Uniwersytet
Wrocławski

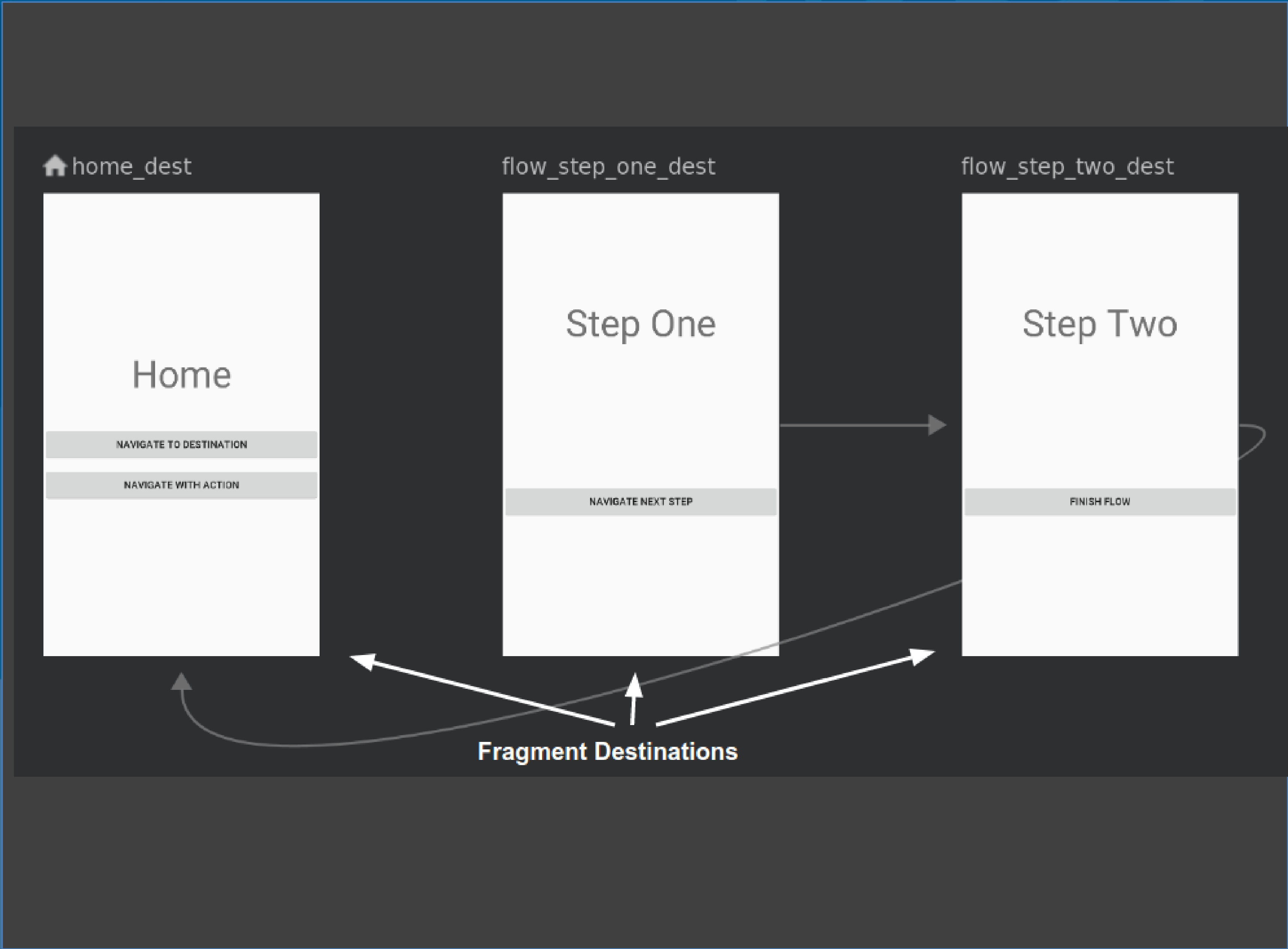
Rafał Lewandków

PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

WYKŁAD 4
Jetpack Navigation

- Nawigacja w aplikacji
- Navigation Component
- Przekazywanie argumentów
- Bottom Navigation
- Nawigacja z TabLayout i ViewPager2

Jetpack Navigation

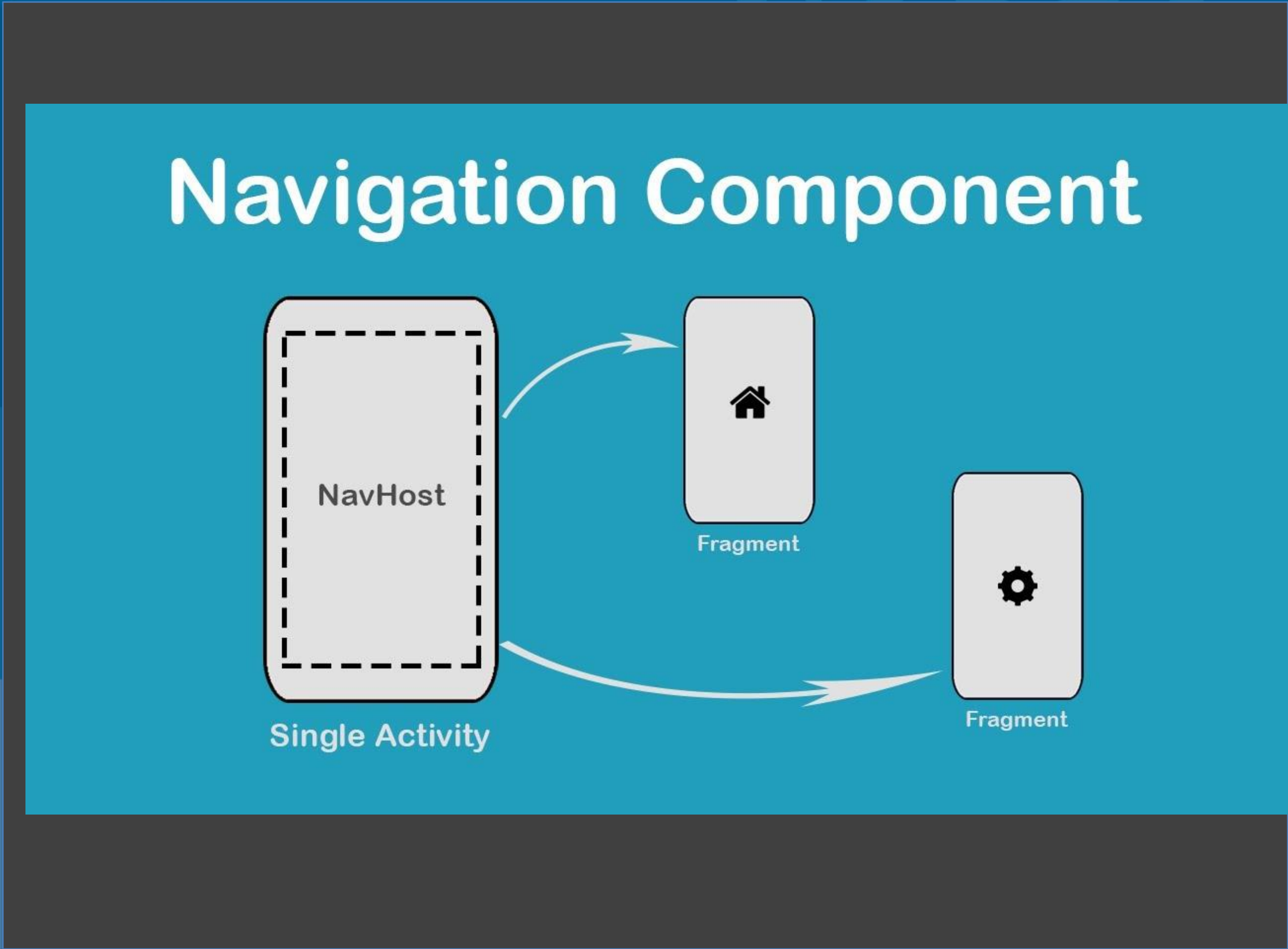


Fragment Destinations

Komponenty

1. **NavHost**: NavHost to kontener, w którym odbywa się nawigacja między różnymi fragmentami. Może to być aktywność lub fragment w Twojej aplikacji, który działa jako miejsce docelowe dla nawigacji.
2. **NavGraph**: NavGraph to reprezentacja struktury nawigacji w Twojej aplikacji. Może to być plik XML, który definiuje połączenia między fragmentami i ekranami oraz opisuje zachowanie nawigacji.
3. **NavDestination**: NavDestination to punkt docelowy w grafie nawigacji. Może to być fragment, aktywność lub inny cel, do którego użytkownik może nawigować.

Komponenty



Navigation Component

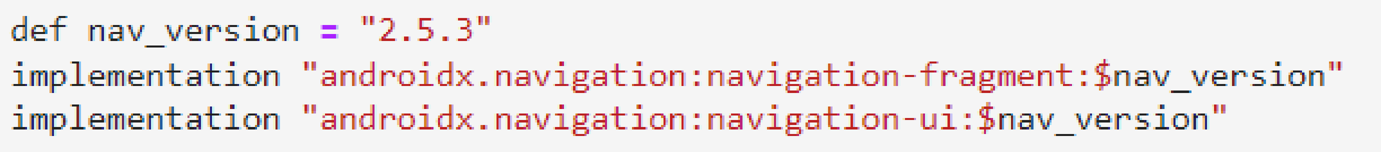
NavHost

Single Activity

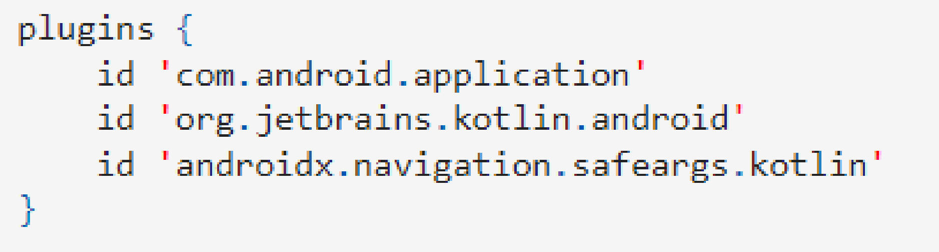
Fragment

Fragment

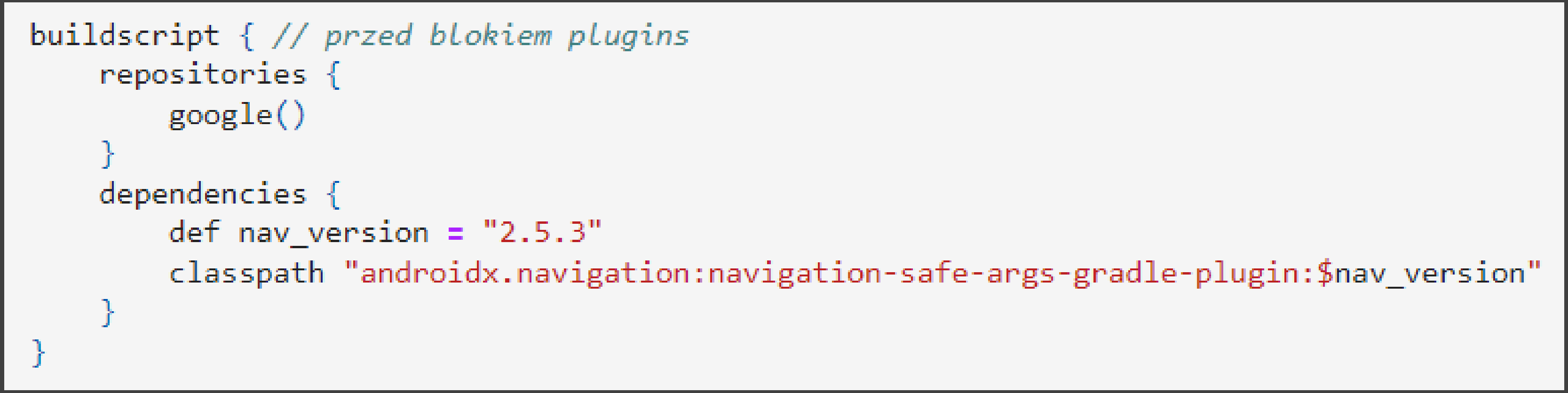
Implementacja



```
def nav_version = "2.5.3"
implementation "androidx.navigation:navigation-fragment:$nav_version"
implementation "androidx.navigation:navigation-ui:$nav_version"
```

