

Họ tên: Lê Phúc Hưng

MSSV: 20215276

Mã lớp: 151902

Môn học: Phát triển ứng dụng cho thiết bị di động (IT4785)

Source Code: <https://github.com/lephuchung/HustMobile>

Bài: Lesson 10 – Fragment – Nhân 2 số

Contents

1. Tạo project android mới và cài đặt thư viện.....	2
2. Tạo InputFragment và multifragment.....	2
3. Tạo file Navigation Graph.....	4
4. Tích hợp Navigation Graph vào MainActivity.....	5
5. Xử lý sự kiện khi người dùng ấn nút Multiply trên InputFragment.....	5
6. Truyền / nhận dữ liệu sử dụng Safe Args	6

1. Tạo project android mới và cài đặt thư viện

- Thêm safeargs vào dự án

```
plugins {  
    alias(libs.plugins.android.application) apply false  
    alias(libs.plugins.kotlin.android) apply false  
    id("androidx.navigation.safeargs") version "2.8.3" apply false  
}
```

```
1 plugins {  
2     alias(libs.plugins.android.application)  
3     alias(libs.plugins.kotlin.android)  
4     id("androidx.navigation.safeargs.kotlin")  
5 }
```

- Hỗ trợ cơ chế view binding:

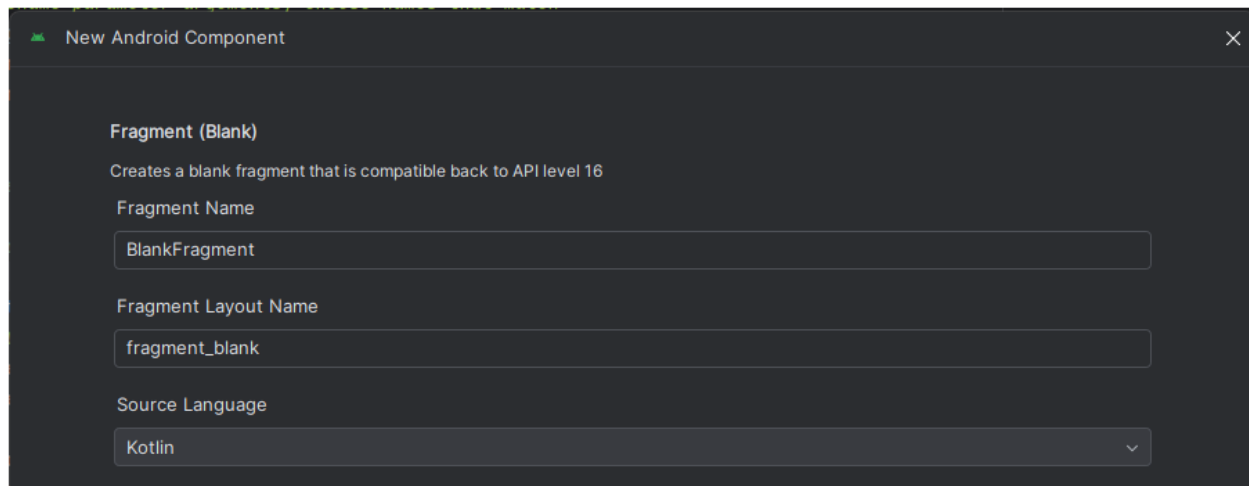
```
// Add  
buildFeatures {  
    dataBinding = true  
}
```

- Thêm 2 dependencies:

