



Fragments

План

- Fragment
- FragmentManager
- Support Fragment
- Вложенные фрагменты
- Lifecycle
- Передача данных



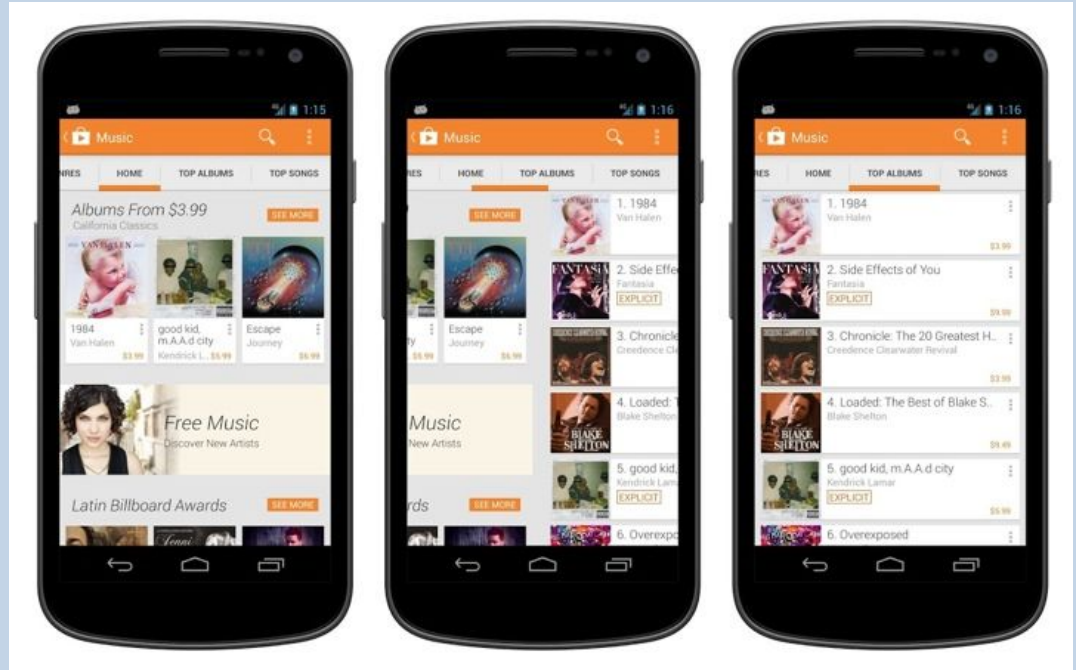
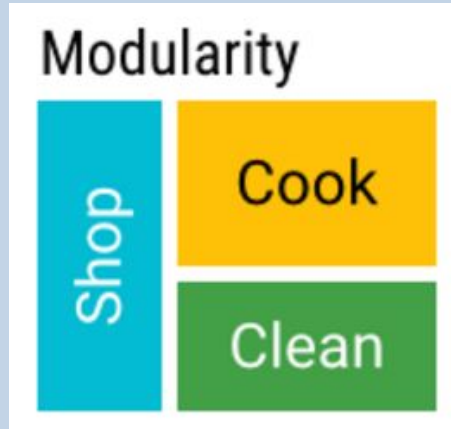
Fragment

- Fragment - сущность инкапсулирующая логику или UI
- Фрагмент можно создать в любой Activity
- Имеет собственный жизненный цикл привязанный к ЖЦ Activity
- Фрагменты могут быть вложенными друг в друга

<https://developer.android.com/guide/components/fragments.html>

Fragment

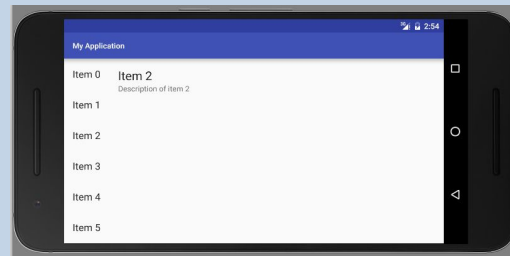
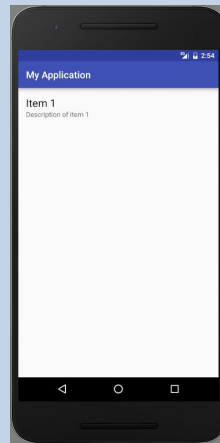
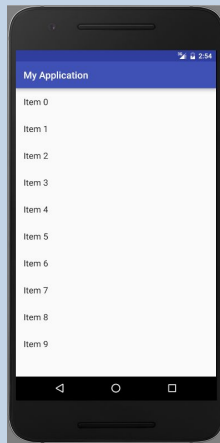
Фрагменты решают



Fragment

Фрагменты решают

Adaptability



Fragment

Фрагменты решают



Fragment

- Статические - добавляются в Activity через XML разметку

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <fragment
        android:id="@+id/my_fragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        tools:layout="@layout/fragment_my" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="That's my UI!" />

</FrameLayout>
```

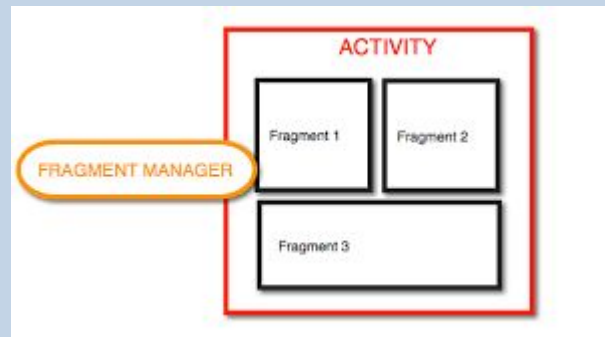
Fragment

- Динамические - добавляются в рантайме с помощью FragmentManager

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getFragmentManager()
        .beginTransaction()
        .add(new MyFragment(), "my")
        .commit();
}
```