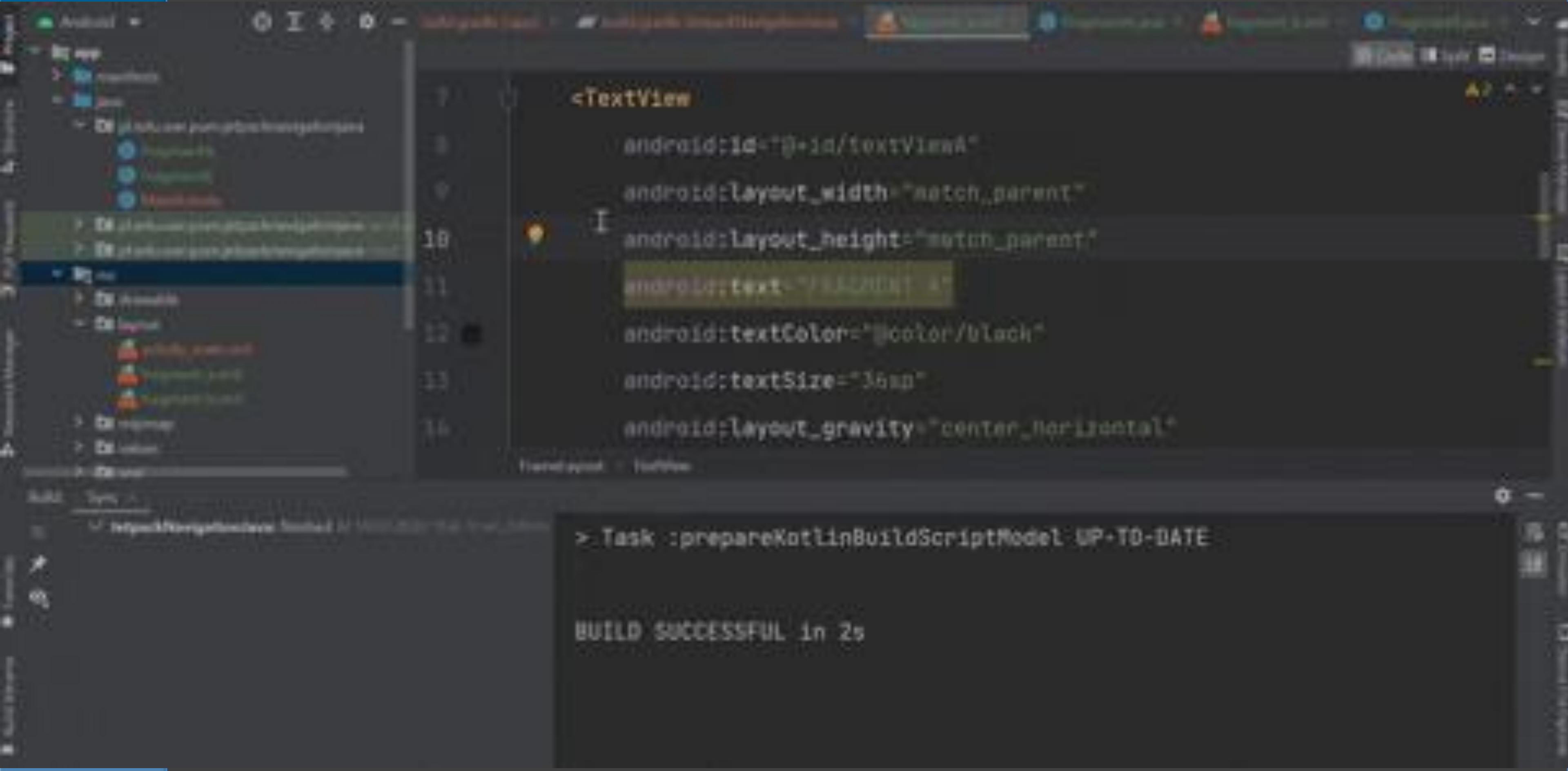


```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'androidx.navigation.safeargs.kotlin'
}
```



```
buildscript { // przed blokiem plugins
    repositories {
        google()
    }
    dependencies {
        def nav_version = "2.5.3"
        classpath "androidx.navigation:navigation-safe-args-gradle-plugin:$nav_version"
    }
}
```

Implementacja

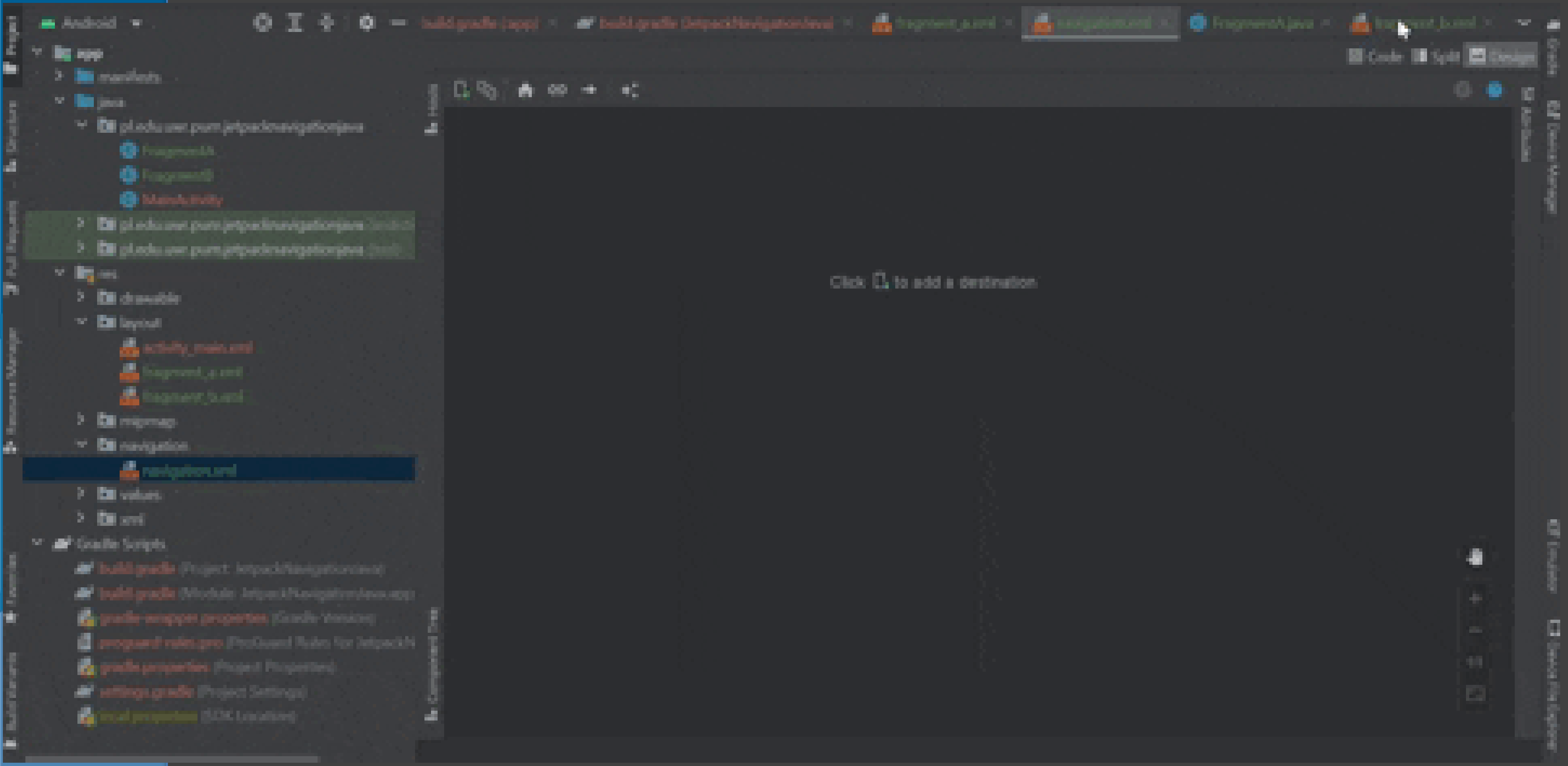


```
<TextView
    android:id="@+id/textViewA"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="FRAGMENT A"
    android:textColor="@color/black"
    android:textSize="36sp"
    android:layout_gravity="center_horizontal"
```

```
> Task :prepareKotlinBuildScriptModel UP-TO-DATE


BUILD SUCCESSFUL in 2s
```

