Practice 10 solunin

Создаем 2 фрагмента и на каждом из них ставим кнопки для передачи информации

Так же нужно дополнительно создать класс отдельным файлом для передави информации с помощью viewModel

DataModel

```
package com.example.myapplication10

import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel

open class DataModel : ViewModel() {
    val message: MutableLiveData<String> by lazy {
        MutableLiveData<String>()
    }
    val message2: MutableLiveData<String> by lazy {
        MutableLiveData<String>()
    }
}
```

MainActivity

```
package com.example.myapplication10

import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.activity.viewModels
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import androidx.fragment.app.Fragment
import com.example.myapplication10.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    private val dataModel: DataModel by viewModels()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) {
v, insets ->
            val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }
        openFragment(BlankOne(),R.id.frame1)
        openFragment(BlankTwo(),R.id.frame2)

        dataModel.message.observe(this, {
            binding.textView.text = it
        })
    }
    private fun openFragment(f: Fragment, idHolder : Int){
        supportFragmentManager
            .beginTransaction()
            .replace(idHolder,f)
            .commit()

    }
}
```

activity_main

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@color/pink">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:textColor="@color/black"/>
    <FrameLayout
        android:id="@+id/frame1"
        android:layout_width="409dp"
        android:layout_height="354dp"
        android:layout_marginStart="1dp"
        android:layout_marginTop="1dp"
        android:layout_marginBottom="1dp"
        app:layout_constraintBottom_toTopOf="@+id/textView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">
    </FrameLayout>
    <FrameLayout
        android:id="@+id/frame2"
        android:layout_width="409dp"
        android:layout_height="354dp"
        android:layout_marginTop="1dp"
        android:layout_marginEnd="1dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView">
    </FrameLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Fragment_blanck_one.xml

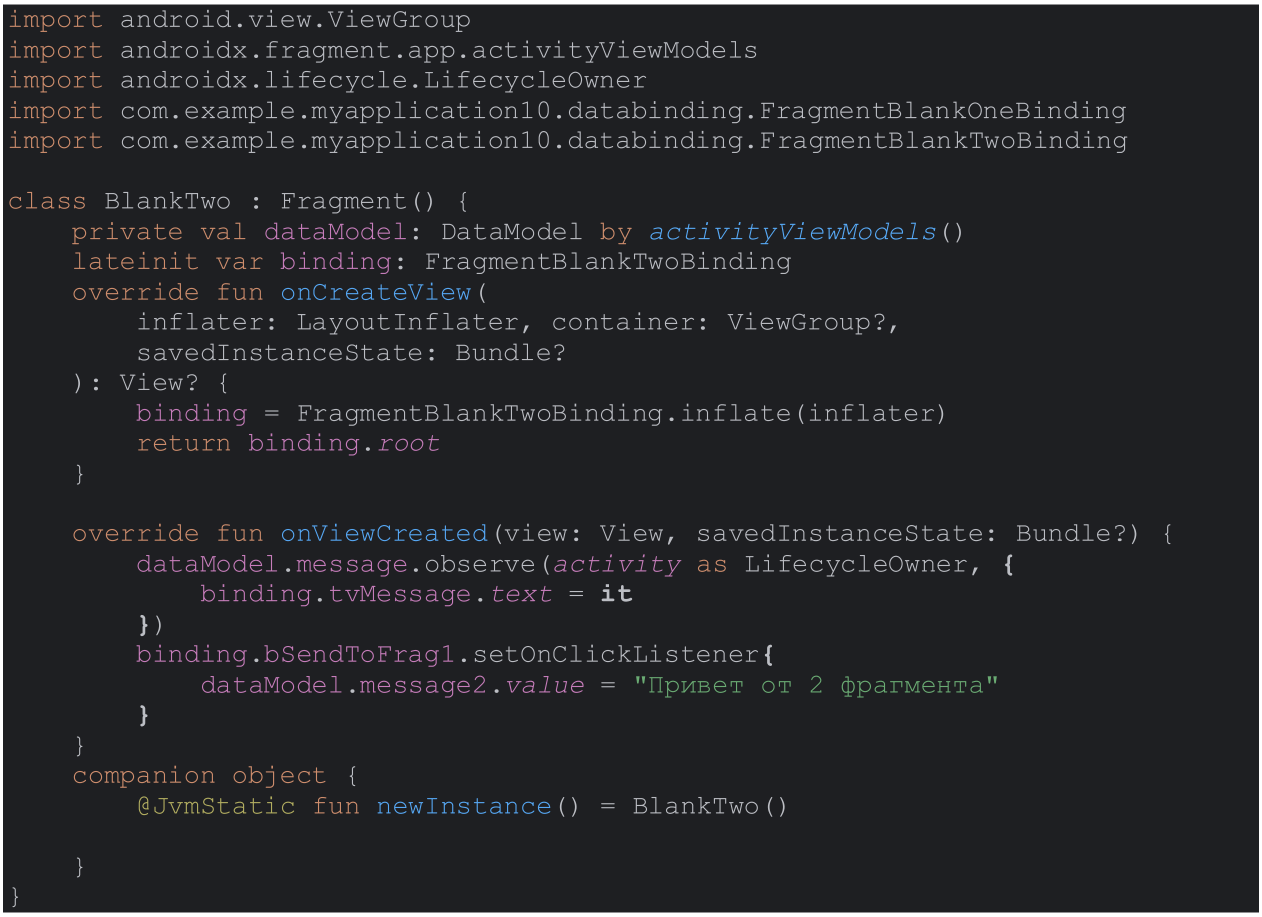
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BlankOne">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="40dp"
        android:gravity="center">
        <TextView
            android:id="@+id/tvMessage2"
            android:layout_width="match_parent"
            android:layout_height="70sp"
            android:text=" Куда передать инфу?"
            android:textSize="20sp"
            android:textStyle="bold"
            android:textColor="@color/black"
            />

