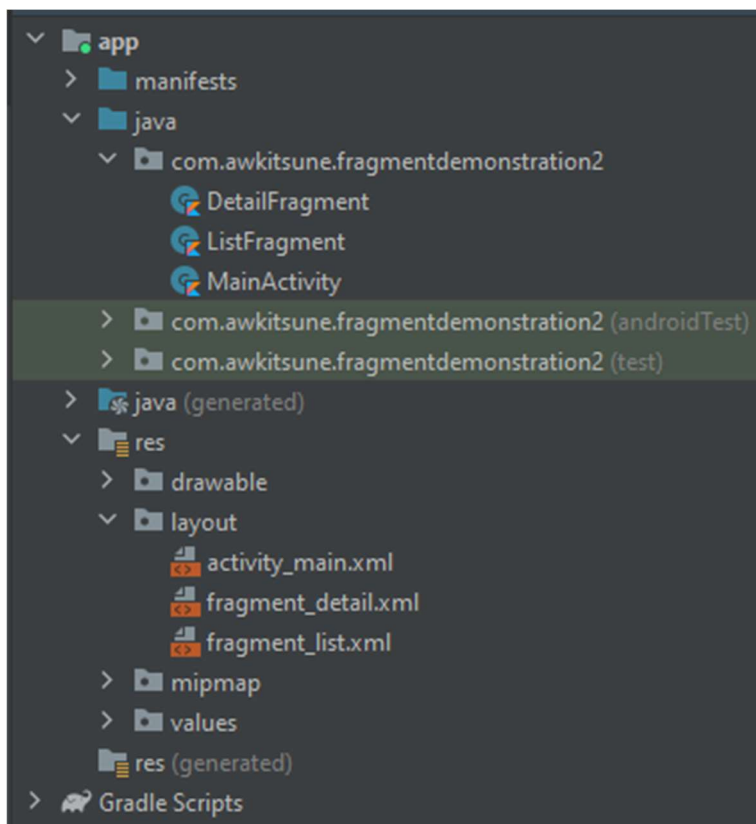


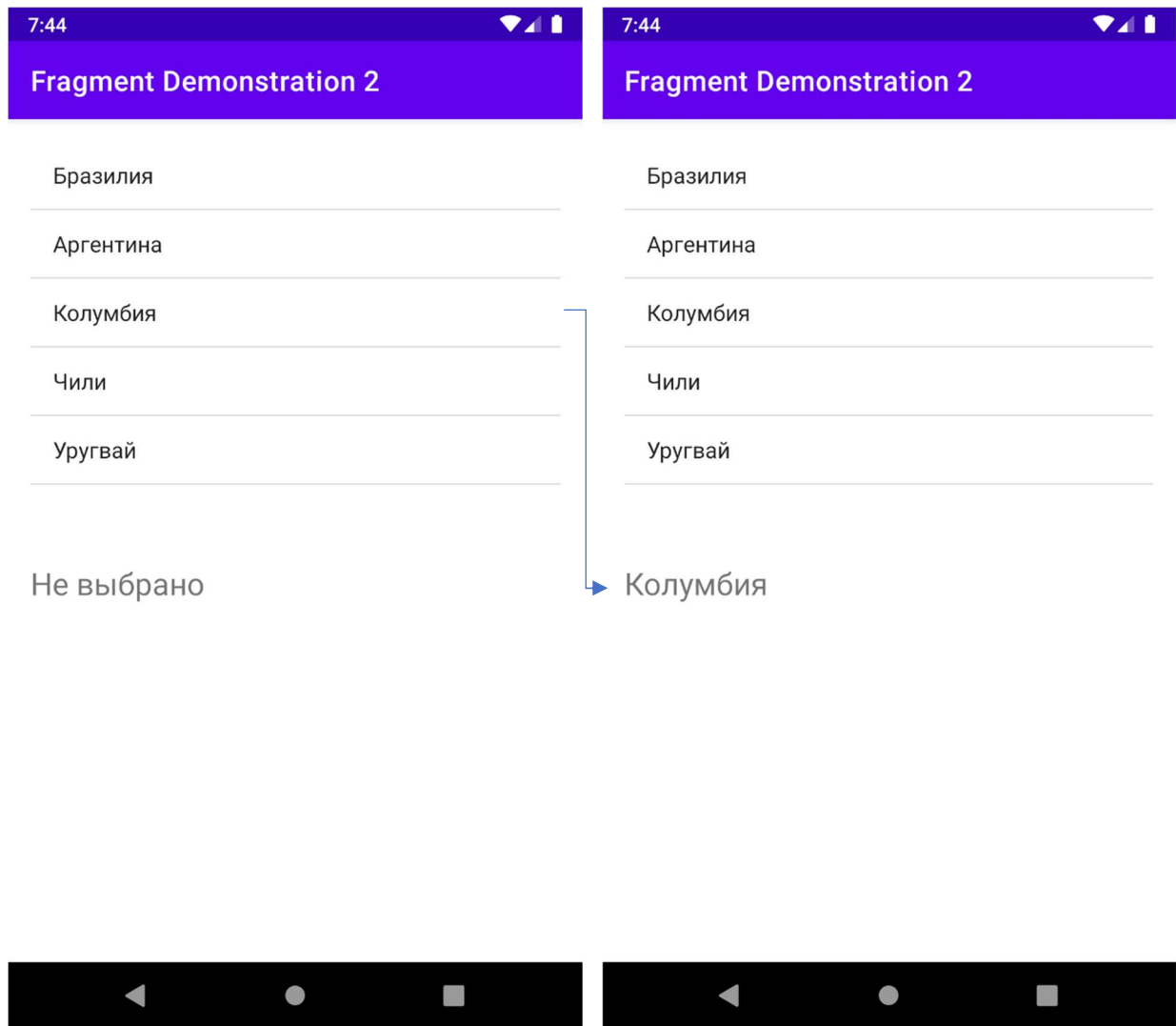


Лекция #28. Фрагменты в альбомном и портретном режиме

В прошлой теме было разработано приложение, которое выводит оба фрагмента на экран. Продолжим работу с этим проектом. Всего было создано два фрагмента: ListFragment для отображения списка и DetailFragment для отображения выбранного элемента в списке. И MainActivity выводила оба фрагмента на экран:



Но отображение двух и более фрагментов при портретной ориентации не очень оптимально. Например, в прошлой теме приложение выглядело так:



Но если список большой, то второй фрагмент, который отображает выбранный элемент, соответственно уходит вниз. При альбомной ориентации получится расположение еще более неоптимально. Поэтому сначала изменим файл **activity_main.xml**, чтобы более удобно располагать фрагменты в альбомной ориентации:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/listFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:name="com.awkitsune.fragmentdemonstration2.ListFragment"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintRight_toLeftOf="@id/detailFragment"
        app:layout_constraintBottom_toBottomOf="parent" />

