BÁO CÁO BÀI TẬP LỚN MÔN LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

Các thành viên trong nhóm:

1. Bùi Minh Nhật - 1651010103
2. Nguyễn Hùng Anh - 1651012005
3. Võ Khắc Nhân - 1651012129
4. Trần Đức Thịnh – 1651010140

Tên đề tài : Music App

1. Giới thiệu Music App:
2. Tạo 2 tab Trang chủ và Tìm kiếm:
3. Giới thiệu ViewPager:

- ViewPager là một trình quản lý Layout cho phép người dùng vuốt tay qua trái hay qua bên phải để chuyển sang nội dung của trang khác một cách đồng thời. Điểm khác nhau của ViewPager và cách sử dụng Gesture đó là, gesture sẽ chỉ thực hiện khi người dùng thực hiện xong thao tác vuốt tay, trong khi ViewPager thực hiện chuyển màn hình đồng thời với cử chỉ tay người dùng.

– ViewPager có thể trượt chuyển đổi giữa các giao diện một cách nhẹ nhàng và khá mượt, thay ví chuyển đổi màn hình qua một sự kiến chớp đen như trên tivi. Màn hình hiển thị trước nó hoặc sau nó sẽ được hiển thị ra ngay tức thì liền với nó.

– ViewPager hỗ trợ từ Android API 13 trở lên.

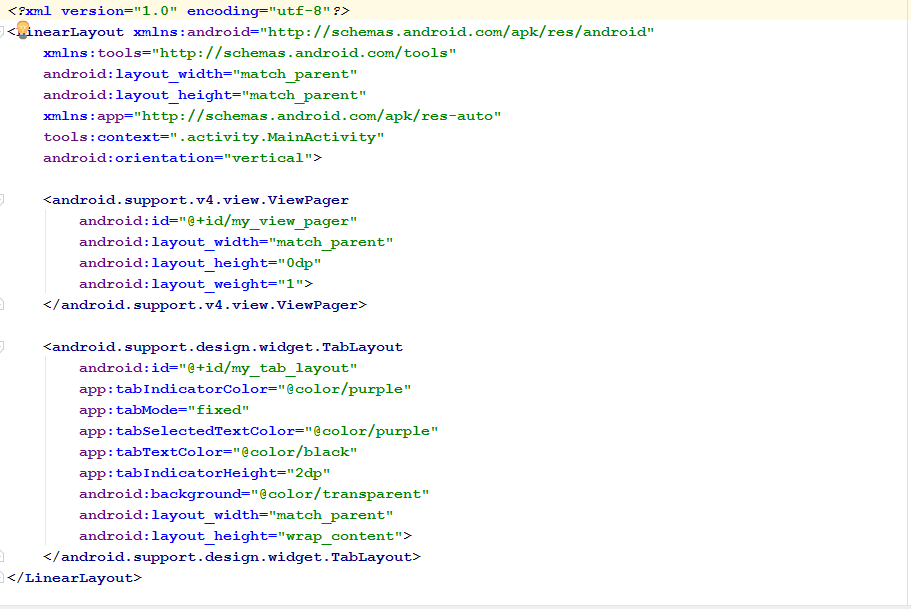
\_ ViewPager không phải là một View chuẩn của Android, mà là một thành phần nằm trong gói android.support:design:xxx

1. Giới thiệu Tab Layout:

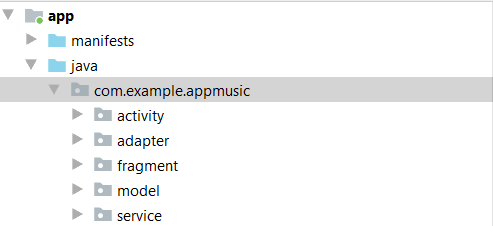
- Tab Layout trong Android được dùng để thiết kế màn hình với nhiều mục (tab) khác nhau trên một màn hình, giúp chuyển đổi qua lại giữa các màn hình một cách nhanh chóng và thuận lợi.

