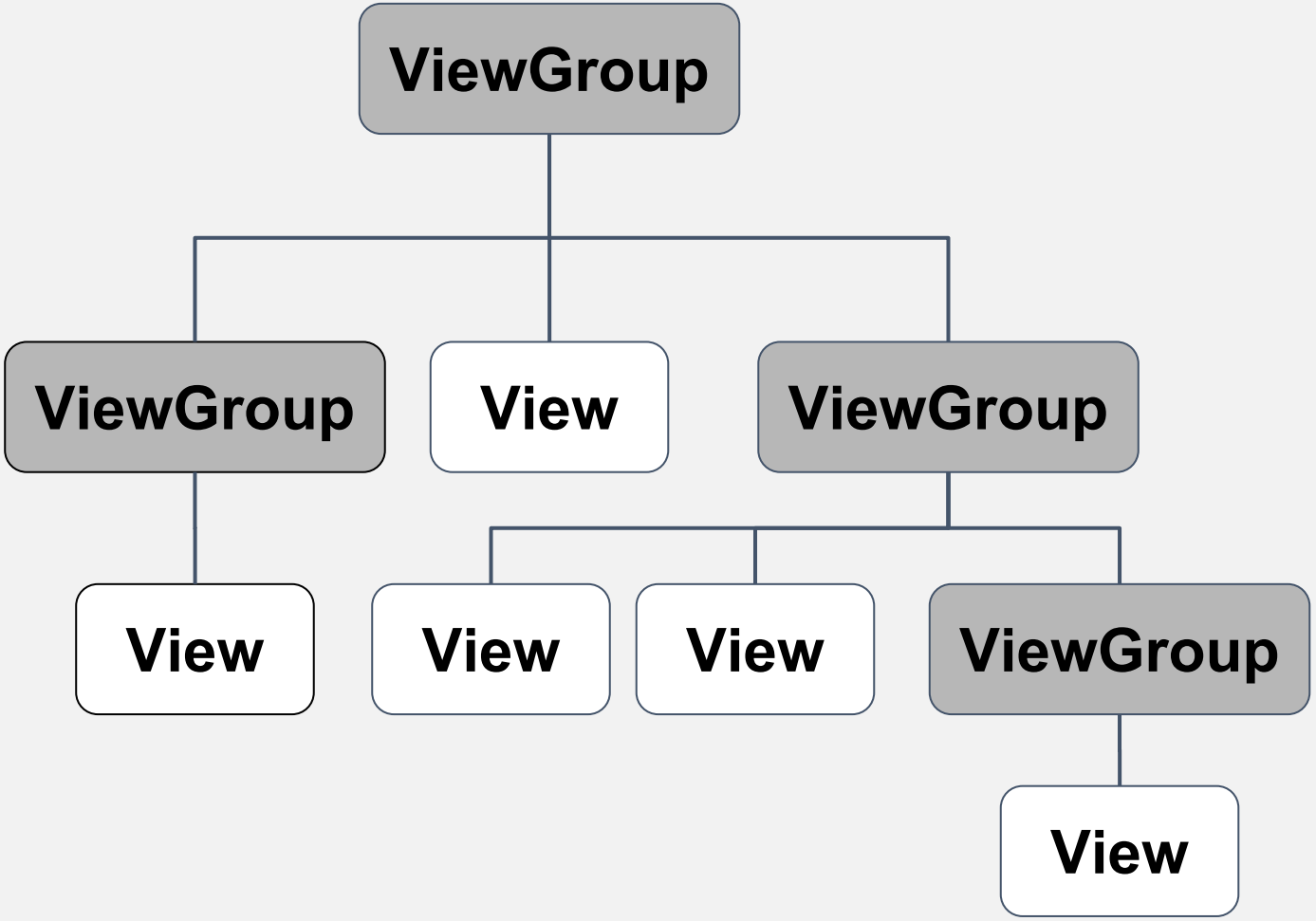
Fragments и UI

Лекция №2

Попов Даниил

Вспомним об иерархии View



Атрибуты View

Ширина и высота	`match_parent`, `wrap_content`, абсолютная величина
Идентификатор	`@+id/name` или `@id/name`
Padding и Margin	абсолютная величина
Видимость элемента	`visible`, `invisible`, `gone`
Выравнивание содержимого	`left`, `right`, `top`, `bottom`, `center`
Фоновый рисунок	цвет или картинка