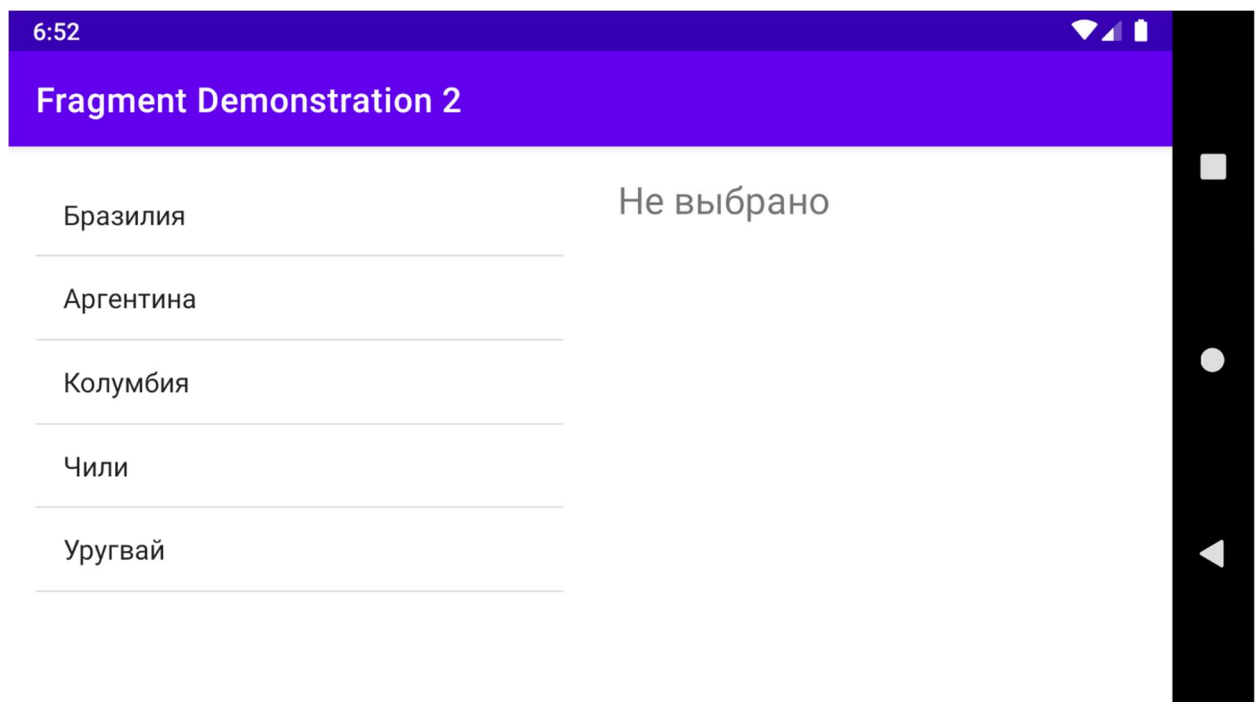
    <androidx.fragment.app.FragmentContainerView
```

```

        android:id="@+id/detailFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:name="com.awkitsune.fragmentdemonstration2.DetailFragment"
        app:layout_constraintLeft_toRightOf="@id/listFragment"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintRight_toRightOf="parent"/>
    </androidx.constraintlayout.widget.ConstraintLayout>

