

Мобайл програмчлал – Лекц 11

Ж.Золжаргал
(j.zoljargal@must.edu.mn)



What are fragments?

- Fragment нь Activity дээр хэрэглэгддэг биеэ даасан компонент юм
- Fragment нь Activity дотор ажилладаг ч өөрийн гэсэн амьдралын муж (life cycle), өөрийн гэсэн хэрэглэгчийн интерфэйстэй
- Fragment-г динамикаар болон статикаар Activity-д нэмж болно



Advantages of using fragments

- Ялгаатай layout-ууд дээр програмын компонентуудыг дахин ашиглахад хялбар болгож өгдөг
- Хэрэглээ:
 - Гар утсанд зориулж single-pane, таблет-д зориулж multi-pane хийх
 - Андроид төхөөрөмж дэх дараалсан олон UI дүрслэхэд тохиромжтой



Advantages of using fragments

- Андройд 3.0 (Honeycomb – API level 11) хувилбарт анх танилцуулагдсан.
- Support library-д агуулагддаг. Андройд 1.6 (API level 4)
- Support library дэх Fragment-ыг ашиглах үед activity нь FragmentActivity Классаас удамших ёстой.



Fragment skeleton code

```
package com.paad.fragments;

import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class MySkeletonFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater,
                           ViewGroup container,
                           Bundle savedInstanceState) {
        // Create, or inflate the Fragment's UI, and return it.
        // If this Fragment has no UI then return null.
        return inflater.inflate(R.layout.my_fragment, container, false);
    }
}
```



Advantages of using fragments

- Fragment-г Activity шиг manifest.xml-д бүртгүүлэх шаардлагагүй
- Fragment-н onCreateView үзэгдлийг даран тодорхойлж UI-ийг дэлгэцэнд дүрслэнэ