Fragment Manager

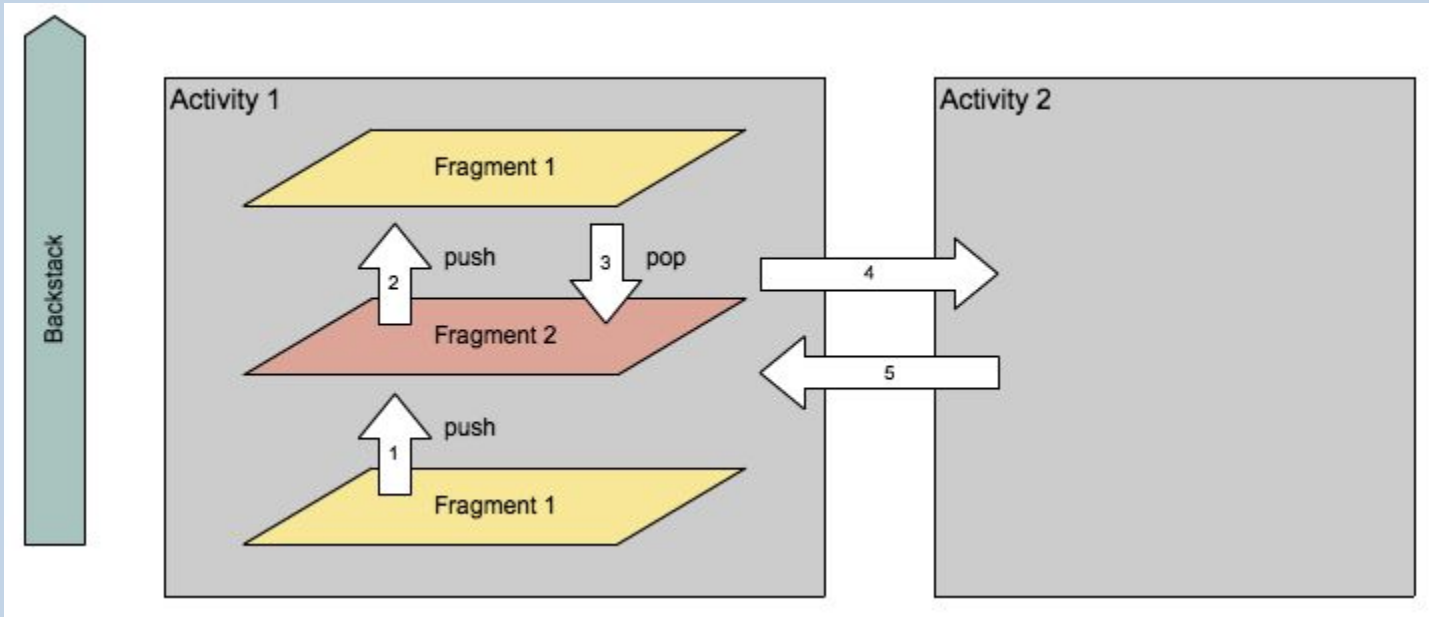
- Управляет фрагментами
 - Добавление
 - Удаление
 - Замена
 - Показ
 - Скрытие
 - Поиск
- Управляет backstack-ом
 - Добавление
 - Удаление
- Операции управления фрагментами выполняются внутри транзакций
- Транзакции выполняются асинхронно



<https://developer.android.com/reference/android/app/FragmentManager.html>

Fragments & Backstack

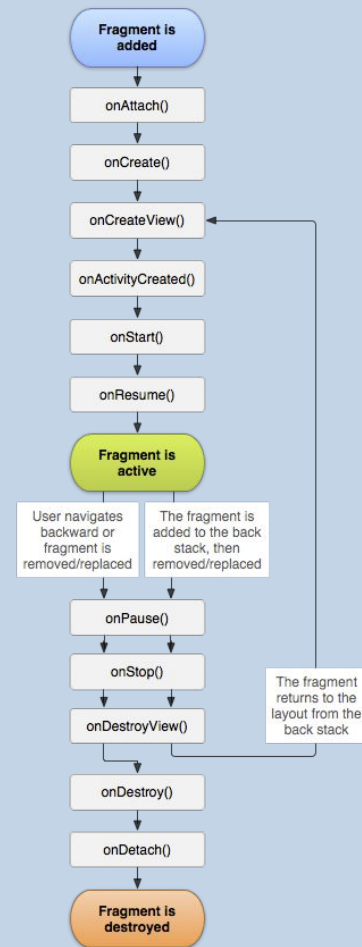
В отличие от Activity фрагменты требуют ручного управления BackStack-ом