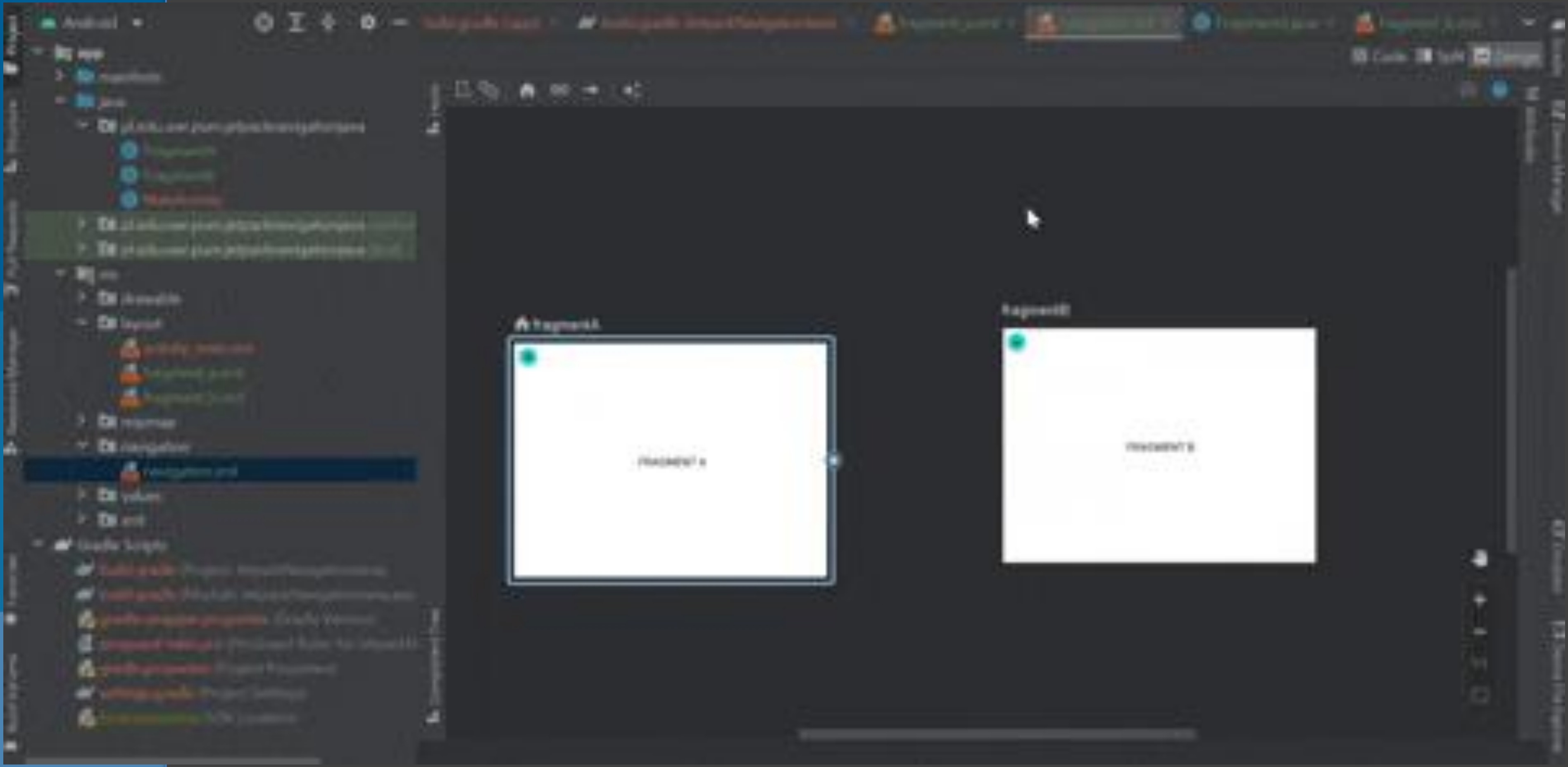
Uniwersytet Wrocławski

Implementacja

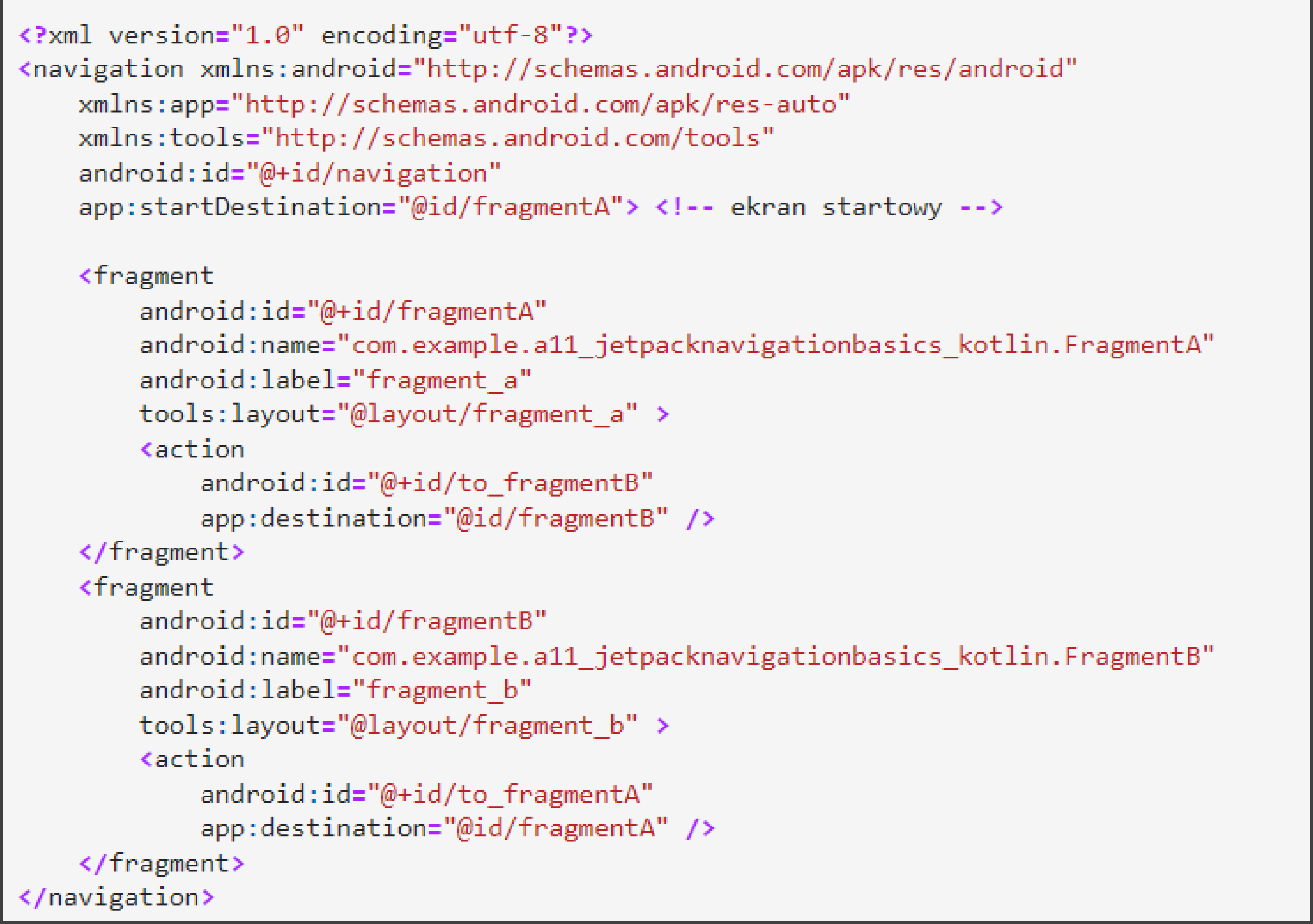

Android
app
manifests
java
pl.edu.uwr.pum.jetpacknavigationjava
FragmentA
FragmentB
MainActivity
pl.edu.uwr.pum.jetpacknavigationjava (androidTest)
pl.edu.uwr.pum.jetpacknavigationjava (test)
res
drawable
layout
activity_main.xml
fragment_a.xml
fragment_b.xml
mipmap
navigation
navigation.xml
values
xml
Gradle Scripts
build.gradle (Project: JetpackNavigationJava)
build.gradle (Module: JetpackNavigationJava.app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for JetpackN
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
build.gradle (:app)
build.gradle (JetpackNavigationJava)
fragment_a.xml
navigation.xml
FragmentA.java
fragment_b.xml
Code
Split
Design
Hosts
Component Tree
Attributes
Project
Structure
Pull Requests
Resource Manager
Favorites
Build Variants
Gradle
Device Manager
Emulator
Device File Explorer

Click to add a destination

Implementacja

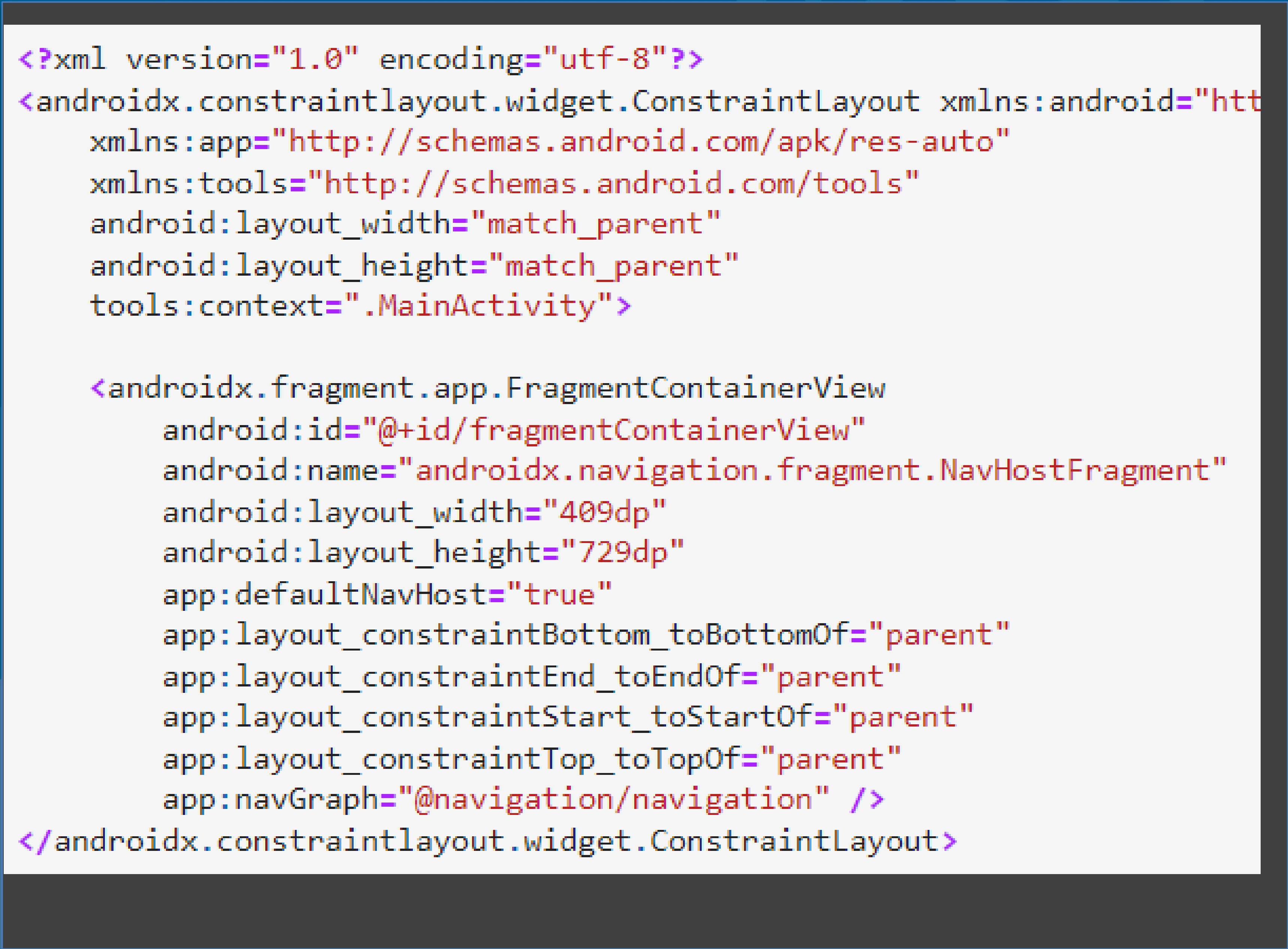


Implementacja



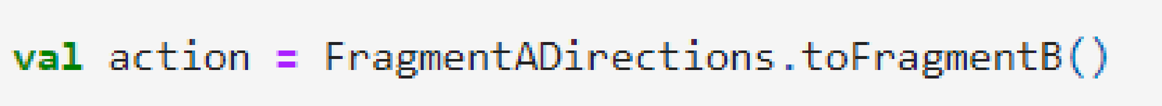
```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/navigation"
    app:startDestination="@id/fragmentA"> <!-- ekran startowy -->

    <fragment
        android:id="@+id/fragmentA"
        android:name="com.example.a11_jetpacknavigationbasics_kotlin.FragmentA"
        android:label="fragment_a"
        tools:layout="@layout/fragment_a" >
        <action
            android:id="@+id/to_fragmentB"
            app:destination="@id/fragmentB" />
    </fragment>
    <fragment
        android:id="@+id/fragmentB"
        android:name="com.example.a11_jetpacknavigationbasics_kotlin.FragmentB"
        android:label="fragment_b"
        tools:layout="@layout/fragment_b" >
        <action
            android:id="@+id/to_fragmentA"
            app:destination="@id/fragmentA" />
    </fragment>
</navigation>
```