Пример View

```
<TextView
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:id="@+id/text"
  android:padding="10dp"
  android:layout_marginLeft="15dp"
  android:visibility="visible"
  android:gravity="center"
  android:background="#FF0000"
  android:text="Hello World!" />
```

Какие бывают ViewGroup

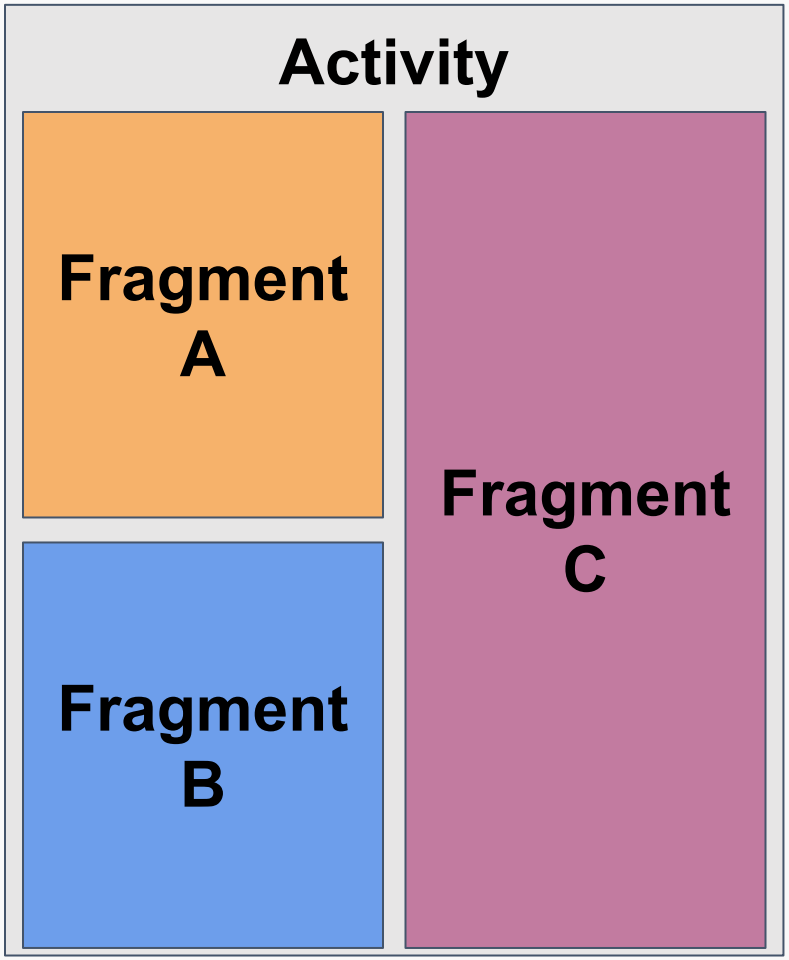
- **LinearLayout** - дочерние элементы располагаются последовательно друг за другом либо вертикально, либо горизонтально

- **RelativeLayout** - дочерние элементы позиционируются относительно друг друга и относительно границ самого *RelativeLayout*.

- **FrameLayout** - дочерние элементы располагаются слоями друг над другом и позиционируются относительно верхнего левого угла

Что такое Fragment?