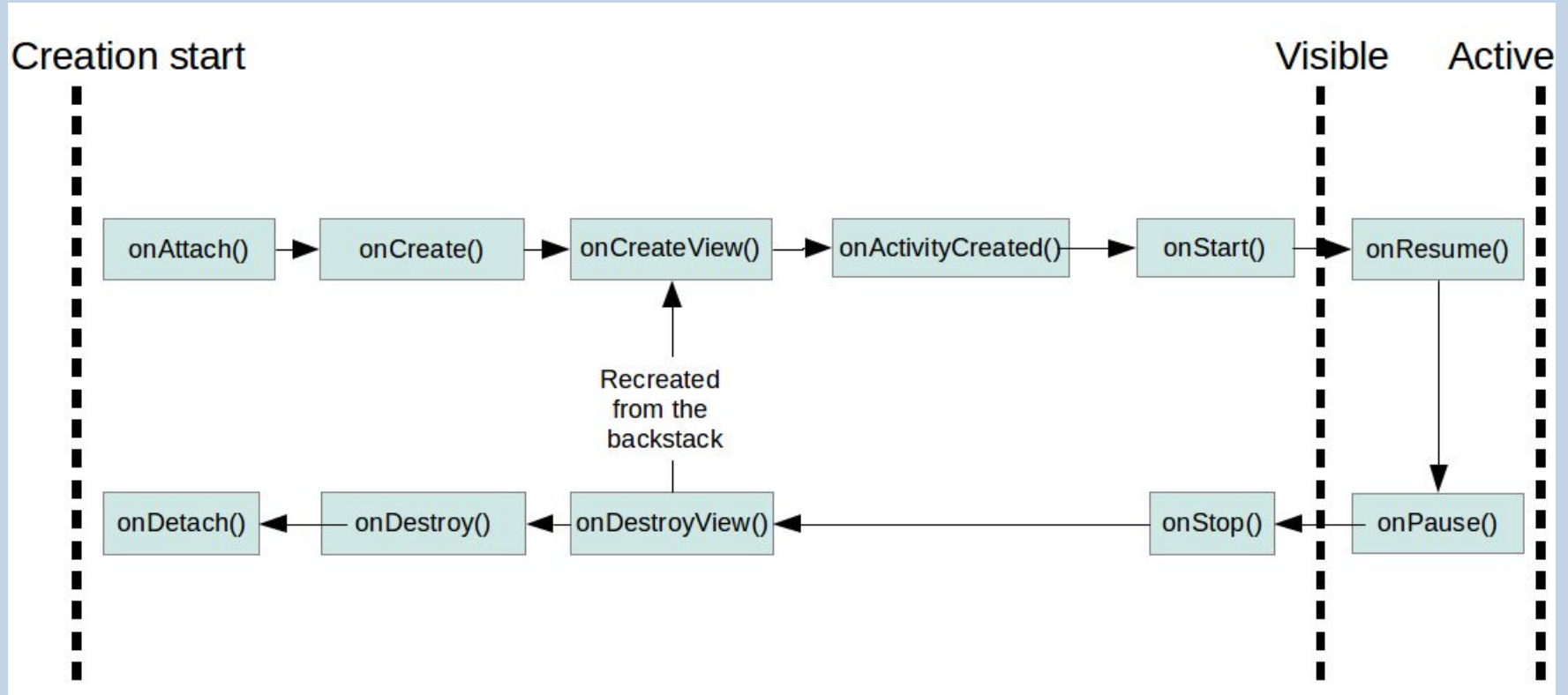
Fragment Lifecycle

Жизненный цикл фрагмента привязан к Activity и в целом очень похож, но

- Имеет несколько дополнительных колбеков
- Фрагменты могут создаваться и разрушаться без пересоздания/разрушения Activity



Fragment Lifecycle



Support Fragment

- Бэкпорт Fragment API для api level < 11 (Pre 3.0)
- Находится в support-v4
- Всегда содержит последние исправления и обновления
- Для того чтобы начать использовать
 - Подключить support package в build.gradle (`com.android.support:support-v4:*`)
 - Наследовать Activity от `android.support.v4.app.FragmentActivity`
 - Или от более нового `androidx.fragment.app.FragmentActivity`
 - Наследовать Fragment от `android.support.v4.app.Fragment`
 - Использовать FragmentManager из `FragmentManager#getSupportFragmentManager()`

<https://developer.android.com/topic/libraries/support-library/features.html>

Nested Fragments

- Фрагменты можно вкладывать друг в друга
- Доступно с Android 4.2 (API Level 17) или в support-v4
- `Fragment.getChildFragmentManager()`
- `Fragment.getParentFragment()`
- Может быть полезно, например, при работе с `ViewPager`



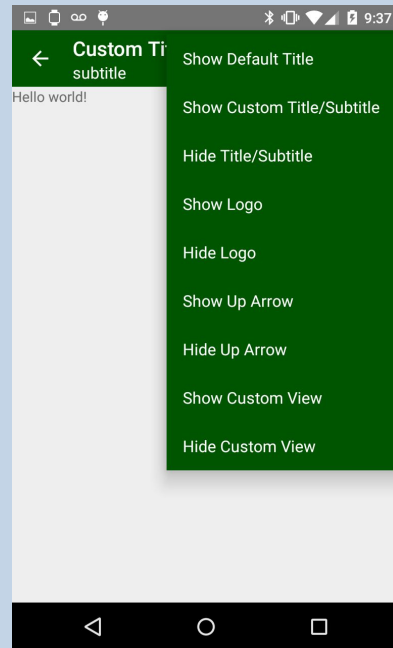
Fragment

- Фрагменты могут содержать собственные пункты меню

<code>void</code>	<code>setHasOptionsMenu(boolean hasMenu)</code> Report that this fragment would like to participate in populating the options menu by receiving a call to <code>onCreateOptionsMenu(Menu, MenuInflater)</code> and related methods.
-------------------	--

- Фрагменты могут запускать Activity и обрабатывать `onActivityResult`

<code>void</code>	<code>onActivityResult(int requestCode, int resultCode, Intent data)</code> Receive the result from a previous call to <code>startActivityForResult(Intent, int)</code> .
-------------------	--



Передача Данных

- Конструктор
- Сеттеры
- Bundle



Передача Данных

- Конструктор
- Сеттеры
- Bundle



Передача Данных

```
public class MyFragment extends Fragment {  
  
    public static MyFragment newInstance(String description, Integer amount) {  
        Bundle args = new Bundle();  
        args.putString("description", description);  
        args.putInt("amount", amount);  
  
        MyFragment fragment = new MyFragment();  
        fragment.setArguments(args);  
        return fragment;  
    }  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        String description = getArguments().getString("description");  
        Integer amount = getArguments().getInt("amount");  
        // do some stuff with arguments  
    }  
}
```

