

# Bài 6: Fragment trong Android

Giáo viên: Lê Quốc Anh

# Nội dung

---

1. Fragment trong Android
2. Tạo một Fragment
3. Thay thế Fragment trong Activity
4. Fragment Backsacks
5. List Fragment
6. DialogFragment

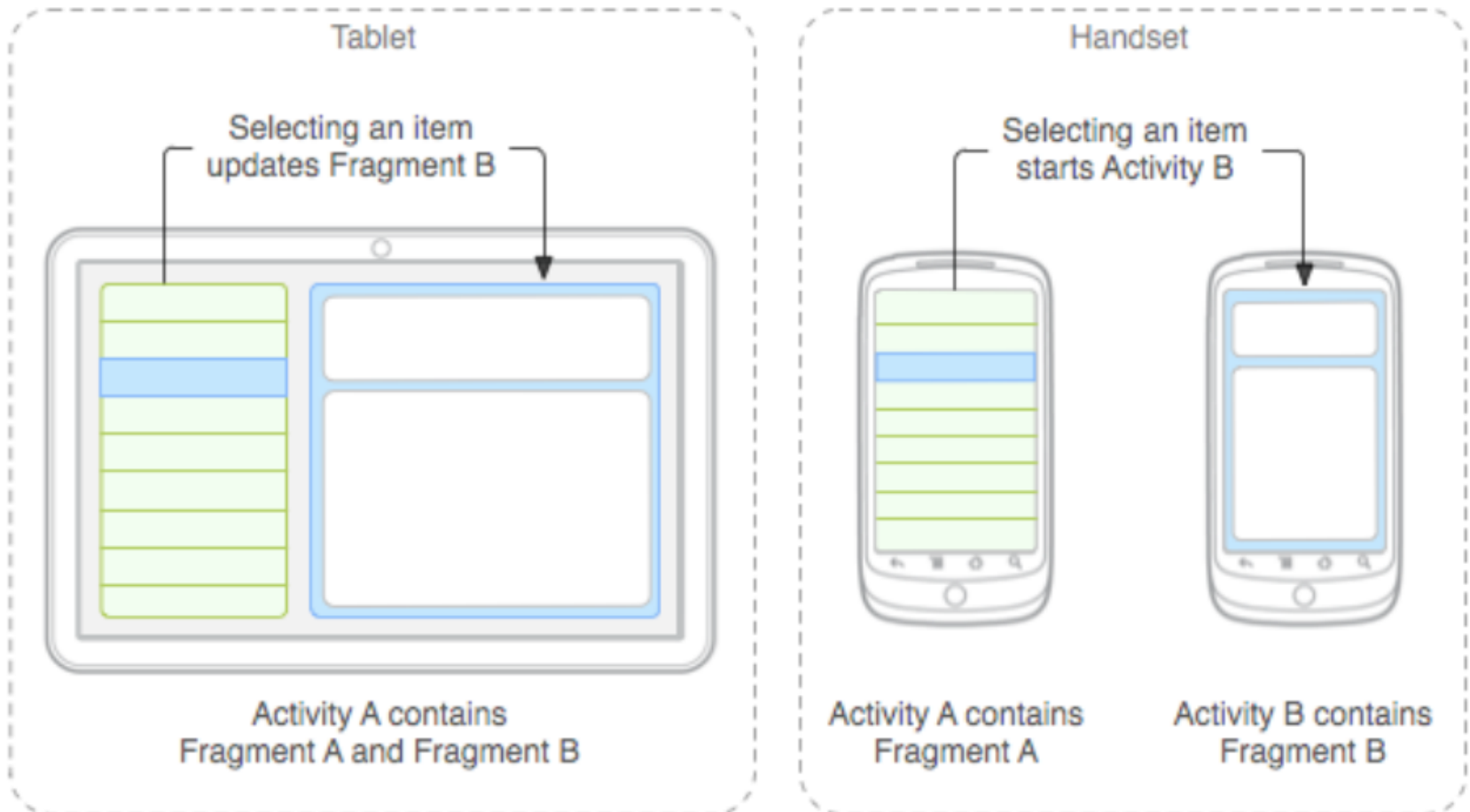
# 1. Fragment trong Android

---

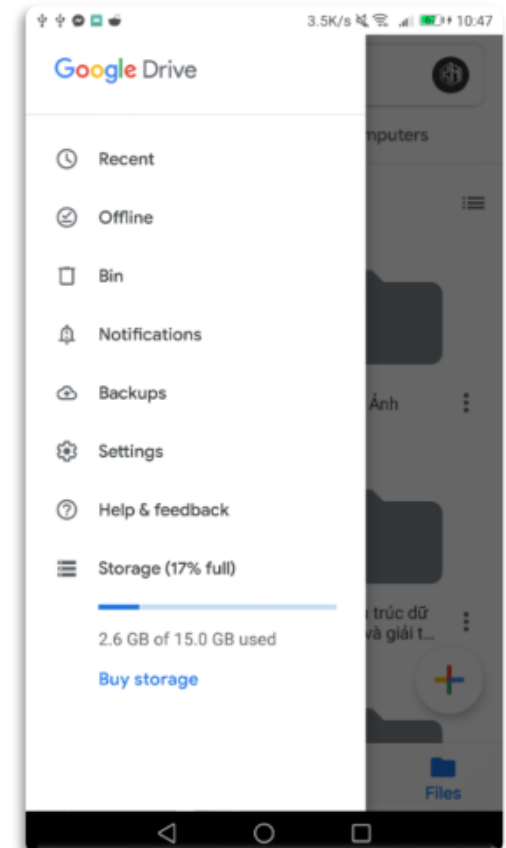
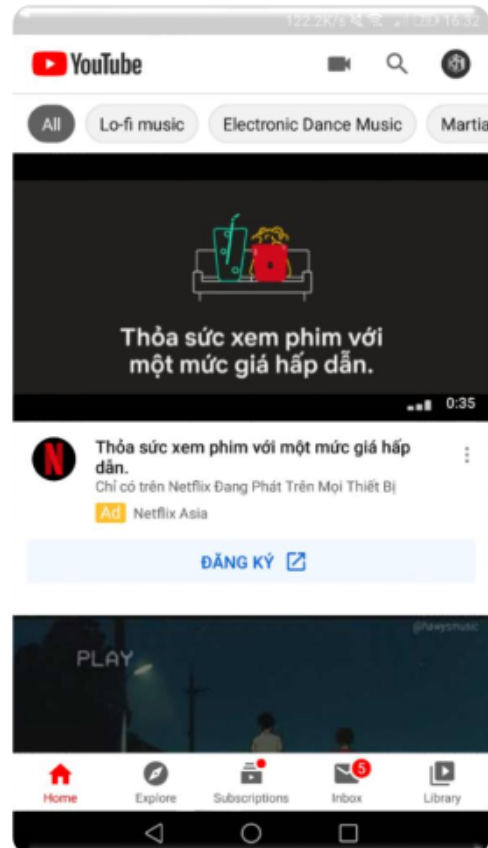
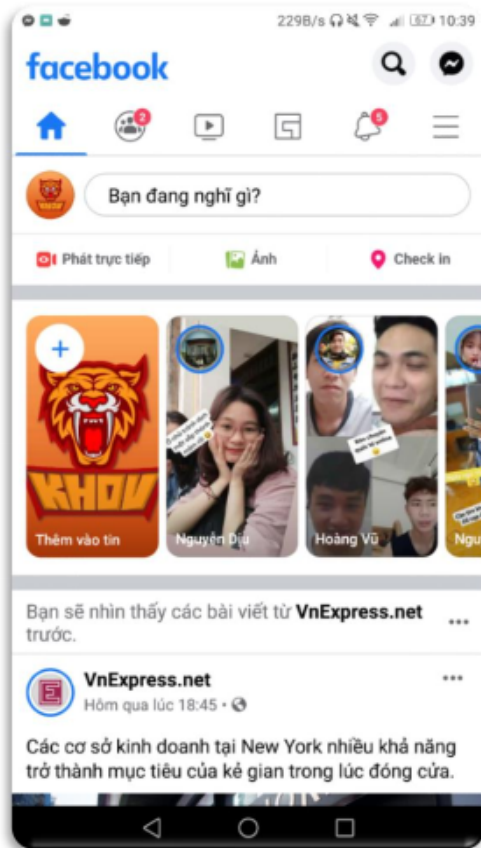
- Fragment là một phần của Activity
- Fragment chịu trách nhiệm quản lý một không gian màn hình.
- Không gian màn hình đó của Fragment phải nằm trong một Activity nào đó
- Một Activity có thể có nhiều Fragment
- Kết hợp nhiều fragment trong một activity để tạo giao diện nhiều khung (multi-pane UI)
- Một Fragment nào đó cũng có thể được khai báo và sử dụng bởi nhiều Activity khác nhau
- Có thể xem Fragment như một phần mô-đun (modular) của Activity

# 1. Fragment trong Android

- Một fragment chỉ có thể hoạt động bên trong một activity và phụ thuộc vào vòng đời của activity đó.

