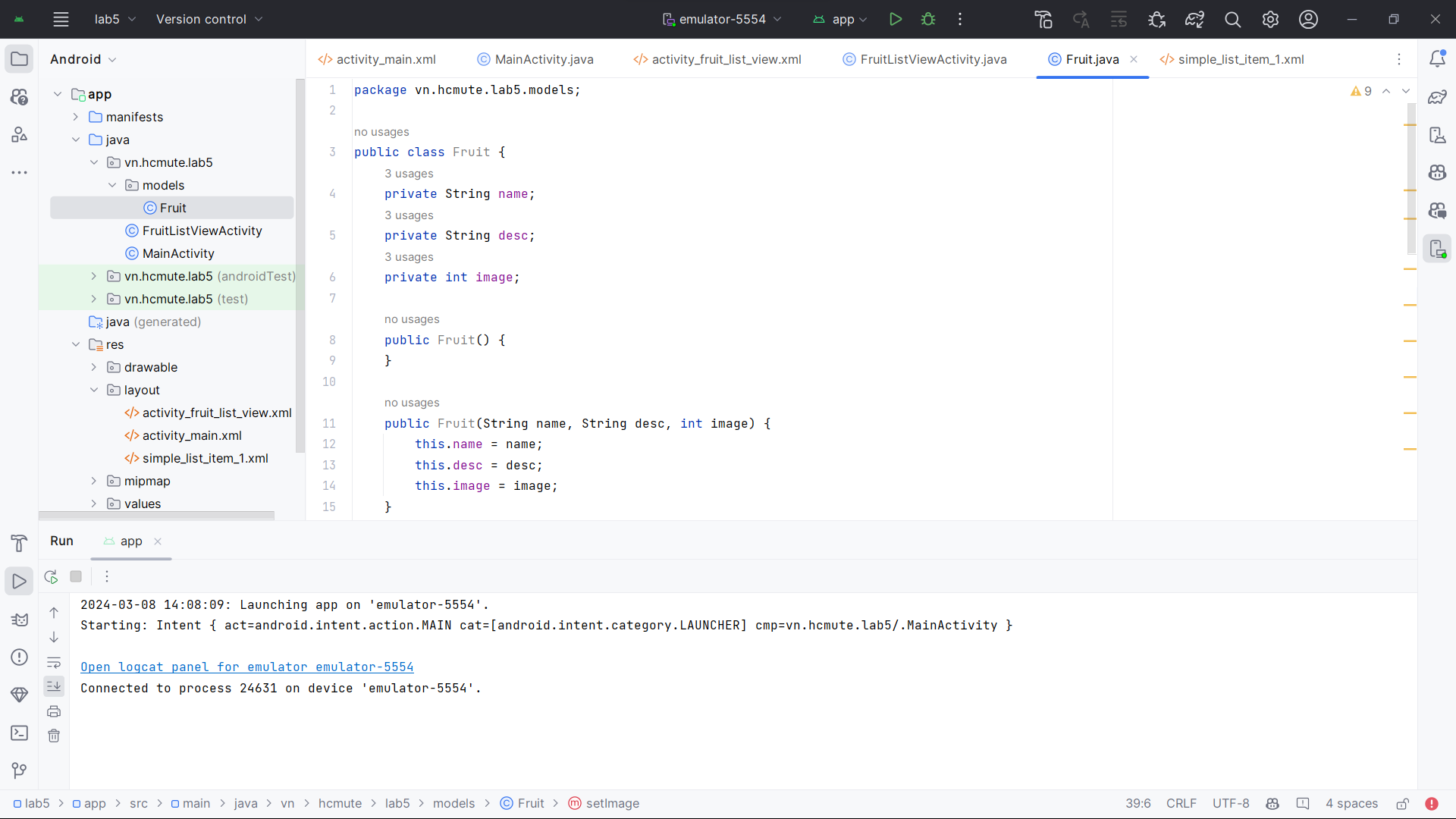
**Trần Hoàng Phúc – 21110606 – Lab 5 (Tuần 6 08/03)**

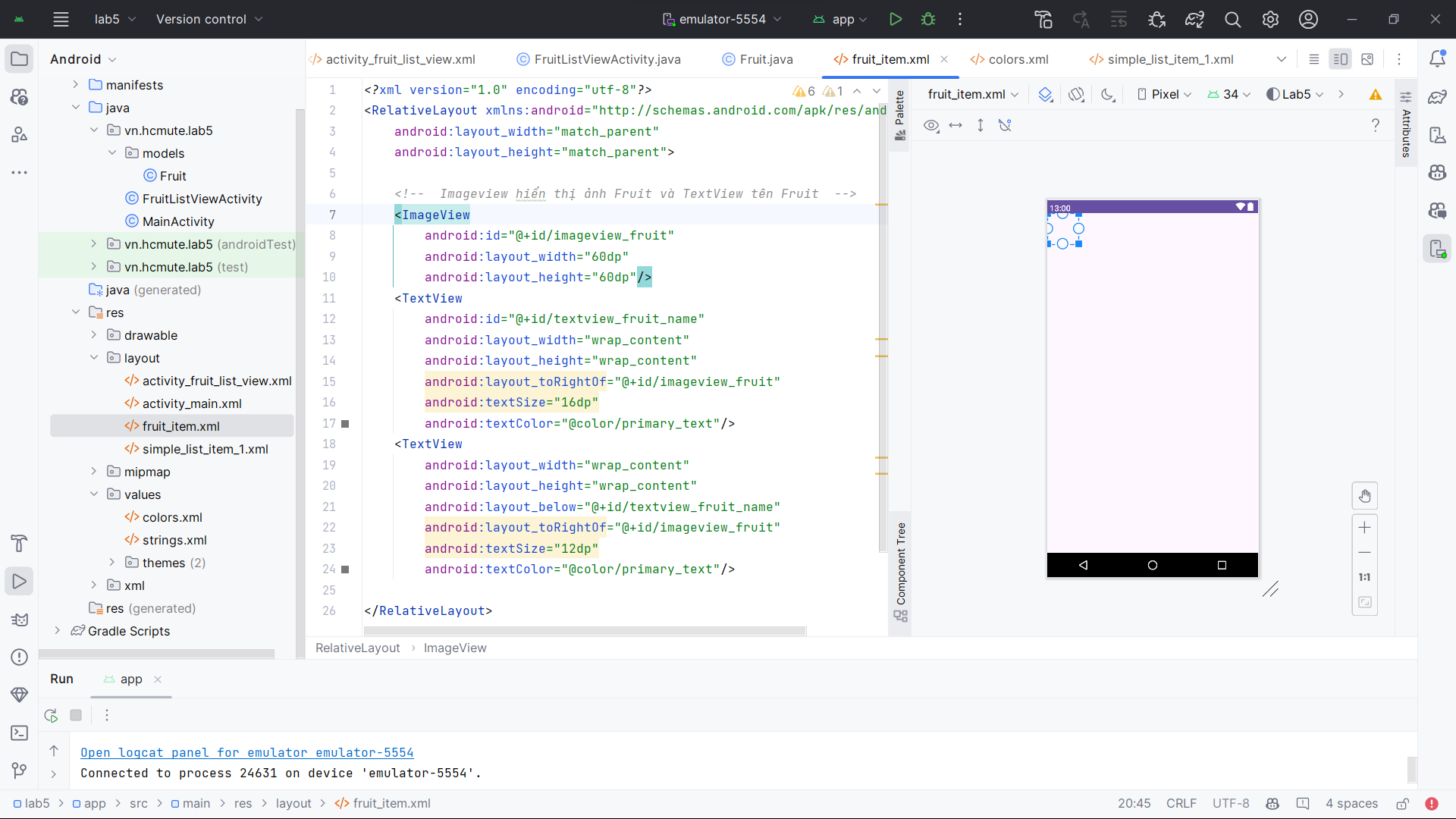
# Custom Adapter Listview

Để sử dụng view phong phú hơn, ta dùng custom adapter extends từ BaseAdapter hoặc ArrayAdapter, cùng với custom Model cho dữ liệu trên view.