        <Button
            android:id="@+id/bSendToFrag2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="На второй фрагмент"
            />

    </LinearLayout>
</FrameLayout>
```

Fragment_blanck_two.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BlankTwo">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="40dp"
        android:gravity="center">
        <TextView
            android:id="@+id/tvMessage"
            android:layout_width="match_parent"
            android:layout_height="70sp"
            android:text=" Куда передать инфу?"
            android:textSize="20sp"
            android:textStyle="bold"
            android:textColor="@color/black"
            />

        <Button
            android:id="@+id/bSendToFrag1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="На Первый фрагмент"
            />

    </LinearLayout>
</FrameLayout>
```

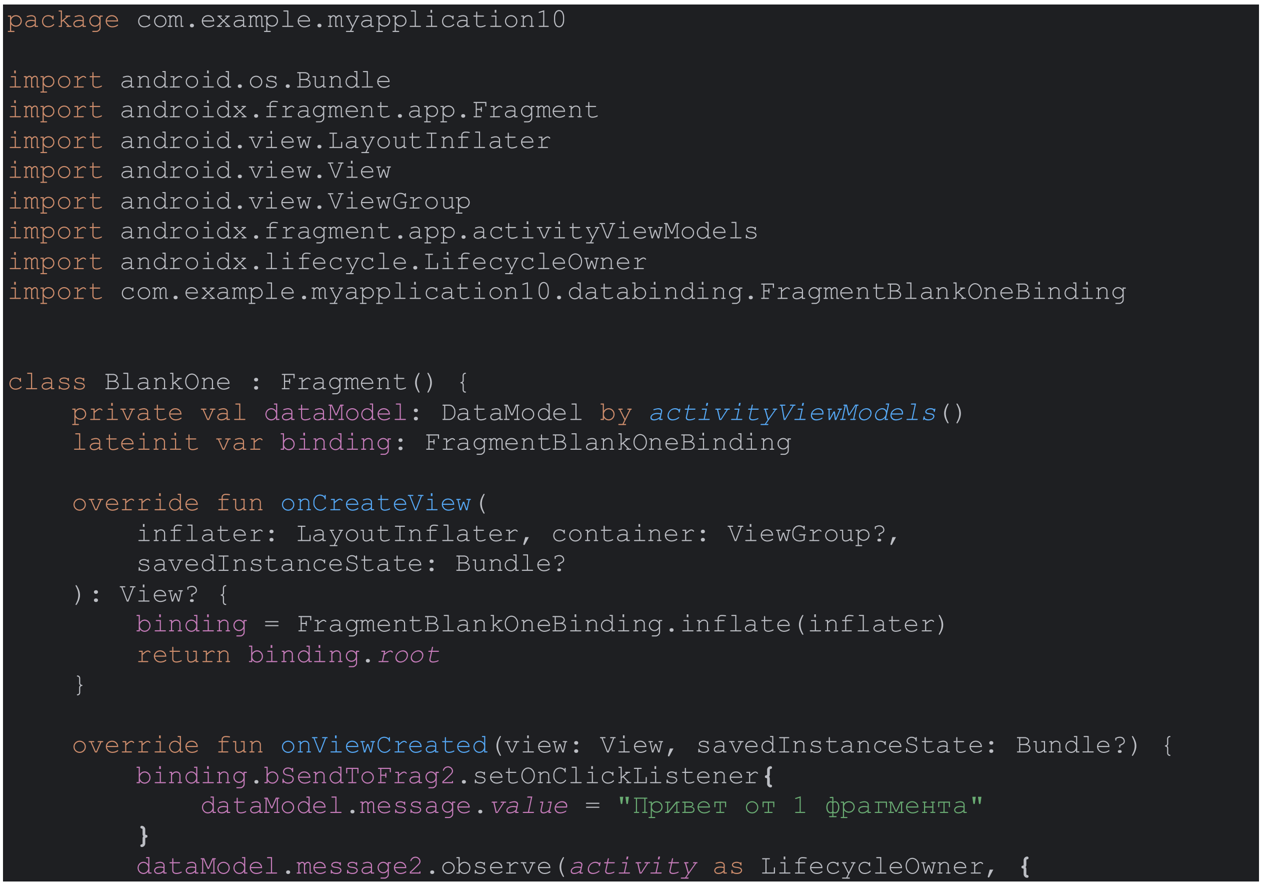
BlankTwo.kt

```
package com.example.myapplication10

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
```