c) Tạo giao diện:

activity\_main.xml

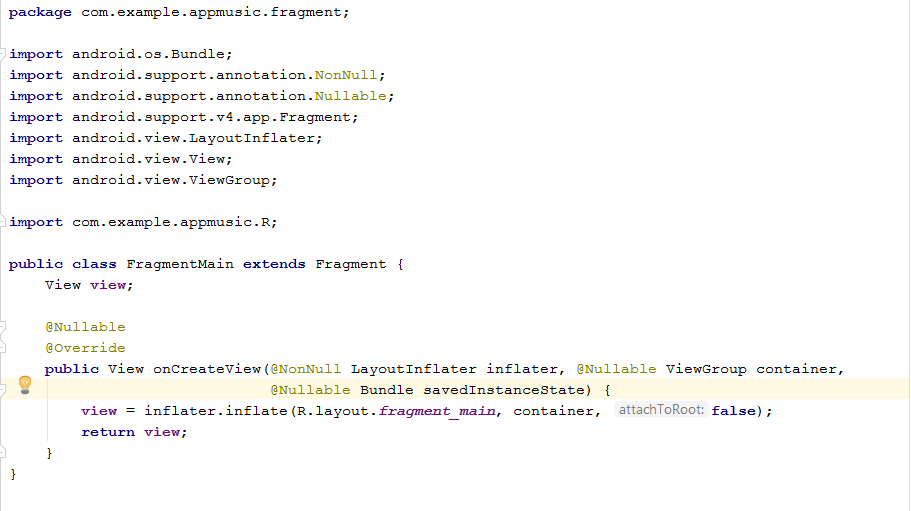


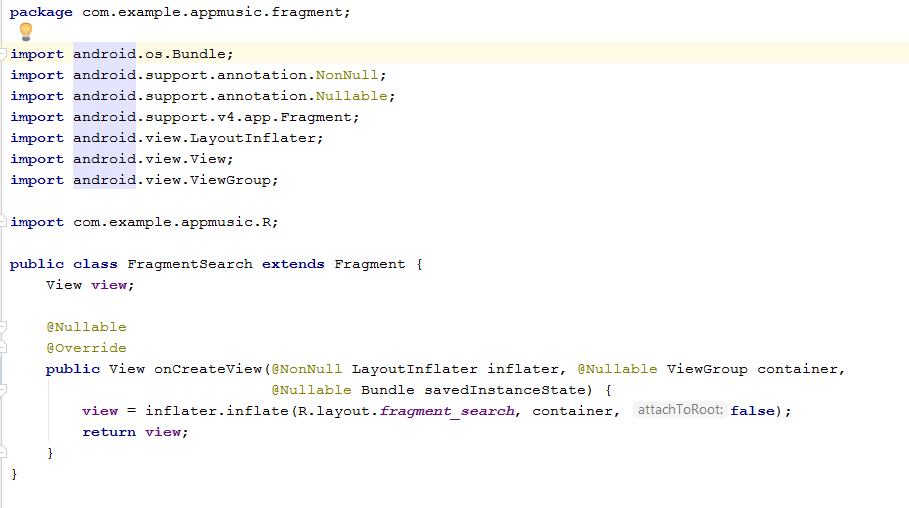
Phân ra cho project 1 số package cho mỗi chức năng khác nhau:



Tạo 2 fragment :Fragment Main & Fragment Search extends Fragment

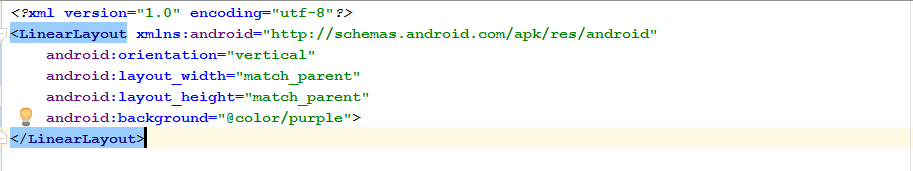
-> import android.support.v4.app.Fragment -> gọi phương thức OnCreateView -> tạo view (view này có chức năng gán layout vào fragment & lấy view này để tương tác được với những view ở bên trong layout của fragment) ->trả về layout sau khi gán (view)