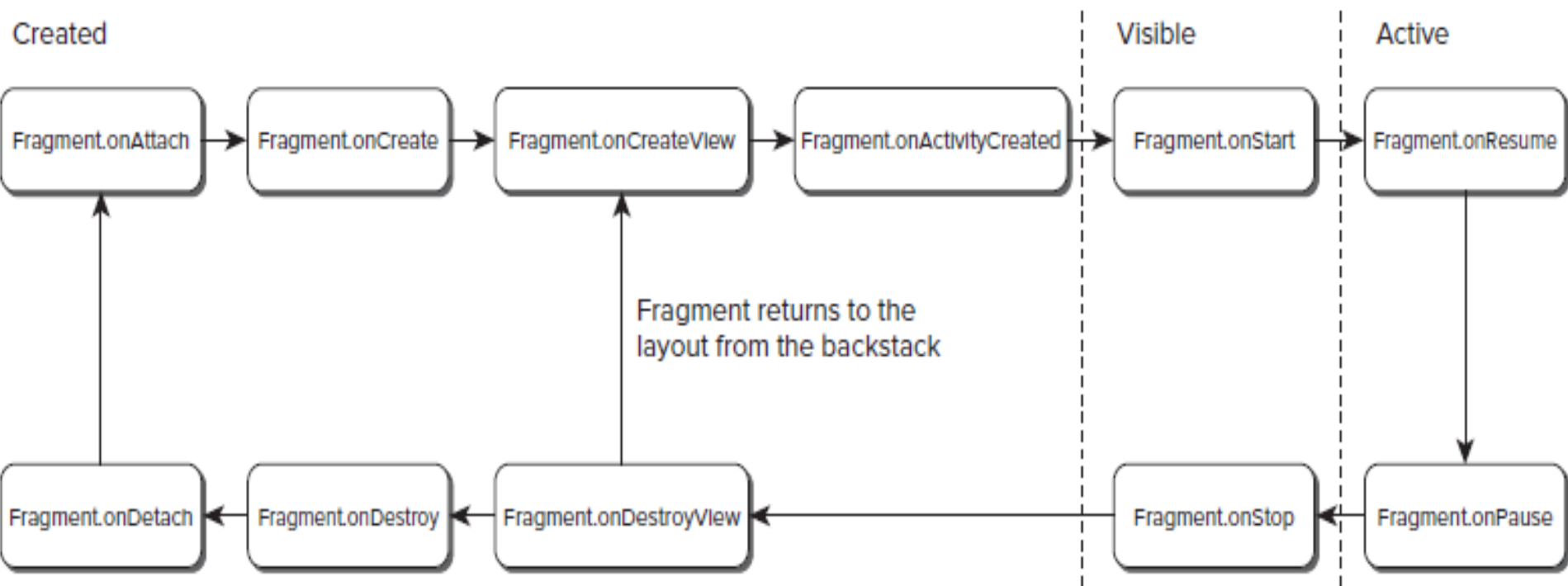
Fragment lifecycle

- Fragment-ийн lifecycle үзэгдлүүд нь түүний эцэг Activity-ийн lifecycle-ийн тусгал байна.
 - Created
 - Started
 - Paused
 - Stopped
 - Destroyed



Fragment lifecycle

- Ихэнх үзэгдлийг даран тодорхойлоход super классын үйлдлийг дуудах шаардлагатай.



Fragment lifecycle

- Fragment-ийн амьдралын муж нь Activity дээр холбогдоход эхэлж, тухайн Activity-ээс салгахад төгсдөг.
 - `onAttach()`, `onDetach()`
- Activity нь `onDestroy()` үзэгдэл биелэхгүйгээр төгсөх тохиолдол цөөнгүй байдаг.
- Тиймээс Fragment-ийн `onDestroy()` үзэгдэлд найдах хэрэггүй.



Fragment states

- Activity нь foreground орчинд идэвхтэй байхад Fragment нь идэвхтэй байна
- Activity нь pause эсвэл stop болоход Fragment мөн pause болон stop болж тус бүр нь идэвхгүй (inactive) болно.
- Activity устахад, түүний агуулсан fragment бүр мөн устана.



Fragment states

- Fragment ашигласнаар идэвхтэй Activity-ийн UI-г Fragment нэмэх болон устгах байдлаар уян хатан үзүүлэх боломжтой.
- Энэ нь эцэг Activity-ийн амьдралын муж нь идэвхтэй төлөвт байхад Fragment нь өөрийн амьдарлын мужуудаараа шилжинэ
- Fragment идэвхгүй төлөвөөс идэвхтэй төлөвт иргэн шилжихэд UI-ийн төлөв болон өгөгдлийг хадгалаад иргэн сэргээх шаардлагатай



Introducing the Fragment manager

- Activity бүр өөрийн агуулсан Fragment-уудаа удирдах Fragment Manager классаар хангагдсан байдаг.

```
FragmentManager fragmentManager = getFragmentManager();
```

- Fragment manager нь одоогын Activity-д нэмэгдсэн Fragment-д хандах функцээр хангаж өгдөг.
 - Fragment-ийг нэмэх, устгах, солих

Adding Fragments to Activity

- Энгийнээр Activity дотор Fragment-г үүсгэх

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment android:name="com.paad.weatherstation.MyListFragment"
        android:id="@+id/my_list_fragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
    />
    <fragment android:name="com.paad.weatherstation.DetailsFragment"
        android:id="@+id/details_fragment"
        android:layout_width="match_parent"

        android:layout_height="match_parent"
        android:layout_weight="3"
    />
</LinearLayout>
```

Adding Fragments to Activity

- Хэрэв динамикаар layout-аа шинэчлэх буюу fragment нэмэх, устгах болон солих үед түүнийг агуулагч элементийг Activity-д тодорхойлж өгнө.
- Fragment-ийг үүсгэх болон нэмэх кодыг түүний эцэг Activity-ийн onCreate үзэгдэлд тодорхойлж өгнө



Adding Fragments to Activity

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <FrameLayout
        android:id="@+id/ui_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
    />
    <FrameLayout
        android:id="@+id/details_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="3"
    />
```

Fragment transaction

- Fragment transaction нь идэвхтэй Activity дотор Fragment-уудыг нэмэх, устгах болон солих үйлдэл хийхэд ашиглагддаг.
- Мөн шилжилт хийхэд тодорхой Animation-г дэмждэг.
- Шинэ Fragment шилжилтийг `beginTransaction()` функцээр үүсгэж, өөрчлөлтийг түүний `commit()` функцээр хэрэгжүүлнэ.



