



## **THIẾT KẾ GIAO DIỆN ANDROID**

### **BÀI 6: FRAGMENT , VIEWPAGER**

- ⊙ Kết thúc bài học này bạn có khả năng
  - ⊙ Hiểu được Fragment
  - ⊙ Biết được cách tạo ViewPager & TabLayout



## Phần I: Fragment

 A Giới thiệu Fragment

 B Vòng đời Fragment

 C Tạo, quản lý Fragment

## Phần II: ViewPager

 D Tạo ViewPager

 E Tạo TabLayout

 F Phối hợp ViewPager và TabLayout



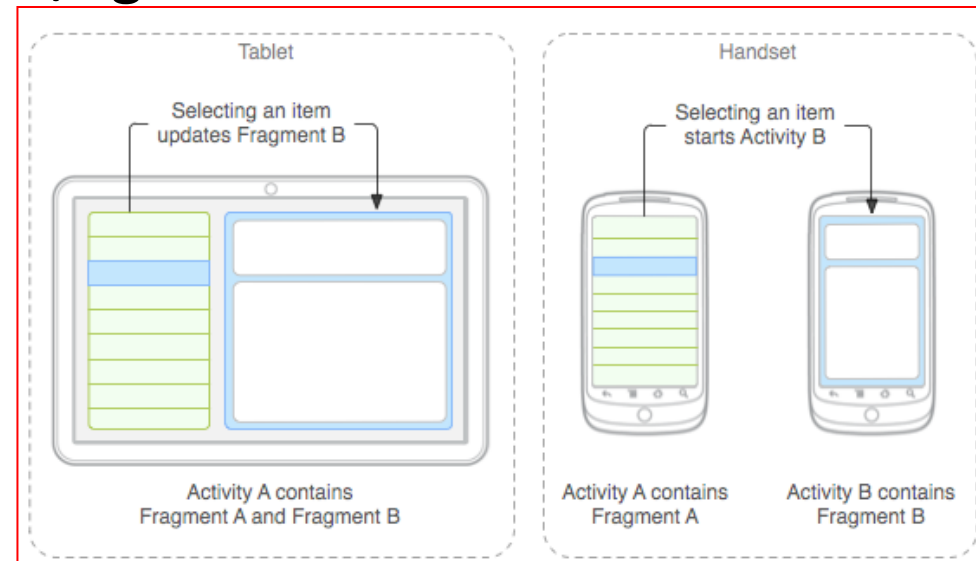


## **BÀI 6: FRAGMENT & VIEWPAGER**

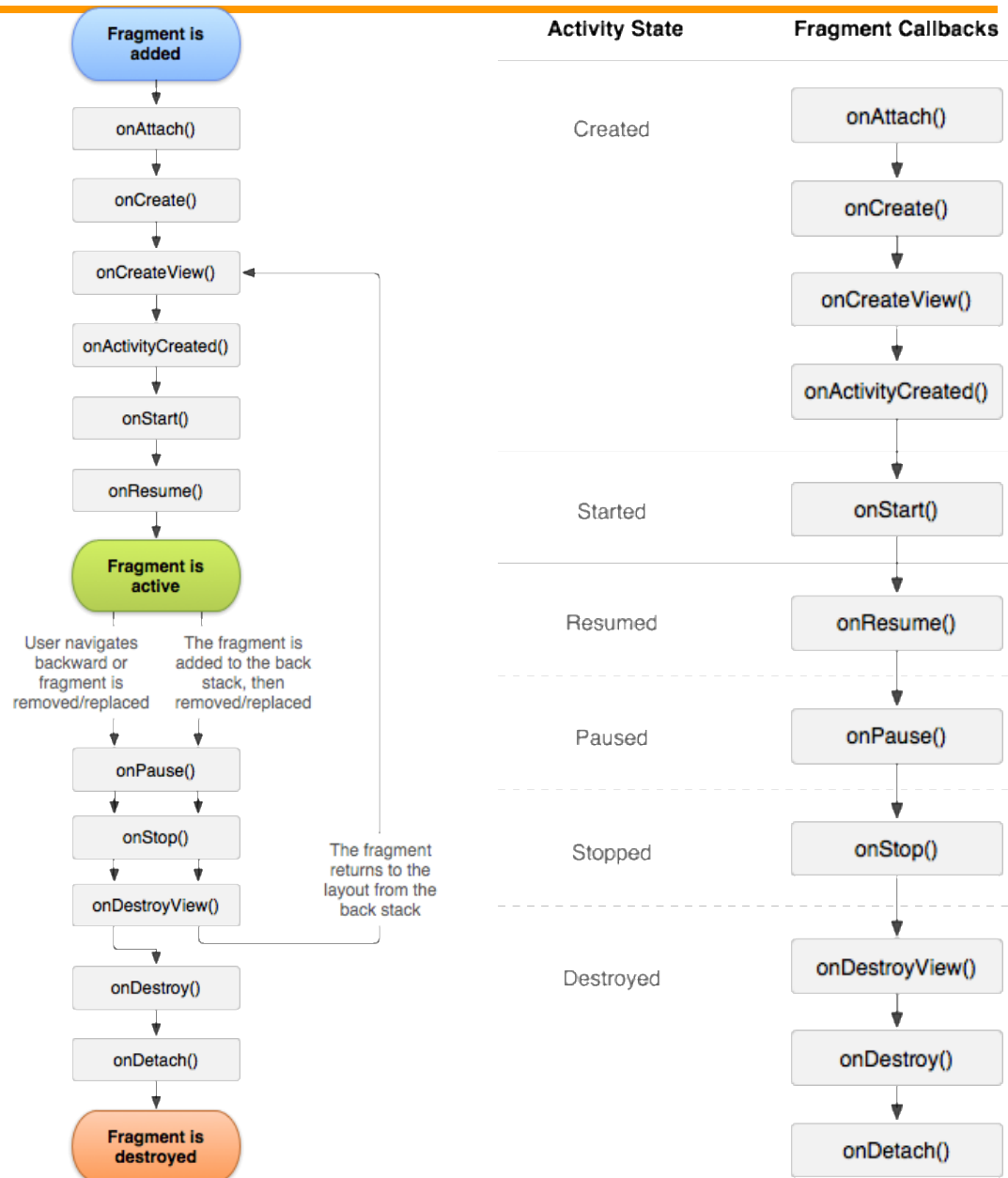
---

### **PHẦN 1: FRAGMENT**

- ❑ Fragment là 1 phân đoạn giao diện được nhúng vào 1 activity, fragment không chạy độc lập với activity được.
- ❑ Fragment giúp chương trình linh động và hướng module hơn.
- ❑ Một fragment có thể chiếm 1 phần hoặc toàn bộ giao diện Activity. Một Activity có thể có nhiều fragment.
- ❑ Fragment có thể cố định hoặc có thể được tạo, thêm, xóa.... Trong quá trình Activity hoặc động
- ❑ Fragment được đưa vào Android 3.0
- ❑ Nên dùng Fragment trong gói support v4 để tương thích ngược



- Fragment cũng có lifecycle riêng của nó và phụ thuộc vào lifecycle của Activity chứa nó