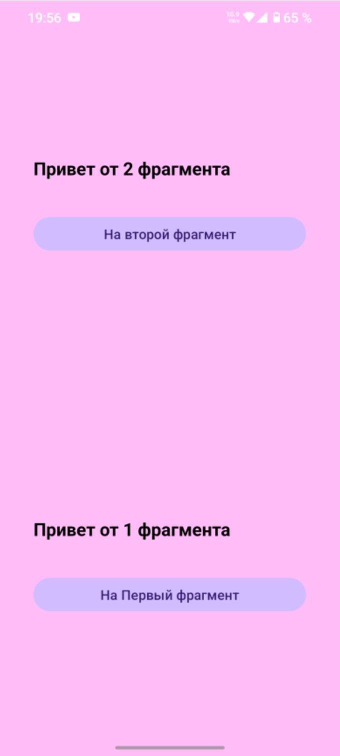
```kotlin
import android.view.ViewGroup
import androidx.fragment.app.activityViewModels
import androidx.lifecycle.LifecycleOwner
import com.example.myapplication10.databinding.FragmentBlankOneBinding
import com.example.myapplication10.databinding.FragmentBlankTwoBinding

class BlankTwo : Fragment() {
    private val dataModel: DataModel by activityViewModels()
    lateinit var binding: FragmentBlankTwoBinding
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        binding = FragmentBlankTwoBinding.inflate(inflater)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        dataModel.message.observe(activity as LifecycleOwner, {
            binding.tvMessage.text = it
        })
        binding.bSendToFrag1.setOnClickListener{
            dataModel.message2.value = "Привет от 2 фрагмента"
        }
    }
    companion object {
        @JvmStatic fun newInstance() = BlankTwo()

    }
}
```

BlankOne.kt



```kotlin
package com.example.myapplication10

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.activityViewModels
import androidx.lifecycle.LifecycleOwner
import com.example.myapplication10.databinding.FragmentBlankOneBinding


class BlankOne : Fragment() {
    private val dataModel: DataModel by activityViewModels()
    lateinit var binding: FragmentBlankOneBinding

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

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        binding.bSendToFrag2.setOnClickListener{
            dataModel.message.value = "Привет от 1 фрагмента"
        }
        dataModel.message2.observe(activity as LifecycleOwner, {
            binding.tvMessage2.text = it
        })
    }

    companion object {
        @JvmStatic fun newInstance() = BlankOne()

    }
}
```

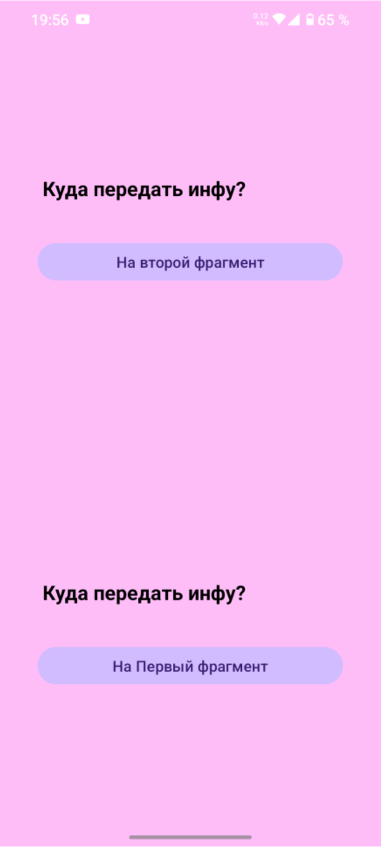
Result



19:56 65 %

Привет от 2 фрагмента

Привет от 1 фрагмента



Куда передать инфу?

Куда передать инфу?