Implementacja



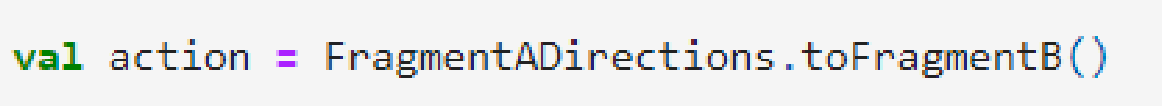
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="htt
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragmentContainerView"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="409dp"
        android:layout_height="729dp"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/navigation" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Implementacja

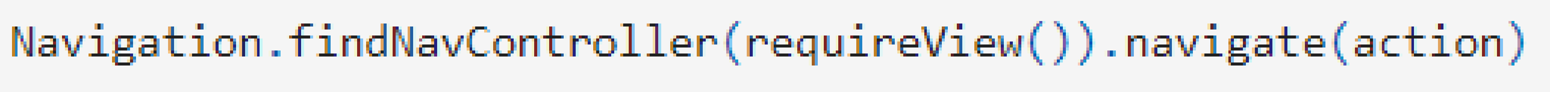


```kotlin
val action = FragmentADirections.toFragmentB()
```

Implementacja

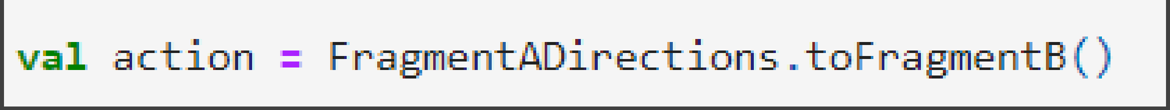


```
val action = FragmentADirections.toFragmentB()
```

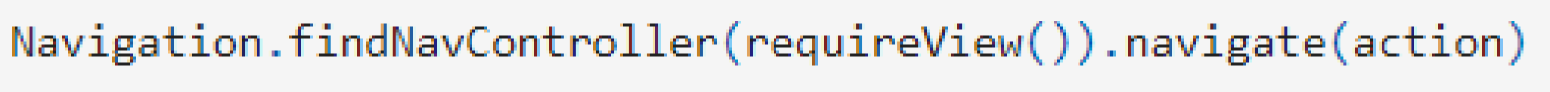


```
Navigation.findNavController(requireView()).navigate(action)
```

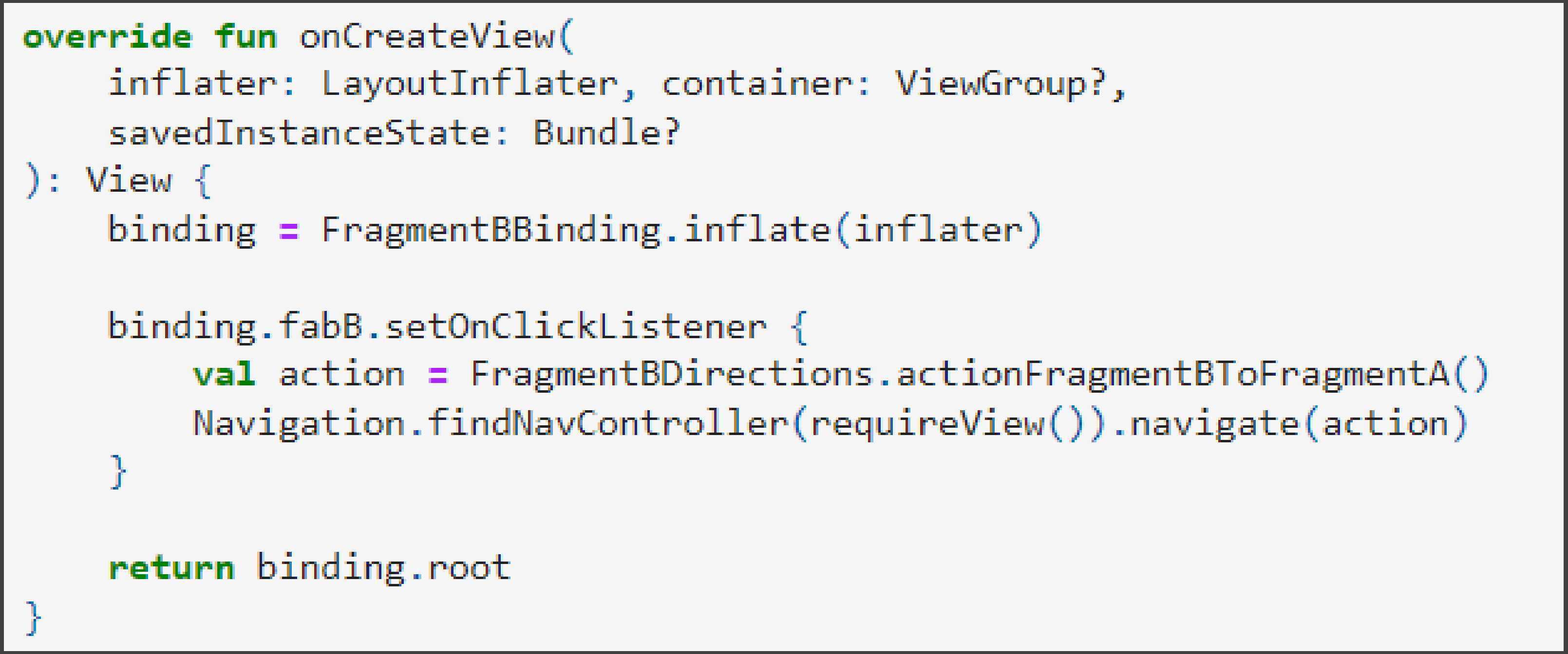
Implementacja



```kotlin
val action = FragmentADirections.toFragmentB()
```

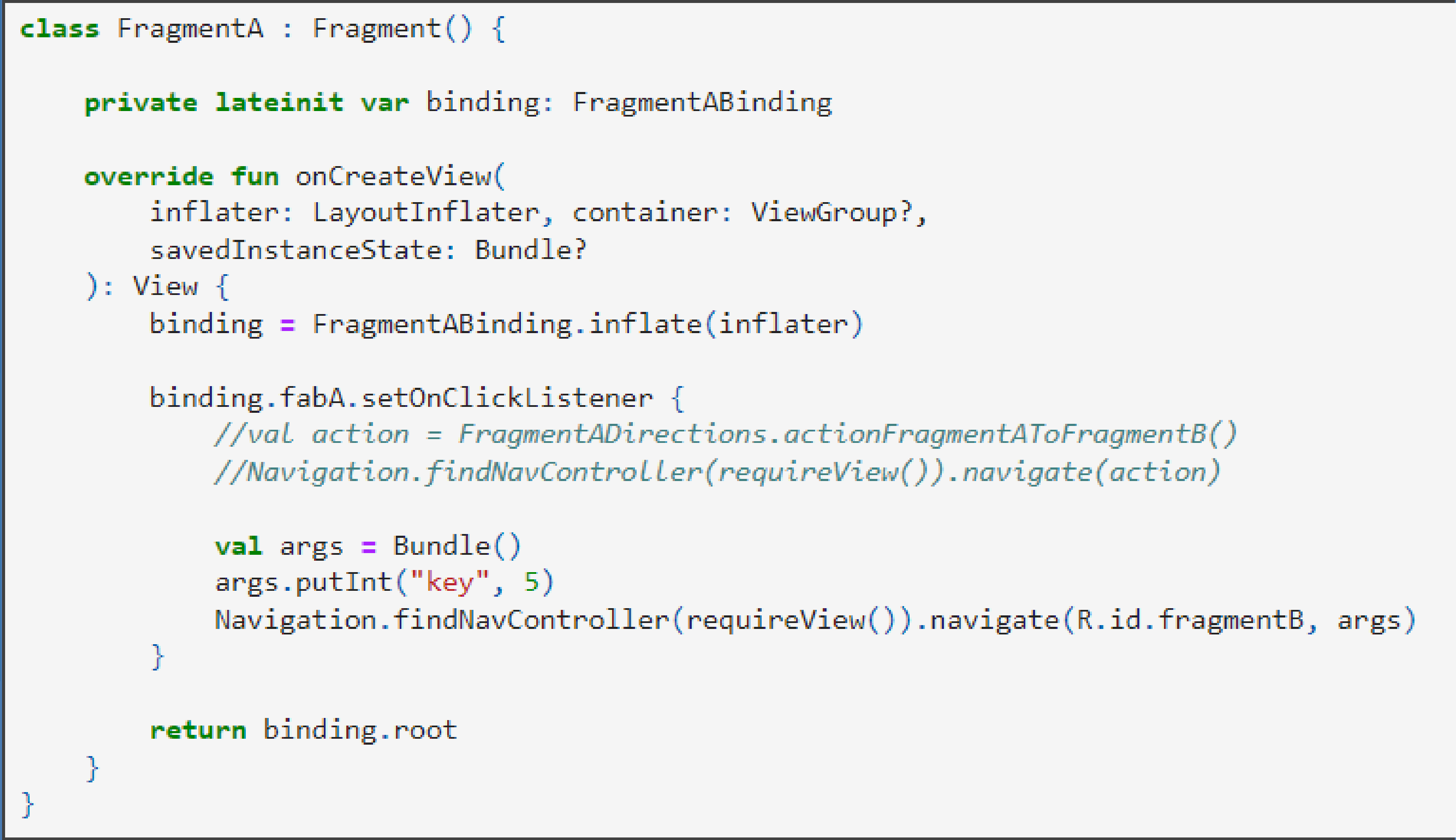


```kotlin
Navigation.findNavController(requireView()).navigate(action)
```