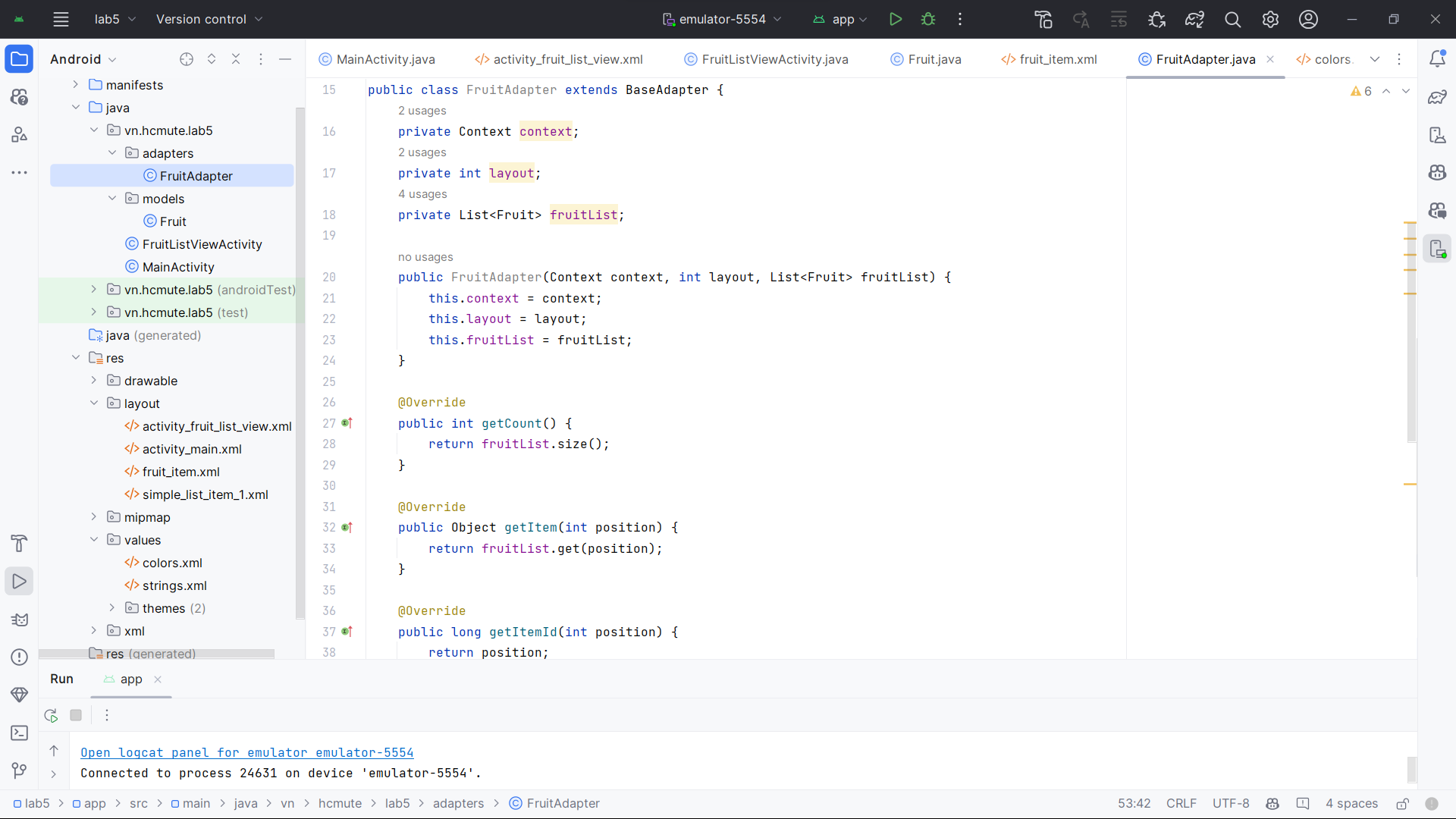
## Class Fruit:

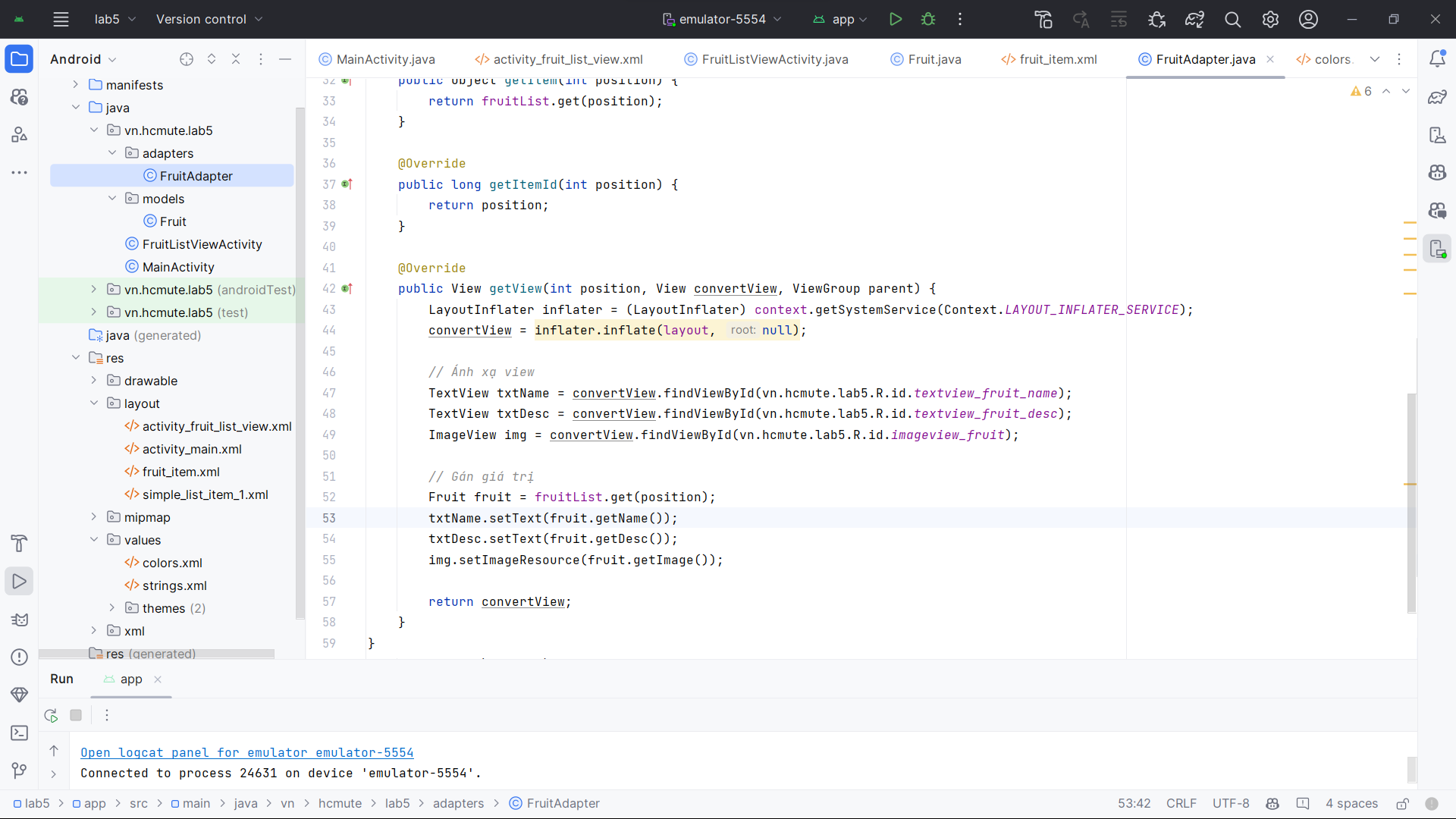


## Tạo layout item view:

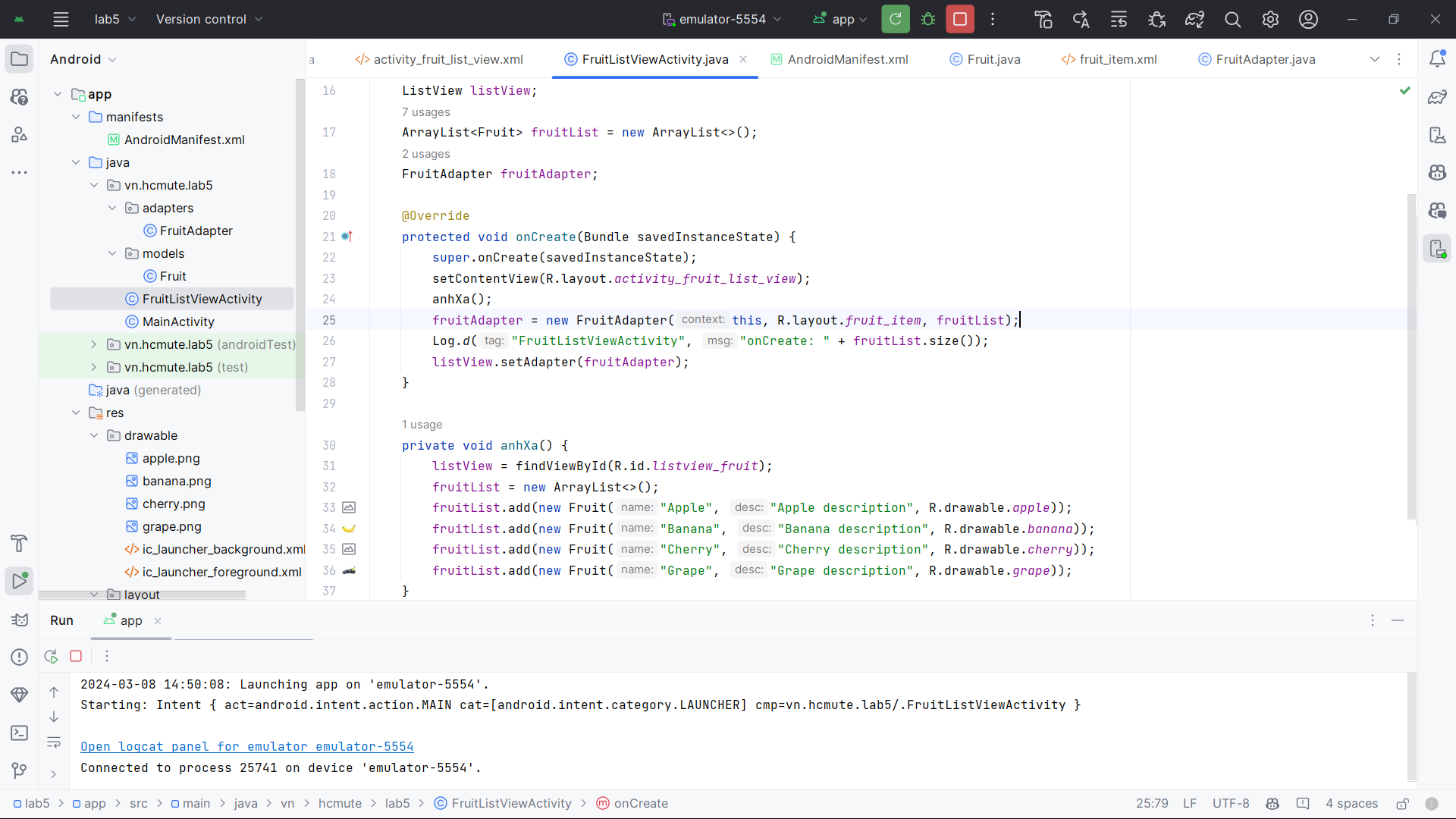


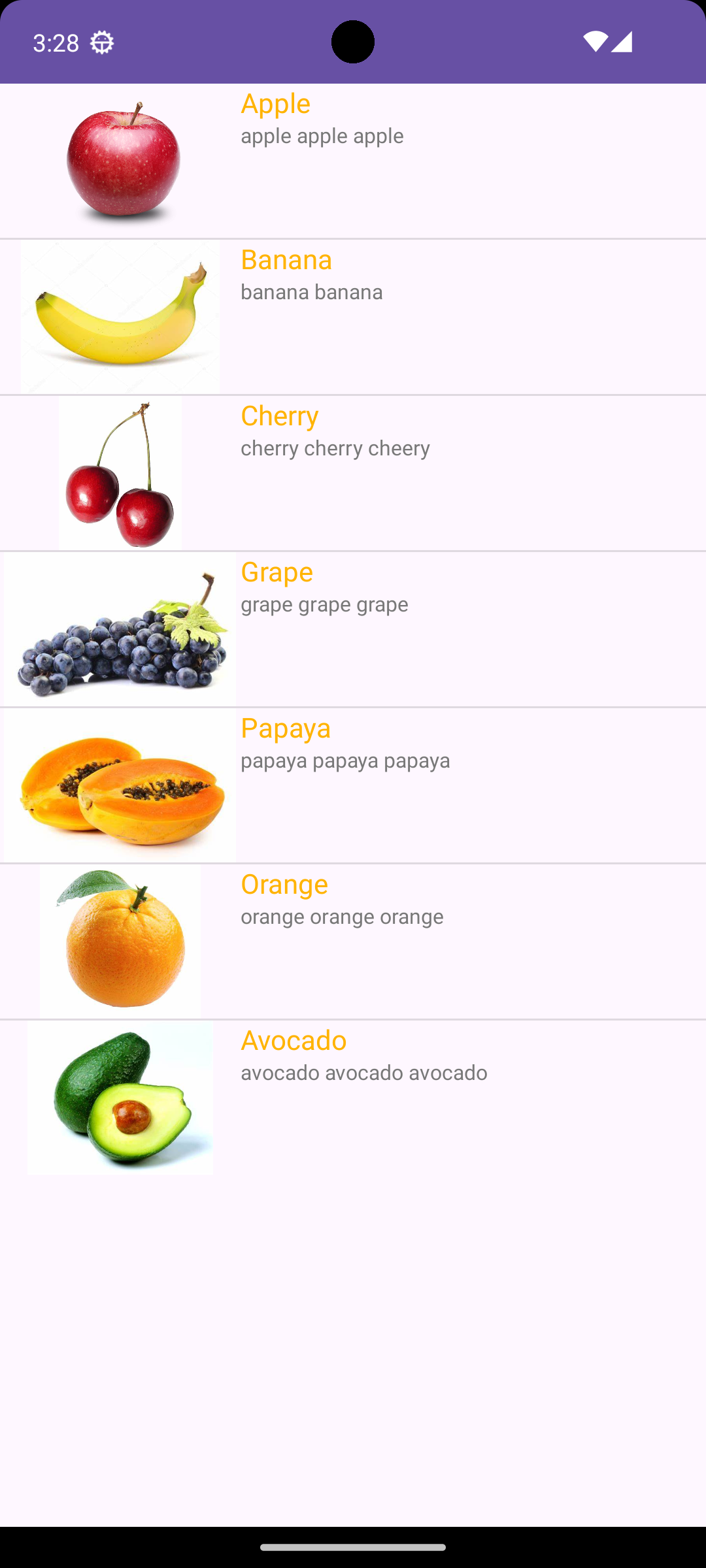
## Tạo custom adapter extends từ class BaseAdapter, sau đó implement các method của class này và tạo constructor gồm Context, layout, arrayList:



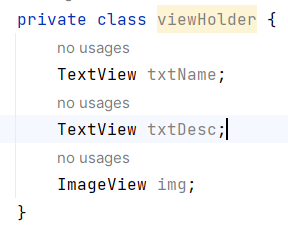


## Đưa dữ liệu vào ListView, trong đó, ta ánh xạ các view, tạo một adapter FruitAdapter và truyền các