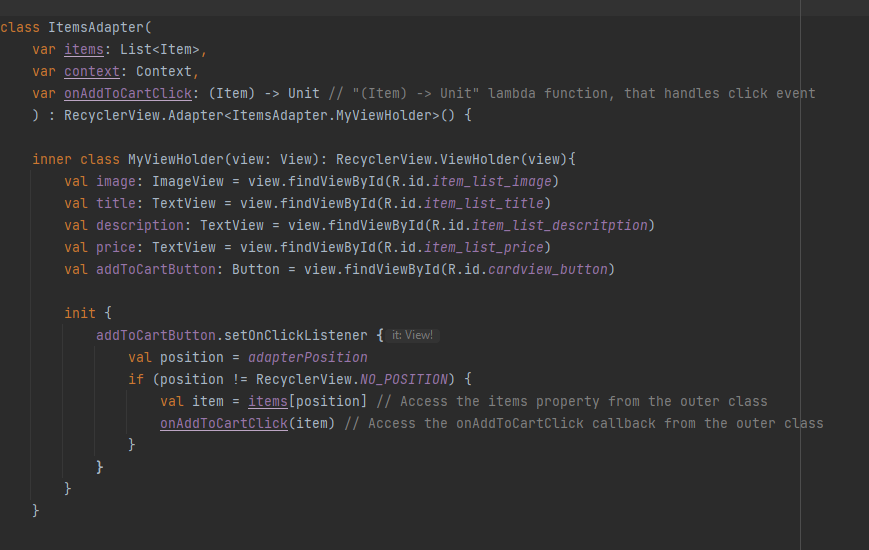


Функция подсчета итоговой суммы

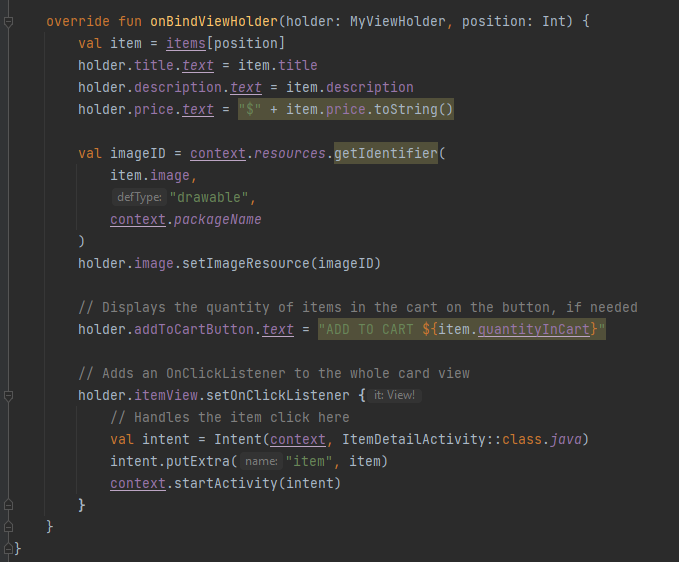


Теперь, остановлюсь подробнее у класса ItemsAdpater.