Tạo 2 layout của 2 fragment vừa tạo:

fragment\_main.xml & fragment\_search.xml



Tạo 1 MainViewPagerAdapter: extends FragmentPagerAdapter

Alt + Enter -> kế thừa lại 2 phương thức getItem() & getCount()  
Alt + Enter -> thêm vào 1 constructor

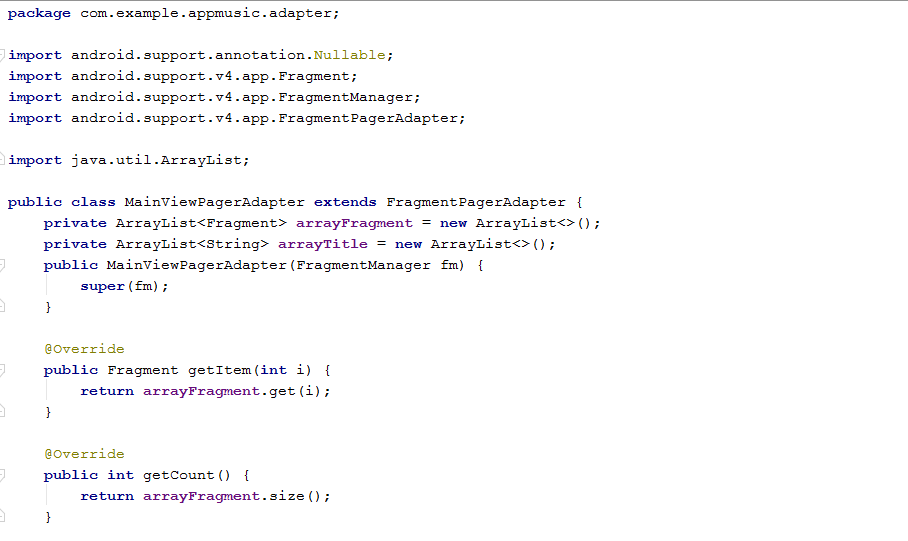
Tạo arrayFragment để thêm các fragment (chọn thư viện android.support.v4.app.Fragment)

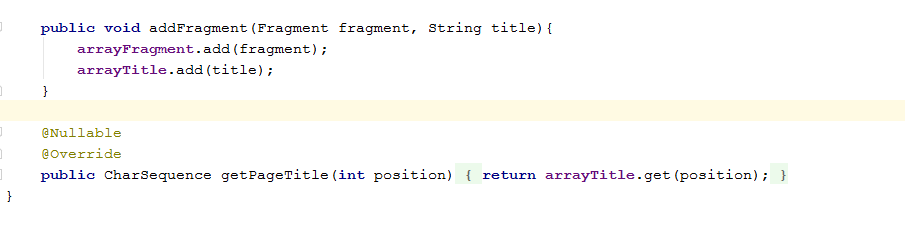
Tạo arrayTitle để hiển thị tên của các Tab

arrayFragment.get(i) -> Click tới đâu trả về vị trí fragment tới đó

Có bao nhiêu fragment muốn hiển thị -> Trả về kích thước arrayFragment.size()

Viết thêm phương thức addFragment(Fragment fragment, String title) cho hiển thị tiêu đề của fragment





