

[RxJava \(http://developer.alexanderklimov.ru/android/rx/\)](http://developer.alexanderklimov.ru/android/rx/)

[Советы \(http://developer.alexanderklimov.ru/android/tips-android.php\)](http://developer.alexanderklimov.ru/android/tips-android.php)

[Статьи \(http://developer.alexanderklimov.ru/android/articles-android.php\)](http://developer.alexanderklimov.ru/android/articles-android.php)

[Книги \(http://developer.alexanderklimov.ru/android/books.php\)](http://developer.alexanderklimov.ru/android/books.php)

[Java \(http://developer.alexanderklimov.ru/android/java/java.php\)](http://developer.alexanderklimov.ru/android/java/java.php)

[Kotlin \(http://developer.alexanderklimov.ru/android/kotlin/\)](http://developer.alexanderklimov.ru/android/kotlin/)

[Дизайн \(http://developer.alexanderklimov.ru/android/design/\)](http://developer.alexanderklimov.ru/android/design/)

[Отладка \(http://developer.alexanderklimov.ru/android/debug/\)](http://developer.alexanderklimov.ru/android/debug/)

[Open Source \(http://developer.alexanderklimov.ru/android/opensource.php\)](http://developer.alexanderklimov.ru/android/opensource.php)

[Полезные ресурсы \(http://developer.alexanderklimov.ru/android/links.php\)](http://developer.alexanderklimov.ru/android/links.php)

Fragment (Фрагменты).

Часть третья



А теперь представим, что котов много (котов много не бывает), и нам придётся добавлять новые кнопки на экран к трём имеющимся. И тогда совсем не останется места картинке и сопроводительному тексту в портретной ориентации, которые уползут вниз и будут не видны. Для решения этой задачи можно кнопки оставить на первой активности, а картинку с текстом перенести на вторую активность. Для альбомной ориентации можно оставить как было.

Создадим вторую активность **SecondActivity** через мастер создания активностей. Скопируем в его разметку вторую нижнюю часть из разметки от первой активности (**activity_second.xml**):

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <fragment
        android:id="@+id/fragment2"
        android:name="ru.alexanderklimov.fragmentdemo.Fragment2"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        tools:layout="@layout/fragment2" />

</LinearLayout>
```

А в файле **activity_main.xml**, наоборот, удалим второй фрагмент, чтобы остался только первый фрагмент с кнопками.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <fragment
        android:id="@+id/fragment1"
        android:name="ru.alexanderklimov.fragmentdemo.Fragment1"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        tools:layout="@layout/fragment1" />

</LinearLayout>
```

Если запустить приложение, то столкнёмся с такой ситуацией. У нас запустится активность с кнопками. Повернём устройство в альбомную ориентацию и увидим знакомый интерфейс, когда слева располагаются кнопки, а справа - картинка и текст. Мы знаем, что на экране сейчас два фрагмента и нажатия на кнопки по-прежнему работают.

Повернём устройство обратно в портретную ориентацию и попробуем нажать на любую кнопку. Программа аварийно закроется. Что произошло? А произошло следующее. На экране первой активности в портретном режиме у нас теперь один фрагмент, а вот код об этом не знает и пытается обратиться к фрагменту, которого не существует для него. Ведь разметка поменялась.

Иными словами, нам нужно отслеживать ориентацию или размеры устройства и выбирать, какой код использовать - с участием фрагментов или запускать вторую активность.

Прежде чем решать эту задачу, добавим код во вторую активность.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);

    // Получим индекс из намерения активности
    Intent intent = getIntent();
    int buttonIndex = intent.getIntExtra("buttonIndex", -1);
    if (buttonIndex != -1) {
        FragmentManager fragmentManager = getSupportFragmentManager();
        Fragment2 fragment2 = (Fragment2) fragmentManager
            .findFragmentById(R.id.fragment2);
        fragment2.setDescription(buttonIndex);
    }
}

```

Мы знаем, что на другую активность можно перейти через намерение и передавать данные через метод **putExtra()**. А вторая активность, соответственно, может запуститься и получить данные от первой активности через принимающий метод **getIntExtra()**.

Вторую активность интересует только индекс нажатой кнопки. Получив его, мы вызываем менеджер фрагментов, находим по идентификатору сам фрагмент **Fragment2** и вызываем его метод **setDescription()**, передав ему индекс. Как видите, мы по-прежнему, не вносим изменений в код для фрагмента, всё делается на уровне активностей.

Перейдём к первой активности. Тут придётся попотеть.

Когда устройство находится в альбомной ориентации, то всё нормально. В активности присутствуют два фрагмента из разметки **res/layout/activity_main_wide.xml** и код соответствует этому состоянию.

Таким образом нам надо отслеживать ситуацию и поправить код в методе **onButtonSelected()**.

```

@Override
public void onButtonSelected(int buttonIndex) {
    // подключаем FragmentManager
    FragmentManager fragmentManager = getSupportFragmentManager();