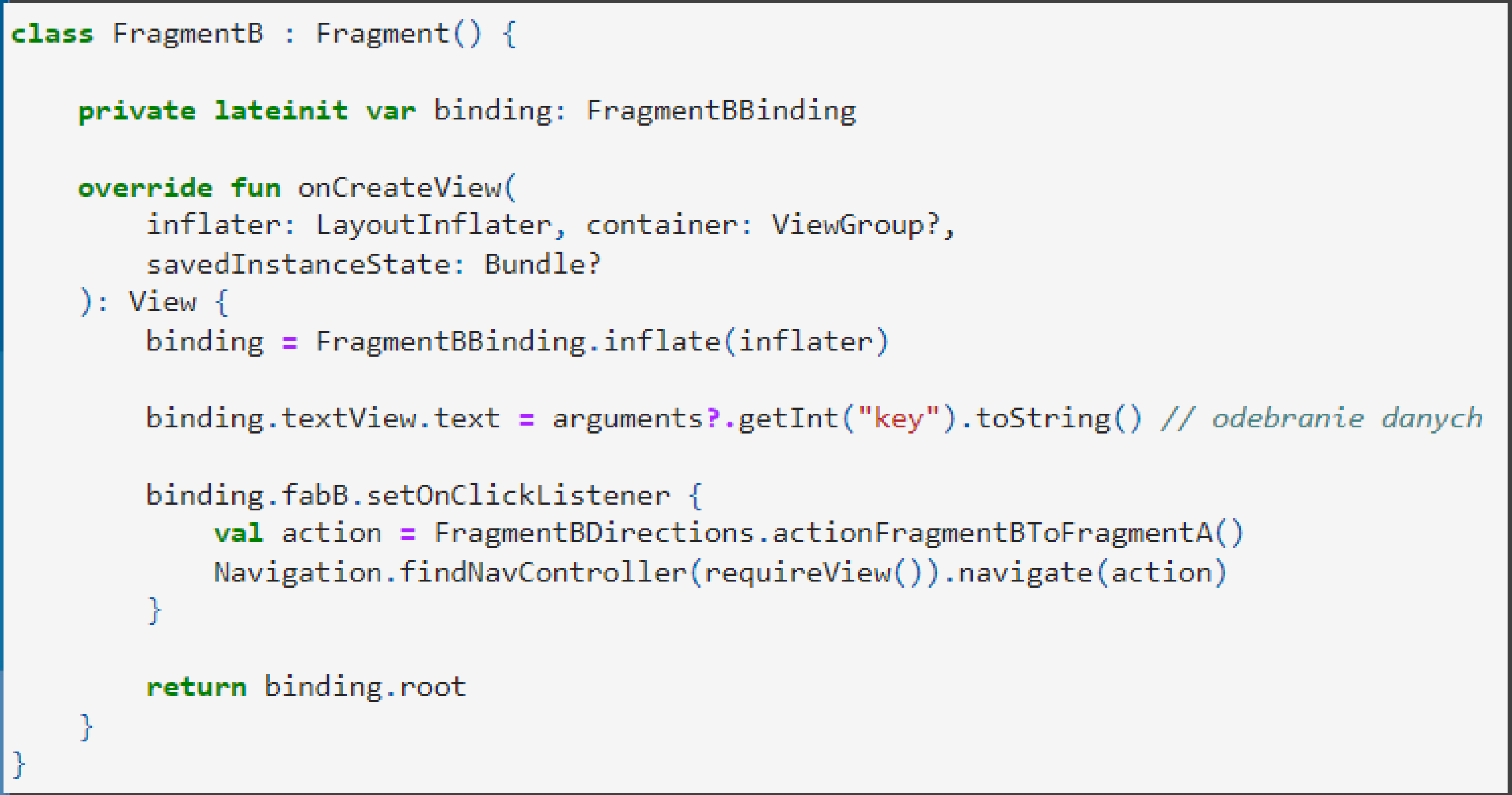
```kotlin
override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    binding = FragmentBBinding.inflate(inflater)

    binding.fabB.setOnClickListener {
        val action = FragmentBDirections.actionFragmentBToFragmentA()
        Navigation.findNavController(requireView()).navigate(action)
    }

    return binding.root
}
```

Argumenty



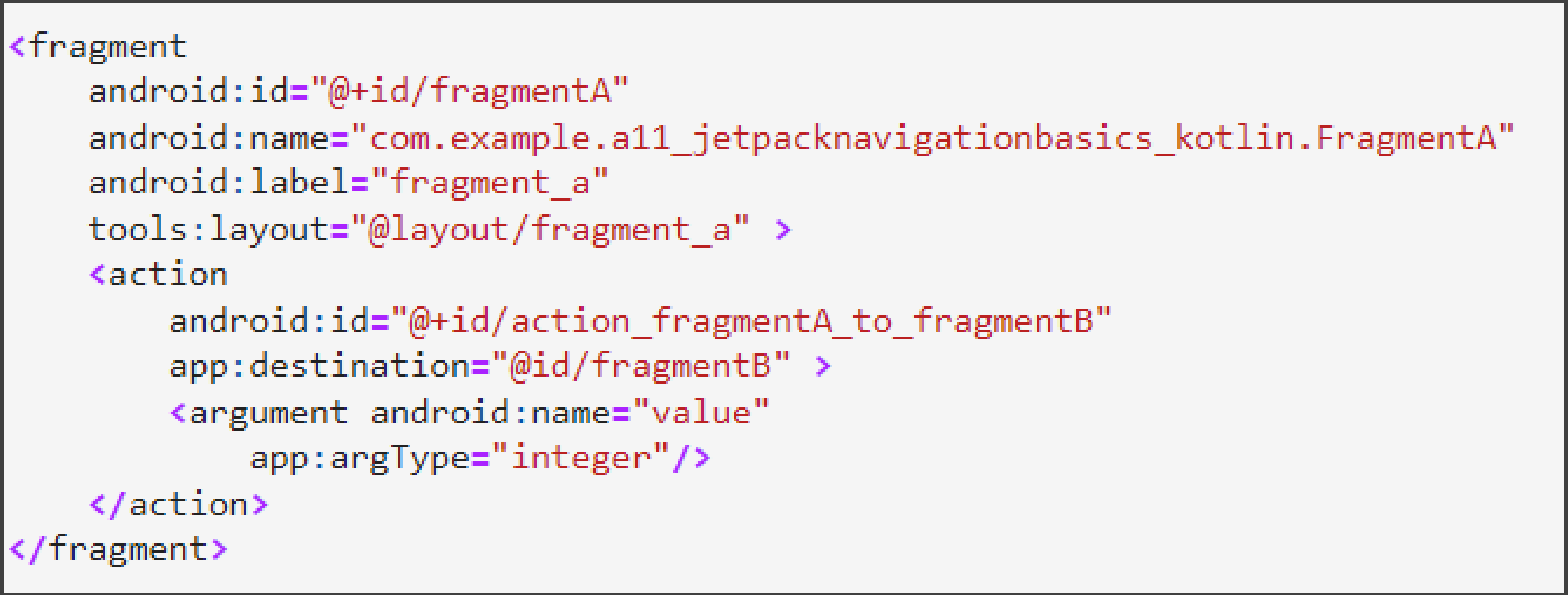
```
class FragmentA : Fragment() {

    private lateinit var binding: FragmentABinding

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        binding = FragmentABinding.inflate(inflater)

        binding.fabA.setOnClickListener {
            //val action = FragmentADirections.actionFragmentAToFragmentB()
            //Navigation.findNavController(requireView()).navigate(action)

            val args = Bundle()
            args.putInt("key", 5)
            Navigation.findNavController(requireView()).navigate(R.id.fragmentB, args)
        }

        return binding.root
    }
}
```

Argumenty



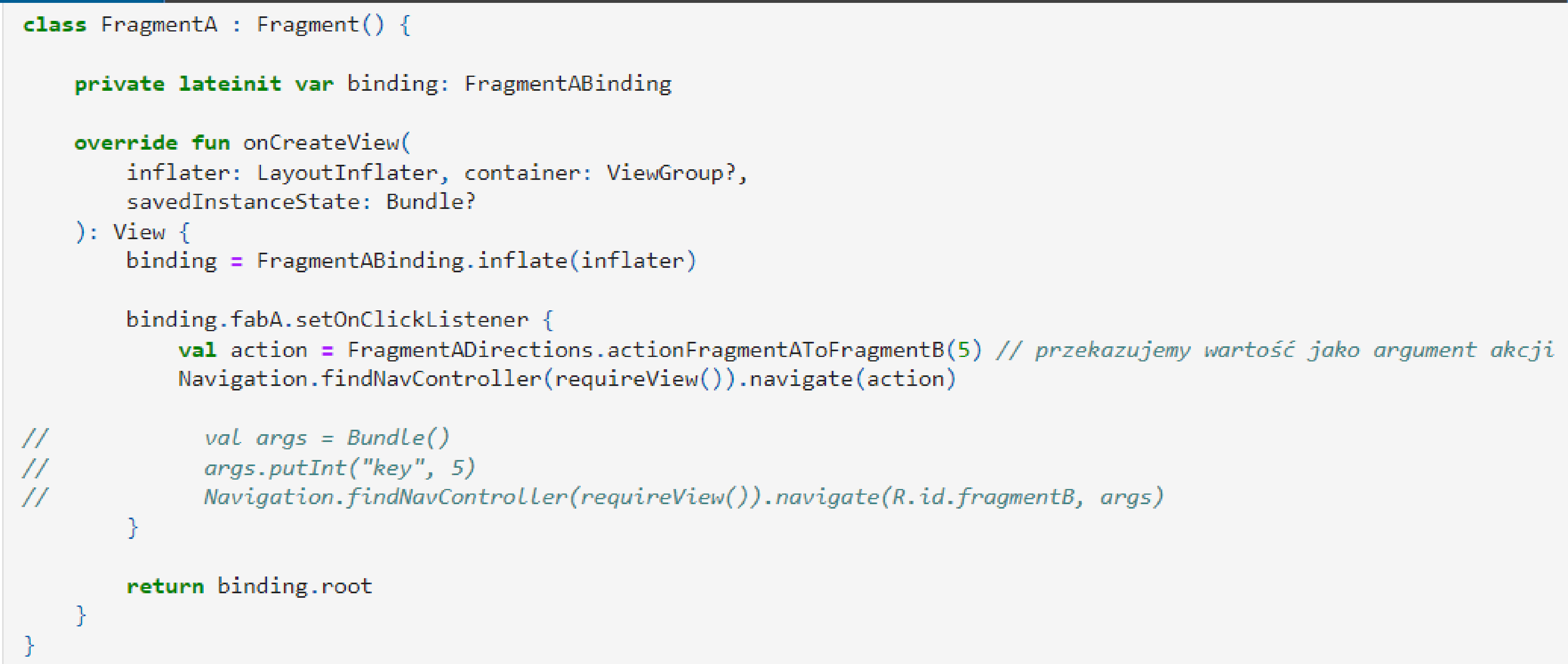
```
class FragmentB : Fragment() {

    private lateinit var binding: FragmentBBinding

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        binding = FragmentBBinding.inflate(inflater)

        binding.textView.text = arguments?.getInt("key").toString() // odebranie danych

        binding.fabB.setOnClickListener {
            val action = FragmentBDirections.actionFragmentBToFragmentA()
            Navigation.findNavController(requireView()).navigate(action)
        }

        return binding.root
    }
}
```