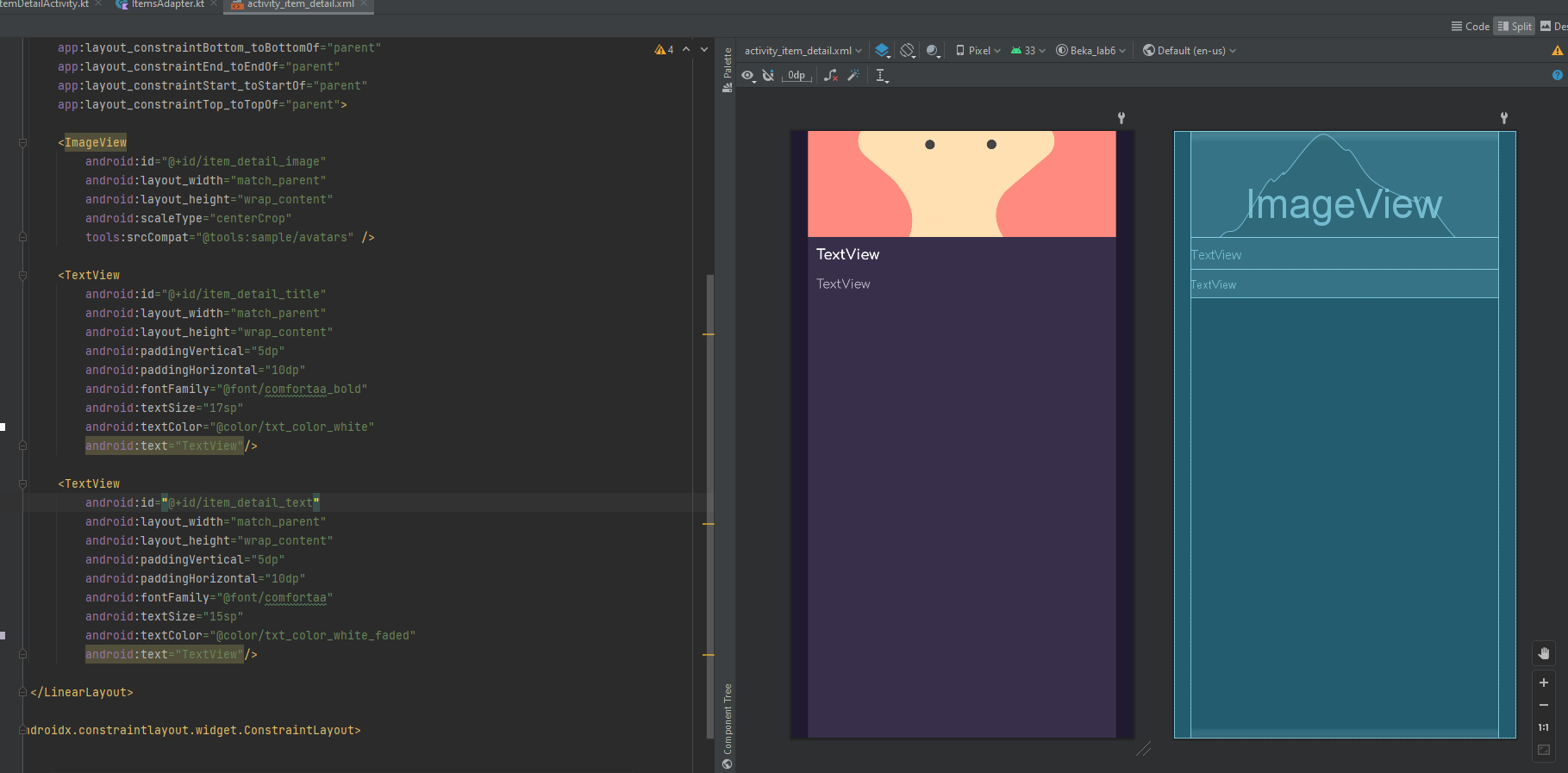
Для добавления более 1го товара в корзину, я добавил возможность при нажатии на кнопку “ADD TO CART” для каждого товара добавлять их дубликаты в корзину. Реализовал я ее с помощью лямбда функции, обрабатывающая нажатие по ней. С помощью позиции адаптера я понимаю с каким конкретно товаром я работаю при нажатии, таким образом я отправляю товар в корзину.



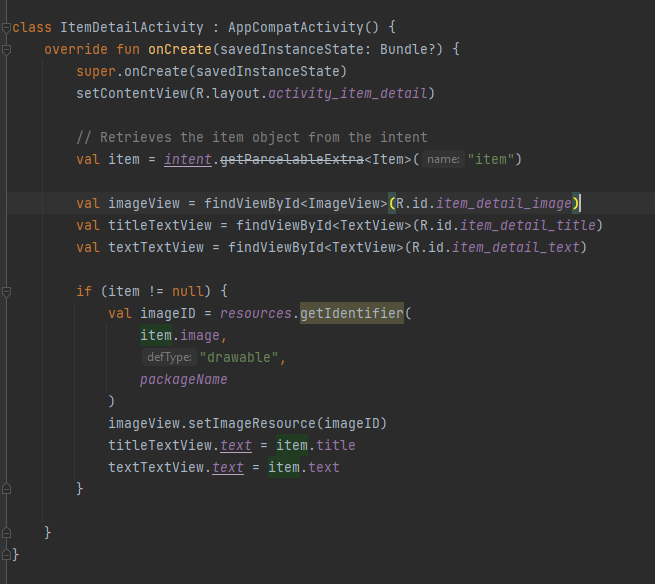
Здесь я показываю, сколько однотипных товаров было добавлено в корзину, а также при нажатии на сам товар(item) показываю его полную информацию переводя пользователя на activity\_item\_detail.xml