Возврат Данных

```
@Override
public void onCreateView(final View view, Bundle savedInstanceState) {
    view.findViewById(R.id.button).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            EditText arg1View = (EditText) view.findViewById(R.id.arg_1);
            EditText arg2View = (EditText) view.findViewById(R.id.arg_2);

            int arg1 = Integer.valueOf(arg1View.getText().toString());
            int arg2 = Integer.valueOf(arg2View.getText().toString());

            MyActivity activity = (MyActivity) getActivity();
            activity.myCallback(arg1 + arg2);
        }
    });
    super.onCreateView(view, savedInstanceState);
}
```

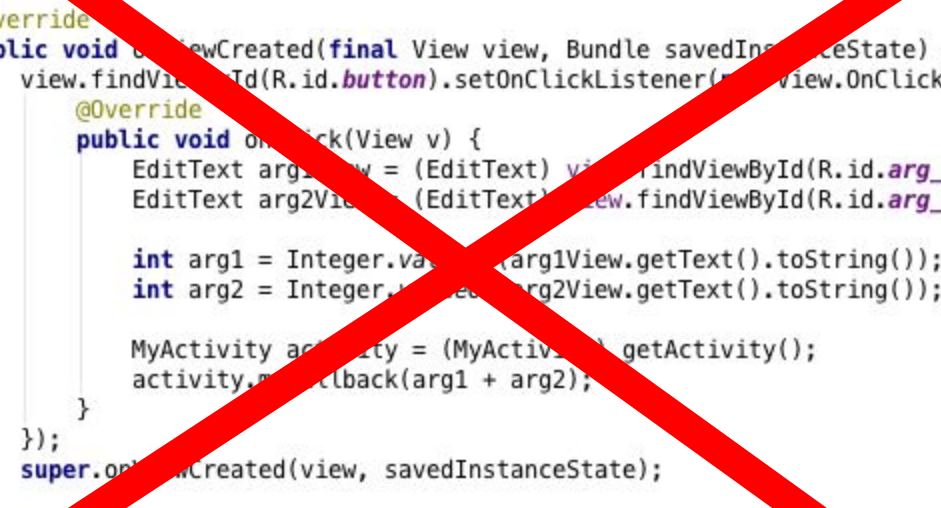
Возврат Данных

```
@Override
public void onCreateView(final View view, Bundle savedInstanceState) {
    view.findViewById(R.id.button).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            EditText arg1View = (EditText) view.findViewById(R.id.arg_1);
            EditText arg2View = (EditText) view.findViewById(R.id.arg_2);

            int arg1 = Integer.valueOf(arg1View.getText().toString());
            int arg2 = Integer.valueOf(arg2View.getText().toString());

            MyActivity activity = (MyActivity) getActivity();
            activity.myCallback(arg1 + arg2);
        }
    });
    super.onCreateView(view, savedInstanceState);
}
```

Возврат Данных



```
@Override
public void onCreate(final View view, Bundle savedInstanceState) {
    view.findViewById(R.id.button).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            EditText arg1View = (EditText) view.findViewById(R.id.arg_1);
            EditText arg2View = (EditText) view.findViewById(R.id.arg_2);

            int arg1 = Integer.parseInt(arg1View.getText().toString());
            int arg2 = Integer.parseInt(arg2View.getText().toString());

            MyActivity activity = (MyActivity) getActivity();
            activity.setResult(RESULT_OK, back(arg1 + arg2));
        }
    });
    super.onCreate(view, savedInstanceState);
}
```

Возврат Данных

```
public interface MyFragmentListener {  
    void myCallback(int result);  
}
```

Возврат Данных

```
public interface MyFragmentListener {  
    void myCallback(int result);  
}
```

```
public class MyActivity extends Activity implements MyFragment.MyFragmentListener {  
  
    @Override  
    public void myCallback(int result) {  
        // process result here...  
    }  
}
```

Возврат Данных

```
public interface MyFragmentListener {  
    void myCallback(int result);  
}
```

```
public class MyActivity extends Activity implements MyFragment.MyFragmentListener {  
  
    @Override  
    public void myCallback(int result) {  
        // process result here...  
    }  
}
```

```
public class MyFragment extends Fragment {  
  
    private MyFragmentListener listener;
```


Возврат Данных

```
public interface MyFragmentListener {  
    void myCallback(int result);  
}
```

```
public class MyActivity extends Activity implements MyFragment.MyFragmentListener {  
  
    @Override  
    public void myCallback(int result) {  
        // process result here...  
    }  
}
```

```
public class MyFragment extends Fragment {  
  
    private MyFragmentListener listener;
```