Argumenty



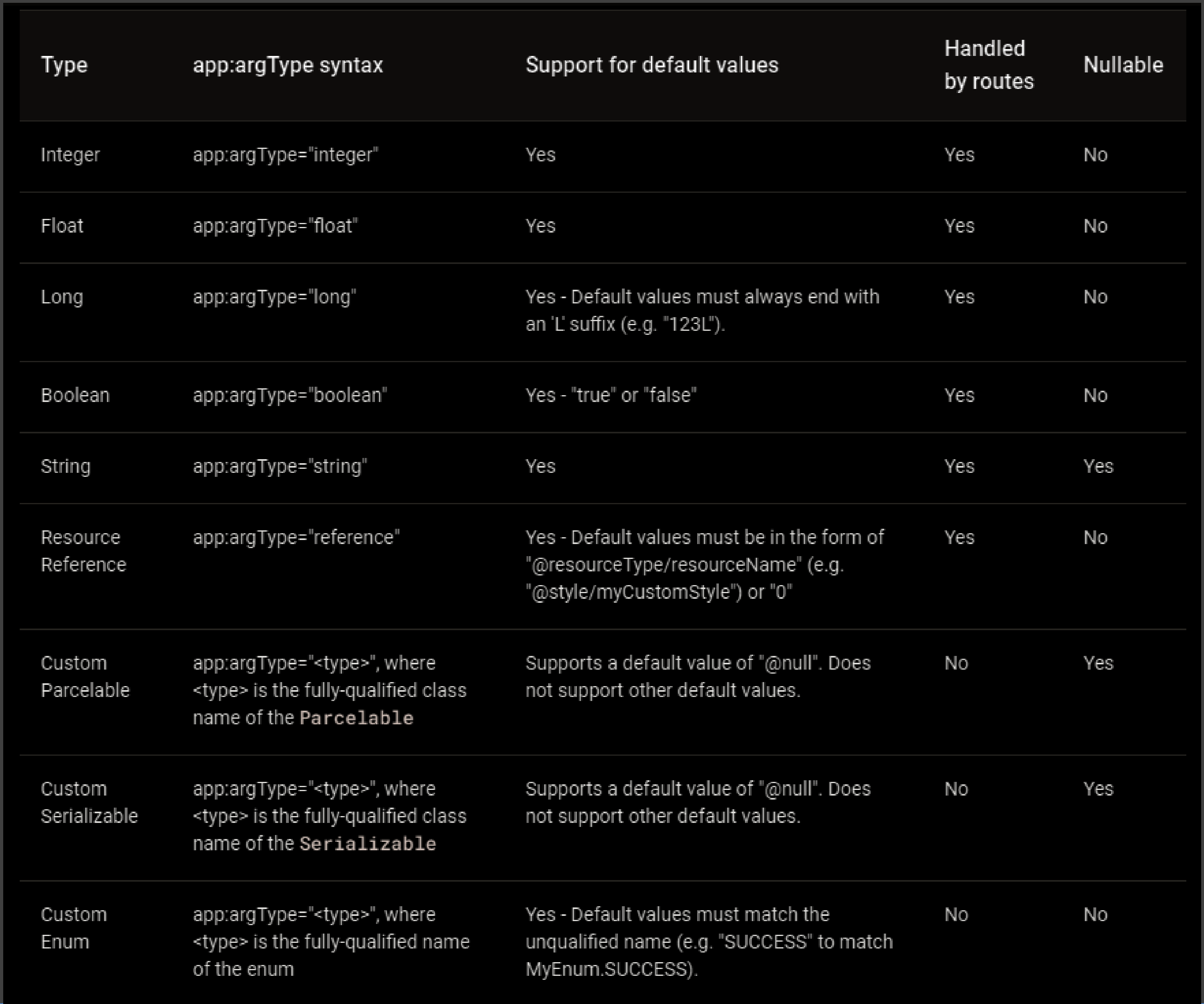
```
<fragment
    android:id="@+id/fragmentA"
    android:name="com.example.a11_jetpacknavigationbasics_kotlin.FragmentA"
    android:label="fragment_a"
    tools:layout="@layout/fragment_a" >
    <action
        android:id="@+id/action_fragmentA_to_fragmentB"
        app:destination="@id/fragmentB" >
        <argument android:name="value"
            app:argType="integer"/>
    </action>
</fragment>
```

Argumenty



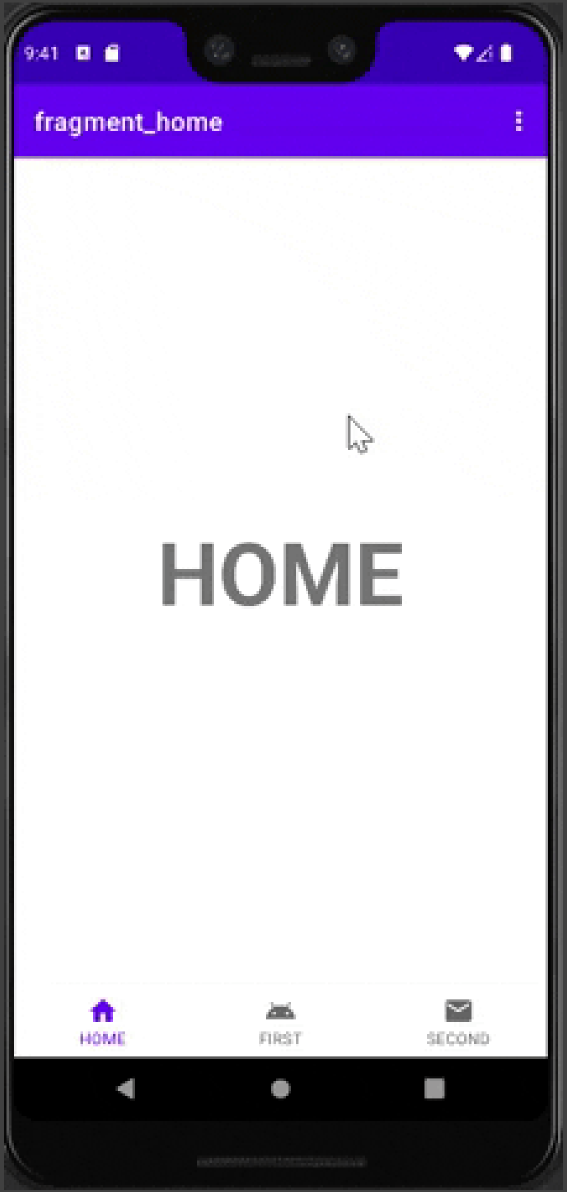
```
class FragmentA : Fragment() {

    private lateinit var binding: FragmentABinding

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        binding = FragmentABinding.inflate(inflater)

        binding.fabA.setOnClickListener {
            val action = FragmentADirections.actionFragmentAToFragmentB(5) // przekazujemy wartość jako argument akcji
            Navigation.findNavController(requireView()).navigate(action)

//            val args = Bundle()
//            args.putInt("key", 5)
//            Navigation.findNavController(requireView()).navigate(R.id.fragmentB, args)
        }

        return binding.root
    }
}
```

Argumenty

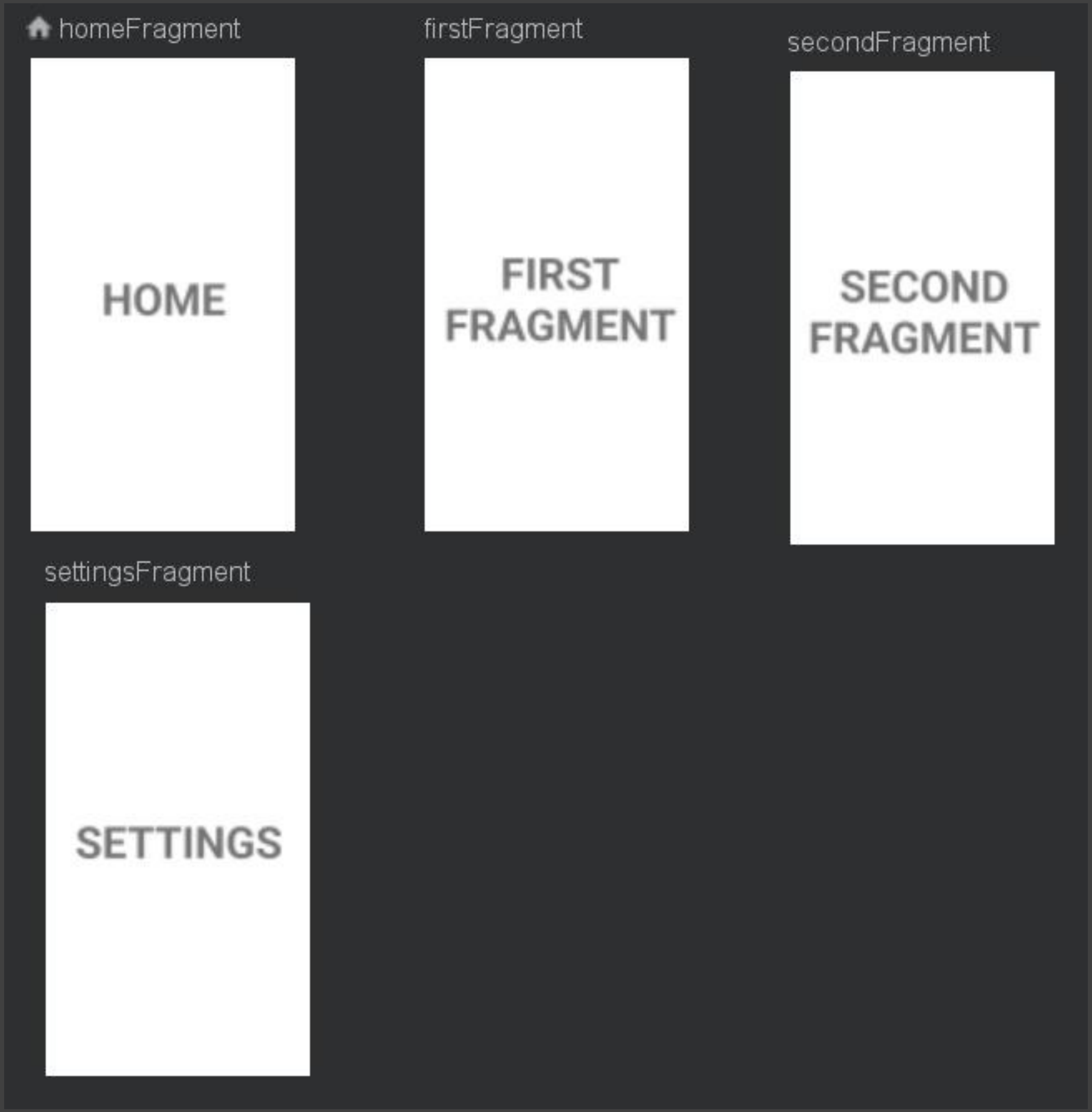


Type	app:argType syntax	Support for default values	Handled by routes	Nullable
Integer	app:argType="integer"	Yes	Yes	No
Float	app:argType="float"	Yes	Yes	No
Long	app:argType="long"	Yes - Default values must always end with an 'L' suffix (e.g. "123L").	Yes	No
Boolean	app:argType="boolean"	Yes - "true" or "false"	Yes	No
String	app:argType="string"	Yes	Yes	Yes
Resource Reference	app:argType="reference"	Yes - Default values must be in the form of "@resourceType/resourceName" (e.g. "@style/myCustomStyle") or "0"	Yes	No
Custom Parcelable	app:argType="<type>", where <type> is the fully-qualified class name of the `Parcelable`	Supports a default value of "@null". Does not support other default values.	No	Yes
Custom Serializable	app:argType="<type>", where <type> is the fully-qualified class name of the `Serializable`	Supports a default value of "@null". Does not support other default values.	No	Yes
Custom Enum	app:argType="<type>", where <type> is the fully-qualified name of the enum	Yes - Default values must match the unqualified name (e.g. "SUCCESS" to match MyEnum.SUCCESS).	No	No

