Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе № 3 + 4**

Дисциплина: Проектирование мобильных приложений

Тема: Навигация в приложении. UI-тестирование

Выполнил студент гр. 3530901/90202 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.С. Плетнев

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Н. Кузнецов

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

1. **Цели работы**

* Познакомиться с Google Codelabs и научиться его использовать как способ быстрого изучения новых фреймворков и технологий
* Изучить основные возможности навигации внутри приложения: создание новых activity, navigation graph
* Познакомиться с принципами и получить практические навыки разработки UI тестов для Android приложений

1. **Задачи работы**

* Ознакомление с Jetpack Compose
* Навигация (startActivityForResult)
* Навигация (флаги Intent/атрибуты Activity)
* Навигация (Fragments, Navigation Graph)
* Тестирование навигации

1. **Ход работы (вариант 16 = Navigation Drawer)**

**Задача 1.**

Был пройден codelab по основам Jetpack Compose. Были изучены:

* базовые компоненты Compose
* как реализовать состояния UI (с помощью mutableState)
* базовые контейнеры.
* базовая анимация DP

**Задача 2.**

Исходный код: <https://github.com/timatifey/alarmper/tree/lab3-activities/navigation_demo>

Реализуем навигацию между экранами одного приложения согласно изображению ниже с помощью Activity, Intent и метода startActivityForResult.

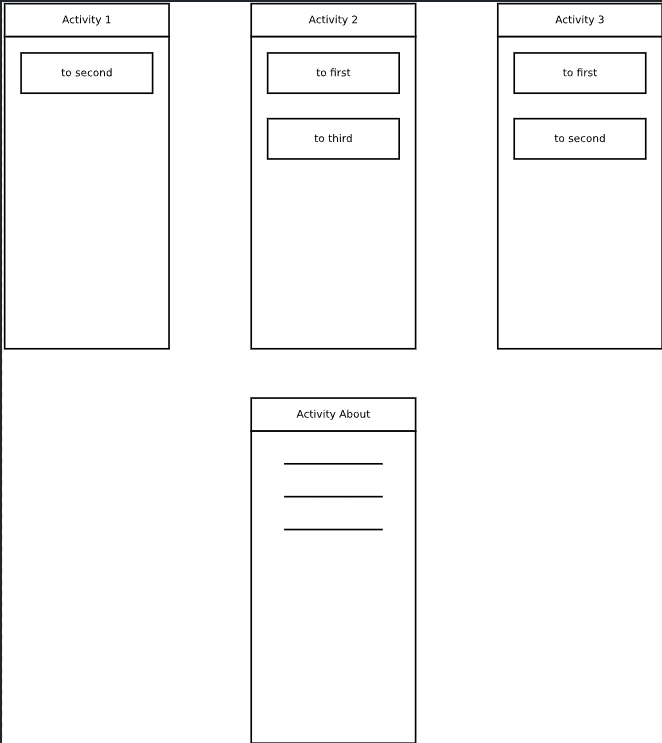


Рис. 2.1 Исходное задание

Для этого создадим 4 Activity и объявим их манифесте (рис. 2) (иначе при попытке открытия Activity будет ActivityNotFoundException).

Изображение выглядит как текст

Автоматически созданное описание

Рис. 2 Application в манифесте проекта

Листинг 2.1 Activity1.kt

|  |
| --- |
| class Activity1 : AppCompatActivity() {   private lateinit var binding: Activity1Binding   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = Activity1Binding.inflate(*layoutInflater*)  setContentView(binding.*root*)   binding.btnToSecond.setOnClickListener **{** navToActivity2()  **}** *setupDrawer*(binding.drawer, binding.drawerNav)  }   private fun navToActivity2() {  startActivity(Intent(this, Activity2::class.*java*))  }  } |

Листинг 2.2 Activity2.kt

|  |
| --- |
| class Activity2 : AppCompatActivity() {   private lateinit var binding: Activity2Binding   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = Activity2Binding.inflate(*layoutInflater*)  setContentView(binding.*root*)   binding.btnToFirst.setOnClickListener **{** navToActivity1()  **}** binding.btnToThird.setOnClickListener **{** navToActivity3()  **}** *setupDrawer*(binding.drawer, binding.drawerNav)  }   override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  if (requestCode == REQUEST\_CODE\_THIRD) {  when (resultCode) {  RESULT\_CODE\_TO\_FIRST -> {  navToActivity1()  }  RESULT\_CODE\_TO\_SECOND -> return  else -> return  }  return  }  super.onActivityResult(requestCode, resultCode, data)  }   private fun navToActivity1() {  finish()  }   private fun navToActivity3() {  startActivityForResult(  Intent(  this,  Activity3::class.*java* ), REQUEST\_CODE\_THIRD  )  }   companion object {  private const val REQUEST\_CODE\_THIRD = 100  const val RESULT\_CODE\_TO\_FIRST = REQUEST\_CODE\_THIRD + 1  const val RESULT\_CODE\_TO\_SECOND = REQUEST\_CODE\_THIRD + 2  } } |