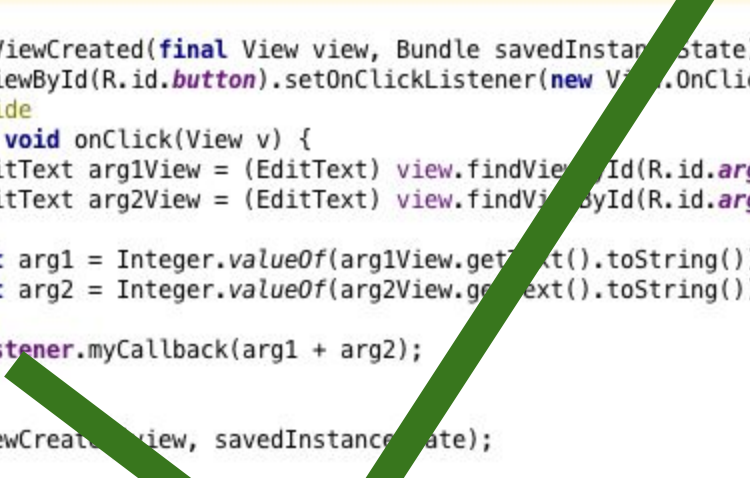
```
    @Override  
    public void onAttach(Context context) {  
        super.onAttach(context);  
        if (getActivity() instanceof MyFragmentListener) {  
            listener = (MyFragmentListener) getActivity();  
        } else {  
            throw new RuntimeException("Activity should implement MyFragmentListener");  
        }  
    }  
}
```

Возврат Данных

```
@Override
public void onCreateView(final View view, Bundle savedInstanceState) {
    view.findViewById(R.id.button).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            EditText arg1View = (EditText) view.findViewById(R.id.arg_1);
            EditText arg2View = (EditText) view.findViewById(R.id.arg_2);

            int arg1 = Integer.valueOf(arg1View.getText().toString());
            int arg2 = Integer.valueOf(arg2View.getText().toString());

            listener.myCallback(arg1 + arg2);
        }
    });
    super.onCreateView(view, savedInstanceState);
}
```

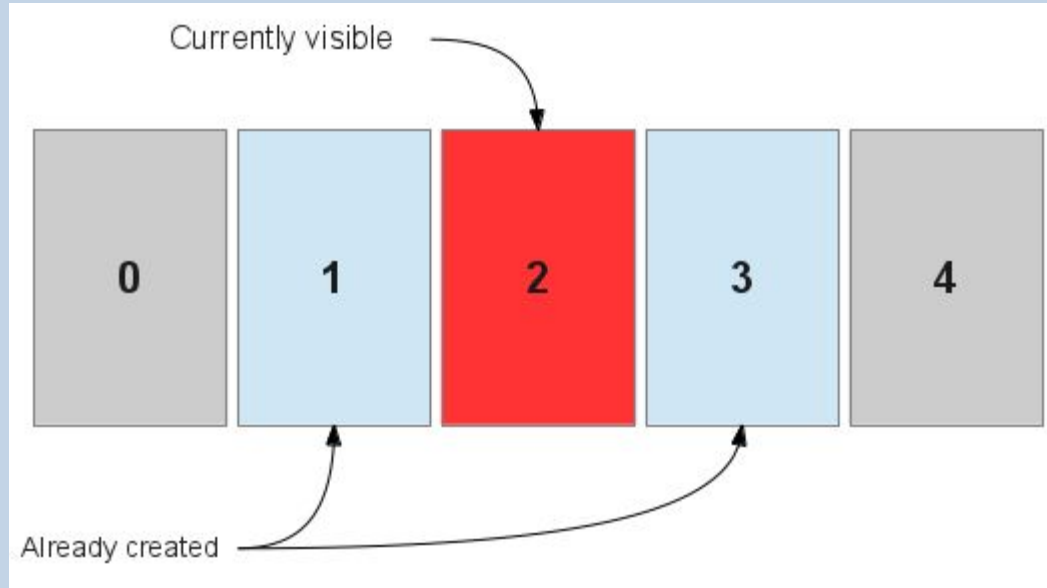


Сохранение Состояния

- Используя механизм `onSaveInstanceState`
 - Автоматически сохраняется состояние view для которых задан `android:id`
 - View должны реализовывать `onSaveInstanceState` / `onRestoreInstanceState`
 - У фрагмента нет коллбека `onRestoreInstanceState`
 - Вместо `onRestoreInstanceState` используется `onCreate`, `onActivityCreated` и др.
- Используя `FragmentManager`
 - `putFragment`
 - `getFragment`
- Используя `Fragment#setRetainInstance`
 - Из Lifecycle “убираются” `onCreate` / `onDestroy`
 - и вызываются по одному разу
 - `onAttach` / `onDetach` остаются