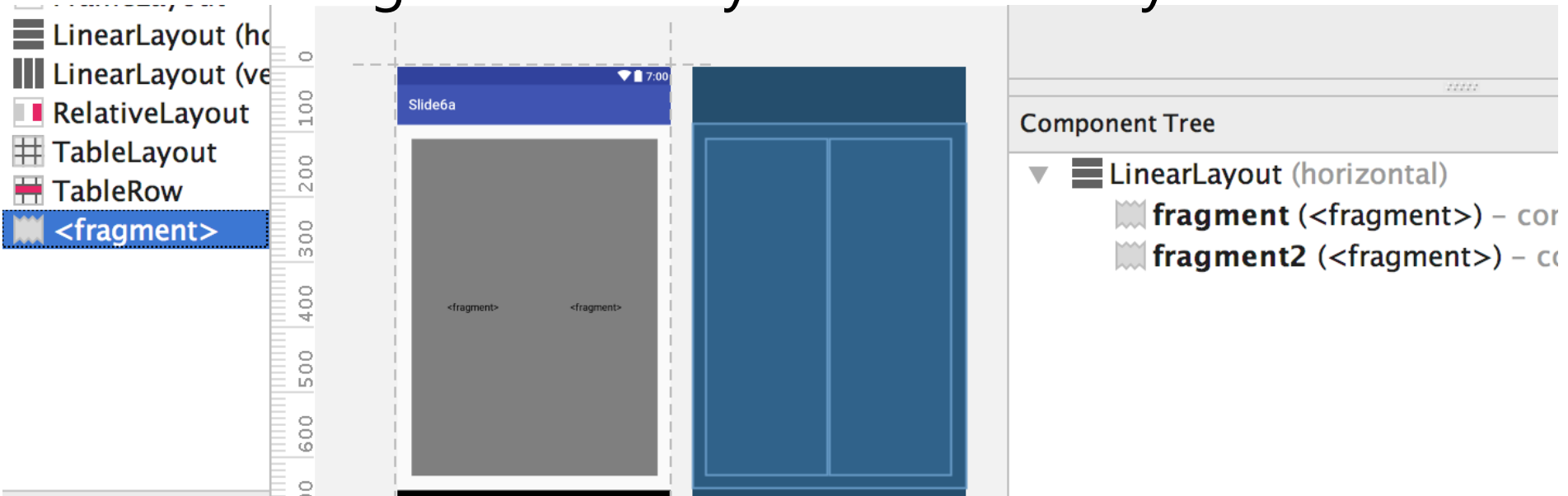


- ❑ Tạo 1 layout mới, thiết kế tùy ý
- ❑ Tạo 1 class mới kế thừa từ Fragment và gán layout vào (chú ý: nên dùng gói support)

```
public class LeftFragment extends Fragment {  
  
    @Override  
    public void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
        View view=inflater.inflate(R.layout.fragment_left,container,false);  
        return view;  
    }  
}
```

- ❑ Thực hiện tương tự để tạo 1 fragment Right

## ❑ Gắn 2 Fragment vào layout của Activity chính



## ❑ Chạy và xem kết quả





❑ Ta có thể quản lý (thêm, thay thế...) các Fragment trong khi đang chạy chương trình với các bước sau

1. Dùng `Activity.getFragmentManager()` để tham chiếu đến `FragmentManager` (nếu dùng "Compatibility package" thì dùng `FragmentManager.supportFragmentManager()`)
2. Dùng `FragmentManager.beginTransaction()` để lấy 1 instance của `FragmentTransaction`.
3. Khởi tạo fragment.
4. Dùng `FragmentTransaction.add()` để thêm fragment vào.
5. Gọi `FragmentTransaction.commit()` để chấp

- ❑ Ngoài ra có thể dùng thêm các lệnh của FragmentTransaction như : add(), remove(), replace(), hide(), show(), detach() (api 13), attach() (api 13).

```
FragmentManager fragmentManager = getFragmentManager()  
// hoặc: FragmentManager fragmentManager = getSupportFragmentManager()  
fragmentManager.beginTransaction()  
    .remove(fragment1)  
    .add(R.id.container, fragment2)  
    .show(fragment3)  
    .hide(fragment4)  
    .commit();
```