tham số context, layout ( của view sẽ đổ vào ListView), fruitList:

Kết quả:



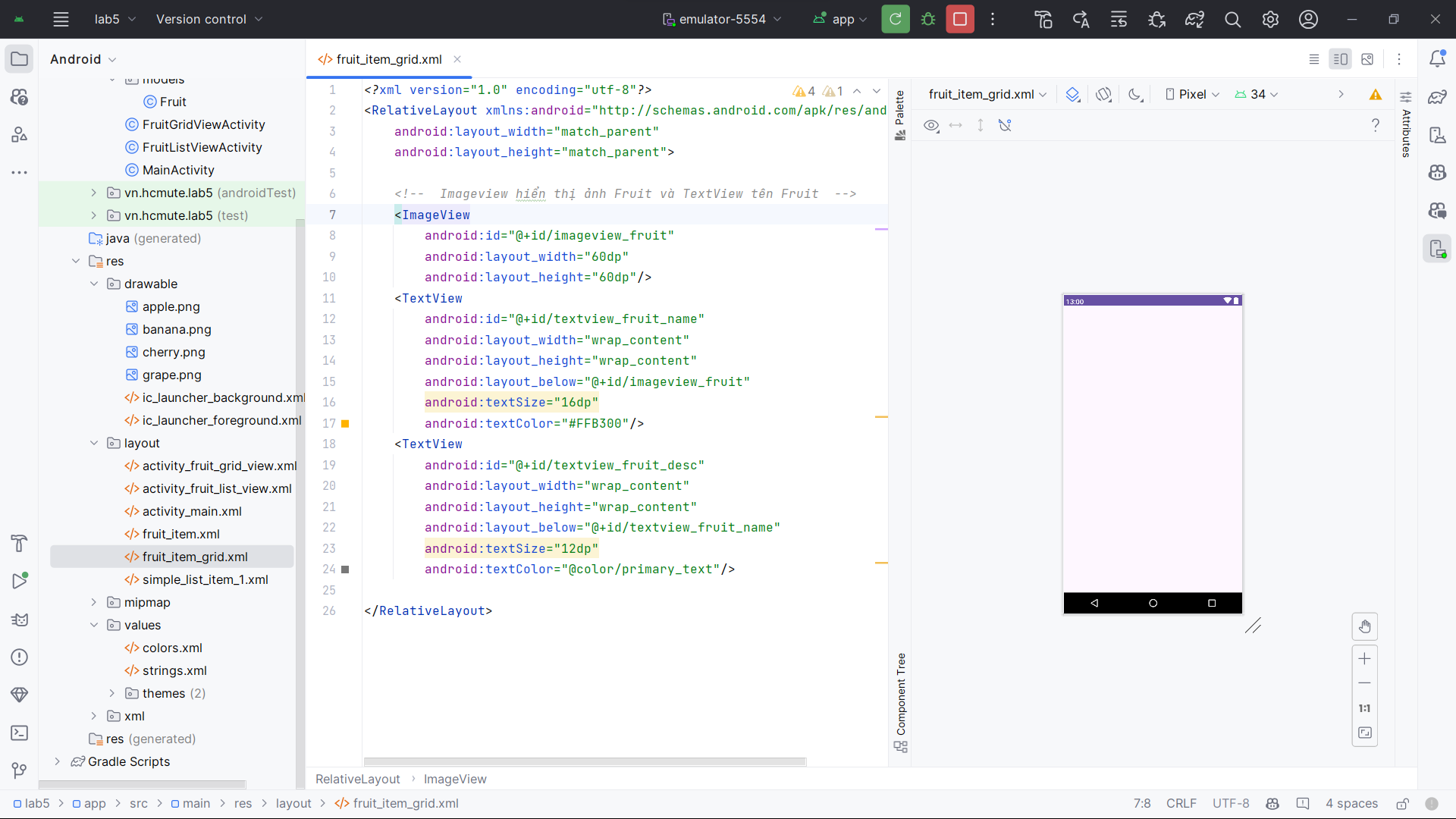
## Viewholder trong ListView: là một design pattern được sử dụng để giảm tải việc truy cập vào các view qua findViewById(). Ta dùng class viewHolder để chứa các control, khi cần cập nhật các view, ta có thể gọi thông qua ViewHolder mà không cần gọi lại findViewById() nhiều lần, việc này làm tăng hiệu suất chương trình.