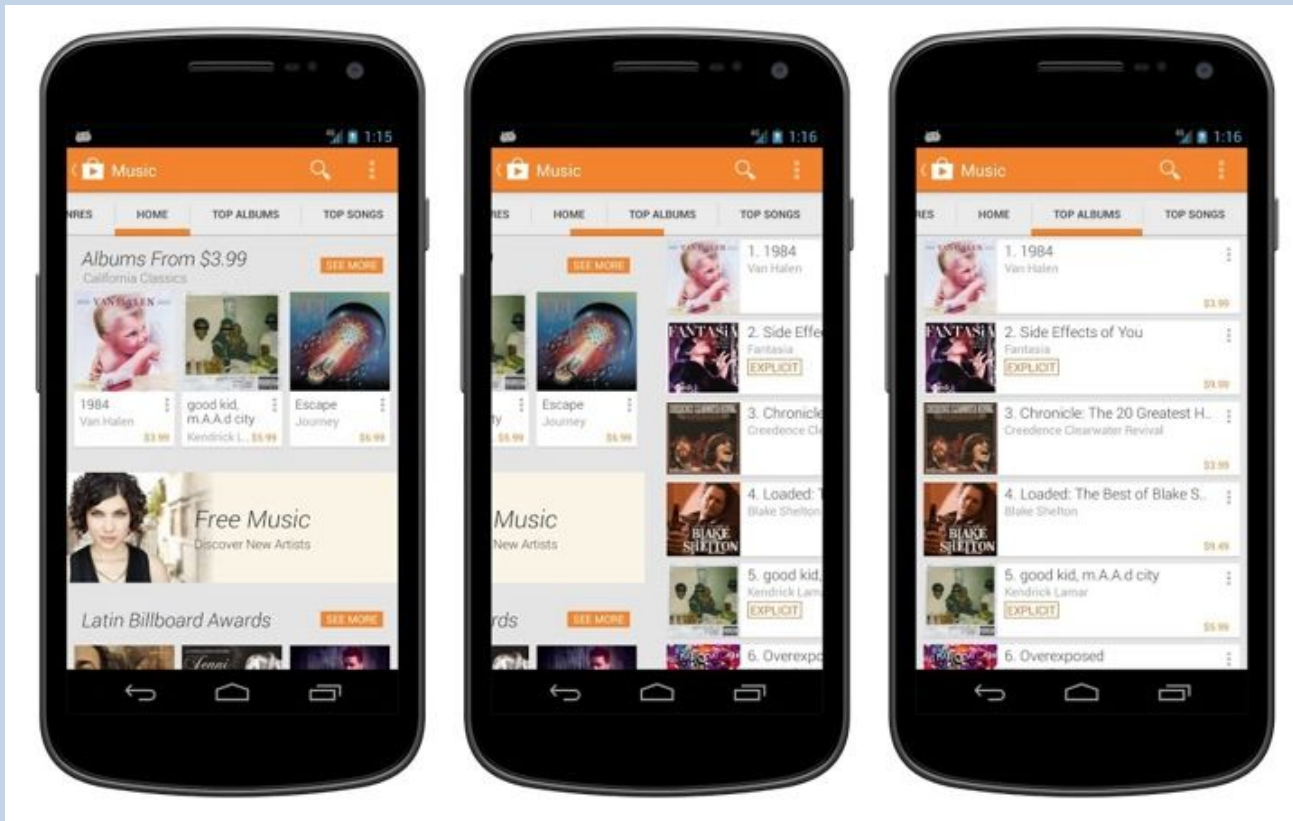


ViewPager



<https://developer.android.com/training/animation/screen-slide.html>

ViewPager



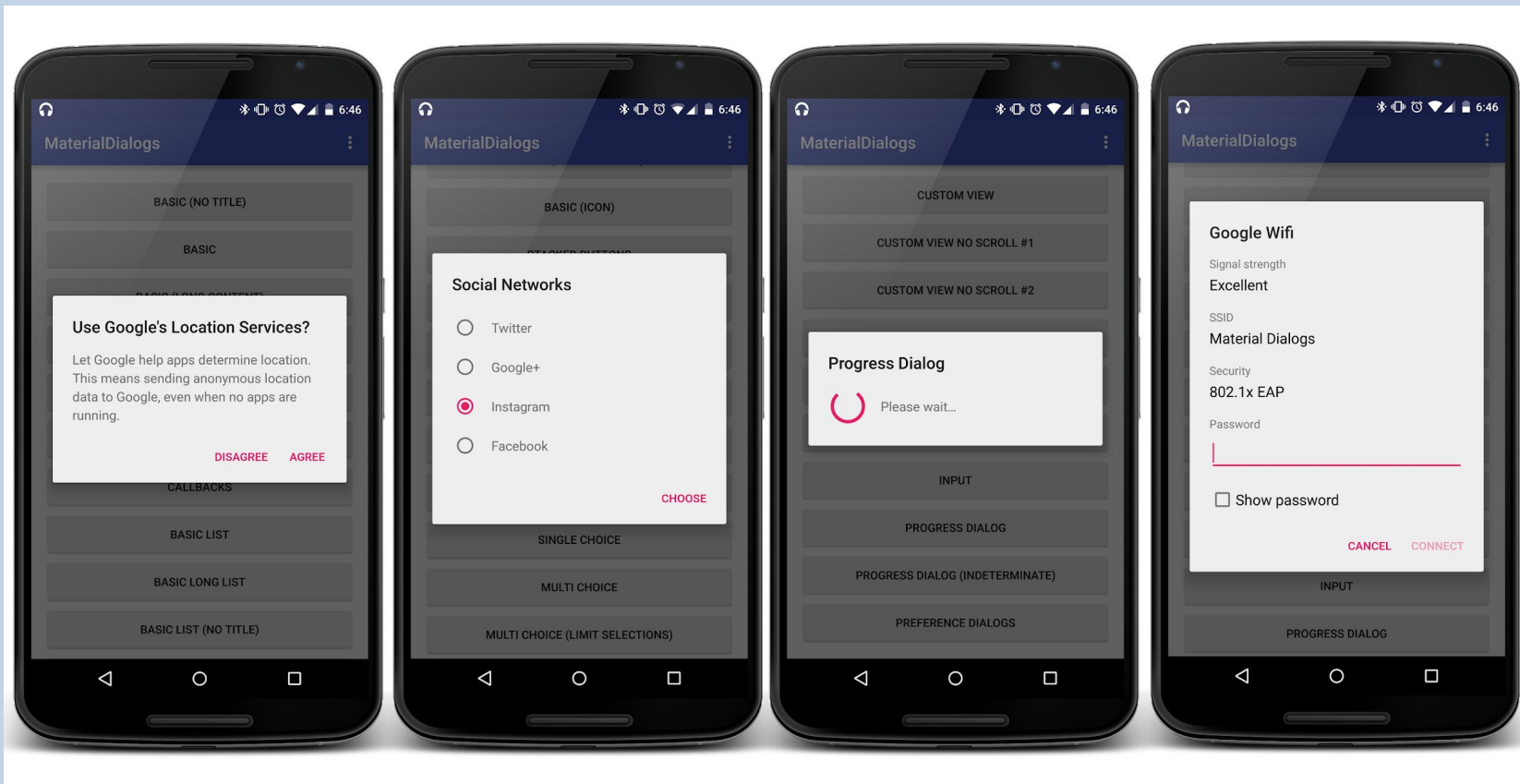
ViewPager



ViewPager

- `android.support.v4.view.ViewPager`
- `android.support.v4.view.PagerAdapter`
 - `android.support.v4.app.FragmentPagerAdapter`
 - Небольшое количество страниц
 - Отображение статичного контента
 - Держит все фрагменты в памяти
 - `android.support.v4.app.FragmentStatePagerAdapter`
 - Много страниц
 - Динамичный контент
 - Держит в памяти несколько соседних фрагментов (по умолчанию +1)