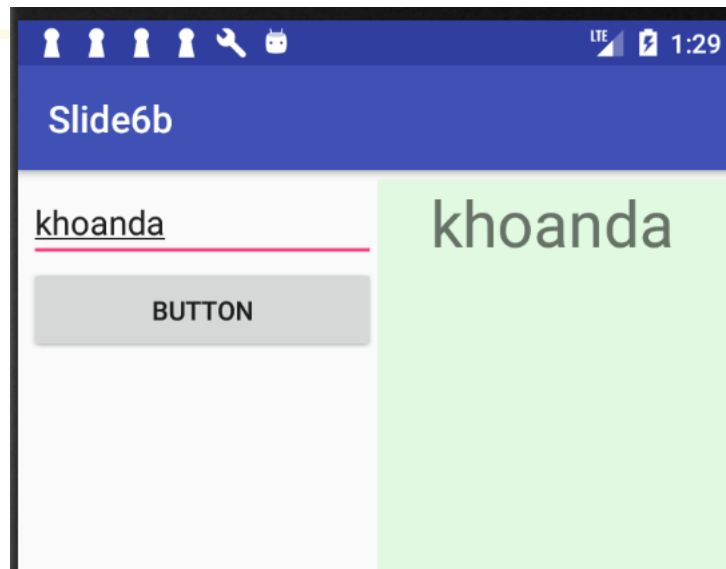


- ❑ Một fragment gắn liền với activity chứa nó. Activity có thể gọi trực tiếp các phương thức public của Fragment thông qua tham chiếu đến đối tượng Fragment.
- ❑ Nếu ta không giữ lại các tham chiếu đến Fragment khi tạo nó ta cũng có thể tìm nó thông qua FragmentManager, dùng `findFragmentById()` hoặc `findFragmentByTag()`.
- ❑ Fragment cũng có thể truy cập tới Activity chứa nó bằng phương thức `Fragment.getActivity()`.

- ❑ Vd: nhấn button ở fragment left cập nhật textview bên fragment right

```
bt.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        String chuoi=et.getText().toString();
        FragmentManager ma=getActivity().getSupportFragmentManager();
        RightFragment rf=(RightFragment)ma.findFragmentById(R.id.fragmentright);
        rf.tv.setText(chuoi);
    }
});
```





# DEMO

Tạo fragment

Quản lý (thêm, đổi) fragment

Giao tiếp giữa các fragment





## **BÀI 6: FRAGMENT & VIEWPAGER**

---

### **PHẦN 2: VIEWPAGER**

- ❑ ViewPager là một dạng view sử dụng Fragment
- ❑ ViewPager cho phép ta dùng thao tác trước tay để chuyển giữa các Fragment. Phối hợp thêm với tab để tạo ra view giống tabhost nhưng không cần tab



- ❑ Tạo 1 Fragment mới có chứa 1 TextView
- ❑ Layout chính thêm vào một ViewPager

```
<android.support.v4.view.ViewPager  
    android:id="@+id/viewpager"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
</android.support.v4.view.ViewPager>
```



- ❑ Thêm một class như sau hình (hàm getCount trả về tổng số Fragment, hàm getItem trả về Fragment tại 1 vị trí index)

```
class MyAdapter extends FragmentStatePagerAdapter
{
    int numeroftab;
    public MyAdapter(FragmentManager fm, int numeroftab) {
        super(fm);
        this.numeroftab=numeroftab;
    }
    @Override
    public Fragment getItem(int position) {
        return null;
    }
    @Override
    public int getCount() {
        return numeroftab;
    }
}
```

- ❑ Trong getItem viết thêm để tạo Fragment và truyền đối số dựa vào position như sau:

```
@Override
public Fragment getItem(int position) {
    TextFragment fragment=new TextFragment();
    Bundle ds_doiso=new Bundle();
    ds_doiso.putString("doiso1",position+"");
    fragment.setArguments(ds_doiso);
    return fragment;
}
```

- ❑ Trong onCreate của Activity:

```
MyAdapter adapter=new MyAdapter(getSupportFragmentManager(),3);
viewPager.setAdapter(adapter);
```

- ❑ Trong onCreateView của Fragment, dùng hàm getArguments để lấy đối số truyền vào như sau:

```
if (getArguments() != null) {  
    String chuỗi = getArguments().get("doisoi").toString();  
    tv.setText(chuoi);  
}
```

- ❑ Chạy chương trình, trực tay để chuyển giữa các Fragment trong ViewPager

- ❑ Có nhiều cách để thêm tab vào cho ViewPager như dùng TabHost, PagerTabStrip, TabLayout.
- ❑ Ta sẽ dùng TabLayout
- ❑ TabLayout nằm trong material design (ta sẽ bàn đến ở bài sau) nên ta cần nhúng nó trong file build.gradle

```
compile 'com.android.support:design:25.3.1'
```