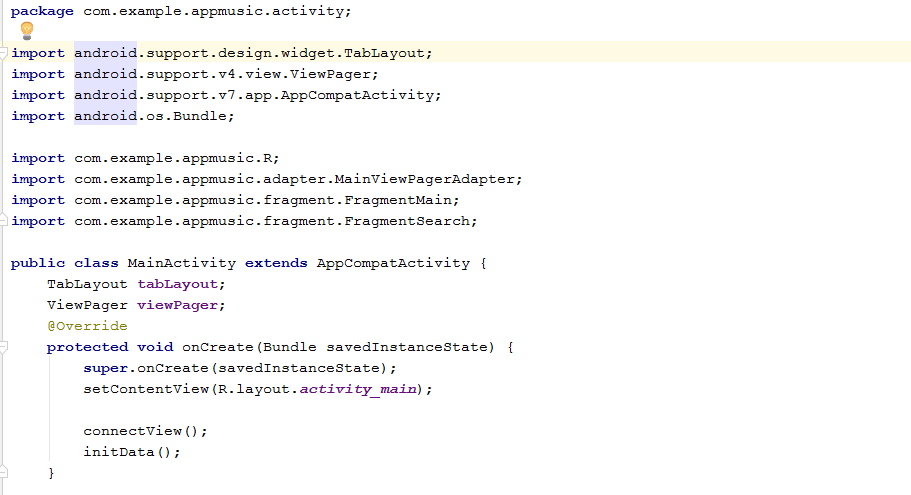
Trở lại với Main Activity: Tạo tabLayout & viewPager

Tạo phương thức connectView() & initData()

Trong initData() tạo mainViewPagerAdapter

-> add lần lượt từng fragment

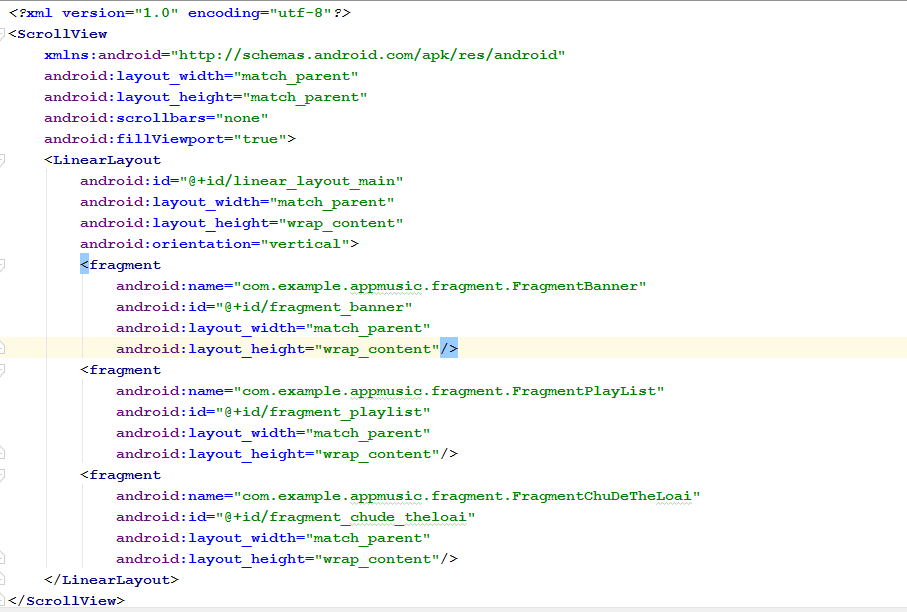
-> setAdapter() cho viewPager

-> setupWithViewPager cho tabLayout -> tabLayout.setTabAt : trỏ đến vị trí muốn set lại



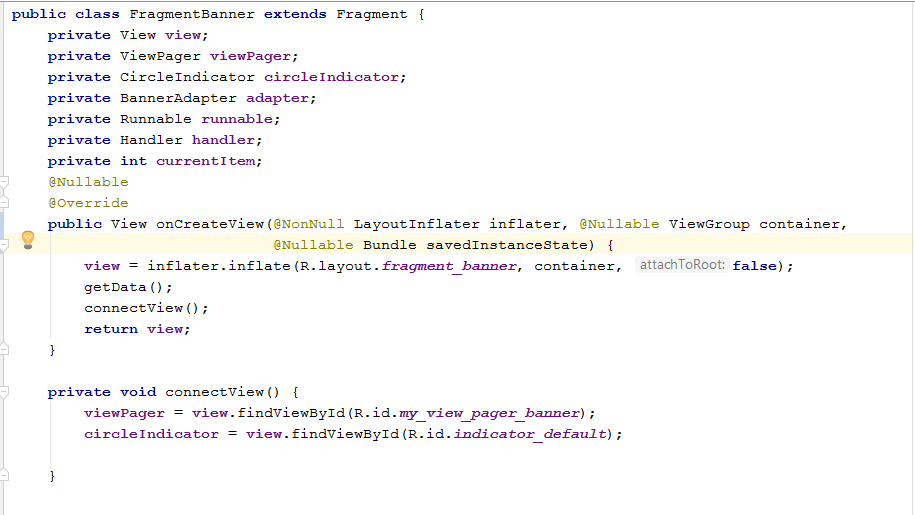
3. Thiết kế layout cho fragment quảng cáo