Fragment transaction

- Жишээ код

```
FragmentManager fragmentManager = getSupportFragmentManager();  
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  
  
// Add, remove, and/or replace Fragments.  
// Specify animations.  
// Add to back stack if required.  
  
fragmentTransaction.commit();
```



Add, remove and replace



- Фрагмент нэмэх

```
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  
fragmentTransaction.add(R.id.ul_container, new MyListFragment());  
fragmentTransaction.commit();
```

- Фрагмент устгах, findFragmentByTag

```
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  
Fragment fragment = fragmentManager.findFragmentById(R.id.details_fragment);  
fragmentTransaction.remove(fragment);  
fragmentTransaction.commit();
```

- Фрагмент солих

```
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  
fragmentTransaction.replace(R.id.details_fragment,  
                           new DetailFragment(selected_index));  
fragmentTransaction.commit();
```

Using the fragment manager to find Fragments

- Activity-ээс Fragment-ийг findFragmentById функцээр хайна.

```
MyFragment myFragment =  
    (MyFragment) fragmentManager.findFragmentById(R.id.MyFragment);
```

- Fragment нэмэхдээ Fragment transaction ашигласан үед түүнийг агуулах элементийн нөөцийн хаягыг ашиглах бөгөөд Fragment-ийг хайхдаа Fragment Transaction-д өөрийн тодорхойлсон Tag-аар хайна.

```
MyFragment myFragment =  
    (MyFragment) fragmentManager.findFragmentByTag(MY_FRAGMENT_TAG);
```

Populating dynamic Activity layouts with Fragments

- Run-time үед Fragment-уудаа динамикаар солих үед XML Layout файлд зөвхөн эцэг элементийг тодорхойлох нь тохиромжтой.
 - Тохиргоо өөрчлөгдөхөд UI дахин үүсгэнэ.
(screen rotation)



Populating dynamic Activity layouts with Fragments



```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    // Inflate the layout containing the Fragment containers  
    setContentView(R.layout.fragment_container_layout);  
  
    FragmentManager fm = getFragmentManager();  
  
    // Check to see if the Fragment back stack has been populated  
    // If not, create and populate the layout.  
    DetailsFragment detailsFragment =  
        (DetailsFragment)fm.findFragmentById(R.id.details_container);  
  
    if (detailsFragment == null) {  
        FragmentTransaction ft = fm.beginTransaction();  
        ft.add(R.id.details_container, new DetailsFragment());  
        ft.add(R.id.ui_container, new MyListFragment());  
        ft.commit();  
    }  
}
```

Fragments and the Back Stack

- Activity stack-ийн зарчимтай төстэй
- Зарим тохиолдолд хэрэглэгч буцах товчлуур дарсан үед өмнө биелэгдсэн Layout-ийг үзүүлэх шаардлага гардаг.
 - Fragment transaction-ний өмнө биелэгдсэнийг буцаах.
- addToBackStack функцын тусламжтай гүйцэтгэнэ.



Fragments and the Back Stack

- Буцах товчлуур дарахад өмнөх UI-руу
ШИЛЖИНЭ

```
FragmentManager.beginTransaction();  
fragmentTransaction.add(R.id.ui_container, new MyListFragment());  
  
FragmentManager.findFragmentById(R.id.details_fragment);  
fragmentTransaction.remove(fragment);  
  
String tag = null;  
fragmentTransaction.addToBackStack(tag);  
  
fragmentTransaction.commit();
```