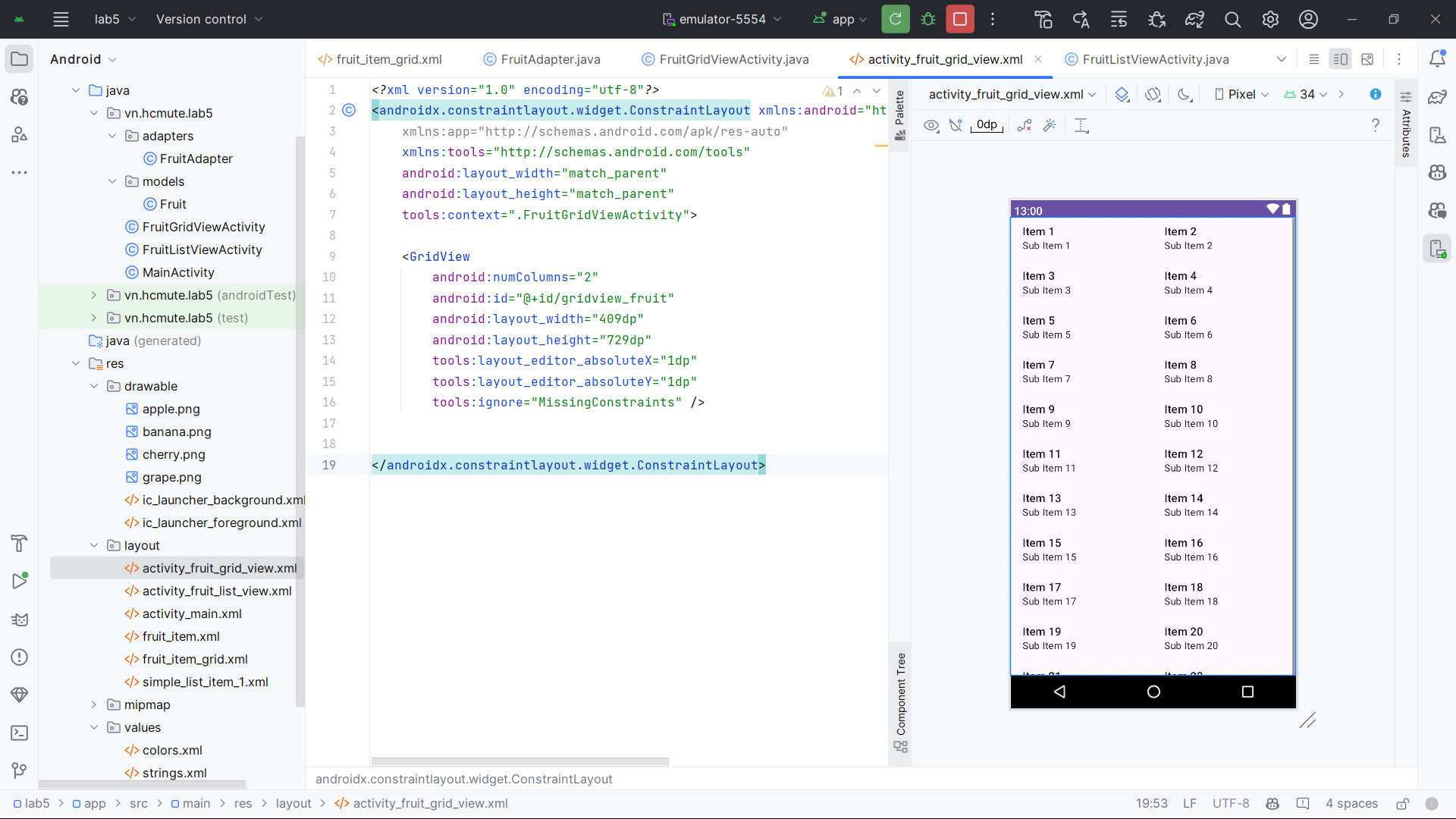


# Custom Adapter GridView

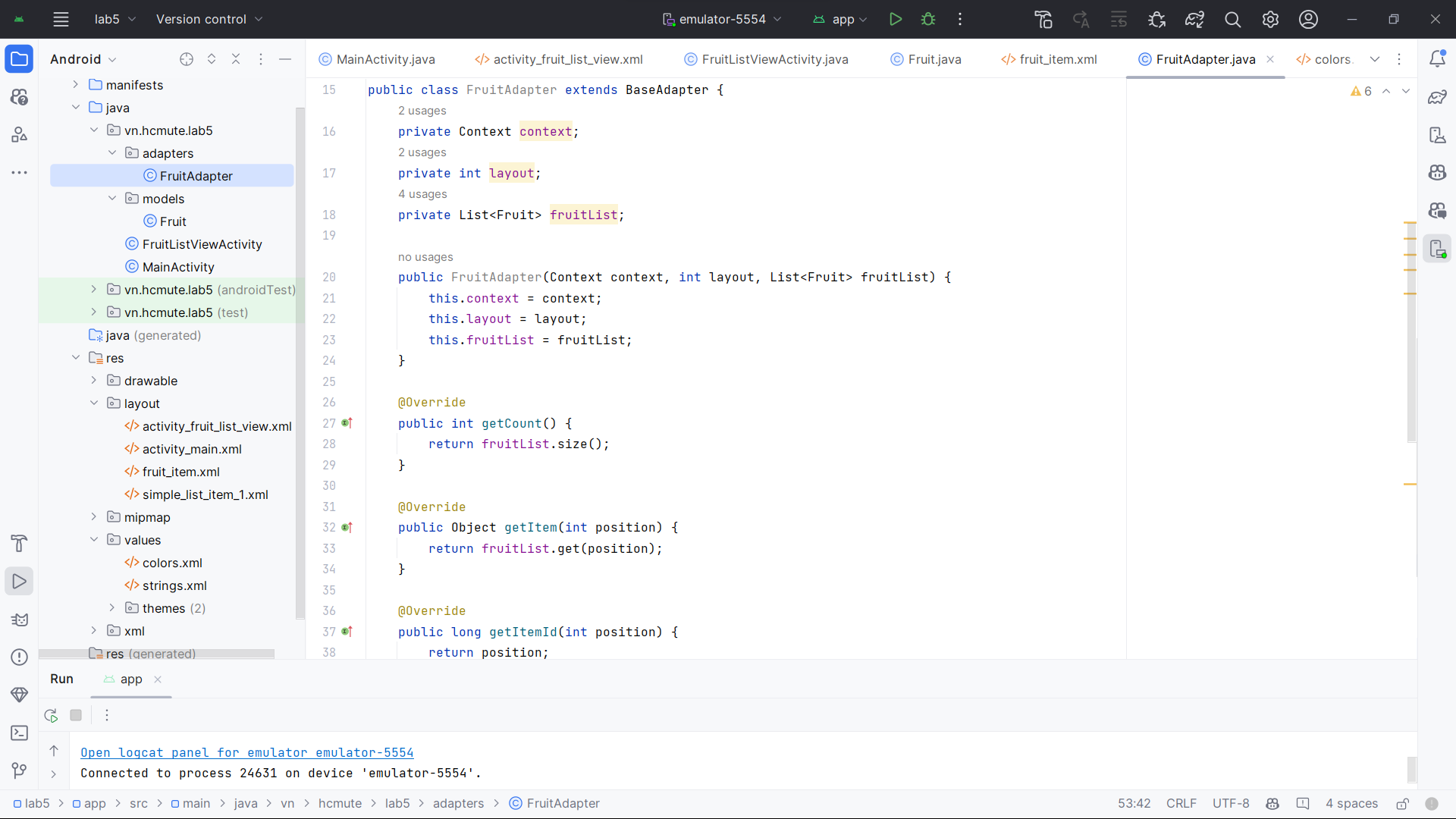
## Tạo layout item view:



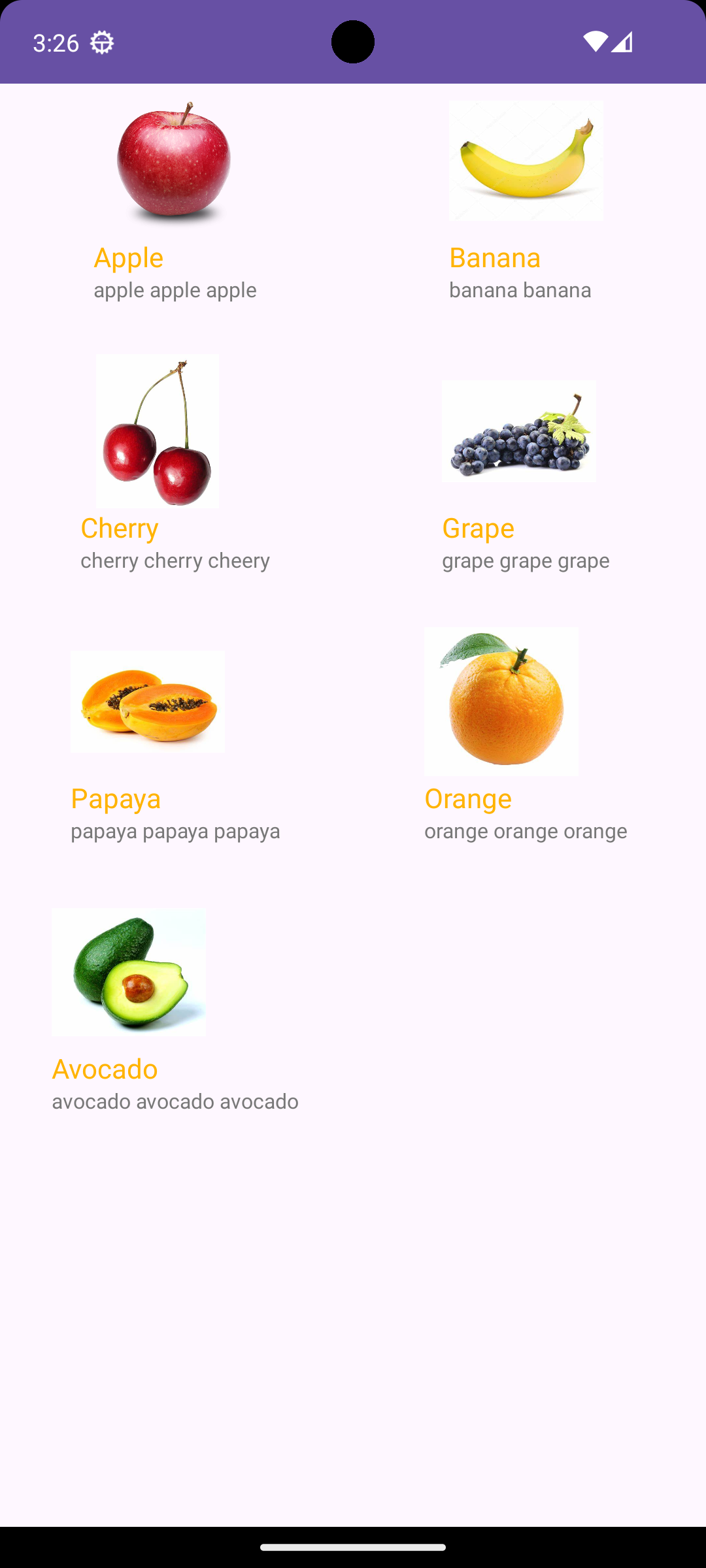
## Gridview:



## Custom adapter:



## Kết quả:



Tương tự với ListView, ta có thể dùng class ViewHolder trong Adapter để giảm tải việc truy cập vào view hierarchy.

# RecyclerView

## 3.1. Giới thiệu RecyclerView

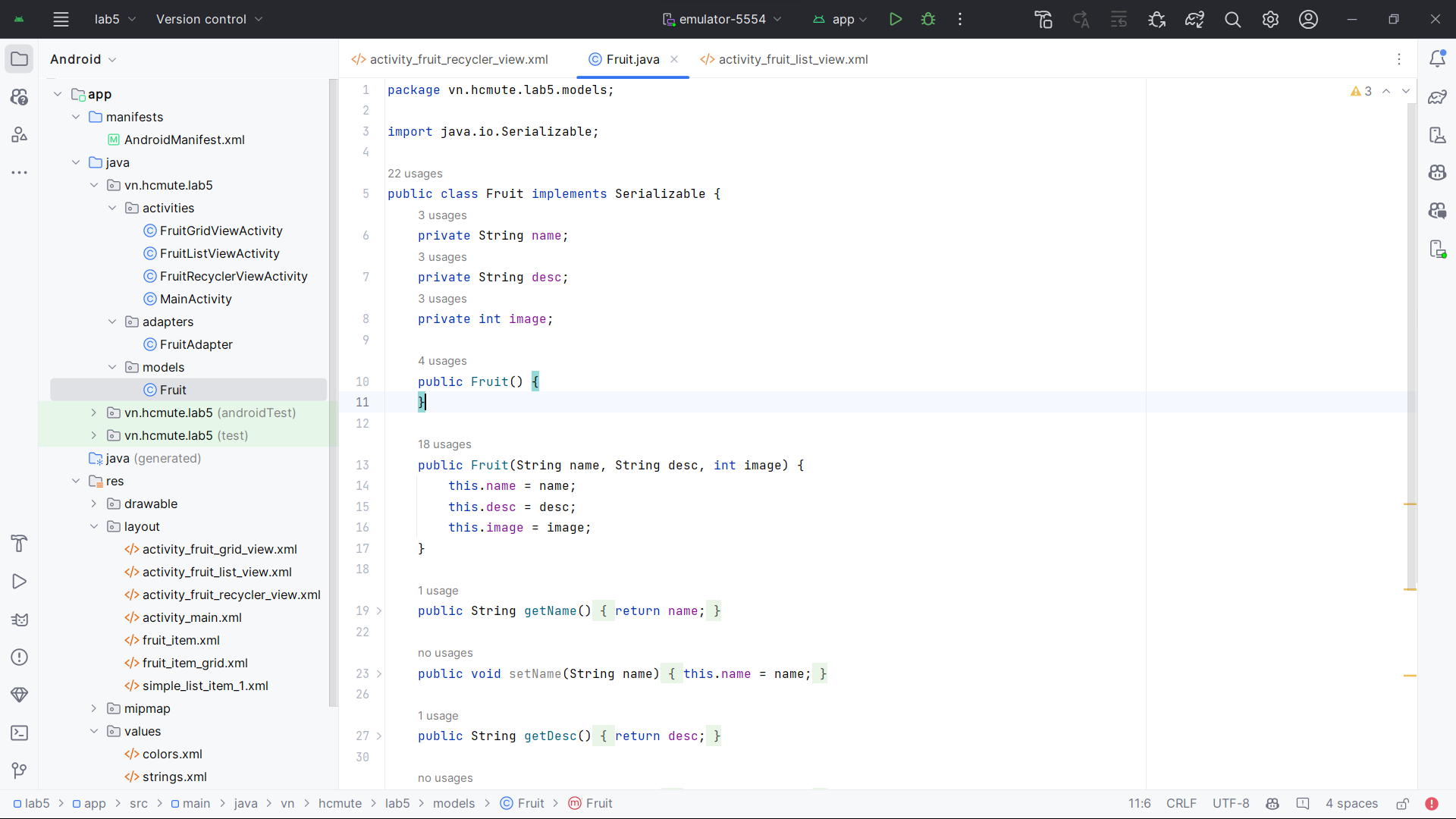
RecyclerView là một thành phần UI trong Android, được giới thiệu trong Android Lollipop (Android 5.0, API 21). Nó là một phiên bản nâng cấp của ListView và GridView với nhiều tính năng và hiệu suất tốt hơn. Dưới đây là một số tính năng chính của RecyclerView:

* Hiệu suất cao: RecyclerView sử dụng một kỹ thuật gọi là "recycling" để tái sử dụng các view đã không còn hiển thị trên màn hình. Điều này giúp giảm bộ nhớ và CPU, đặc biệt khi hiển thị danh sách lớn.
* Layout linh hoạt: RecyclerView hỗ trợ nhiều loại layout như danh sách dọc (vertical list), danh sách ngang (horizontal list), lưới (grid), và thậm chí có thể tùy chỉnh layout với LayoutManager.
* Hỗ trợ animation: RecyclerView cung cấp ItemAnimator để thực hiện animation khi thêm, xóa hoặc sắp xếp lại các item.
* Hỗ trợ tương tác người dùng: RecyclerView hỗ trợ các tương tác người dùng như kéo để làm mới (pull to refresh), kéo để xóa (swipe to dismiss), kéo và thả để sắp xếp lại (drag and drop).

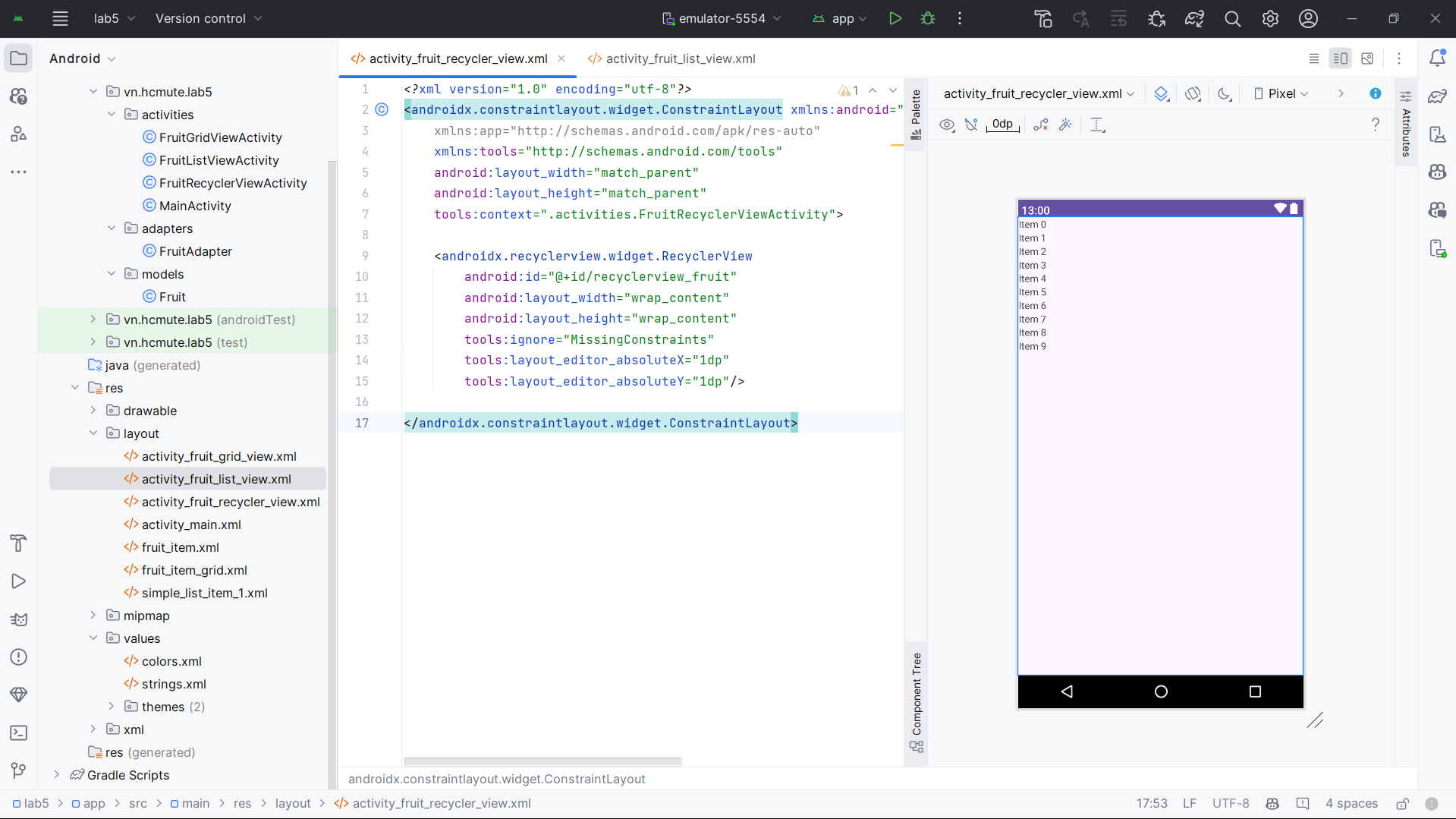
Để sử dụng RecyclerView, ta cung cấp một Adapter để quản lý dữ liệu và tạo view cho mỗi item. Ta cũng cần cung cấp một LayoutManager để quy định cách các item được sắp xếp trên màn hình.