Trở lại fragment\_main.xml:



Tạo Fragment\_Banner: extends Fragment

-> import android.support.v4.app.Fragment -> gọi phương thức OnCreateView -> tạo view (view này có chức năng gán layout vào fragment & lấy view này để tương tác được với những view ở bên trong layout của fragment) ->trả về layout sau khi gán (view)

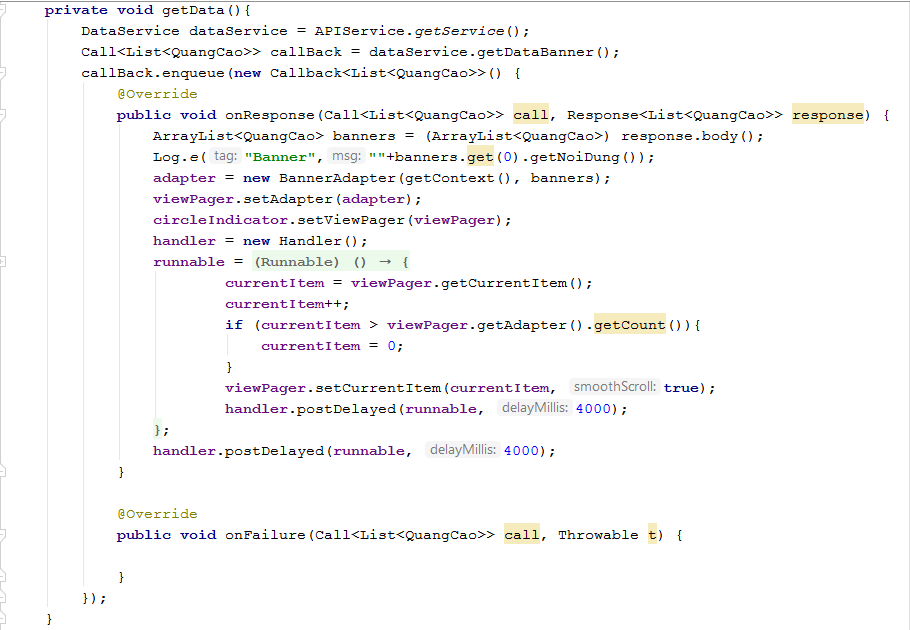


Truyền dữ liệu vào cho adapter -> gọi lại viewPager.setAdapter(adapter) -> circleIndicator.setViewPager(viewPager) hiện ra số lượng Indicator

Tạo handler & runnable(dùng để lắng nghe thực hiện những hành động khi handler đổi) -> Alt + Enter -> implements phương thức run()

Tạo biến currentItem để khi nó chạy qua từng item này biết giá trị item trước là bao nhiêu cộng thêm giá trị để chạy thêm page sau lưng

Khi gọi lại hành động gọi currentItem kiểm tra item hiện tại đang đứng ở đâu -> tăng currentItem -> if() nếu vượt qua kích thước của pager -> trở về page đầu -> mỗi lần chạy sẽ set dữ liệu lên -> cho handler chạy ->



Tạo layout item\_banner.xml:

Khi đã có phần pha màu ở custom\_background\_banner nhìn nó đẹp hơn, bắt mắt hơn trong phần View