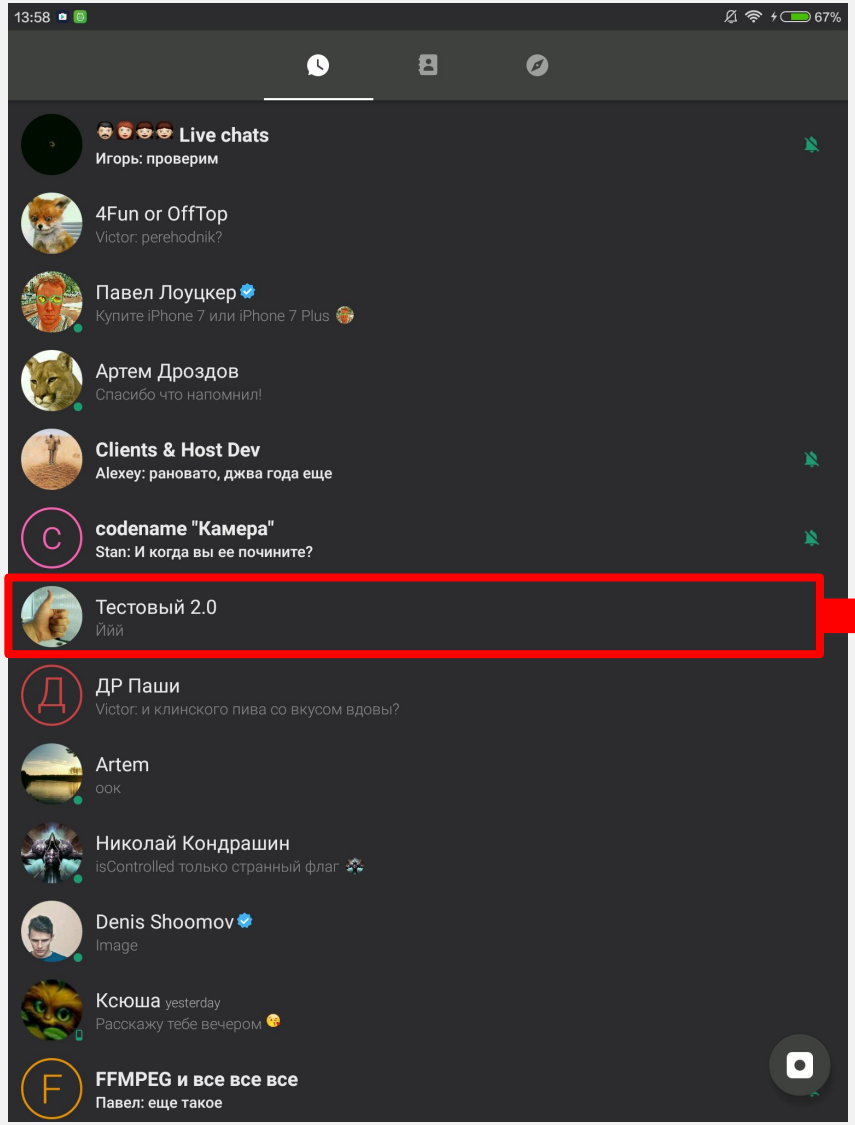
1. ***Fragment*** - “кусочек” *Activity*;
2. Имеет свой жизненный цикл, связанный с *Activity*;
3. Взаимодействует с пользователем;
4. Могут добавлять на *Activity* в *runtime*;
5. Появились в *Api Level 11*.