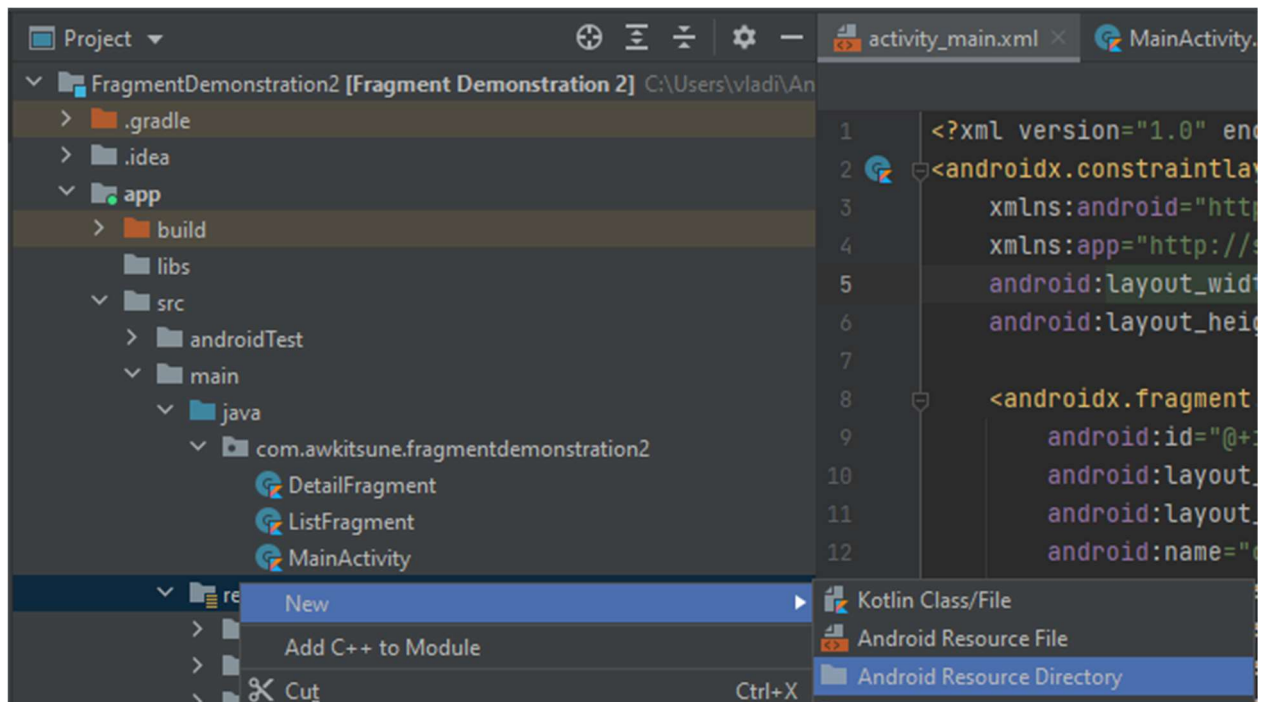
```

Для более удобного расположения при альбомной ориентации, как правило, решение довольно простое - два фрагмента располагаются горизонтально в ряд.

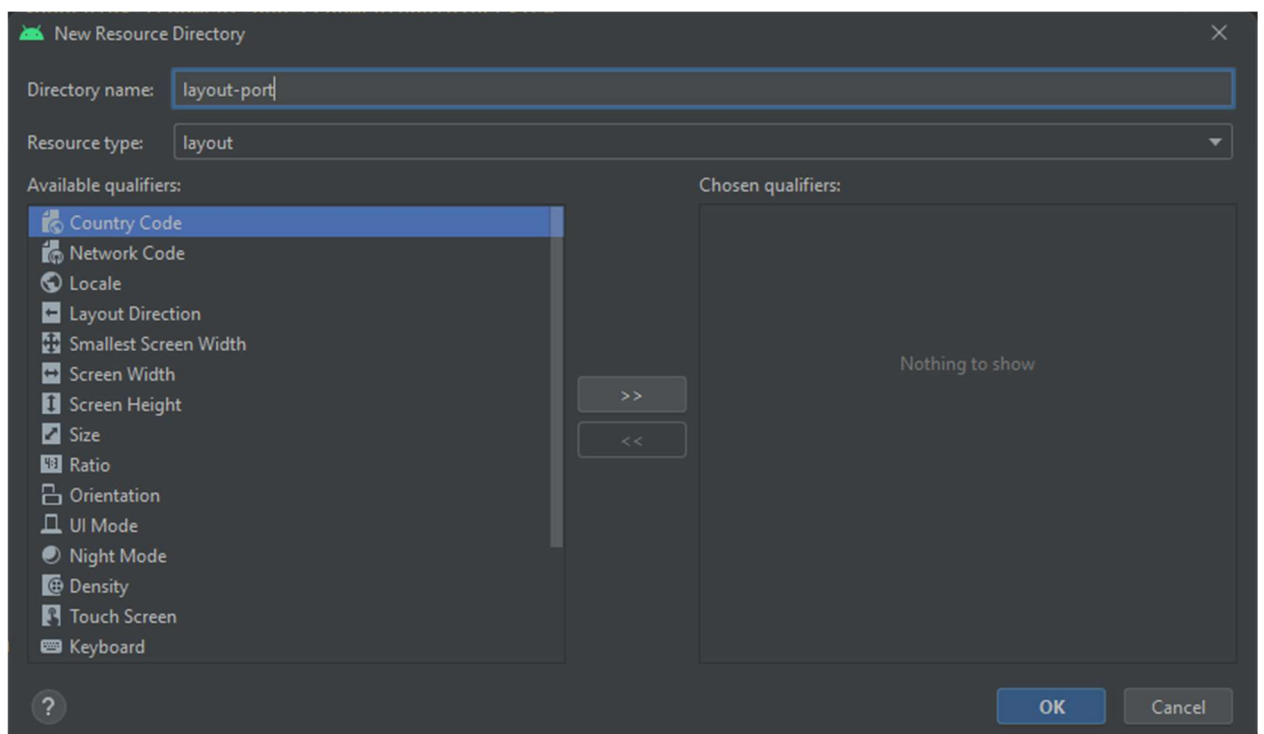


Теперь рассмотрим как более удачно расположить фрагменты при портретной ориентации. Нередко в этом случае решение следующее - одновременно экран отображает только один фрагмент.

Итак, создадим в проекте в папке **res**, где хранятся все ресурсы, подкаталог **layout-port**, который будет хранить файлы интерфейса для портретной ориентации. Для этого