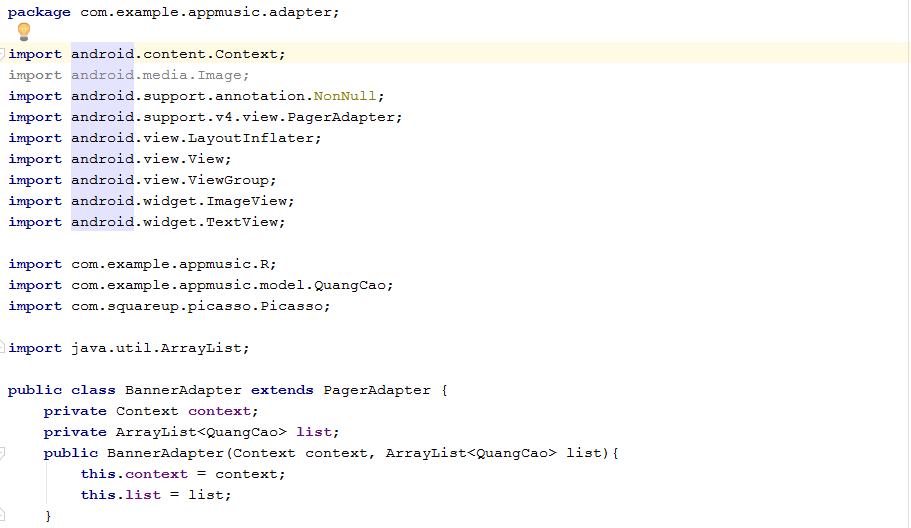
Tạo BannerAdapter: extends PagerAdapter

-> import android.support.v4.view.PagerAdapter

-> Alt + Enter -> implements method getCount() & isViewFromObject()

-> truyền vào context & list

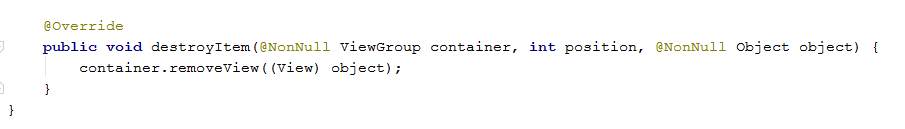
-> Alt + Insert chọn Constructor đè Shift chọn 2 dòng context & list



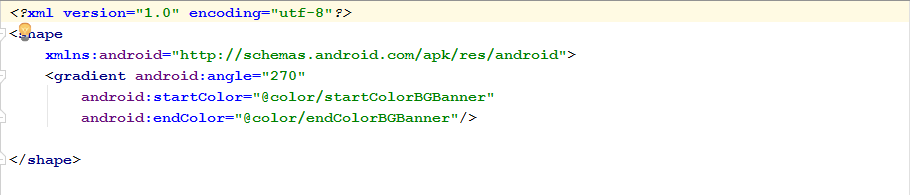
Đổ dữ liệu vào cho view -> định hình lại các layout -> ánh xạ những cái view của layout item\_banner vào -> gọi thư viện Picasso -> tvTitleSong.setText() & tvContent.setText() của mảng list.

Sẽ có 1 trường hợp nếu chuyển page sang page cuối nếu k cho finish thì sẽ bị lỗi -> gọi phương thức onDestroyItem() -> container.removeView() xóa view đi sau khi đã thực hiện xong rồi -> container.addView(view)



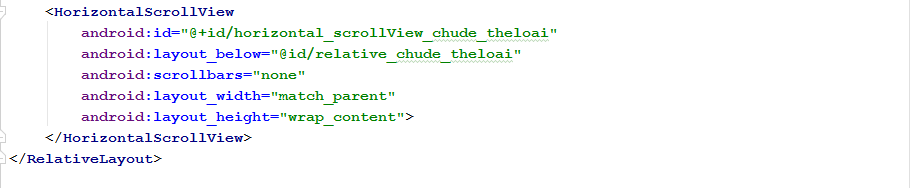


Tạo drawable custom\_background\_banner:



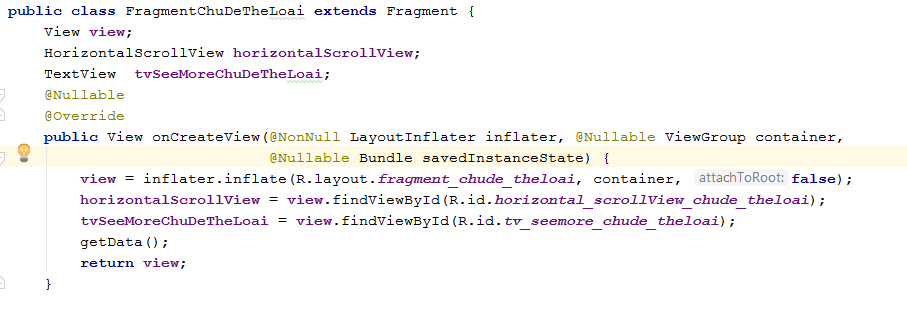
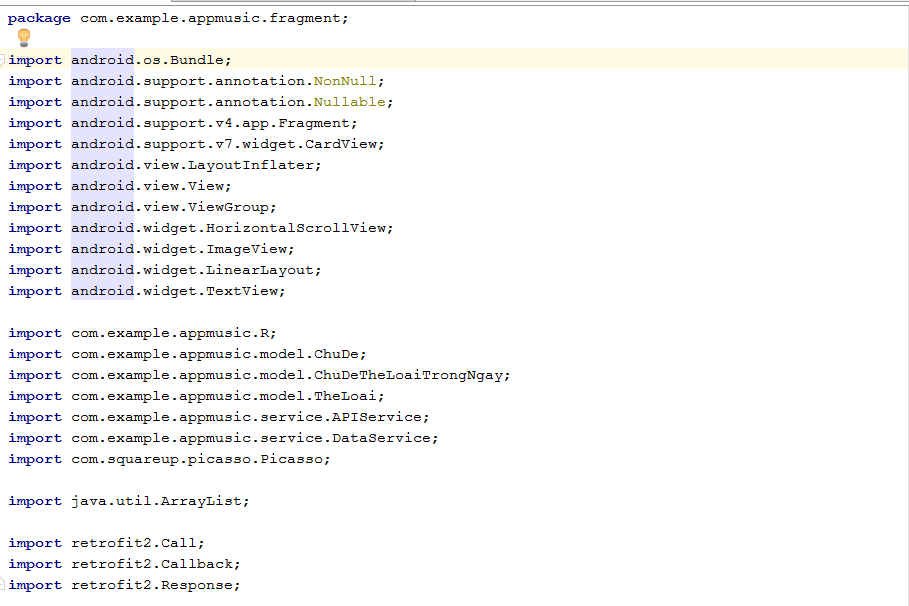
Tạo layout fragment\_chude\_theloai.xml





Tạo FragmentChuDeTheLoai: extends Fragment

Tạo horizontalScrollView & tvSeeMoreChuDeTheLoai



Tạo mảng chudeArrayList/theloaiArrayList -> chudeArrayList.addAll()/theloaiArrayList.addAll() (phương thức này add thêm 1 mảng cùng kiểu dữ liệu)

Gọi linearLayout -> Khởi tạo 1 ViewGroup -> Khi view được đưa vào linear muốn cho nó thực hiện theo chiều nào thì sẽ gọi linearLayout.setOrientation()

Sử dụng vòng lặp for cho hình ảnh chủ đề và hình ảnh thể loại

Khởi tạo cardView -> dùng tham số getActivity() -> để bo xung quanh dùng cardView.setRadius() -> đưa hình ảnh vào cardView dùng imageView, khởi tạo = tham số getActivity() -> gọi lại imageView.setScaleType() thì khi hình lỡ đâu nhỏ quá thì sẽ tự phóng ra vừa với kích thước của imageView -> xét if() -> gọi lại linearLayout.LayoutParams layout để set kích thước lại -> layout.setMargins() để dãn cách xung quanh -> gọi lại cardView.setLayoutParams(layout) ->đưa phần hình ảnh của imageView vào cardView cardView.addView(imageView)

-> làm tương tự cho phần mảng theloaiArrayList

-> gọi lại horizontalScrollView.addView(linearLayout)



