

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе № 3

Дисциплина: Проектирование мобильных приложений

Тема: Lifesycle компоненты. Навигация в приложении

Выполнил студент гр. 3530901/90201 _____ Д.Е. Бакин
(подпись)

Принял преподаватель _____ А.Н. Кузнецов
(подпись)

“ ____ ” _____ 2021 г.

Санкт-Петербург
2021

Репозиторий GitHub:

https://github.com/donebd/Android_spbstu2021/tree/main/labs/lab3

1. Цели

- 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

2. Задачи

- Ознакомление с Jetpack Compose и выполнение Jetpack Compose basics
- Навигация (startActivityForResult)
- Навигация (флаги Intent/атрибуты Activity)
- Навигация (Fragments, Navigation Graph)

3. Ход работы

3.1. Задача 1.

Был изучен интересный инструмент Google Codelabs для быстрого и эффективного ихучения новых инструментов и фреймворков на примере ознокомления с технологией Jetpack Compose.

В Jetpack Compose basics codelab были изучены:

- Декларативный принцип описания верстки
- Базовые компоненты Compose
- Реализация состояний UI (mutableState)
- Базовые контейнеры (Row, Column, etc)
- Анимция (animateDpAsState)
- Инструменты для работы с Compose в AndroidStudio

3.2. Задача 2.

Реализуем навигацию между экранами одного приложения согласно сехмы представленной ниже с помощью Activity, Intent, методов startActivity и startActivityForResult. Также Activity About должна быть доступна из любой другой Activity через Options Menu (Вариант 3).



Рис. 1 Задание по навигации.

Для начала объявим в манифесте эти 4 активности, иначе будем получать `ActivityNotFoundException` при переходе на любой из них.

```
<activity
    android:name=".ActivityAbout"
    android:label="About" />
<activity
    android:name=".Activity3"
    android:label="Activity 3" />
<activity
    android:name=".Activity2"
    android:label="Activity 2" />
<activity
    android:name=".Activity1"
    android:exported="true"
    android:label="Activity 1">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Рис. 1 Application в манифесте проекта.

Из первого активности при помощи стандартного метода `startActivity` мы записываем первое активности в стек и переходим на второе активности.

Для перехода в activity about, используем option menu, для этого переопределяем два метода `onCreateOptionsMenu` и `onOptionsItemSelected`.

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.button.setOnClickListener {
            navToActivity2()
        }
    }

    private fun navToActivity2() {
        startActivity(Intent(this, Activity2::class.java))
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        val inflater: MenuInflater = menuInflater
        inflater.inflate(R.menu.option_menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return if (item.itemId == R.id.about_item) {
            startActivity(Intent(this, ActivityAbout::class.java))
            true
        } else {
            super.onOptionsItemSelected(item)
        }
    }
}
```

Во втором активити нам нужно перейти либо в первое активити или в третье. Для перехода в первое активити используем метод `finish` – завершение работы текущего Activity. Для перехода в активити `about` используем аналогичные методы из предыдущего активити.

Для перехода в третье активити используем пару методов `startActivityResult` и `onActivityResult`. Т.к. из третьего активити мы можем перейти в первое или во второе активити, соответственно можем вернуть код возврата в первое или во второе активити. В методе `onActivityResult` определяем остаться ли в текущем активити или перейти в первое.

Листинг 2.2. Activity2.kt

```
class Activity2 : AppCompatActivity() {

    private val REQUEST_TO_THIRD = 226
    private val REQUEST_TO_SECOND = 227
    private val REQUEST_TO_FIRST = 228

    private lateinit var binding: Activity2Binding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = Activity2Binding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.button.setOnClickListener {
            navToActivity1()
        }

        binding.button2.setOnClickListener {
            navToActivity3()
        }

    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        if (requestCode == REQUEST_TO_THIRD && resultCode == REQUEST_TO_FIRST) {
            navToActivity1()
        }
        super.onActivityResult(requestCode, resultCode, data)
    }

    private fun navToActivity1() {
        finish()
    }

    private fun navToActivity3() {
        startActivity(
            Intent(this, Activity3::class.java), REQUEST_TO_THIRD
        )
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean { //from activity1 }

    override fun onOptionsItemSelected(item: MenuItem): Boolean { //activity1 }

}
```

В третьем активити нам нужно либо вернуть код возврата во второе активити или в первое при помощи метода `setResult`. Логика возврата в первое активити обрабатывается во втором активити. Активити `about` обрабатывается аналогично.