## 3.2. Sử dụng RecyclerView

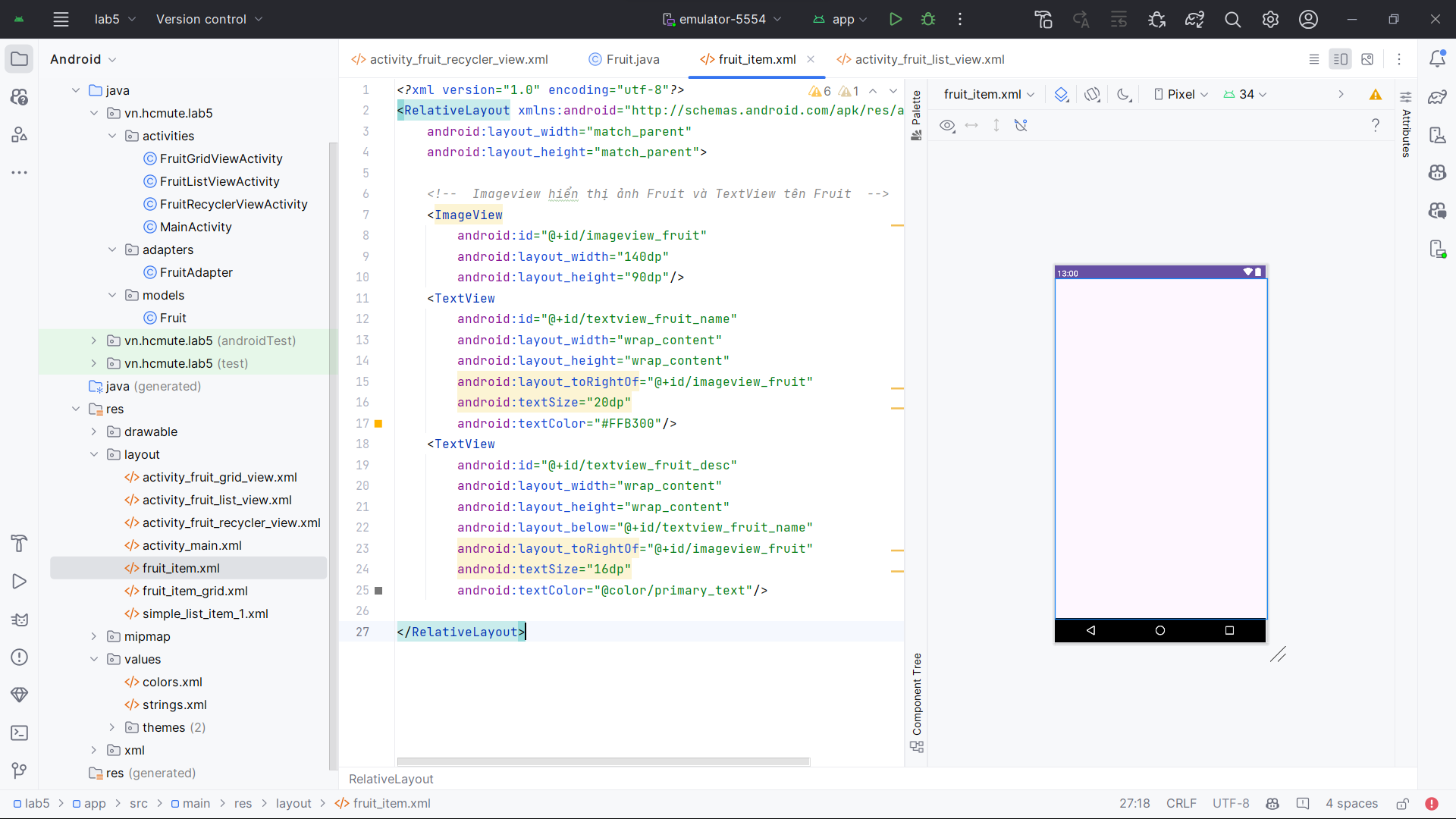
Tạo model chứa dữ liệu:



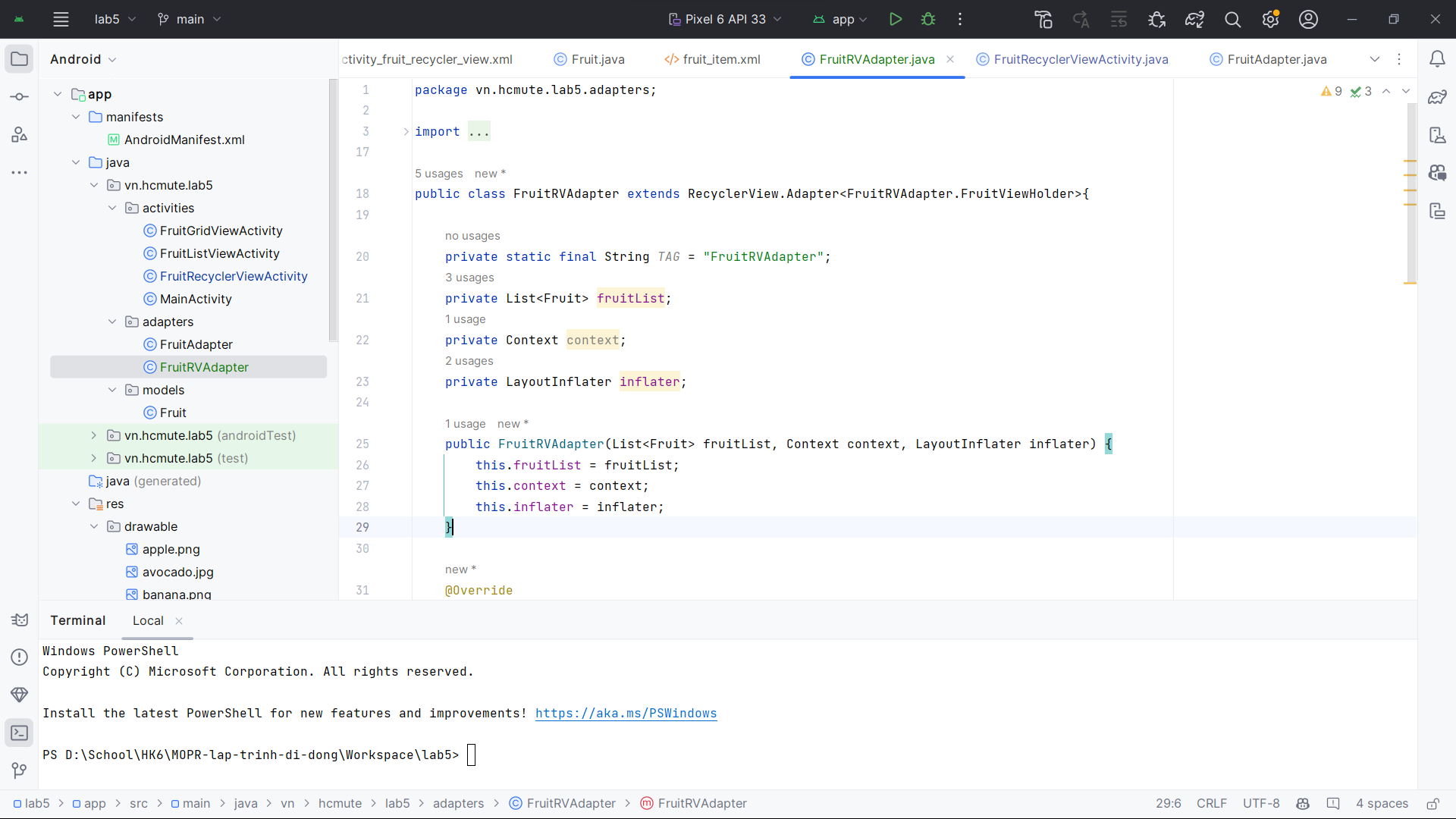
Tạo RecyclerView trong layout activity:



Tạo layout cho 1 item trong view group:



Tạo Custom Adapter extends từ RecyclerView.Adapter:



Class ViewHolder:



Các phương thức override:



Sử dụng RecyclerView với Adapter và LayoutManager:

Kết quả:



Chọn item trong RecyclerView, bằng cách setOnClickListenner cho itemView, trong constructor của ViewHolder như sau:

