Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №3**

**Дисциплина**: Проектирование мобильных приложений

**Тема**: Lifecycle компоненты. Навигация в приложении.

Выполнил студент гр. 3530901/80202 Сухачев Н.А

(подпись)

Преподаватель Алексюк А.О

(подпись)

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Санкт-Петербург

2020

**1. Цели** Ознакомиться с методом обработки жизненного цикла activity/fragment при помощи Lifecycle-Aware компонентов

Изучить основные возможности навигации внутри приложения: создание новых activity, navigationgraph

Вариант 14.

**2. Программа работы**

**2.1. Задача 1. Обработка жизненного цикла с помощью Lifecycle-Aware компонентов**

Ознакомьтесь с Lifecycle-AwareComponents по документации и выполните codelabs.

После выполнения codelabs я узнал что:  
ViewModel - предоставляет способ создавать и извлекать объекты, привязанные к определенному жизненному циклу.

LifecycleOwner - это интерфейс, реализованный классами AppCompatActivity и Fragment. Мы можем подписать другие компоненты на объекты-владельцы, которые реализуют данный интерфейс, чтобы наблюдать за изменениями в жизненном цикле владельца.

LiveData - позволяет наблюдать за изменениями данных в нескольких компонентах приложения, не создавая явных жёстких путей зависимости между ними. Он учитывает сложные жизненные циклы компонентов приложения, включая действия, фрагменты, службы или любой LifecycleOwner, определённый в нашем приложении. LiveData управляет подписками наблюдателей, приостанавливая подписки на остановленные объекты LifecycleOwner и отменяя подписки на завершённые объекты LifecycleOwner.

**2.2. Задача 2. Навигация (startActivityForResult)**

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

Во всех вариантах Activity “About” должна быть доступна из любой другой Activity одним из указанных способов:

Вариант 14: Navigation Drawer

Задача решена с применением Navigation Drawer:

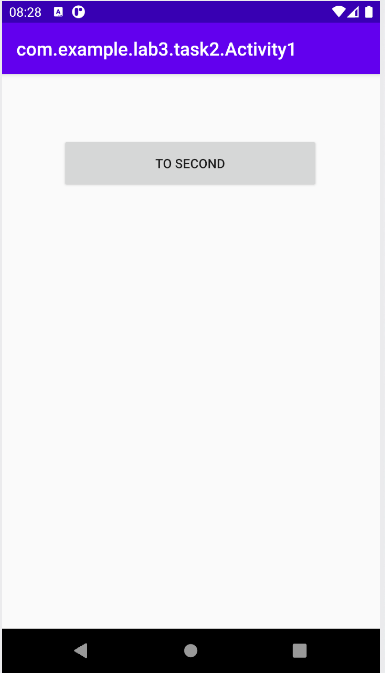
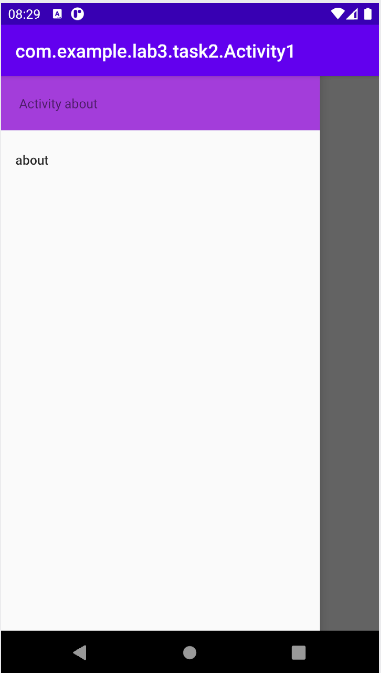


Рис.1. Вид activity с Navigation Drawer

Для правильной (в рамках задания, не более 4 activity в backstack) навигации в первом задание были использованы следующие методы:

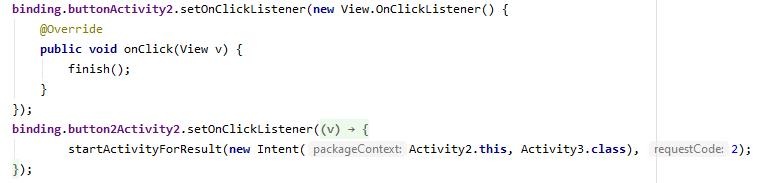
* startActivity, startActivityForResult
* setResult, onActivityResult
* finish

**В Activity1:**

****

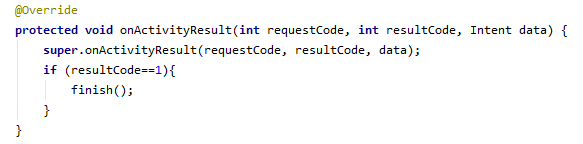
По нажатию на кнопку запускаем Activity2.