DialogFragment



DialogFragment

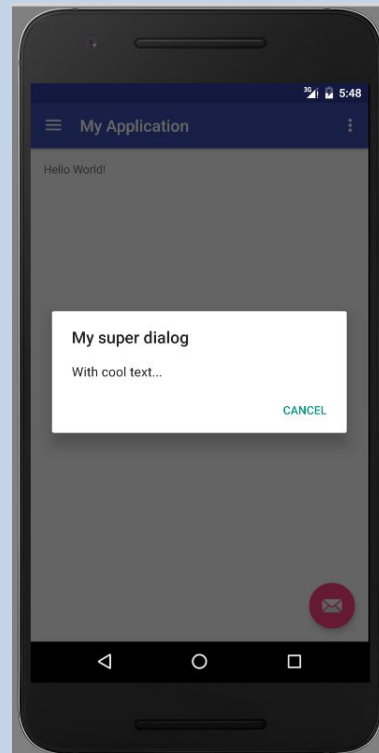
```
package ru.tinkoff.school.myapplication;

import android.app.Dialog;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.DialogFragment;
import android.support.v4.app.FragmentManager;
import android.support.v7.app.AlertDialog;

/**
 * @author Dmitriy Tarasov
 */
public class MyDialogFragment extends DialogFragment {

    public static MyDialogFragment show(FragmentManager fm) {
        MyDialogFragment dialog = new MyDialogFragment();
        dialog.show(fm, null);
        return dialog;
    }

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        return new AlertDialog.Builder(getContext())
            .setTitle("My super dialog")
            .setMessage("With cool text...")
            .setNegativeButton("Cancel", null)
            .create();
    }
}
```

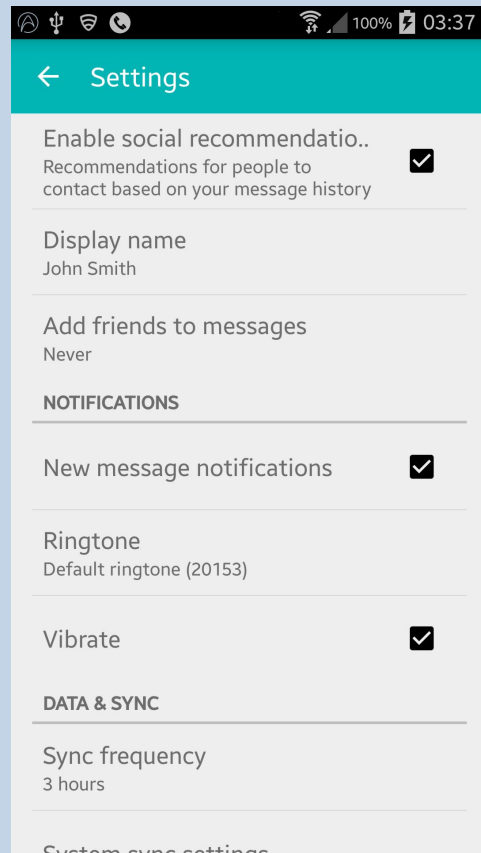


<https://developer.android.com/reference/android/support/v4/app/DialogFragment.html>

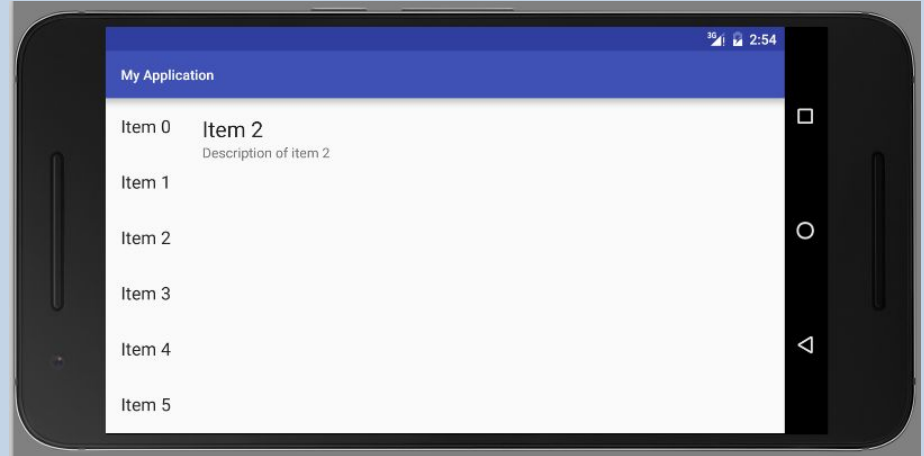
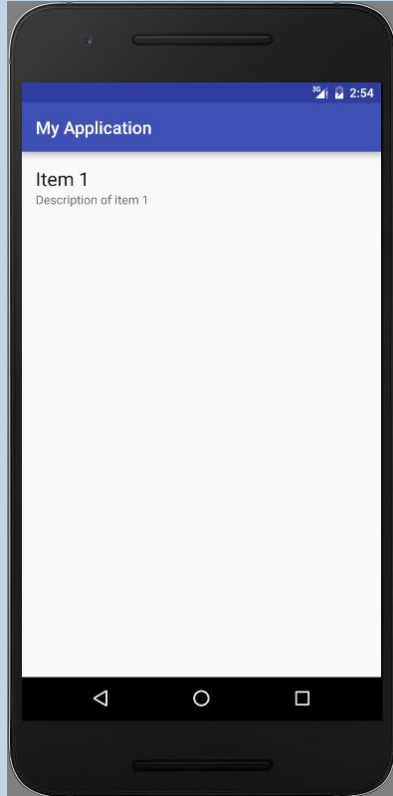
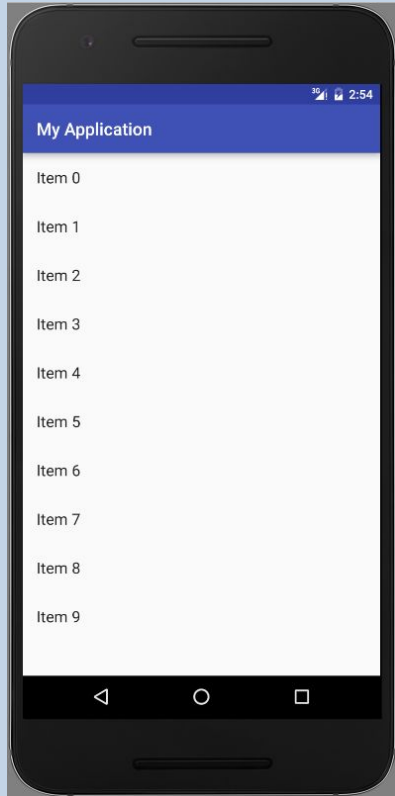
PreferenceFragment