Листинг 2.3 Activity3.kt

|  |
| --- |
| class Activity3 : AppCompatActivity() {   private lateinit var binding: Activity3Binding   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)   binding = Activity3Binding.inflate(*layoutInflater*)  setContentView(binding.*root*)   binding.btnToFirst.setOnClickListener **{** navToActivity1()  **}** binding.btnToSecond.setOnClickListener **{** navToActivity2()  **}** *setupDrawer*(binding.drawer, binding.drawerNav)  }   private fun navToActivity1() {  setResult(Activity2.RESULT\_CODE\_TO\_FIRST)  finish()  }   private fun navToActivity2() {  setResult(Activity2.RESULT\_CODE\_TO\_SECOND)  finish()  }  } |

Листинг 2.4 Utils.kt

|  |
| --- |
| fun Activity.setupDrawer(drawer: DrawerLayout, navigationView: NavigationView) {  navigationView.setNavigationItemSelectedListener **{** if (**it**.*itemId* == R.id.*nav\_about*) {  drawer.closeDrawer(GravityCompat.*START*, true)  startActivity(Intent(this, ActivityAbout::class.*java*))  }  true  **}** } |

Проверим работу со стэком. Откроем приложение и нажмем на кнопку.

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #1: ActivityRecord{1fb871f u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t44}

\* Hist #0: ActivityRecord{2d4e968 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t44}

Откроем ActivityAbout через Drawer.

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #2: ActivityRecord{c477afe u0 dev.timatifey.navigation\_demo/.about.ActivityAbout t44}

\* Hist #1: ActivityRecord{1fb871f u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t44}

\* Hist #0: ActivityRecord{2d4e968 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t44}

Нажмем кнопку назад и перейдем к Activity3.

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #2: ActivityRecord{9636ed1 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity3 t44}

\* Hist #1: ActivityRecord{1fb871f u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t44}

\* Hist #0: ActivityRecord{2d4e968 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t44}

Нажмем кнопку to first.

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #0: ActivityRecord{2d4e968 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t44}

Убеждаемся, что все работает корректно.

**Задача 3.**

Исходный код: <https://github.com/timatifey/alarmper/tree/lab3-activities-flag/navigation_demo>

Переделаем написанный код без использования startActivityForResult.

Листинг 3.1 Activity2.kt

|  |
| --- |
| class Activity2 : AppCompatActivity() {   private lateinit var binding: Activity2Binding   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = Activity2Binding.inflate(*layoutInflater*)  setContentView(binding.*root*)   binding.btnToFirst.setOnClickListener **{** navToActivity1()  **}** binding.btnToThird.setOnClickListener **{** navToActivity3()  **}** *setupDrawer*(binding.drawer, binding.drawerNav)  }   private fun navToActivity1() {  finish()  }   private fun navToActivity3() {  startActivity( Intent(this, Activity3::class.*java*))  } } |

Листинг 3.2 Activity3.kt

|  |
| --- |
| class Activity3 : AppCompatActivity() {   private lateinit var binding: Activity3Binding   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)   binding = Activity3Binding.inflate(*layoutInflater*)  setContentView(binding.*root*)   binding.btnToFirst.setOnClickListener **{** navToActivity1()  **}** binding.btnToSecond.setOnClickListener **{** navToActivity2()  **}** *setupDrawer*(binding.drawer, binding.drawerNav)  }   private fun navToActivity1() {  startActivity(Intent(this, Activity1::class.*java*)  .addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*)  )  }   private fun navToActivity2() {  finish()  }  } |

Проделаем проверку с помощью adb

Откроем приложение и перейдем на Activity 2

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #1: ActivityRecord{cb2c533 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t54}

\* Hist #0: ActivityRecord{6cd51db u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t54}

Перейдем на Activity 3

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #2: ActivityRecord{895d19b u0 dev.timatifey.navigation\_demo/.main\_screens.Activity3 t54}

\* Hist #1: ActivityRecord{cb2c533 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t54}

\* Hist #0: ActivityRecord{6cd51db u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t54}

Перейдем на ActivityAbout через Drawer

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #3: ActivityRecord{991e144 u0 dev.timatifey.navigation\_demo/.about.ActivityAbout t54}

\* Hist #2: ActivityRecord{895d19b u0 dev.timatifey.navigation\_demo/.main\_screens.Activity3 t54}

\* Hist #1: ActivityRecord{cb2c533 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t54}

\* Hist #0: ActivityRecord{6cd51db u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t54}

Вернемся нажатием кнопки назад

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #2: ActivityRecord{895d19b u0 dev.timatifey.navigation\_demo/.main\_screens.Activity3 t54}

\* Hist #1: ActivityRecord{cb2c533 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t54}

\* Hist #0: ActivityRecord{6cd51db u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t54}

Перейдем на Activity 1 с помощью нажатия на кнопку to first

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #0: ActivityRecord{91b2b09 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t54}

Убеждаемся, что все работает корректно.