Это activity\_item\_detail.xml для оторбражение подробной информации о товаре

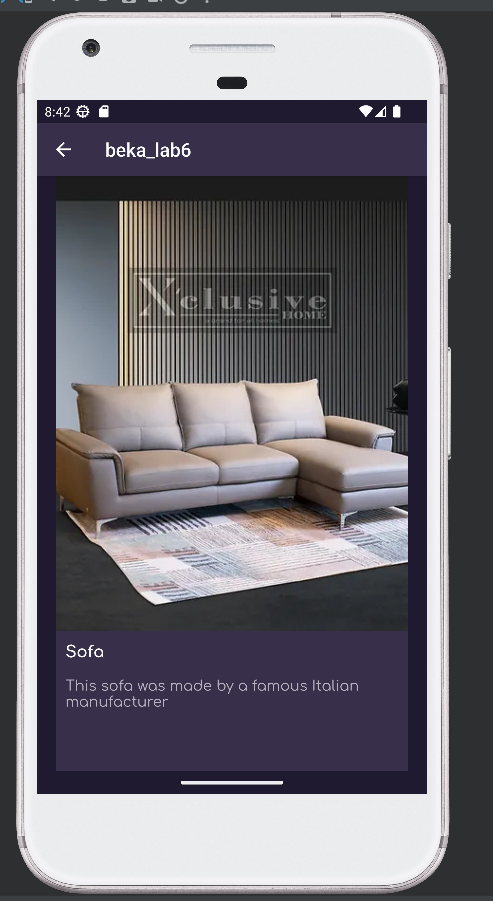
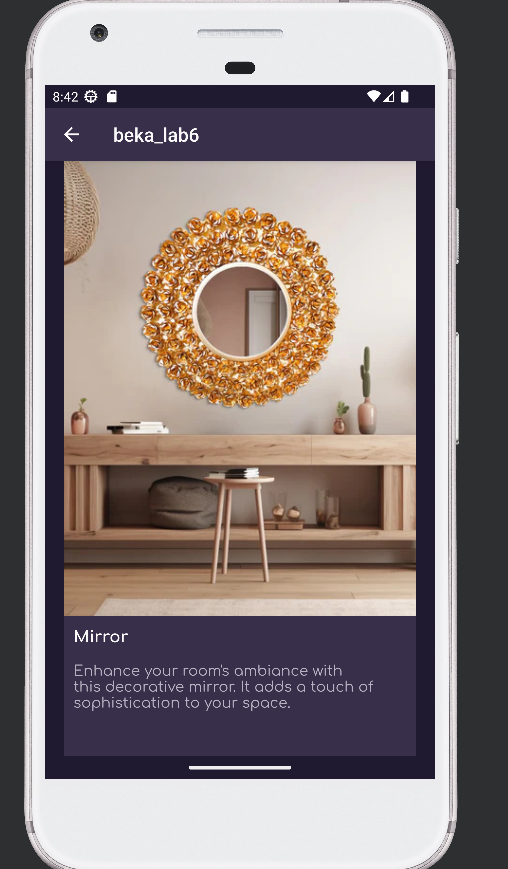


Это класс ItemDetailActivity, в котором я указываю, что конкретно я хочу показать в activity\_item\_detail.xml и какие данные я использую для отображения из item