переключимся к полному виду проекта. Нажмем на папку **res** правой кнопкой мыши и в контекстном меню выберем **New -> Android Resource Directory**:



Назовем новую папку **layout-port**:

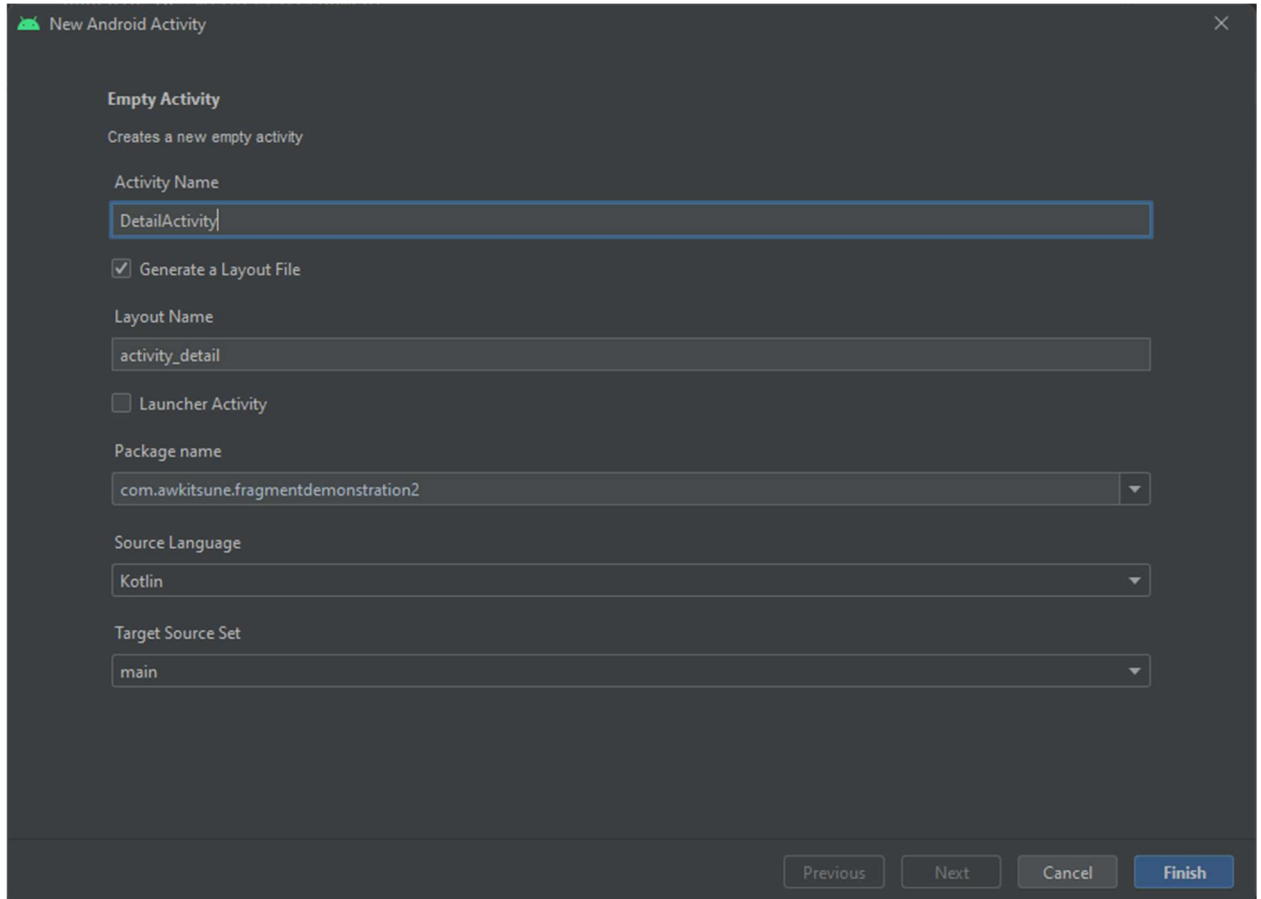


Далее добавим в **res/layout-port** новый файл **activity_main.xml** и определим в нем следующий код:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.fragment.app.FragmentContainerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/listFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.awkitsune.fragmentdemonstration2.ListFragment" />
```