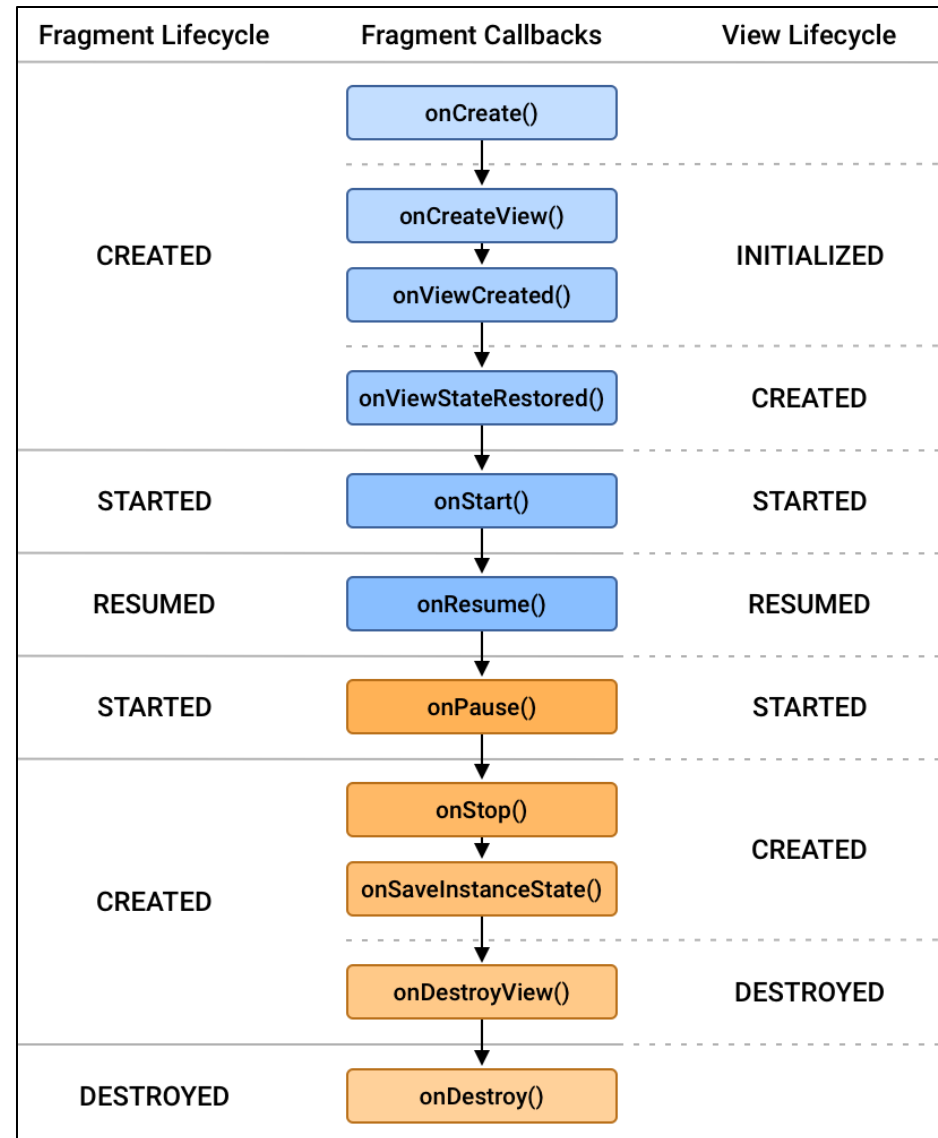


# Fragment được sử dụng

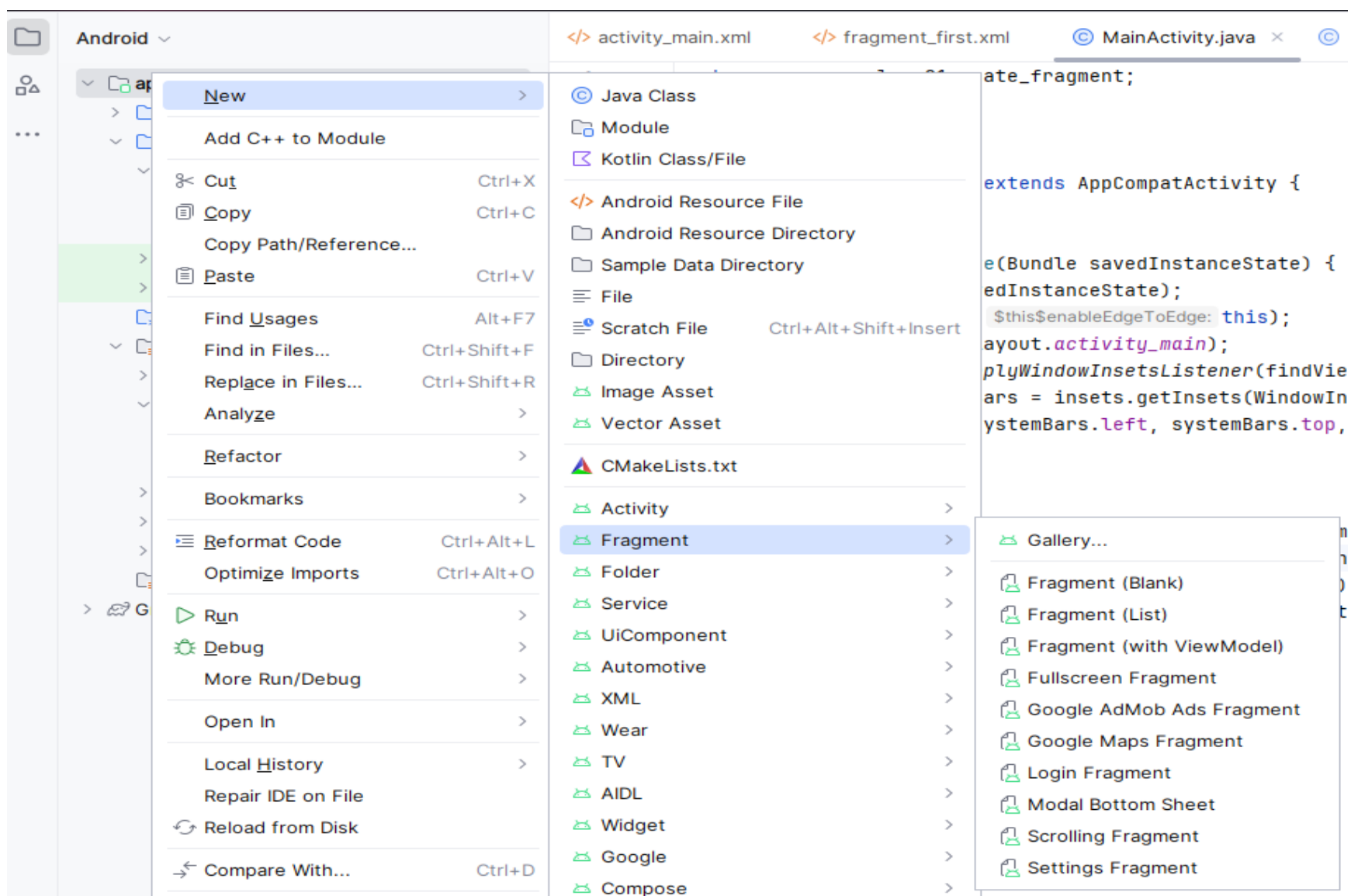


# Vòng đời của Fragment

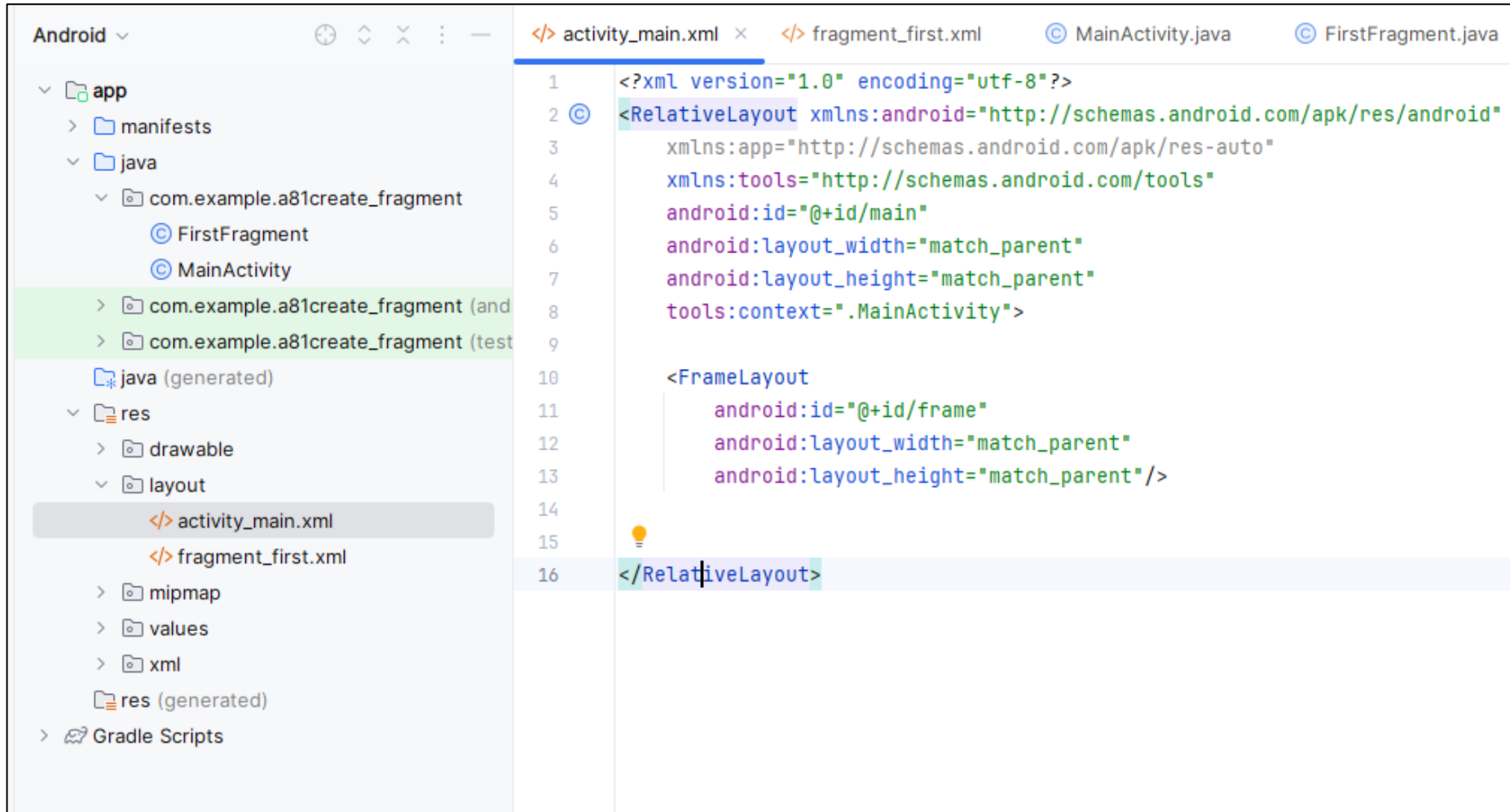
- **Initialized:** Đã tạo Fragment, chưa có View.
- **Created:** Có View, chưa hiển thị.
- **Started:** Hiển thị nhưng chưa hoạt động tối đa.
- **Resumed:** Đang hoạt động, tương tác được.
- **Paused:** Không tương tác, nhưng còn hiển thị.
- **Stopped:** Không còn hiển thị.
- **View Destroyed:** View bị hủy, Fragment vẫn còn.
- **Destroyed:** Fragment bị xóa hoàn toàn.