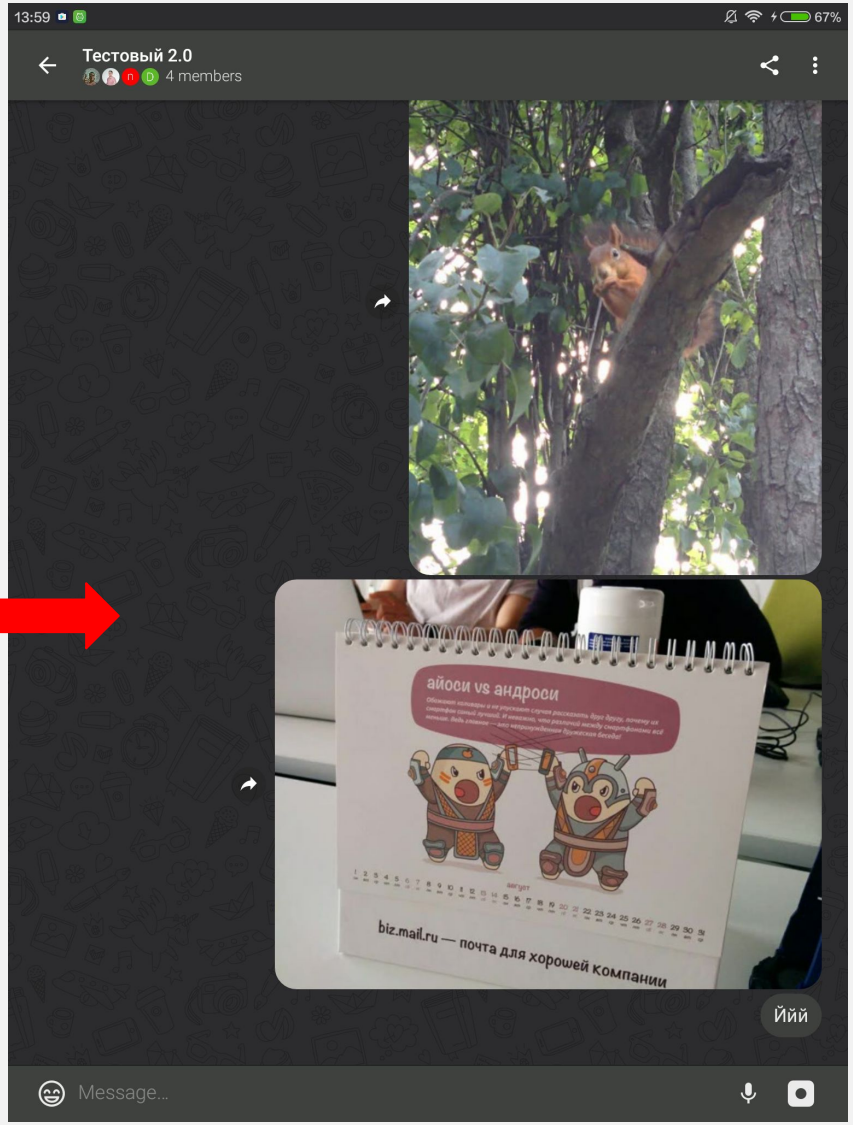


Философия Fragment’ов

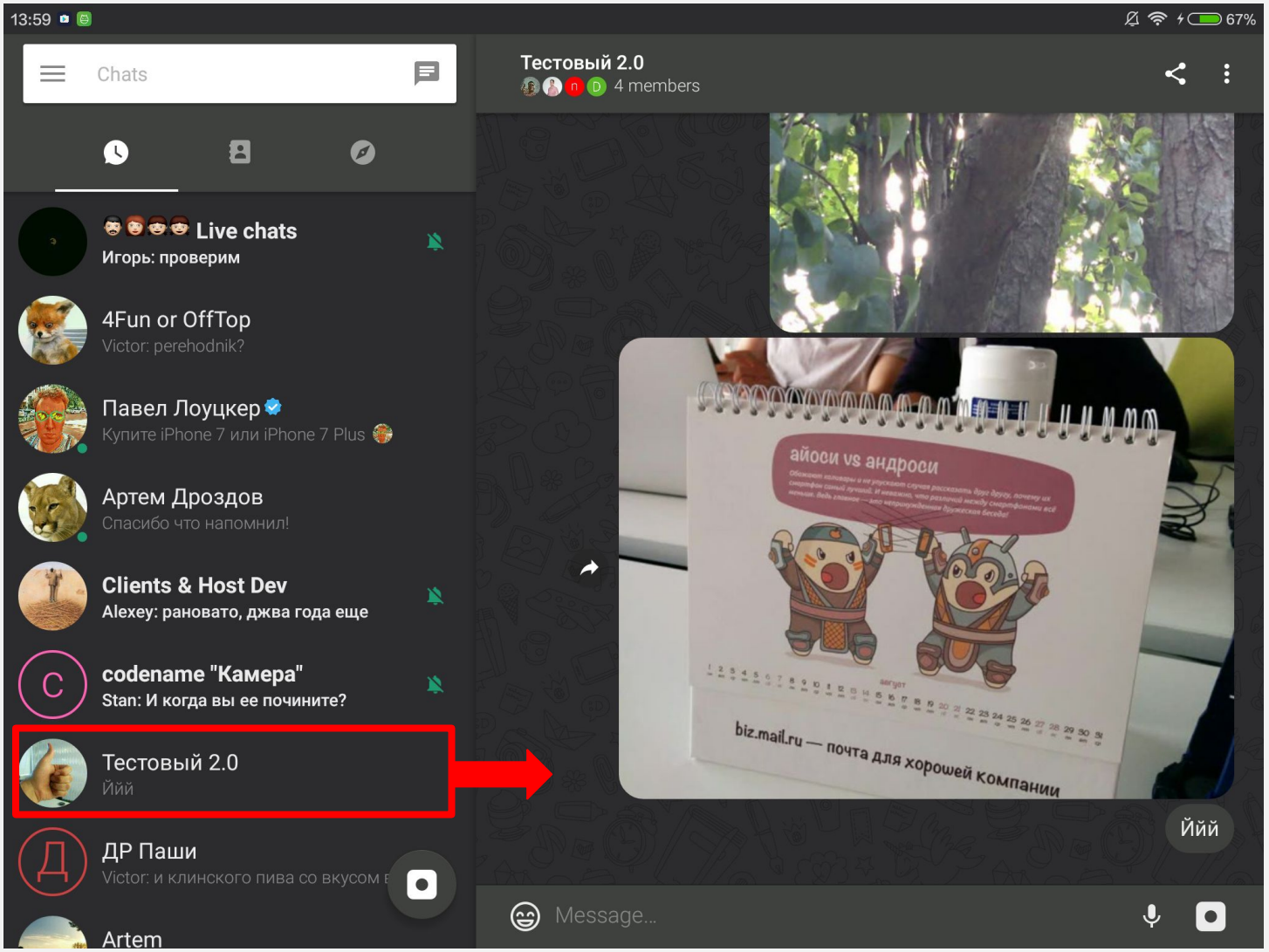
1. Разработка “гибкого” пользовательского интерфейса, реагирующего на действия пользователя;
2. Рациональное использование пространства экрана на планшетах, а также в альбомной ориентации устройства;
3. Переиспользование кода и верстки.