Этот файл будет использоваться для портретной ориентации MainActivity. Таким образом, MainActivity в портретном режиме будет отображать только один список.

Теперь добавим новую activity, которую назовем **DetailActivity**:



В папке **res/layout** в файле **activity_detail.xml** определим для DetailActivity следующий интерфейс:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.fragment.app.FragmentContainerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/detailFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.awkitsune.fragmentdemonstration2.DetailFragment" />
```

Таким образом, интерфейс DetailActivity будет определяться загружаемым фрагментом DetailFragment.

Далее в коде **DetailActivity** определим следующий код:

```
package com.awkitsune.fragmentdemonstration2

import android.content.res.Configuration
import androidx.appcompat.app.AppCompatActivity
```

```

import android.os.Bundle

class DetailActivity : AppCompatActivity() {
    companion object{
        const val SELECTED_ITEM = "SELECTED_ITEM"
    }
    var selectedItem = "Не выбрано"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        if(resources.configuration.orientation ==
        Configuration.ORIENTATION_LANDSCAPE) {
            finish()
            return
        }

        setContentView(R.layout.activity_detail)

        val extras: Bundle? = intent.extras
        selectedItem = extras?.getString(SELECTED_ITEM).toString()
    }

    override fun onResume() {
        super.onResume()

        val fragment: DetailFragment =
            supportFragmentManager.findFragmentById(R.id.detailFragment) as
            DetailFragment
        fragment?.setSelectedItem(selectedItem)
    }
}

```

Здесь в первую очередь проверяем конфигурацию. Так как эта activity предназначена только для портретного режима, то при альбомной ориентации осуществляем выход:

```

if(resources.configuration.orientation ==
    Configuration.ORIENTATION_LANDSCAPE) {
    finish()
    return
}

```

Если устройство находится в портретном режиме, то получаем переданные данные по ключу **"SELECTED_ITEM"**:

Предполагается, что по ключу **"SELECTED_ITEM"** будет передаваться выбранные элемент списка из MainActivity, когда она будет находиться в портретной ориентации.

И очень важный момент - нам надо передать это значение в текстовое поле, определенное во фрагменте. Однако надо учитывать особенности жизненного цикла представления фрагмента. В данном случае переопределяется метод **onResume()**, потому что при вызове этого метода DetailActivity уже будет видима на экране, и пользователь сможет с ней взаимодействовать. А это также

значит, что в этой точке уже будет активен и фрагмент и его представление. К примеру, в методе **onCreate()** представление фрагмента еще полностью не создано, поэтому мы не можем в нем получить виджеты, которые определены во фрагменте. Зато можем все это сделать в методе **onResume()**.

```
override fun onResume() {
    super.onResume()

    val fragment: DetailFragment =
        supportFragmentManager.findFragmentById(R.id.detailFragment) as
        DetailFragment
    fragment?.setSelectedItem(selectedItem)
}
```

И в данном случае мы получаем через метод **getSupportFragmentManager()** фрагмент **DetailFragment** и вызываем его метод **setSelectedItem()**. В качестве аргумента в этот метод передается строковое значение, переданное через **Intent**.

И также изменим главную activity - **MainActivity**:

```
package com.awkitsune.fragmentdemonstration2

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity(),
    ListFragment.OnFragmentSendDataListener {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    override fun sendData(selectedItem: String?) {
        val fragment = supportFragmentManager
            .findFragmentById(R.id.detailFragment) as DetailFragment?