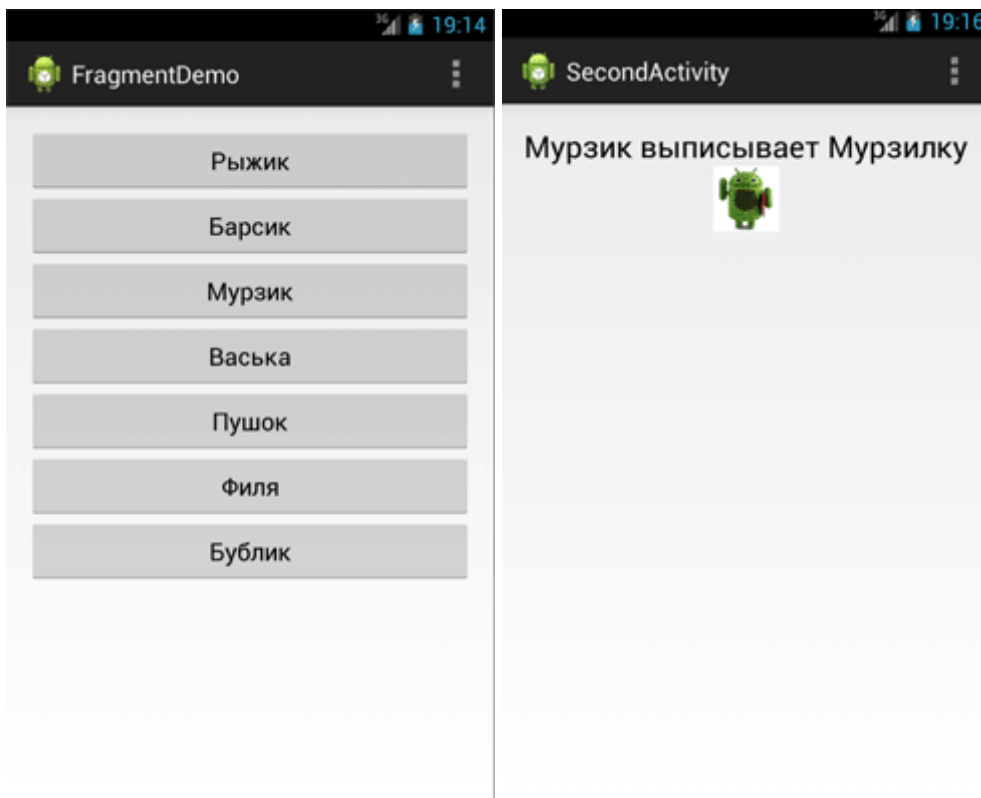
    // Получаем ссылку на второй фрагмент по ID
    Fragment2 fragment2 = (Fragment2) fragmentManager
        .findFragmentById(R.id.fragment2);

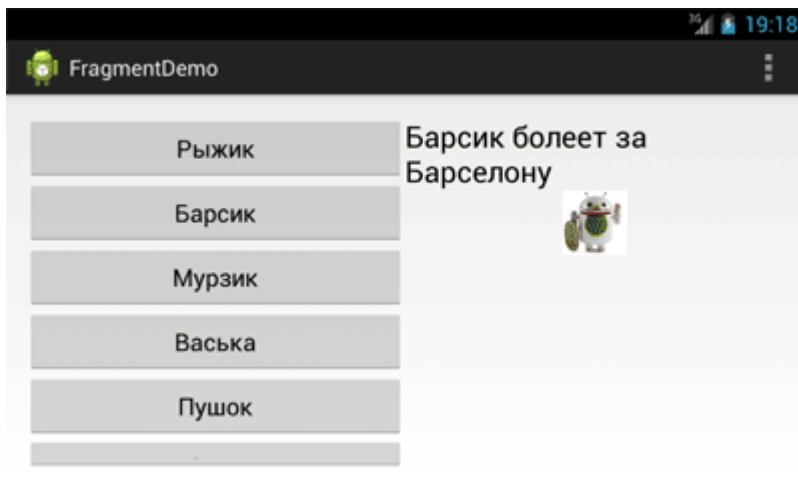
    // если фрагмента не существует или он невидим
    if (fragment2 == null || !fragment2.isVisible()) {
        // запускаем активность
        Intent intent = new Intent(this, SecondActivity.class);
        intent.putExtra("buttonIndex", buttonIndex);
        startActivity(intent);
    } else {
        // Выводим нужную информацию
        fragment2.setDescription(buttonIndex);
    }
}

```

А отслеживать мы будем через условие **if**. Если фрагмента не существует или он невидим, то запускаем вторую активность. В противном случае фрагмент присутствует в активности, и тогда работаем по старому сценарию.

Запустив код, вы теперь можете нажимать на кнопки в любой ориентации. Программа в состоянии определить, какой код следует выполнять - запускать новую активность или работать в одной активности с двумя фрагментами.





Остался один штрих. Если в портретной ориентации перейти на вторую активность и повернуть устройство в альбомную ориентацию, то увидим не тот результат, на который рассчитывали. Мы ожидаем увидеть двухпанельный режим, а видим альбомную ориентации второй активности. Добавим в метод **onCreate()** второй активности проверку текущего состояния ориентации. Если обнаружим, что находимся в альбомной ориентации, то просто выходим из активности. Конечно, можно дополнительно запоминать, какая кнопка была нажата, чтобы при возврате на первую активность вывести соответствующие картинки, но это уже детали.

```
...
setContentView(R.layout.activity_second);

if (getResources().getConfiguration().orientation == Configuration.ORIENTATION_LANDSCAPE) {
    finish();
    return;
}

...
```

Для новичков тема с фрагментами может показаться сложной из-за обилия кода с применением интерфейсов и всякой ерунды. Тем более, что мы рассмотрели только часть возможностей фрагментов.

На последнем рисунке видно, что не все кнопки поместились на экран. Можно было добавить контейнер **ScrollView** для решения проблемы. Но в большинстве случаев используют специальный фрагмент **ListFragment**. Если помните, когда на экране имеется только один компонент **ListView**, то можно использовать готовую активность **ListActivity**. Аналогично, если фрагмент состоит только из **ListView**, то нужно использовать **ListFragment**.

Часть четвёртая. Динамическое управление фрагментами (fragment4.php)

Дополнительное чтение

Обсуждение статьи (<http://forum.alexanderklimov.ru/viewtopic.php?id=31>) на форуме.

Реклама