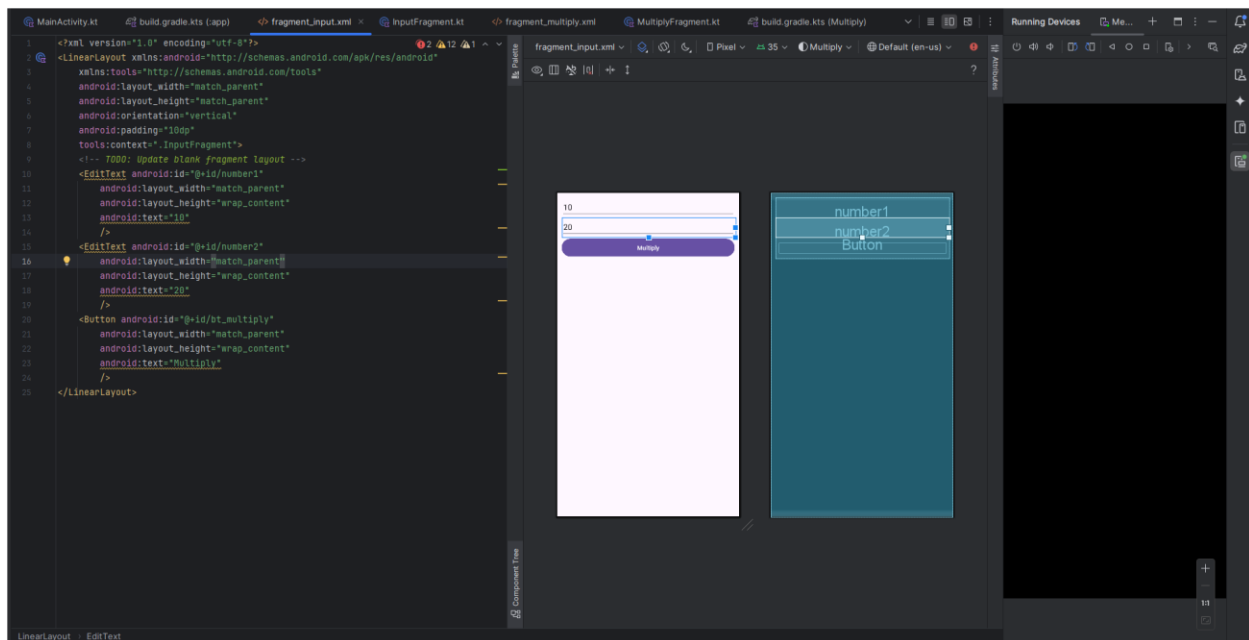
```
44 dependencies {  
45  
46     implementation(libs.androidx.core.ktx)  
47     implementation(libs.androidx.appcompat)  
48     implementation(libs.material)  
49     implementation(libs.androidx.activity)  
50     implementation(libs.androidx.constraintlayout)  
51     testImplementation(libs.junit)  
52     androidTestImplementation(libs.androidx.junit)  
53     androidTestImplementation(libs.androidx.espresso.core)  
54  
55     // Add  
56     implementation("androidx.navigation:navigation-fragment-ktx:2.8.3")  
57     implementation("androidx.navigation:navigation-ui-ktx:2.8.3")  
58 }
```

2. Tạo InputFragment và multifragment

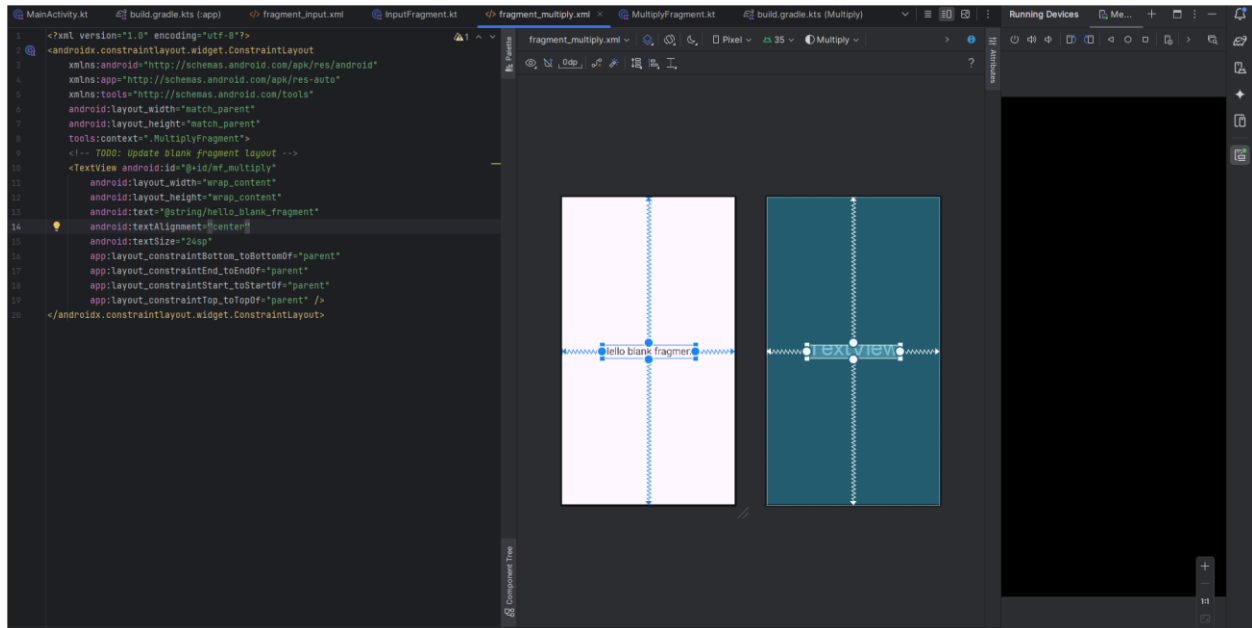
- Tạo 2 fragment mới:



Cập nhật giao diện cho fragment_input.xml



Tiếp theo, cập nhật giao diện fragment_multiply.xml



Sửa lại hàm onCreateView trong InputFragment.kt

```

32  override fun onCreateView(
33      inflater: LayoutInflater,
34      container: ViewGroup?,
35      savedInstanceState: Bundle?
36  ): View? {
37      return inflater.inflate(R.layout.fragment_input, container, attachToRoot: false)
38  }

```

Tương tự, sửa lại hàm onCreateView trong MultiplyFragment.kt

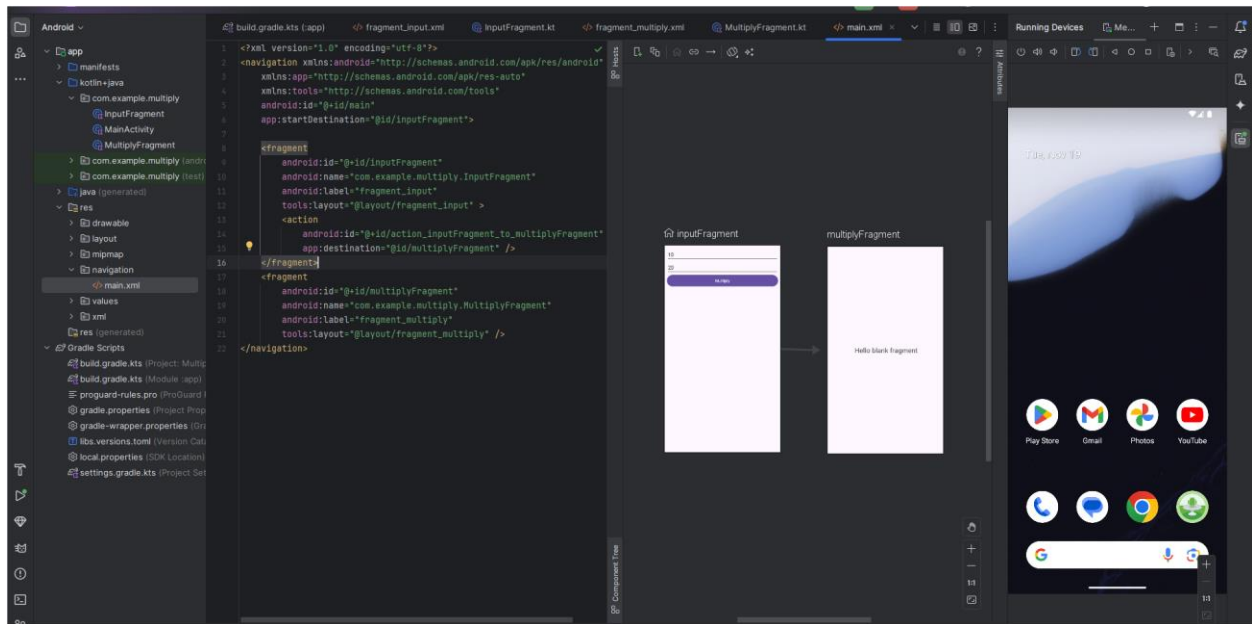
```

32  override fun onCreateView(
33      inflater: LayoutInflater,
34      container: ViewGroup?,
35      savedInstanceState: Bundle?
36  ): View? {
37      return inflater.inflate(R.layout.fragment_multiply, container, attachToRoot: false)
38  }