**В Activity2:**

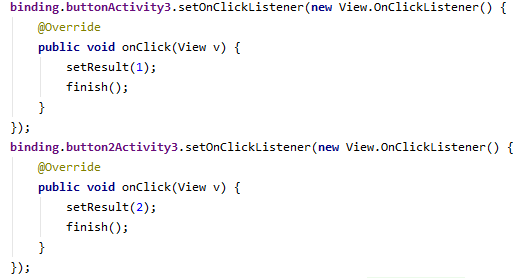
****

Для перехода в Activity1 завершаем Activity2 методом finish.Для перехода в Activity3 используем startActivityForResult().

Также изменяем метод onActivityResult() , для того чтобы завершать Activity2, когда из Activity3 переходим в Activity1.



**В Activity3:**

****

Для перехода из Activity3 в Activity1 или Activity2 устанавливаем результат 1 или 2 соответственно и вызываем finish(), в Activity2 уже есть метод вызывающий finish если был переход в Activity1 .

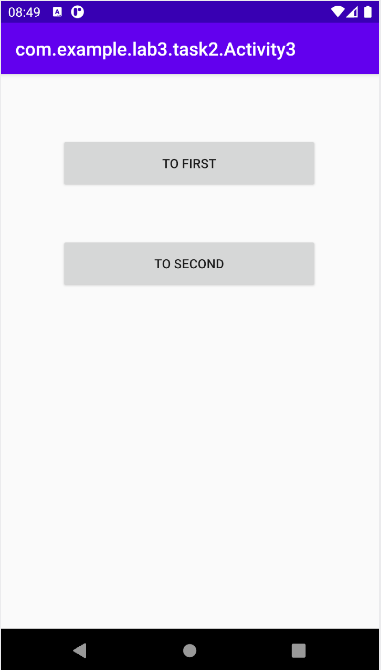
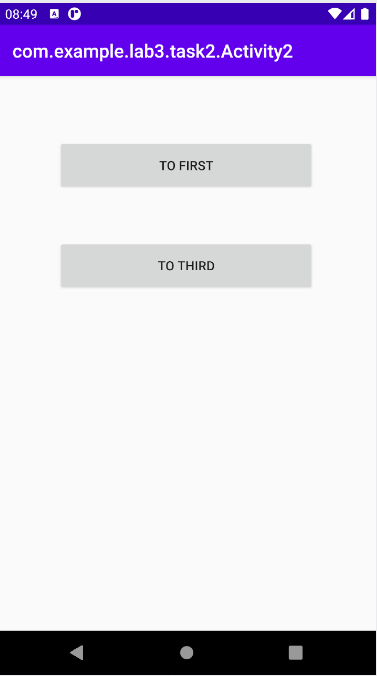
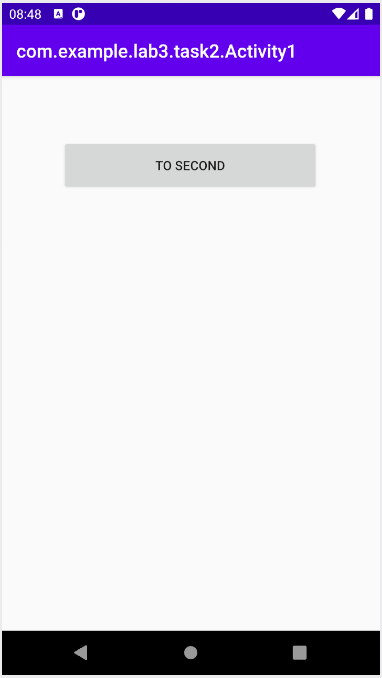


Рис.2. Первая, вторая и третья activity

Также мы можем, как и требуется по условию, открыть activityAbout из любого окна с помощью Navigation Drawer:



Рис.3. Activity About

Activity3 вызывается методом startActivityForResult() так как в зависимости от перехода необходимо определить закрывать Activity2 или нет.

**2.3. Задача 3. Навигация (флаги Intent/атрибуты Activity)**

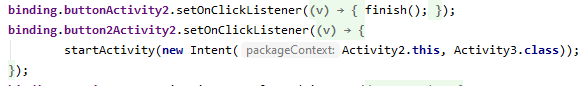
Решите предыдущую задачу с помощью Activity, Intent и флагов Intent либо атрибутов Activity. НеиспользуйтеstartActivityForResultвэтомзадании.

**В Activity1:**

****

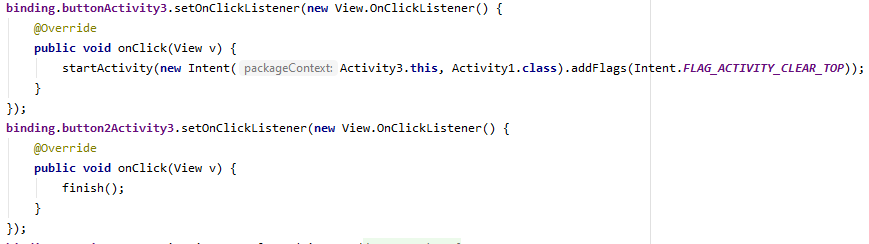
При нажатии на кнопку запускается Activity2 с помощью метода startActivity.