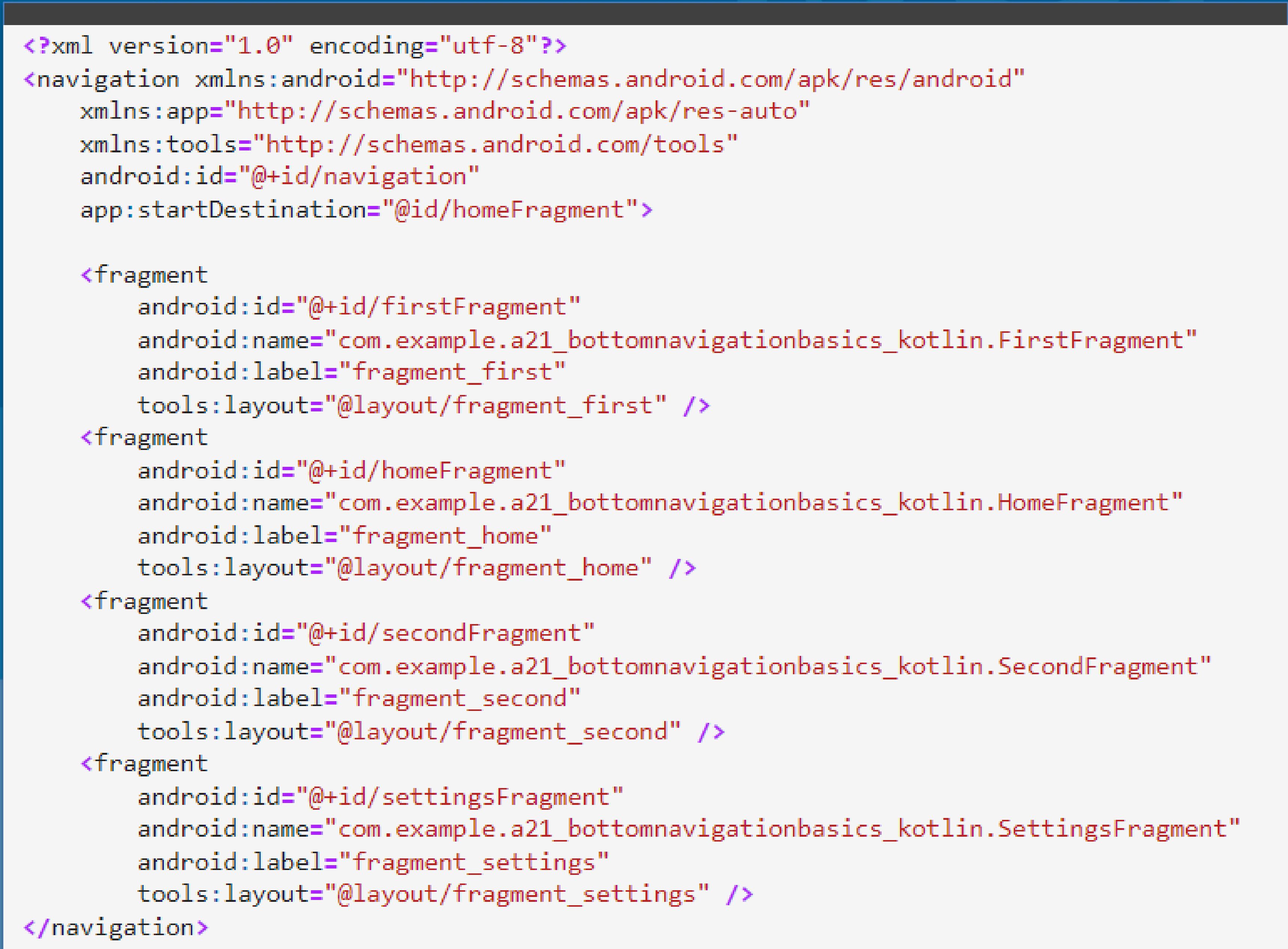
BottomNavigation



BottomNavigation

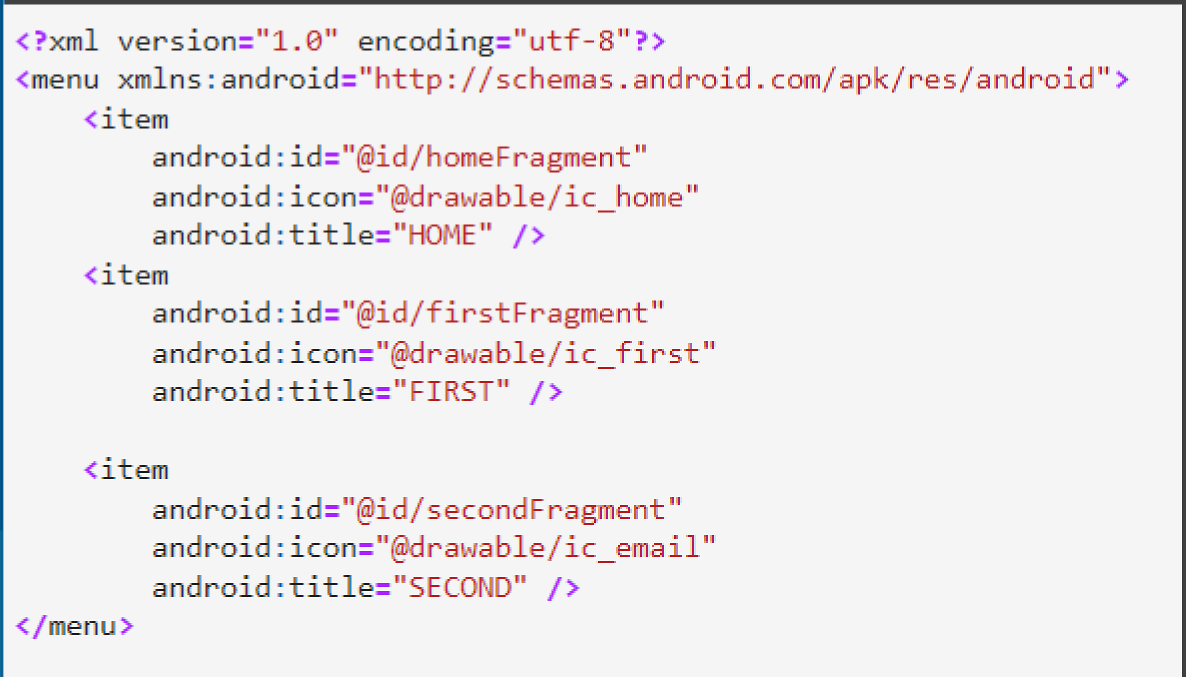


BottomNavigation

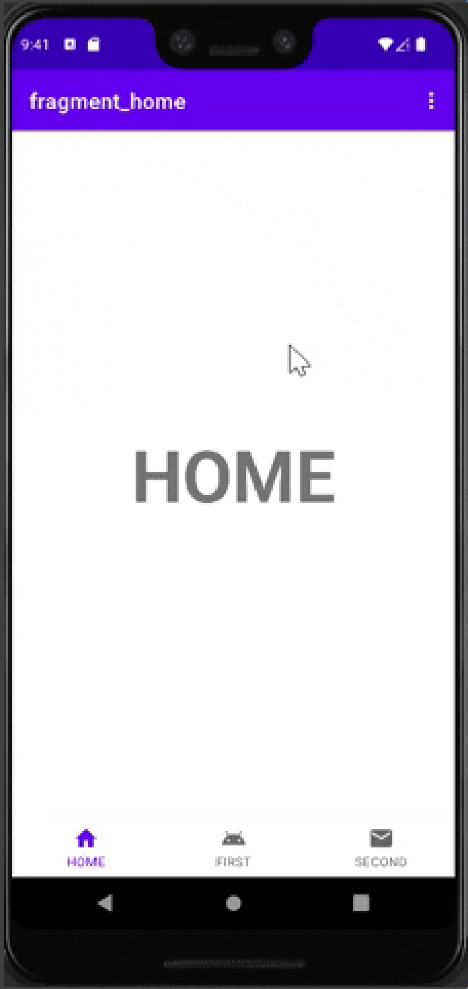


```xml
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/navigation"
    app:startDestination="@id/homeFragment">

    <fragment
        android:id="@+id/firstFragment"
        android:name="com.example.a21_bottomnavigationbasics_kotlin.FirstFragment"
        android:label="fragment_first"
        tools:layout="@layout/fragment_first" />
    <fragment
        android:id="@+id/homeFragment"
        android:name="com.example.a21_bottomnavigationbasics_kotlin.HomeFragment"
        android:label="fragment_home"
        tools:layout="@layout/fragment_home" />
    <fragment
        android:id="@+id/secondFragment"
        android:name="com.example.a21_bottomnavigationbasics_kotlin.SecondFragment"
        android:label="fragment_second"
        tools:layout="@layout/fragment_second" />
    <fragment
        android:id="@+id/settingsFragment"
        android:name="com.example.a21_bottomnavigationbasics_kotlin.SettingsFragment"
        android:label="fragment_settings"
        tools:layout="@layout/fragment_settings" />
</navigation>
```