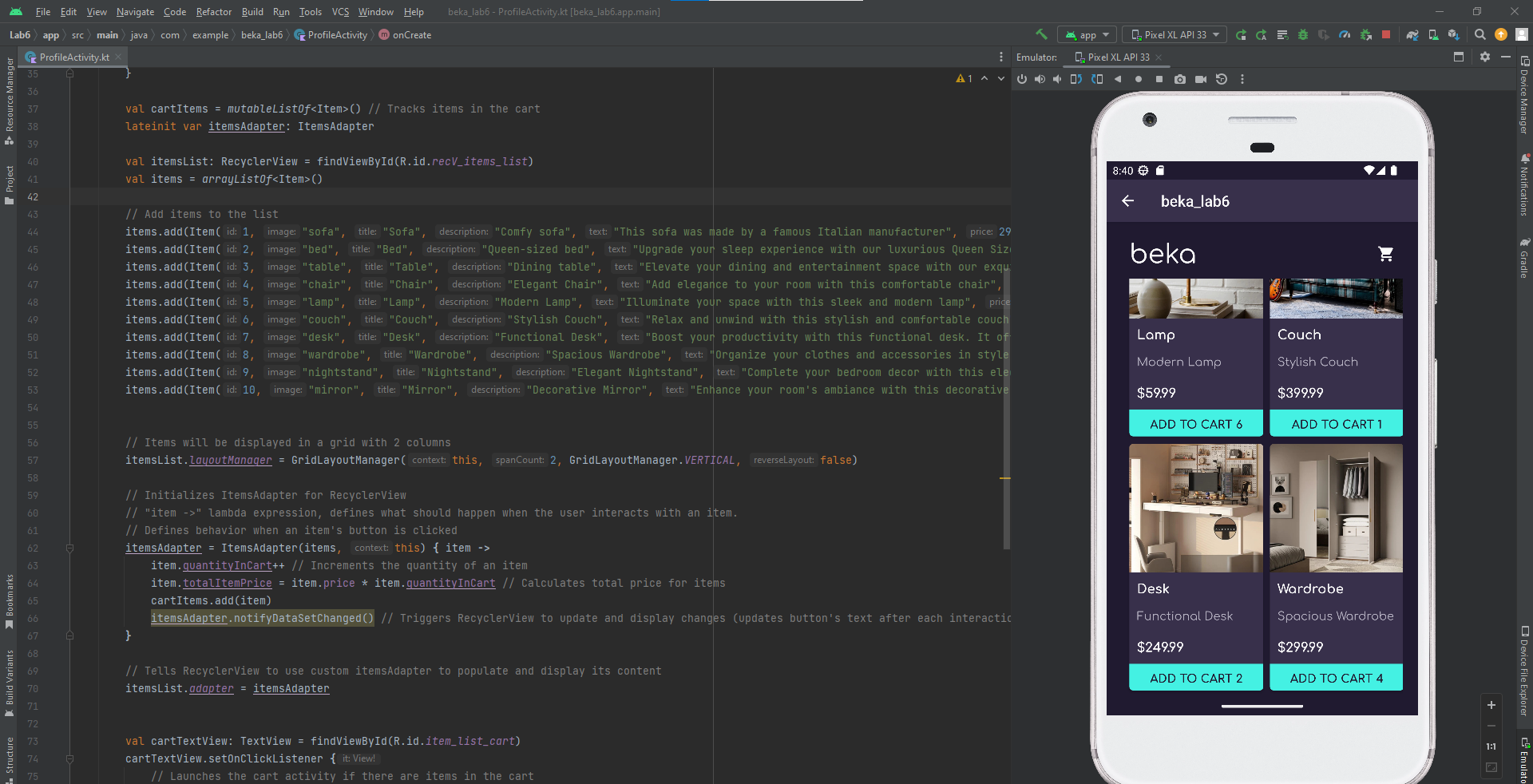


Проверка на эмуляторе

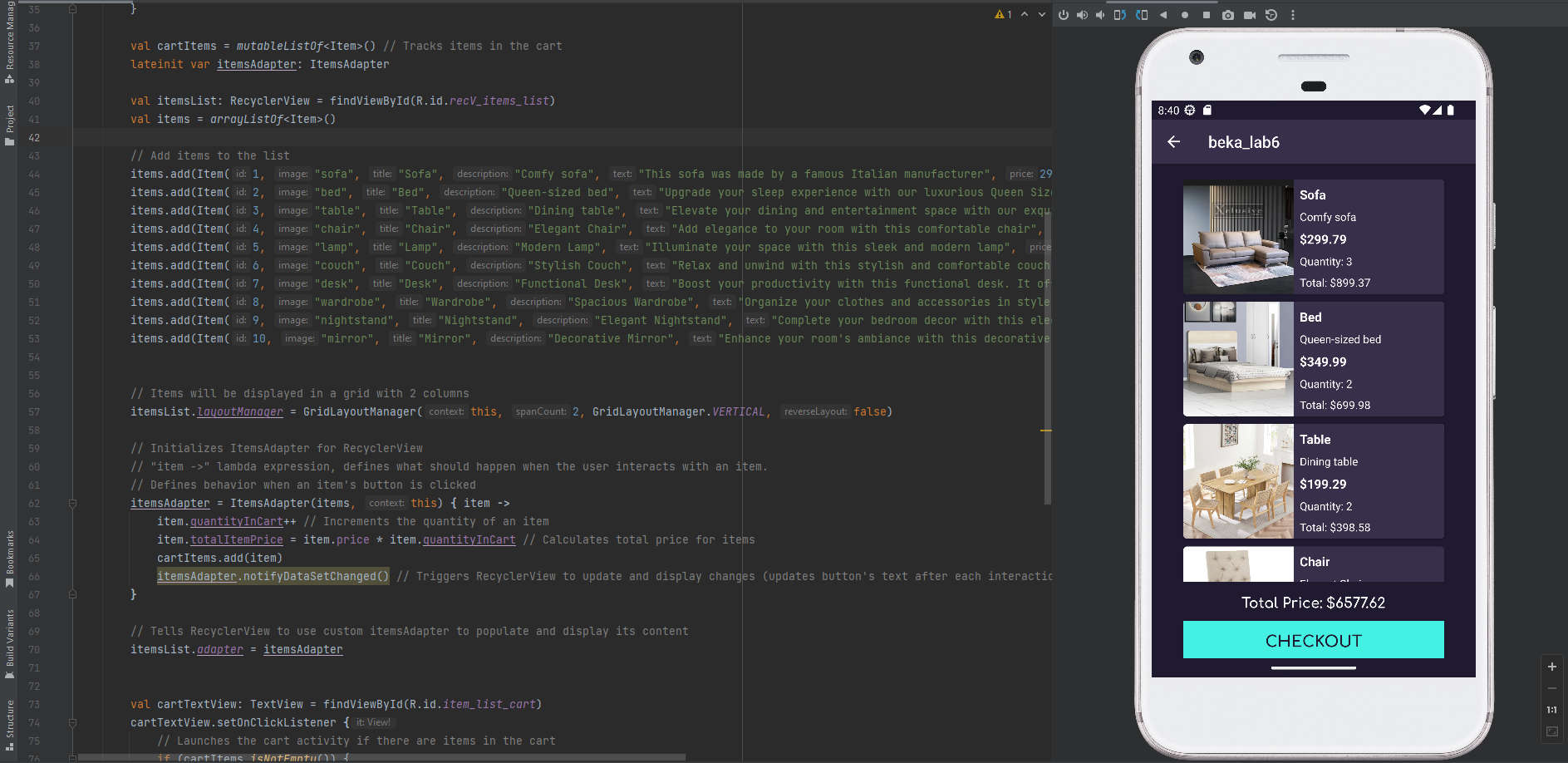
Информация о товаре

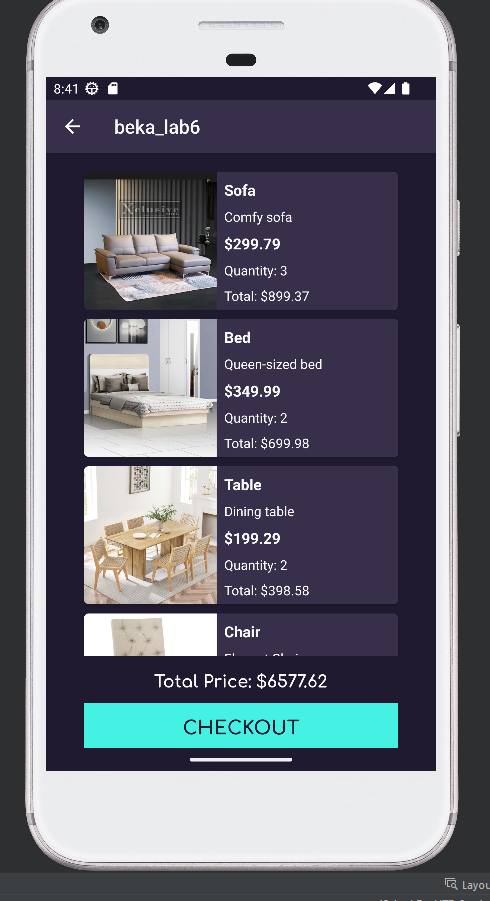