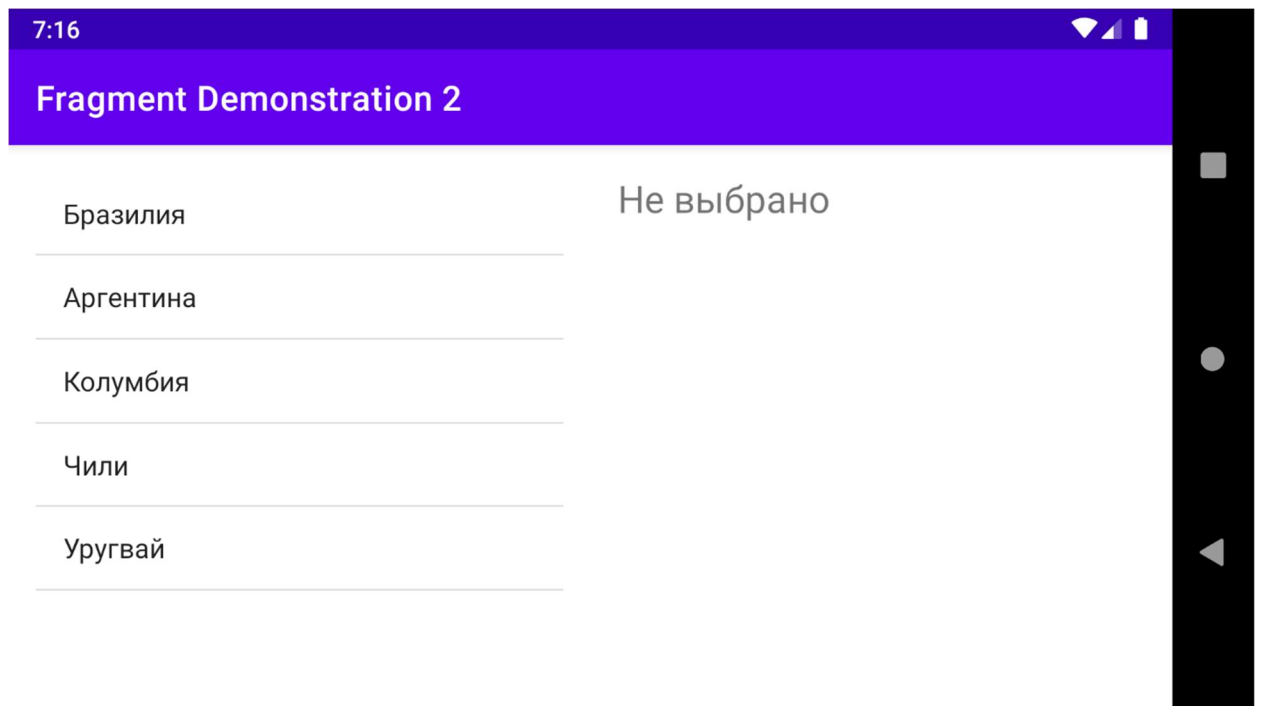
        if (fragment != null && fragment.isVisible) {
            fragment.setSelectedItem(selectedItem)
        }
        else {
            val intent = Intent(applicationContext,
                DetailActivity::class.java)
            intent.putExtra(DetailActivity.SELECTED_ITEM, selectedItem)
            startActivity(intent)
        }

        fragment?.setSelectedItem(selectedItem)
    }
}
```

С помощью метода **fragment.isVisible()** мы можем узнать, активен ли определенный фрагмент в разметке интерфейса. Если фрагмента **DetailFragment** в данный момент не видим, то используется портретный режим, и поэтому запускается **DetailActivity**. Иначе идет работа с фрагментом внутри

MainActivity, которая в альбомном режиме отображает сразу два фрагмента - ListFragment и DetailFragment.

Запустим приложение и перейдем в альбомный режим:



А при портретной ориентации экран будет выглядеть иначе:

Бразилия

Аргентина

Колумбия

Чили

Уругвай

▶ Уругвай