## ❑ Layout chính thêm 1 TabLayout trên

```
<android.support.design.widget.TabLayout
    android:id="@+id/tab_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:elevation="6dp"
    android:minHeight="?attr/actionBarSize"
    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"/>
```

## ❑ Java chính thêm các tab

```
tabLayout.addTab(tabLayout.newTab().setText("Tab 1"));
tabLayout.addTab(tabLayout.newTab().setText("Tab 2"));
tabLayout.addTab(tabLayout.newTab().setText("Tab 3"));
tabLayout.setTabGravity(TabLayout.GRAVITY_FILL);
```

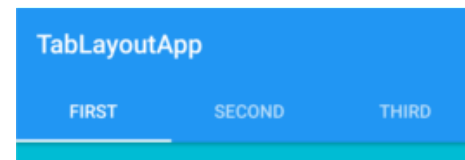
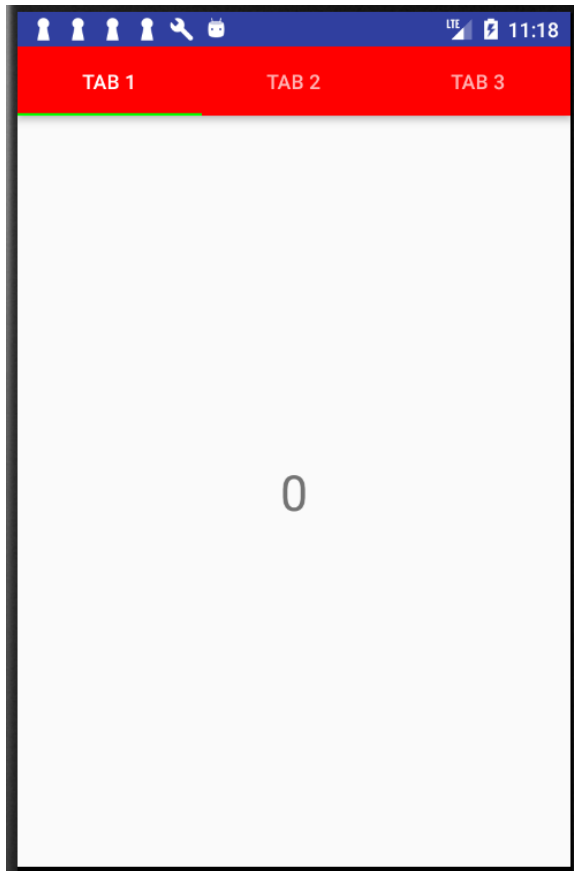
❑ Trước viewPager đổi Tab:

```
viewPager.addOnPageChangeListener(new  
    TabLayout.TabLayoutOnPageChangeListener(tabLayout));
```

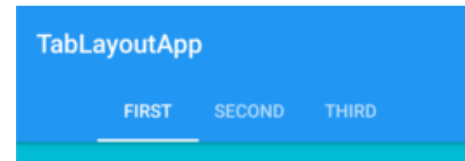
❑ Click Tab đổi ViewPager:

```
tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListene  
    @Override  
    public void onTabSelected(TabLayout.Tab tab) {  
        viewPager.setCurrentItem(tab.getPosition());  
    }  
    @Override  
    public void onTabUnselected(TabLayout.Tab tab) {  
    }  
    @Override  
    public void onTabReselected(TabLayout.Tab tab) {  
    }  
});
```

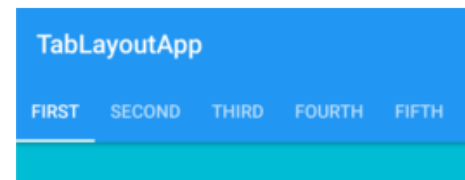
- ❑ Đổi màu tab trong file style, đổi kiểu tab bằng thuộc tính `tabMode`, `tabGravity` :



`tabMode = fixed`  
`tabGravity = fill`



`tabMode = fixed`  
`tabGravity = center`



`tabMode = scrollable`



# DEMO

ViewPager & TabLayout





- ☐ Fragment
- ☐ ViewPager
- ☐ TabLayout





**Cảm ơn**