- PreferenceFragment - отображает настройки в виде списка
- Оформление повторяет системное оформление экрана настроек
- Оформление можно “корректировать” через темы
- При взаимодействии с пользователем настройки автоматически сохраняются в SharedPreferences
- Список настроек описывается в XML файле (res/xml) вместо layout

<https://developer.android.com/reference/android/preference/PreferenceFragment.html>



Fragments



Fragments

fragment_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.RecyclerView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/recycler"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" />
```

fragment_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0"
        android:textAppearance="@android:style/TextAppearance.Large"
        tools:text="Title" />

    <TextView
        android:id="@+id/description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        tools:text="Lorem ipsum .... " />

</LinearLayout>
```

Fragments

layout/activity_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <fragment
        android:id="@+id/list"
        class="ru.tinkoff.school.myapplication.fragments.ListFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout="@layout/fragment_list" />

</LinearLayout>
```

layout-land/activity_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <fragment
        android:id="@+id/list"
        class="ru.tinkoff.school.myapplication.fragments.ListFragment"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        tools:layout="@layout/fragment_list" />

    <fragment
        android:id="@+id/details"
        class="ru.tinkoff.school.myapplication.fragments.DetailsFragment"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        tools:layout="@layout/fragment_details" />

</LinearLayout>
```

Fragments

ListActivity.java

```
public class ListActivity extends AppCompatActivity implements ListFragment.ListFragmentListener {  
  
    private boolean twoPaneMode;  
  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_list);  
  
        if (findViewById(R.id.title) != null) {  
            twoPaneMode = true;  
        }  
    }  
  
    @Override  
    public void onClick(String title, String description) {  
        if (twoPaneMode) {  
            DetailsFragment details = (DetailsFragment) getSupportFragmentManager().findFragmentById(R.id.details);  
            details.setValuesToFragment(title, description);  
        } else {  
            DetailsActivity.start(this, title, description);  
        }  
    }  
}
```

Fragments

DetailsActivity.java

```
public class DetailsActivity extends AppCompatActivity {

    public static void start(Context context, String title, String description) {
        Intent i = new Intent(context, DetailsActivity.class);
        i.putExtra("title", title);
        i.putExtra("description", description);
        context.startActivity(i);
    }

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        String title = getIntent().getExtras().getString("title");
        String description = getIntent().getExtras().getString("description");

        DetailsFragment fragment = DetailsFragment.newInstance(title, description);
        getSupportFragmentManager().beginTransaction().add(android.R.id.content, fragment).commit();
    }
}
```

Fragment vs View

- В отличие от View - Fragment имеет привязку к жизненному циклу Activity
- View не может пережить переворот экрана
- View не имеет backstack
- И тд.

To be, or not to be...

Когда фрагменты только появились было множество попыток построить архитектуру целого приложения на основе всего одной Activity и множества фрагментов, которые динамически подменяются в этой Activity.

Такой подход порождает множество проблем при навигации. Backstack-ом активностей управляет система, backstack-ом фрагментов - программист.

Есть библиотеки, которые решают эту задачу, но зачем?

Подробнее почему фрагменты “зло”:

<https://corner.squareup.com/2014/10/advocating-against-android-fragments.html>

Ссылки

- <https://www.raywenderlich.com/117838/introduction-to-android-fragments-tutorial>
- <http://www.vogella.com/tutorials/AndroidFragments/article.html>
- <https://inthecheesefactory.com/blog/fragment-state-saving-best-practices/en>



Спасибо за внимание

Домашняя работа

По нажатию на кнопку добавить документ, я должен добавить на экран документ с текстом “Документ №”, где номер увеличивается в зависимости от кол-ва фрагментов.

По нажатию на удалить документ, я убираю верхний документ и вижу нижний либо “Документов нет”, если мы удалили фрагмент под номером 1.

Фрагменты должны добавляться в бэкстек.

При перевороте экрана, я должен видеть то же состояние.