**Задача 4.**

Добавим на Activity 1 открытие Activity 2 без сохранения истории. То есть, например, после перехода из Activity 2 в Activity 3 и нажатия кнопки назад мы должны сразу переходить на Activity 1.

Листинг 4.1 Activity1.kt

|  |
| --- |
| class Activity1 : AppCompatActivity() {   private lateinit var binding: Activity1Binding   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  binding = Activity1Binding.inflate(*layoutInflater*)  setContentView(binding.*root*)   binding.btnToSecond.setOnClickListener **{** navToActivity2()  **}** binding.btnToSecondNoHistory.setOnClickListener **{** navToActivity2NoHistory()  **}** *setupDrawer*(binding.drawer, binding.drawerNav)  }   private fun navToActivity2() {  startActivity(Intent(this, Activity2::class.*java*))  }   private fun navToActivity2NoHistory() {  startActivity(  Intent(this, Activity2::class.*java*)  .addFlags(Intent.*FLAG\_ACTIVITY\_NO\_HISTORY*)  )  }  } |

Запустим приложение

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #0: ActivityRecord{fb11148 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t56}

Нажмем на кнопку с новым переходом без сохранения истории

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #1: ActivityRecord{b31ee4e u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t56}

\* Hist #0: ActivityRecord{fb11148 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t56}

Перейдем для примера в ActivityAbout

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #2: ActivityRecord{20c3b62 u0 dev.timatifey.navigation\_demo/.about.ActivityAbout t56}

\* Hist #1: ActivityRecord{b31ee4e u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t56 f}

\* Hist #0: ActivityRecord{fb11148 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t56}

mLastNoHistoryActivity: ActivityRecord{b31ee4e u0 dev.timatifey.navigation\_demo/.main\_screens.Activity2 t56 f}

Нажмем кнопку назад – открылась Activity 1

D:\Projects\alarmper\navigation\_demo>adb shell dumpsys activity activities | findStr timatifey | findStr Hist

\* Hist #0: ActivityRecord{fb11148 u0 dev.timatifey.navigation\_demo/.main\_screens.Activity1 t56}

Убеждаемся, что работает корректно.

**Задача 5.**

Исходный код: <https://github.com/timatifey/alarmper/tree/lab3-fragments/navigation_demo>

Реализуем исходную задачу через Fragment и Arch Nav Component от Google.

Листинг 5.1 MainActivity.kt

|  |
| --- |
| class MainActivity : AppCompatActivity() {   private lateinit var binding: ActivityMainBinding  private lateinit var navController: NavController   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)   binding = ActivityMainBinding.inflate(*layoutInflater*)  setContentView(binding.*root*)   val navHostFragment = *supportFragmentManager* .findFragmentById(R.id.*fragmentContainerView*) as NavHostFragment  navController = navHostFragment.*navController   setupActionBarWithNavController*(  navController,  AppBarConfiguration.Builder(navController.*graph*).build()  )   *with*(binding) **{** drawerNav.setNavigationItemSelectedListener **{** if (**it**.*itemId* == R.id.*nav\_about*) {  drawer.closeDrawer(GravityCompat.*START*, true)  navController.navigate(R.id.*action\_to\_activityAbout*)  }  true  **}  }** }   override fun onSupportNavigateUp(): Boolean {  return navController.navigateUp() || super.onSupportNavigateUp()  }  } |

Листинг 5.2 Fragment1.kt

|  |
| --- |
| class Fragment1 : Fragment() {   private var binding: Fragment1Binding? = null   override fun onCreateView(  inflater: LayoutInflater,  container: ViewGroup?,  savedInstanceState: Bundle?  ): View {  binding = Fragment1Binding.inflate(inflater, container, false)  return binding!!.*root* }   override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  super.onViewCreated(view, savedInstanceState)  *with*(binding!!) **{** btnToSecond.setOnClickListener **{** *findNavController*().navigate(R.id.*action\_fragment1\_to\_fragment2*)  **}  }** }   override fun onDestroyView() {  super.onDestroyView()  binding = null  }  } |