```

3. Tạo file Navigation Graph

Tạo navigation và sửa file main.xml như sau:



4. Tích hợp Navigation Graph vào MainActivity

Sửa lại file activity_main.xml để tích hợp navigation vào mainActivity

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <navigation xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/main"
6   app:startDestination="@id/inputFragment">
7   <fragment
8     android:id="@+id/inputFragment"
9     android:name="com.example.multiply.InputFragment"
10    android:label="fragment_input"
11    tools:layout="@layout/fragment_input">
12    <action
13      android:id="@+id/action_inputFragment_to_multiplyFragment"
14      app:destination="@id/multiplyFragment" />
15  </fragment>
16  <fragment
17    android:id="@+id/multiplyFragment"
18    android:name="com.example.multiply.MultiplyFragment"
19    android:label="fragment_multiply"
20    tools:layout="@layout/fragment_multiply" />
21 </navigation>

```

Chạy chương trình ta chỉ được màn input

5. Xử lý sự kiện khi người dùng ấn nút Multiply trên InputFragment

```

override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    super.onCreateView(view, savedInstanceState)
    val button: Button = view.findViewById(R.id.bt_multiply)
    button.setOnClickListener {
        view.findNavController().navigate(R.id.action_inputFragment_to_multiplyFragment)
    }
}

```

Biên dịch và chạy chương trình, khi đó nếu người dùng ấn vào nút Multiply, chương trình sẽ chuyển sang và hiển thị MultiplyFragment.

6. Truyền / nhận dữ liệu sử dụng Safe Args

Trong file `res/navigation/main.xml`, bổ sung hai Argument cho fragment `multiplyFragment`:

```
16 <fragment
17     android:id="@+id/multiplyFragment"
18     android:name="com.example.multiply.MultiplyFragment"
19     android:label="fragment_multiply"
20     tools:layout="@layout/fragment_multiply">
21     <argument
22         android:name="number1"
23         android:defaultValue="1.0"
24         app:argType="float" />
25     <argument
26         android:name="number2"
27         android:defaultValue="1.0"
28         app:argType="float" />
29 </fragment>
```

Để truyền dữ liệu, trong `InputFragment.kt`, cập nhật hàm `onViewCreated`:

```
64 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
65     super.onViewCreated(view, savedInstanceState)
66     val button: Button = view.findViewById(R.id.bt_multiply)
67     val number1: EditText = view.findViewById(R.id.number1)
68     val number2: EditText = view.findViewById(R.id.number2)
69     button.setOnClickListener {
70         //Toast.makeText(this.requireContext(), n1.toString(), Toast.LENGTH_SHORT).show()
71         val n1:Float = number1.text.toString().toFloatOrNull() ?: 0.0f
72         val n2:Float = number2.text.toString().toFloatOrNull() ?: 0.0f
73         //view.findNavController().navigate(R.id.action_inputFragment_to_multiplyFragment)
74         val action =
75             InputFragmentDirections.actionInputFragmentToMultiplyFragment(n1, n2)
76         view.findNavController().navigate(action)
77     }
78 }
```

Để nhận dữ liệu, trong file `MultiplyFragment`, viết chồng hàm `onViewCreated`:

```
61 val args: MultiplyFragmentArgs by navArgs()
62 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
63     super.onViewCreated(view, savedInstanceState)
64     val number1 = args.number1
65     val number2 = args.number2
66     val result = number1 * number2
67     val output: TextView = view.findViewById(R.id.mf_multiply)
68     output.text = "${number1} * ${number2} = ${result}"
69 }
```

Chạy lại chương trình được kết quả như demo ban đầu
Chạy lại chương trình được kết quả như demo ban đầu