Листинг 2.3. Activity3.kt

```
class Activity3 : AppCompatActivity() {

    private val REQUEST_TO_SECOND = 227
    private val REQUEST_TO_FIRST = 228

    private lateinit var binding: Activity3Binding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = Activity3Binding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.button.setOnClickListener {
            navToActivity1()
        }

        binding.button2.setOnClickListener {
            navToActivity2()
        }

    }

    private fun navToActivity1() {
        setResult(REQUEST_TO_FIRST)
        finish()
    }

    private fun navToActivity2() {
        setResult(REQUEST_TO_SECOND)
        finish()
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        //code from activity1
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        //code from activity1
    }

}
```

Посмотрим работу стека, в течение работы приложения.

Открываем приложение, переходим в активити `about`:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_2>adb shell dumpsys activity activities | findStr
Hist | findStr lab3_2
* Hist #1: ActivityRecord{a7a2dc u0 com.example.lab3_2/.ActivityAbout t81}
* Hist #0: ActivityRecord{87dd2c0 u0 com.example.lab3_2/.Activity1 t81}
```

Нажимаем кнопку назад и поочередно переходим в третье активити:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_2>adb shell dumpsys activity activities | findStr
Hist | findStr lab3_2
* Hist #2: ActivityRecord{d8af6d3 u0 com.example.lab3_2/.Activity3 t81}
```

```
* Hist #1: ActivityRecord{1f53c55 u0 com.example.lab3_2/.Activity2 t81}
```

```
* Hist #0: ActivityRecord{87dd2c0 u0 com.example.lab3_2/.Activity1 t81}
```

Перейдем в activity about:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_2>adb shell dumpsys activity activities | findStr  
Hist | findStr lab3_2
```

```
* Hist #3: ActivityRecord{f1cb9d4 u0 com.example.lab3_2/.ActivityAbout t81}
```

```
* Hist #2: ActivityRecord{d8af6d3 u0 com.example.lab3_2/.Activity3 t81}
```

```
* Hist #1: ActivityRecord{1f53c55 u0 com.example.lab3_2/.Activity2 t81}
```

```
* Hist #0: ActivityRecord{87dd2c0 u0 com.example.lab3_2/.Activity1 t81}
```

Нажмем кнопку назад и перейдем через кнопку в первое активити:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_2>adb shell dumpsys activity activities | findStr  
Hist | findStr lab3_2
```

```
* Hist #0: ActivityRecord{87dd2c0 u0 com.example.lab3_2/.Activity1 t81}
```

3.3. Задача 3.

Переделаем предыдущий проект без использования `startActivityForResult`, с помощью флагов на `Intent`. Ипользуем флаг `FLAG_ACTIVITY_CLEAR_TOP`. Он позволяет переходить к `Activity`, если она уже лежит в стэке.

Листинг 3.2. `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.button.setOnClickListener {  
            navToActivity1()  
        }  
  
        binding.button2.setOnClickListener {  
            navToActivity3()  
        }  
  
    }  
  
    private fun navToActivity1() {  
        finish()  
    }  
  
    private fun navToActivity3() {  
        startActivity(Intent(this, Activity3::class.java))  
    }  
  
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {  
        //code from activity1  
    }  
  
    override fun onOptionsItemSelected(item: MenuItem): Boolean {  
        //code from activity1  
    }  
}
```

Листинг 3.3. 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.button.setOnClickListener {
            navToActivity1()
        }

        binding.button2.setOnClickListener {
            navToActivity2()
        }

    }

    private fun navToActivity1() {
        startActivity(Intent(this, Activity1::class.java)
            .addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP))
    }

    private fun navToActivity2() {
        finish()
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        //code from activity1
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        //code from activity1
    }

}
```

Посмотрим работу стека, в течение работы приложения.