Листинг 5.3 Fragment2.kt

|  |
| --- |
| class Fragment2 : Fragment() {   private var binding: Fragment2Binding? = null   override fun onCreateView(  inflater: LayoutInflater,  container: ViewGroup?,  savedInstanceState: Bundle?  ): View {  binding = Fragment2Binding.inflate(inflater, container, false)  return binding!!.*root* }   override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  super.onViewCreated(view, savedInstanceState)  *with*(binding!!) **{** btnToFirst.setOnClickListener **{** *findNavController*().navigate(R.id.*action\_fragment2\_to\_fragment1*)  **}** btnToThird.setOnClickListener **{** *findNavController*().navigate(R.id.*action\_fragment2\_to\_fragment3*)  **}  }** }   override fun onDestroyView() {  super.onDestroyView()  binding = null  }  } |

Листинг 5.4 Fragment3.kt

|  |
| --- |
| class Fragment3: Fragment() {   private var binding: Fragment3Binding? = null   override fun onCreateView(  inflater: LayoutInflater,  container: ViewGroup?,  savedInstanceState: Bundle?  ): View {  binding = Fragment3Binding.inflate(inflater, container, false)  return binding!!.*root* }   override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  super.onViewCreated(view, savedInstanceState)  *with*(binding!!) **{** btnToFirst.setOnClickListener **{** *findNavController*().navigate(R.id.*action\_fragment3\_to\_fragment1*)  **}** btnToSecond.setOnClickListener **{** *findNavController*().navigate(R.id.*action\_fragment3\_to\_fragment2*)  **}  }** }   override fun onDestroyView() {  super.onDestroyView()  binding = null  } } |

Листинг 5.5 app\_graph.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/app\_graph"  app:startDestination="@id/fragment1">   <fragment  android:id="@+id/fragment1"  android:name="dev.timatifey.navigation\_demo.main\_screens.Fragment1"  android:label="Fragment1">  <action  android:id="@+id/action\_fragment1\_to\_fragment2"  app:destination="@id/fragment2" />  </fragment>  <fragment  android:id="@+id/fragment2"  android:name="dev.timatifey.navigation\_demo.main\_screens.Fragment2"  android:label="Fragment2">  <action  android:id="@+id/action\_fragment2\_to\_fragment3"  app:destination="@id/fragment3" />  <action  android:id="@+id/action\_fragment2\_to\_fragment1"  app:destination="@id/fragment1"  app:popUpTo="@id/fragment1"  app:popUpToInclusive="true" />  </fragment>  <fragment  android:id="@+id/fragment3"  android:name="dev.timatifey.navigation\_demo.main\_screens.Fragment3"  android:label="Fragment3">  <action  android:id="@+id/action\_fragment3\_to\_fragment1"  app:destination="@id/fragment1"  app:popUpTo="@id/fragment1"  app:popUpToInclusive="true" />  <action  android:id="@+id/action\_fragment3\_to\_fragment2"  app:destination="@id/fragment2"  app:popUpTo="@id/fragment2"  app:popUpToInclusive="true" />  </fragment>  <activity  android:id="@+id/activityAbout"  android:name="dev.timatifey.navigation\_demo.about.ActivityAbout"  android:label="activity\_about"  tools:layout="@layout/activity\_about" />  <action  android:id="@+id/action\_to\_activityAbout"  app:destination="@id/activityAbout" /> </navigation> |

**Задача 6.**

Проведем тестирование написанного кода в 5-ой задаче, предварительно проведя изменения в проекте:

* изменим манифест – добавим атрибуты parentActivtyName для About (тем самым добавив up Navigation) и launchMode для Main
* изменим id View элементов, чтобы те совпадали с тестами
* добавим отсутствующие id для тестов (например id корневых элементов layout’ов)

Ссылка на исходный код написанных тестов: <https://github.com/timatifey/alarmper/tree/lab4-my_implement/navigation_demo/app/src/androidTest/java/dev/timatifey/navigation_demo>

Ссылка на измененный код: <https://github.com/timatifey/alarmper/tree/lab4-my_implement/navigation_demo/app/src/main>

1. **Статистика (работа выполнялось с параллельным написанием отчета)**

* Задача 1: 15 мин (знакомство с JetPack Compose было еще до лабораторной работы)
* Задача 2: 1 ч 30 мин (не мог вспомнить как быстро и правильно использовать Drawer)
* Задача 3: 45 мин
* Задача 4: 10 мин
* Задача 5: 30 мин
* Задача 6: ~6 ч (изучение основ нового фреймворка тестирования Kaspresso + изучение лучших практик автоматизированного UI тестирования)

**Итого:** 9 ч 10 мин

1. **Вывод**

В ходе работы были продемонстрированы способы навигации в Android приложении. Была изучена навигация между Activity и управление стеком с помощью Intent-флагов. Использовали activityForResult. Также была решена исходная задача с помощью Google Nav Component и фрагментов. Дополнительно ко всему были написаны UI тесты к написанному коду, с помощью фреймворка Kaspresso. Были изучены некоторые лучшие практики UI-тестирования (PageObject, ScenarioRule)