## 2. Tạo một Fragment



## 2. Tạo một Fragment



The screenshot displays the Android Studio interface. On the left, the 'app' directory is expanded, showing the 'com.example.a81create\_fragment' package. The 'FirstFragment' and 'MainActivity' classes are visible. The 'activity\_main.xml' file is selected in the 'layout' directory. The main editor shows the XML layout for 'activity\_main.xml' with the following content:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10     <FrameLayout
11         android:id="@+id/frame"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"/>
14
15
16 </RelativeLayout>
```

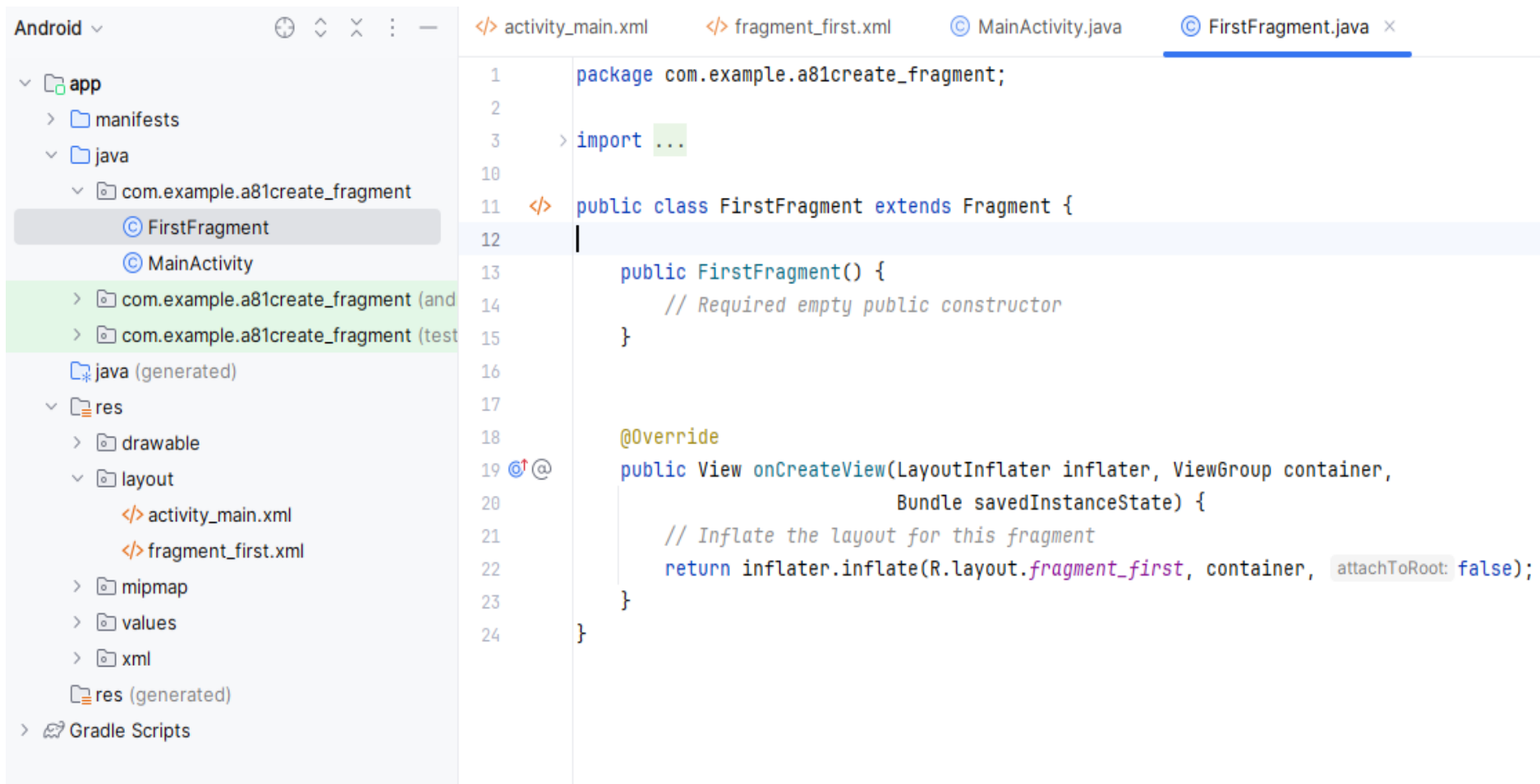


# 2. Tạo một Fragment

The screenshot displays the Android Studio interface. On the left, the 'Project' view shows the package structure: `com.example.a81create_fragment` containing `FirstFragment` and `MainActivity`. The `MainActivity` file is selected. The main editor window shows the Java code for `MainActivity`, which extends `AppCompatActivity`. The code includes an `onCreate` method that calls `super.onCreate`, enables `EdgeToEdge`, sets the content view to `R.layout.activity_main`, and adds a `FirstFragment` instance to the `SupportFragmentManager`. The code is as follows:

```
1 package com.example.a81create_fragment;
2
3 import ...
4
12
13 public class MainActivity extends AppCompatActivity {
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
19         setContentView(R.layout.activity_main);
20         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
21             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
22             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
23             return insets;
24         });
25
26         FragmentManager fragmentManager = getSupportFragmentManager();
27         FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
28         FirstFragment firstFragment = new FirstFragment();
29         fragmentTransaction.add(R.id.frame, firstFragment);
30         fragmentTransaction.commit();
31
32     }
33 }
```

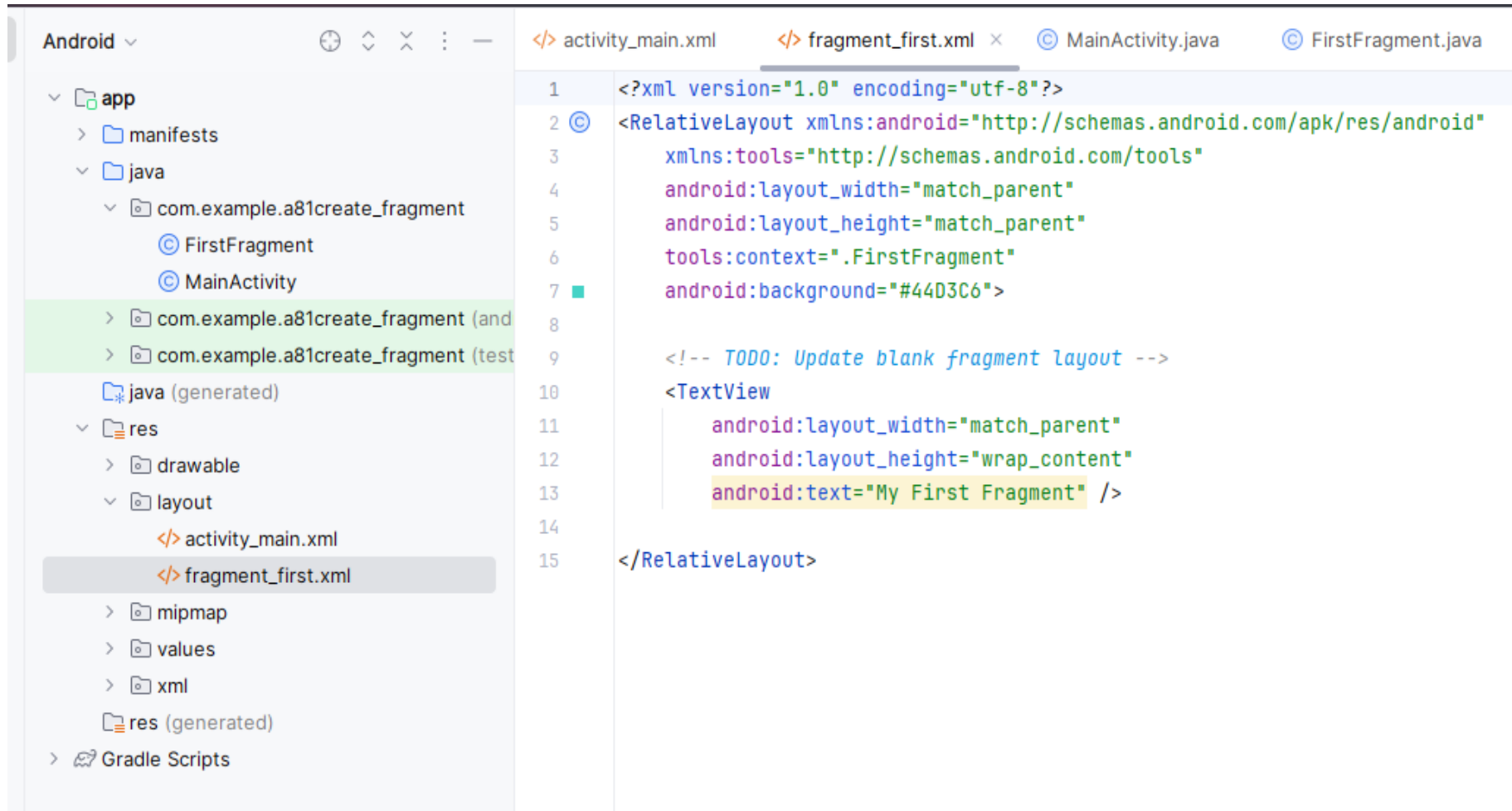
## 2. Tạo một Fragment



The screenshot displays the Android Studio IDE interface. On the left, the 'Project' view shows the package structure: `com.example.a81create_fragment` containing `FirstFragment` and `MainActivity`. The `FirstFragment` class is selected. The main editor window shows the code for `FirstFragment.java`.