**В Activity2:**

****

Если переходим в Activity1 , то завершаемся методом finish, иначе переходим в Activity3 методом startActivity() .

**В Activity3:**

****

При переходе на Activity1 делаем переход с установкой флага FLAG\_ACTIVITY\_CLEAR\_TOP, который закроет все Activity, кроме Activity1.

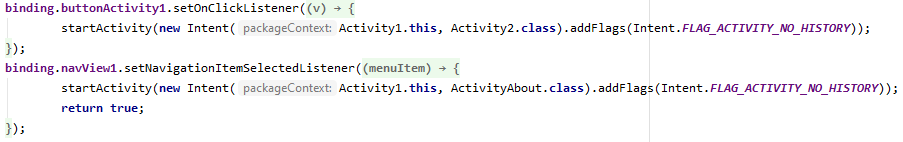
Если же переход на Activity2, то завершаемся методом finish()

**2.4. Задача 4. Навигация (флаги Intent/атрибуты Activity, доп. условия)**

Дополните граф навигации новым(-и) переходом(-ами) с целью демонстрации какого-нибудь (на свое усмотрение) атрибута Activity или флага Intent, который еще не использовался для решения задачи. Поясните пример и работу флага/атрибута.

Ограничение на размер backstack к этому и следующему заданию не применяется.

Для демонстрации работы был выбран флаг Intent.FLAG\_ACTIVITY\_NO\_HISTORY. Его работу можно продемонстрировать путём перехода в любую activity и нажатием «назад». При нажатии назад, флаг будет возвращать каждый раз к 1 activity, то есть не сохранять историю передвижения по activity**.**



Этот флаг был добавлен во все Activity, на скриншоте представлена Activity1.

**2.5. Задача 5. Навигация (Fragments, Navigation Graph)**

Решите предыдущую задачу (с расширенным графом) с использованием navigationgraph. Все Activity должны быть заменены на фрагменты, кроме Activity 'About', которая должна остаться самостоятельной Activity. В отчете сравните все решения.

Для решения данной задачи были созданы фрагменты, была создана MainActivity, в которой находится NavController, который помогает переходить между activity.Nav\_host\_fragment является контейнером для фрагментов и они переключаются внутри него.

Чтобы переключаться между фрагментами нужно заполнить nav\_graph, а также указать переходы между фрагментами.

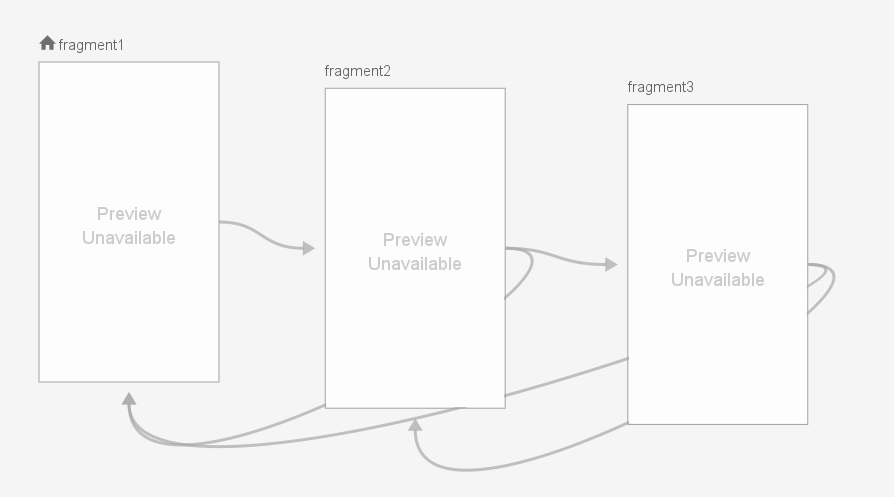
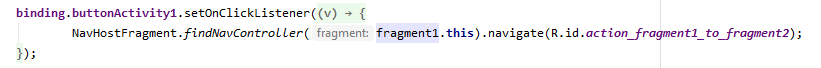


Рис.7. NavigationGraph

Навигация осуществлялась с помощью получение в коде NavController:

NavigationComponent является современной альтернативой стандартному SDK для навигации в андроид приложении.



**3. Выводы**

В ходе выполнения лабораторной работы были изучены: Lifecycle-AwareComponents, разные интерфейсы для навигации, например: Navigation Drawer, навигация с помощью стандартных методов Android SDK, навигация с помощью стандартных методов SDK и флагов Intent, навигация c использованием fragments + NavigationComponent. Визуально переход лучше выглядел с использованием navigation\_graph так как не создавались новые activity и переход делались с помощью переключения фрагментов. Все эти способы навигации были продемонстрированы и протестированы на примере приложения.