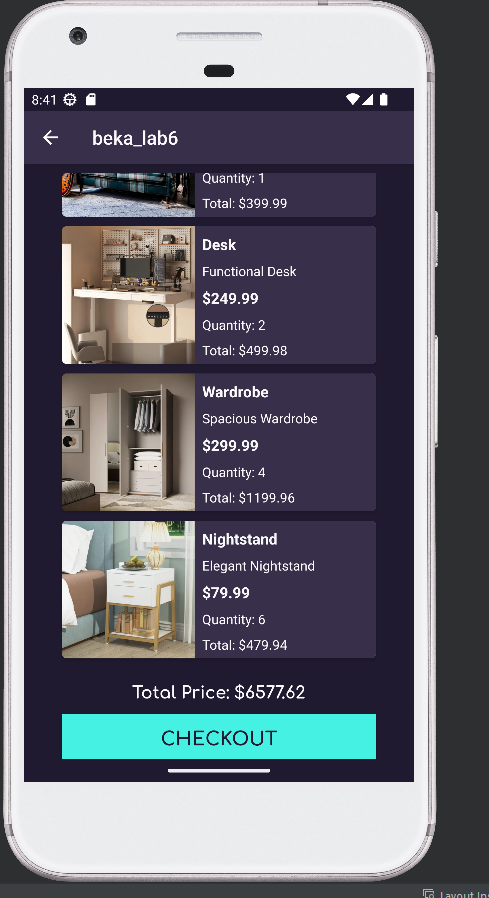
 

Я добавил по каждому товару



В корзине



Код

Classes





XML















Colors



Strings



Manifest