```
1 package com.example.a81create_fragment;
2
3 > import ...
10
11 </> public class FirstFragment extends Fragment {
12
13     public FirstFragment() {
14         // Required empty public constructor
15     }
16
17
18     @Override
19     public View onCreateView(LayoutInflater inflater, ViewGroup container,
20                             Bundle savedInstanceState) {
21         // Inflate the layout for this fragment
22         return inflater.inflate(R.layout.fragment_first, container, attachToRoot: false);
23     }
24 }
```

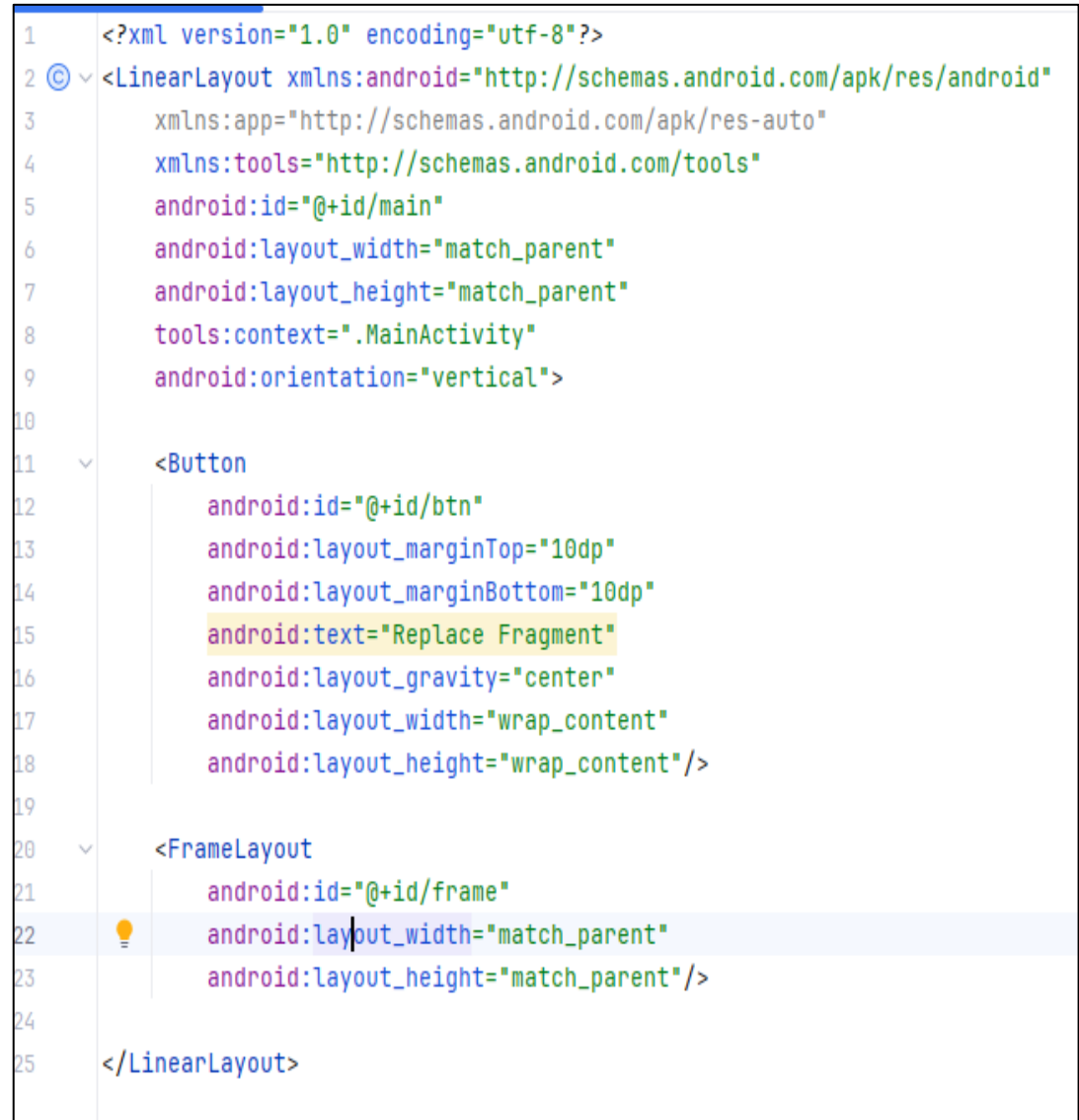
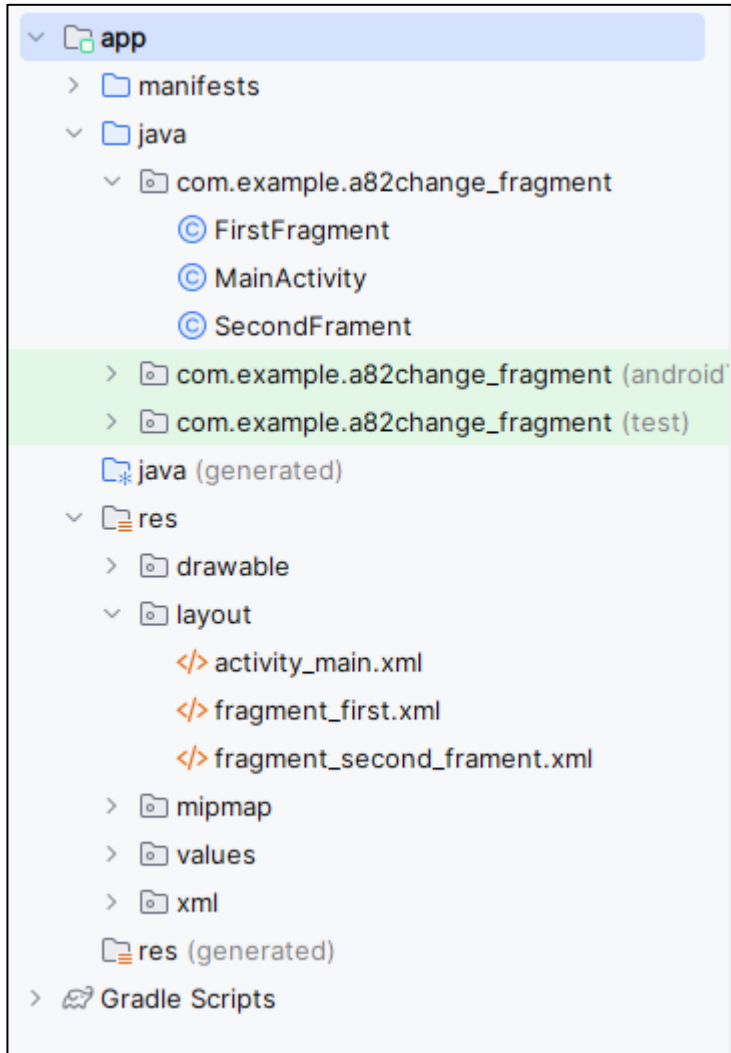
## 2. Tạo một Fragment



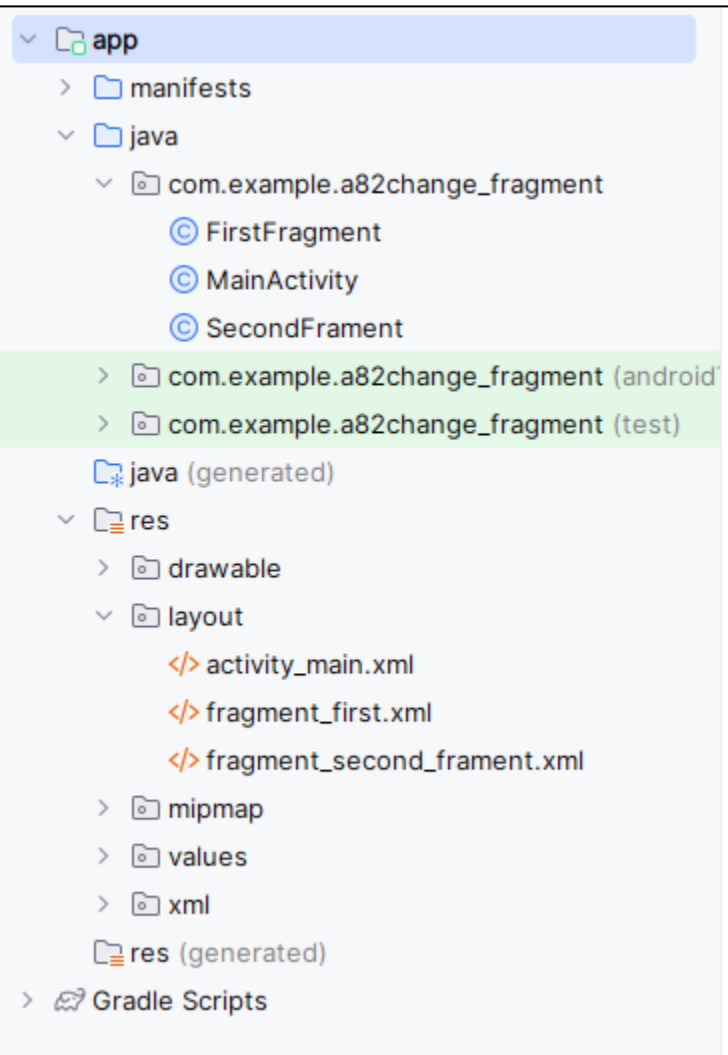
The screenshot displays the Android Studio IDE interface. On the left, the 'app' folder is expanded, showing the 'res' directory and its subfolders: 'drawable', 'layout', 'mipmap', 'values', and 'xml'. The 'layout' folder is selected, and the 'fragment\_first.xml' file is highlighted. The main editor area shows the XML code for the 'fragment\_first.xml' file, which is a RelativeLayout containing a TextView with the text 'My First Fragment'.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     tools:context=".FirstFragment"
7     android:background="#44D3C6">
8
9     <!-- TODO: Update blank fragment layout -->
10    <TextView
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:text="My First Fragment" />
14
15 </RelativeLayout>
```