Пример использования

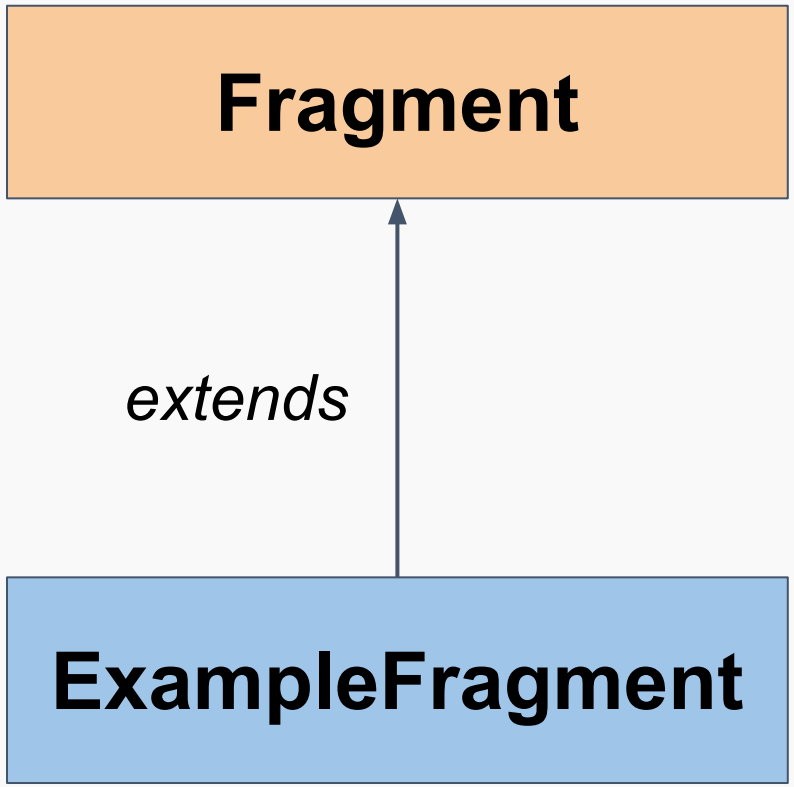




… в альбомной ориентации



Создание Fragment'а



- **DialogFragment** - диалоговое "окно" поверх *Activity*, не закрывающее весь экран
- **ListFragment** - фрагмент для отображения скролящихся списков

Жизненный цикл Fragment’a

Activity State	Fragment Lifecycle Callback
Created	• onAttach() • onCreate() • onCreateView() • onActivityCreated()
Started	• onStart()
Resumed	• onResume()