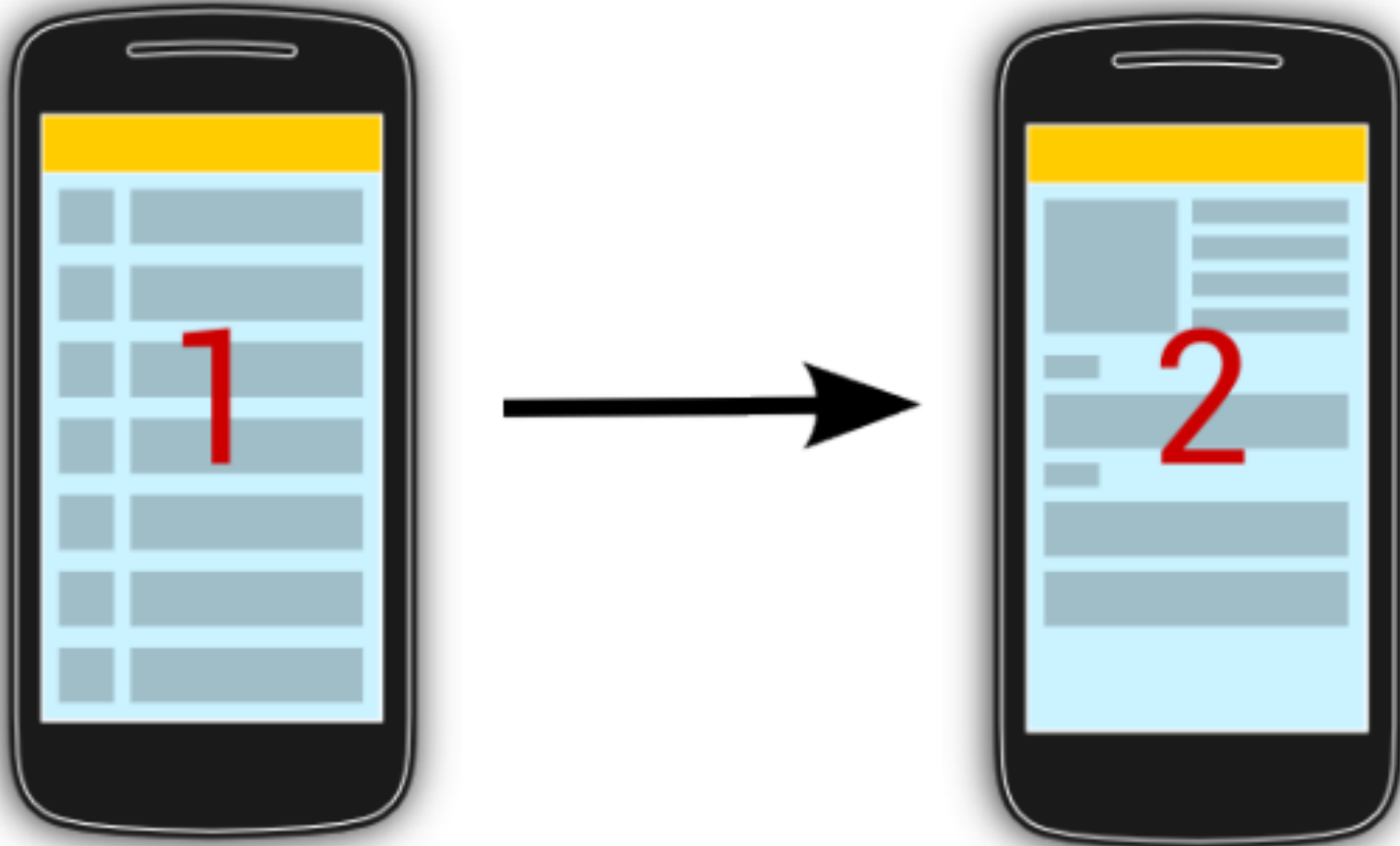

Interfacing between Fragments and Activity

- getActivity функцын тусламжтай fragment дотор Activity-тай харьцана.

```
TextView textView = (TextView) getActivity().findViewById(R.id.textview);
```

- Энэ нь Fragment нь өөрийн байрлаж буй Activity-ийн Fragment Manager-рүү хандах боломжтой

Advantages of using fragments



Advantages of using fragments