# 3. Thay thế Fragment trong Activity



# 3. Thay thế Fragment trong Activity



```
public class MainActivity extends AppCompatActivity {  
    2 usages  
    private Button btn;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);  
        setContentView(R.layout.activity_main);  
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
            return insets;  
        });  
  
        FragmentManager fragmentManager = getSupportFragmentManager();  
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  
  
        FirstFragment firstFragment = new FirstFragment();  
        fragmentTransaction.add(R.id.frame, firstFragment);  
        fragmentTransaction.commit();  
  
        btn = findViewById(R.id.btn);  
  
        btn.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                FragmentManager fragmentManager1 = getSupportFragmentManager();  
                FragmentTransaction fragmentTransaction1 = fragmentManager1.beginTransaction();  
                SecondFrament secondFrament = new SecondFrament();  
                fragmentTransaction1.replace(R.id.frame, secondFrament);  
                fragmentTransaction1.commit();  
            }  
        });  
    }  
}
```

## 4. Fragment Backsacks

---

Khi bạn quản lý fragment bởi các phương thức `replace()` hay `remove()` trên đây, bạn có thể sử dụng thêm phương thức `addToBackStack()` này. Nếu bạn gọi đến phương thức này trước khi gọi `commit()` sẽ được nói ở dưới, thì hệ thống sẽ đưa fragment ở transaction hiện tại vào Back Stack. Điều này có ý nghĩa là, fragment bị thay thế hay bị gỡ ra khỏi `FrameLayout` ở transaction này sẽ không bị xóa khỏi hệ thống mà vẫn còn được quản lý bên trong Back Stack. Và do fragment không bị hủy khỏi back stack, nên nếu user nhấn nút back sau đó, họ hoàn toàn có thể quay trở lại với fragment trước đó đã bị gỡ ra.



# 4. Fragment Backsacks

---

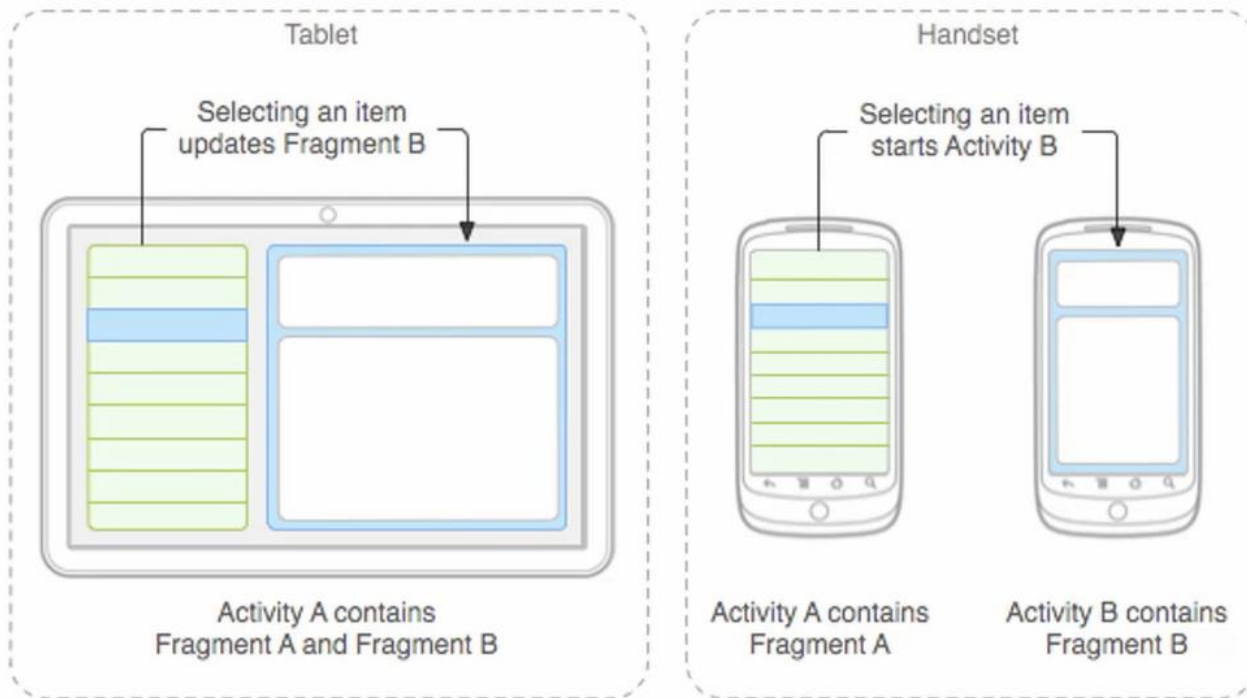
## Stack



- Giả sử trong trường hợp chỉ có một khung Fragment, và muốn hoán đổi nhiều Fragment khác lần lượt vào cái khung đó?
- Nút **Back** trên thiết bị Android mặc định sẽ chỉ có tác dụng với Activity hiện thời chứ không phải Fragment. Mà ở đây nút **Back** để quay lại Fragment trước? Ta sẽ cần sử dụng Fragment Backstack: Đưa các Fragment vào một ngăn xếp (giống như khi ta duyệt trang web vậy, nhấn **back** thì trình duyệt sẽ quay lại trang trước đó).
- Ta sử dụng phương thức `addToBackStack()` để quay lại fragment trước đó

# 5. List Fragment

- List Fragment được sử dụng để hiển thị dữ liệu từ một danh sách list
- Khi click vào một thành phần của list, thì sẽ thực hiện một Transaction





# 6. DialogFragment

- Dialog Fragment thường được sử dụng để nhận dữ liệu từ người dùng hoặc giao tiếp với người dùng
- Dialog Fragment được đặt trong Activity



---

**THANK YOU**  
**for**  
**YOUR ATTENTION**