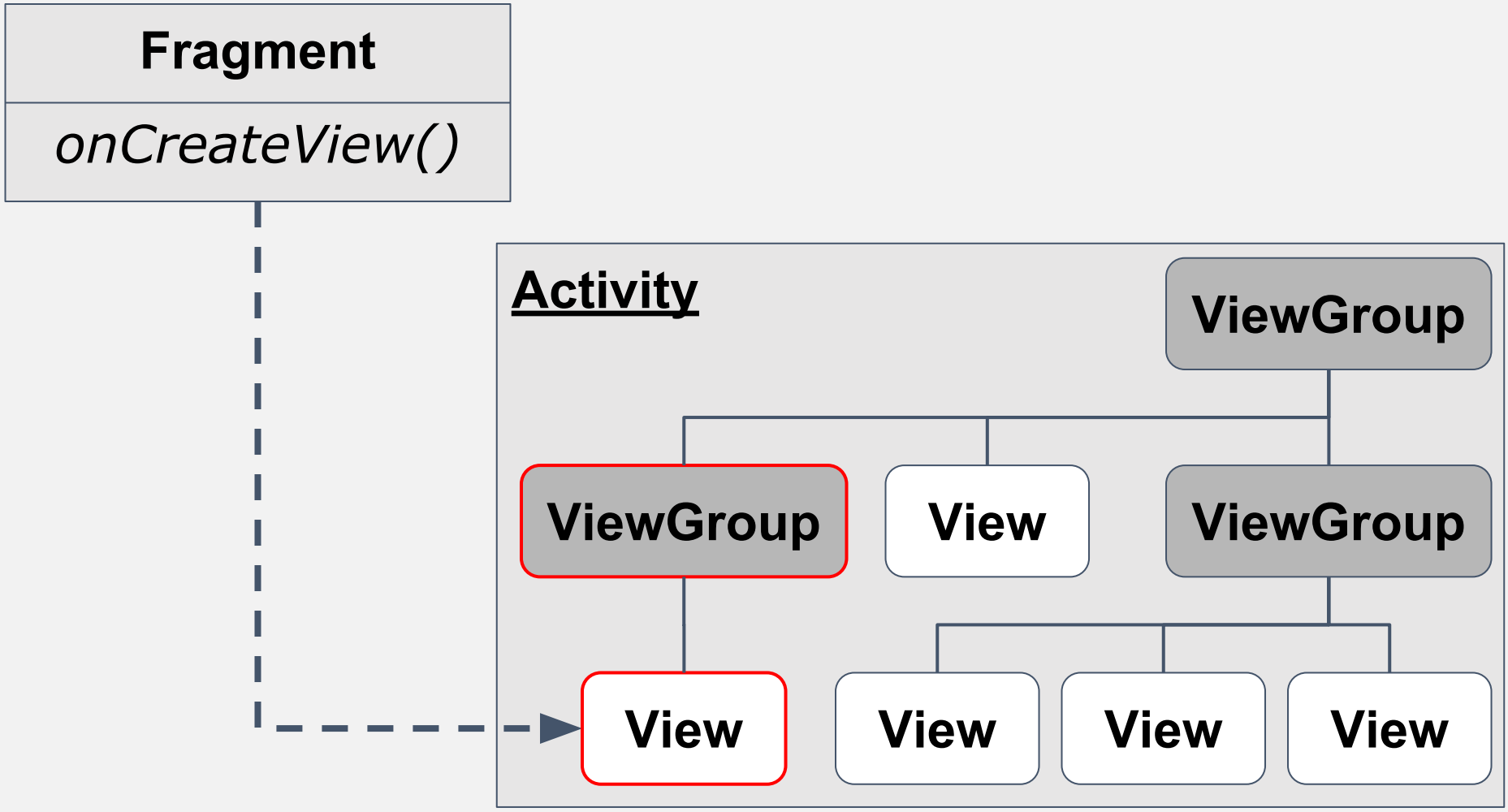
Жизненный цикл Fragment’a

Activity State	Fragment Lifecycle Callback
Paused	• onPause()
Stopped	• onStop()
Destroyed	• onDestroyView() • onDestroy() • onDetach()

Пользовательский интерфейс Fragment'а


```
public static class ExampleFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater,
                             ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.example_fragment,
                                container,
                                false);
    }
}
```

Как это работает?



Добавление Fragment'а в Activity

Существует два способа:

1. Через xml-разметку с помощью тега `<fragment>`
2. Программно, используя `FragmentManager`

Через xml-разметку


```
<?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.example.news.ArticleListFragment"
            android:id="@+id/list"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="match_parent" />
      <fragment android:name="com.example.news.ArticleReaderFragment"
            android:id="@+id/viewer"
            android:layout_weight="2"
            android:layout_width="0dp"
            android:layout_height="match_parent" />
</LinearLayout>
```

Программно

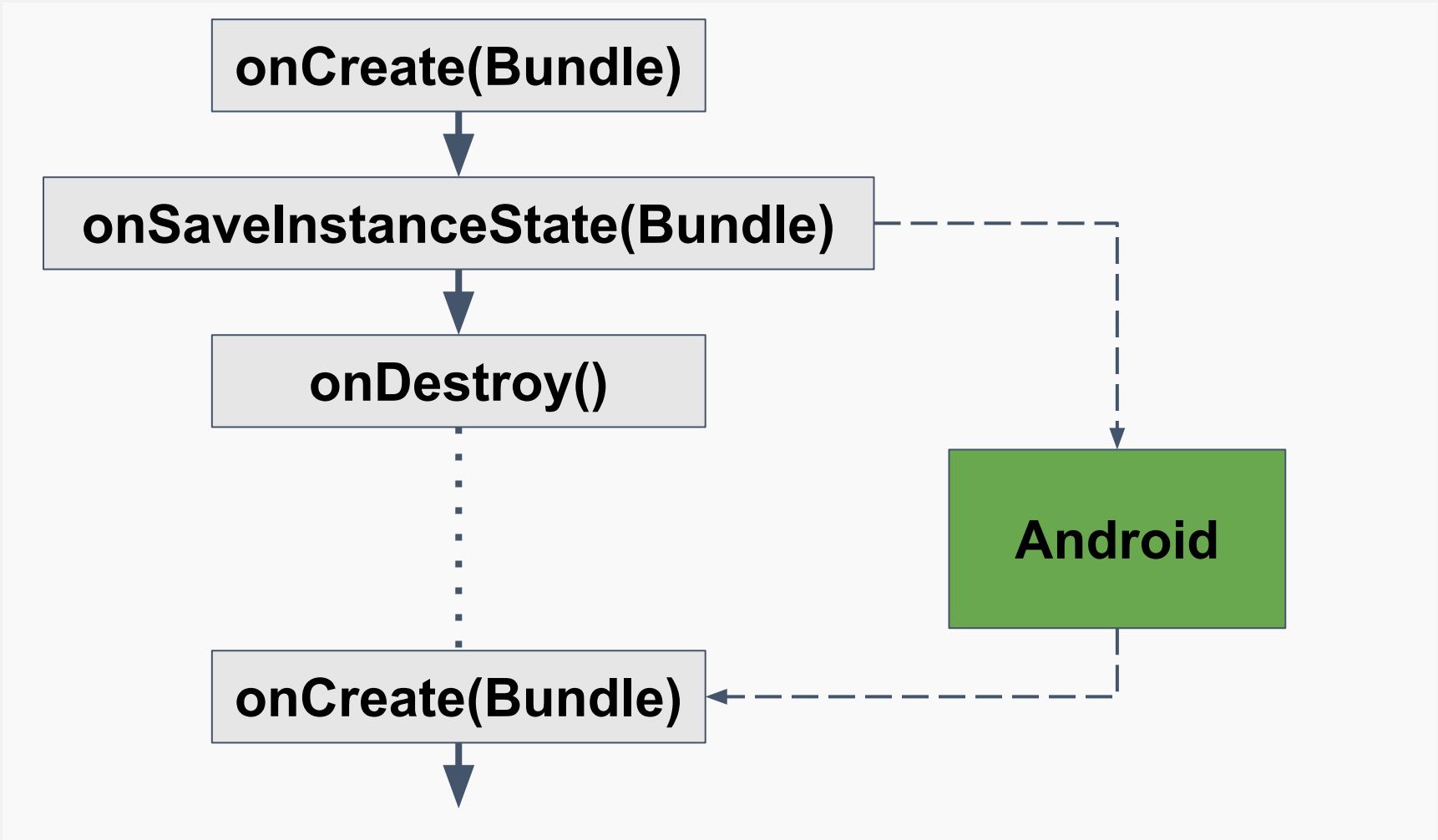

```
FragmentManager fragmentManager = getFragmentManager();
FragmentTransaction fragmentTransaction =
                                fragmentManager.beginTransaction();
ExampleFragment fragment = new ExampleFragment();
fragmentTransaction.add(R.id.fragment_container, fragment);
fragmentTransaction.commit();
```

FragmentTransaction

Любые действия с фрагментами обязательно выполняются в рамках транзакции. Доступны следующие операции:

- Добавление фрагмента;
- Удаление фрагмента;
- Замена фрагмента;
- Добавление транзакции в *back stack*;
- Установка анимации для транзации.

Немного о сохранении состояния



Завершение транзакции

Существует два способа:

- `commit()` - завершение транзакции с учетом потери состояния (бросает исключение)
- `commitAllowingStateLoss()` - завершение транзакции с игнорированием потери состояния

Взаимодействие Activity и Fragment


```
public class ExampleFragment extends Fragment {
    protected OnActionListener callback;

    public interface OnActionListener {
        void onActionDone(String action);
    }

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        if (activity instanceof OnActionListener) {
            callback = (OnActionListener) activity;
        }
    }
}
```

Спасибо за внимание!

А теперь к практике!