Открываем приложение, переходим в активити about:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_3>adb shell dumpsys activity activities | findStr
Hist | findStr lab3_3
```

```
* Hist #1: ActivityRecord{952761b u0 com.example.lab3_3/.ActivityAbout t82}
* Hist #0: ActivityRecord{4ebb035 u0 com.example.lab3_3/.Activity1 t82}
```

Нажимаем кнопку назад и поочередно переходим в третье активити:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_3>adb shell dumpsys activity activities | findStr
Hist | findStr lab3_3
```

```
* Hist #2: ActivityRecord{505cb2c u0 com.example.lab3_3/.Activity3 t82}
* Hist #1: ActivityRecord{71d3501 u0 com.example.lab3_3/.Activity2 t82}
* Hist #0: ActivityRecord{4ebb035 u0 com.example.lab3_3/.Activity1 t82}
```

Перейдем в activity about:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_3>adb shell dumpsys activity activities | findStr
Hist | findStr lab3_3
```

```
* Hist #3: ActivityRecord{d15aca9 u0 com.example.lab3_3/.ActivityAbout t82}
* Hist #2: ActivityRecord{505cb2c u0 com.example.lab3_3/.Activity3 t82}
* Hist #1: ActivityRecord{71d3501 u0 com.example.lab3_3/.Activity2 t82}
```



```
* Hist #0: ActivityRecord{4ebb035 u0 com.example.lab3_3/.Activity1 t82}
```

Нажмем кнопку назад и перейдем через кнопку в первое активити:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_3>adb shell dumpsys activity activities | findStr  
Hist | findStr lab3_3
```

```
* Hist #0: ActivityRecord{cd61219 u0 com.example.lab3_3/.Activity1 t82}
```

Поведение приложение аналогичное предыдущему пункту.

3.4. Задача 4.

Дополним наше первое активити кнопкой с переходом на активити 2 с флагом `FLAG_ACTIVITY_NO_HISTORY`, который позволяет не сохранять в стэке активити на которое переходим (если с него будут переходы).

Таким образом, мы сможем вернуться к первому активити из любого прямого перехода из активити 2.

Листинг 3.1 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.button.setOnClickListener {  
            navToActivity2()  
        }  
  
        binding.buttonNoHistory.setOnClickListener {  
            navToActivity2NoHistory()  
        }  
    }  
  
    private fun navToActivity2() {  
        startActivity(Intent(this, Activity2::class.java))  
    }  
  
    private fun navToActivity2NoHistory() {  
        startActivity(Intent(this,  
Activity2::class.java).addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY))  
    }  
}
```

Проверим поведение стэка.

Перейдем в активити 2 с флагом, и из него перейдем в активити about:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_3>adb shell dumpsys activity activities | findStr  
Hist | findStr lab3_3
```

```
* Hist #0: ActivityRecord{f9a844f u0 com.example.lab3_3/.Activity1 t87}
```

Перейдем в активити 2 с флагом, из него перейдем в активити 3, а оттуда в активити about:

```
C:\Users\Daniil\IdeaProjects\Android_spbstu2021\labs\lab3\lab3_3>adb shell dumpsys activity activities | findStr  
Hist | findStr lab3_3
```

```
* Hist #2: ActivityRecord{511343c u0 com.example.lab3_3/.ActivityAbout t87}
```

```
* Hist #1: ActivityRecord{985f4a4 u0 com.example.lab3_3/.Activity3 t87}
```

* Hist #0: ActivityRecord{f9a844f u0 com.example.lab3_3/.Activity1 t87}

Видим, как флаг успешно позволяет не сохранять в стеке новое активити.

3.5. Задача 5.

4. Вывод

В работе я познакомился с жизненным циклом Activity. Поработал с его callback-методами, выполняя различные пункты лабораторной работы. Узнал о различных переходах состояний Activity в различных штатных/нештатных ситуациях.

Также познакомился с таким понятием как alternative resources, как они используются, где применяются, как выбираются. Прочтение документации позволило выполнить задачи лабораторной работы.

На выполнение работы ушло в общей сумме часов 6.