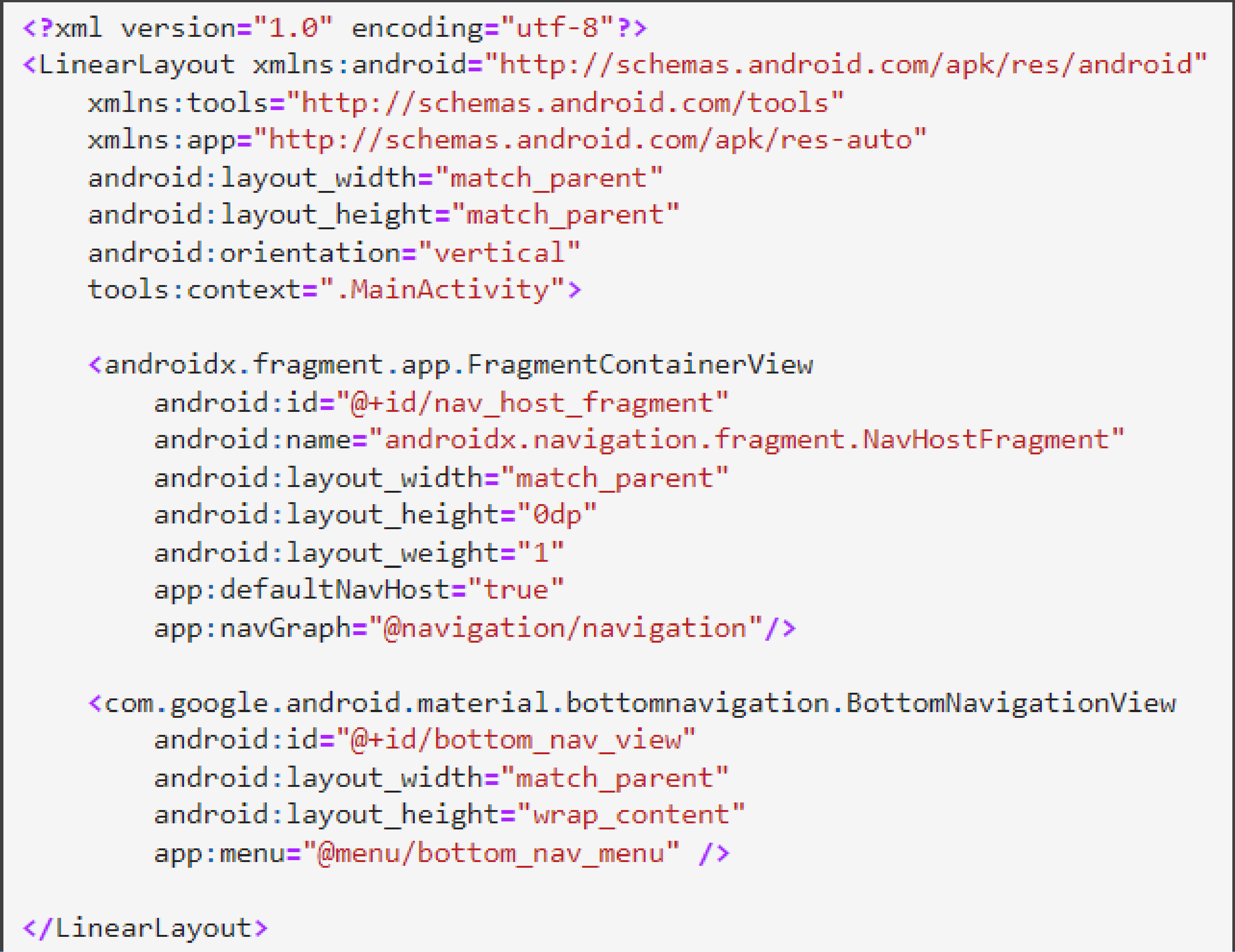
BottomNavigation



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@id/homeFragment"
        android:icon="@drawable/ic_home"
        android:title="HOME" />
    <item
        android:id="@id/firstFragment"
        android:icon="@drawable/ic_first"
        android:title="FIRST" />

    <item
        android:id="@id/secondFragment"
        android:icon="@drawable/ic_email"
        android:title="SECOND" />
</menu>
```

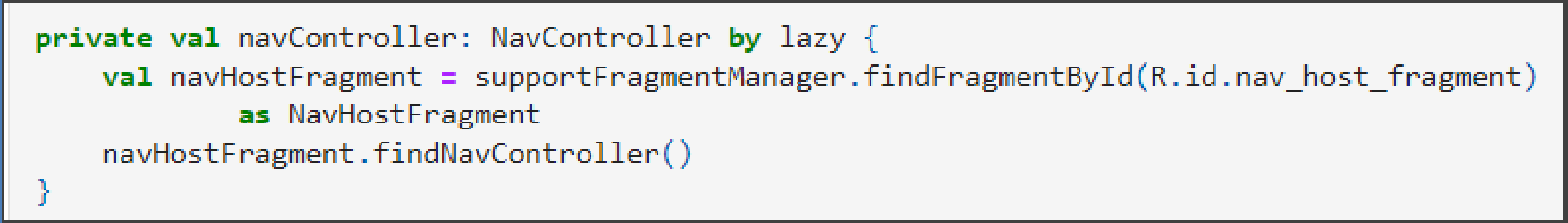


BottomNavigation



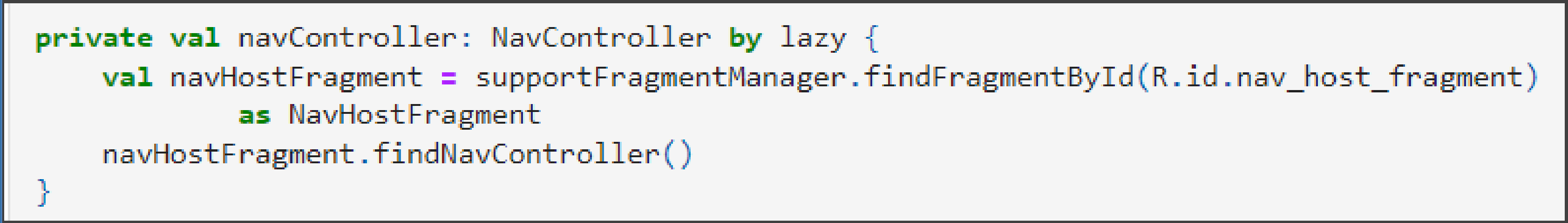
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        app:defaultNavHost="true"
        app:navGraph="@navigation/navigation"/>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_nav_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:menu="@menu/bottom_nav_menu" />

</LinearLayout>
```

BottomNavigation

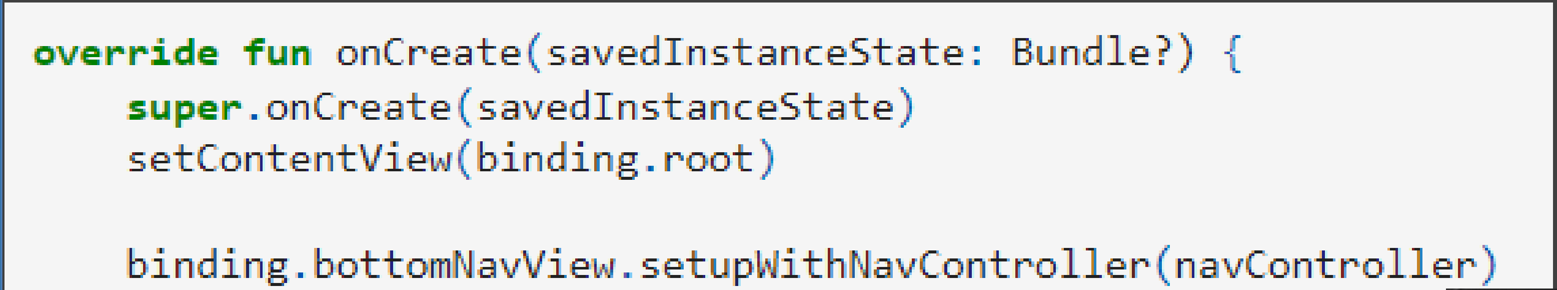


```
private val navController: NavController by lazy {
    val navHostFragment = supportFragmentManager.findFragmentById(R.id.nav_host_fragment)
            as NavHostFragment
    navHostFragment.findNavController()
}
```

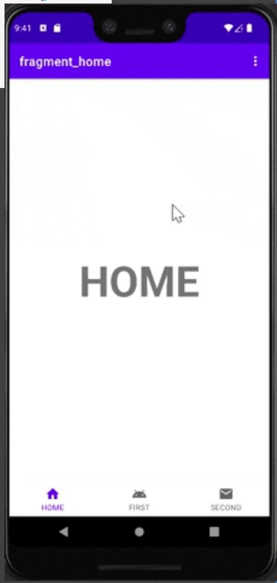
BottomNavigation



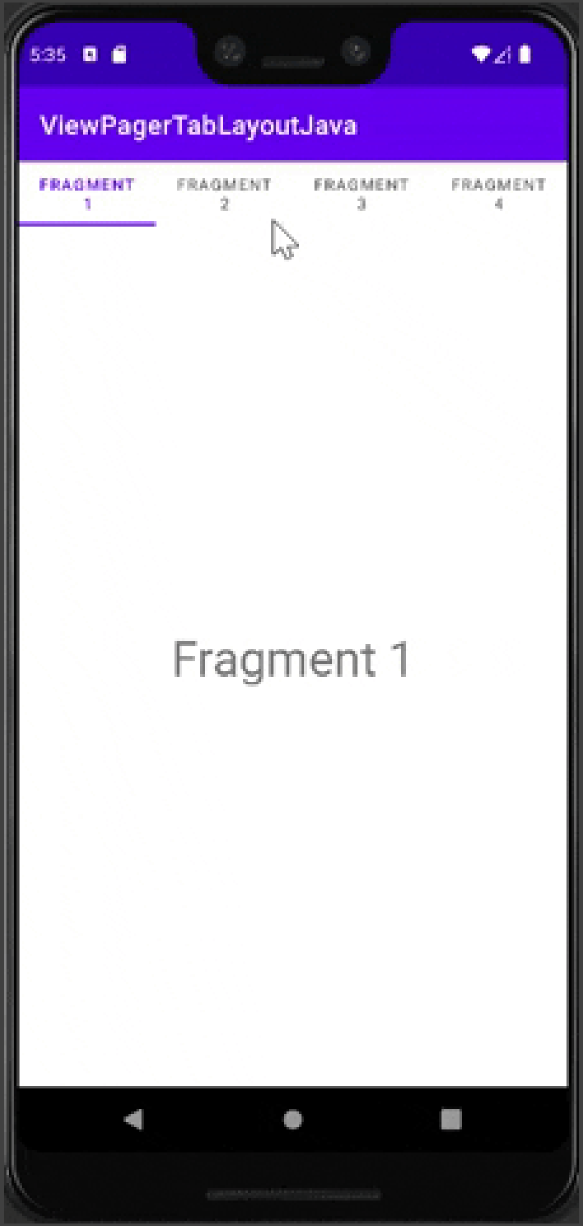
```kotlin
private val navController: NavController by lazy {
    val navHostFragment = supportFragmentManager.findFragmentById(R.id.nav_host_fragment)
            as NavHostFragment
    navHostFragment.findNavController()
}
```



```kotlin
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(binding.root)

    binding.bottomNavView.setupWithNavController(navController)
}
```



ViewPager2 + TabLayout



5:35
ViewPagerTabLayoutJava
FRAGMENT 1
FRAGMENT 2
FRAGMENT 3
FRAGMENT 4
Fragment 1