

К вопросу использования библиотеки «The Navigation Component» для организации навигации в мобильном приложении для Android и ускорения процесса разработки проекта

В.А. Жжонов, В.А. Евсина, С.Н. Широбокова

*Южно-Российский государственный политехнический университет (НПИ)
имени М.И. Платова, Новочеркасск*

Аннотация: Любое мобильное приложение состоит из рабочих экранов, которые содержат различную информацию. С каждым годом требований к приложениям становится все больше. Это обусловлено постоянным ростом количества пользователей различных сервисов, которые требуют для своего функционирования определенные программные компоненты. Использование примитивных методов разработки, как и устаревших компонентов, приводит к увеличению сроков проектирования, реализации и внедрения проекта в работу. Но развитие технологий не стоит на месте, что позволяет решать появляющиеся трудоемкие задачи в более легкой форме и значительно быстрее. И примером этому является разработанная библиотека «*The Navigation Component*», что позволяет создавать удобную и понятную схему навигации внутри приложения, а это просто необходимо, когда мобильное приложение насчитывает множество рабочих окон. В статье будет приведен список компонентов библиотеки «*The Navigation Component*». Показаны основные атрибуты. Рассмотрен принцип работы каждого компонента.

Ключевые слова: *навигационная схема приложения, The Navigation Component, приложение, Android.*

Разработка любого проекта – это сложный процесс, который затрагивает множество сфер деятельности [1]. Использование устаревших технологий при разработке программного обеспечения приводит не только к ухудшению качеств самого приложения, но и также, во многих случаях, увеличивает и время на разработку проекта, затрудняет его масштабируемость и дальнейшее сопровождение. Разумеется, это актуально и для мобильных приложений, которые, что ни для кого не секрет, стали частью современного мира [2-3]. Множество различных сервисов, без которых трудно представить наш быт, адаптированы под экраны смартфонов. Но с каждым годом требования к приложениям растут. Поэтому так важно разрабатывать новые технологические решения, которые могут упростить и ускорить разработку мобильных программ [4-6].

Как мы знаем, каждое приложение состоит из нескольких экранов, где отображается различная информация. Например, в приложении магазина мы

можем видеть меню, где представлены списки товаров, а также и экран с более подробной информацией о конкретном товаре. Подобных экранов может быть огромное множество, что значительно может усложнить процесс разработки навигации между ними. В компании *Google* хорошо это понимают, поэтому они решили разработать средство для формирования грамотной схемы навигации между окнами приложения, которое, в свою очередь, получило название «*The Navigation Component*». В состав данной библиотеки входят следующие компоненты:

- 1) *NavigationGraph* – представляет собой *XML*-файл, который хранит в себе информацию о стартовом экране, экранах назначения, действиях при переходах, выбранную анимацию перехода.
- 2) *NavHostFragment* – является контейнером, который помещается в разметку страницы приложения. В нем отображаются выбранные фрагменты из «*NavigationGraph*».
- 3) *NavController* – управляет навигацией приложения.

Как уже говорилось ранее, «*NavigationGraph*» содержит основную информацию, необходимую для переходов между экранами, и является навигационной схемой, в соответствии с которой будут осуществляться переходы между экранами приложения [7]. На рис. 1 представлен пример содержимого файла «*NavigationGraph*». Как можно заметить, в нем перечислены два фрагмента, у каждого из них указан соответствующий идентификатор, класс, название, файл верстки. Кроме того, к каждому из фрагментов добавлен «*action*», который обозначает собой действие перехода. Для каждого «*action*» обязательно задается уникальный идентификатор и место назначения для перехода [8]. Кроме того, данный набор атрибутов может быть дополнительно расширен, например, указанием анимации для перехода, только нужно указать атрибуты «*...Anim*», которые определяют анимацию для соответствующего перехода [9]. Указание в

«*popUpTo*» необходимой страницы позволяет удалить из стека окна, которые находятся между текущей страницей и указанной. Если включить атрибут «*popUpToInclusive*», то и указанная в «*popUpTo*» страница будет удалена из стека. Атрибут «*launchSingleTop*» позволяет запустить страницу, как однопользовательскую. Для того, чтобы обозначить данные, которые могут быть переданы на фрагмент, используется «*argument*», где задается название передаваемых данных, тип, а также значение по умолчанию, на случай если данные при переходе не будут указаны. На рис. 2 представлен схематичный вид «*NavigationGraph*» в среде разработки «*Android Studio*».

```
<?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/nav_graph"
    app:startDestination="@id/First">

    <fragment
        android:id="@+id/First"
        android:name="com.example.myapplication.FirstFragment"
        android:label="First Fragment"
        tools:layout="@layout/fragment_first">

        <action
            android:id="@+id/action_FirstFragment_to_Second"
            app:destination="@id/Second"
            app:enterAnim="@anim/nav_default_enter_anim"
            app:exitAnim="@anim/nav_default_exit_anim"
            app:popEnterAnim="@anim/nav_default_pop_enter_anim"
            app:popExitAnim="@anim/nav_default_pop_exit_anim" />
        <argument
            android:name="sendStringName"
            app:argType="string"
            android:defaultValue="Пусто" />
    </fragment>
    <fragment
        android:id="@+id/Second"
        android:name="com.example.myapplication.SecondFragment"
        android:label="@string/second_fragment_label"
        tools:layout="@layout/fragment_second">
        <action
            android:id="@+id/action_Second_to_Third"
            app:destination="@id/Third"
            app:launchSingleTop="true"
            app:popUpTo="@id/First"
            app:popUpToInclusive="false" />
    </fragment>
```

Рис. 1. – Файл «*NavigationGraph*»

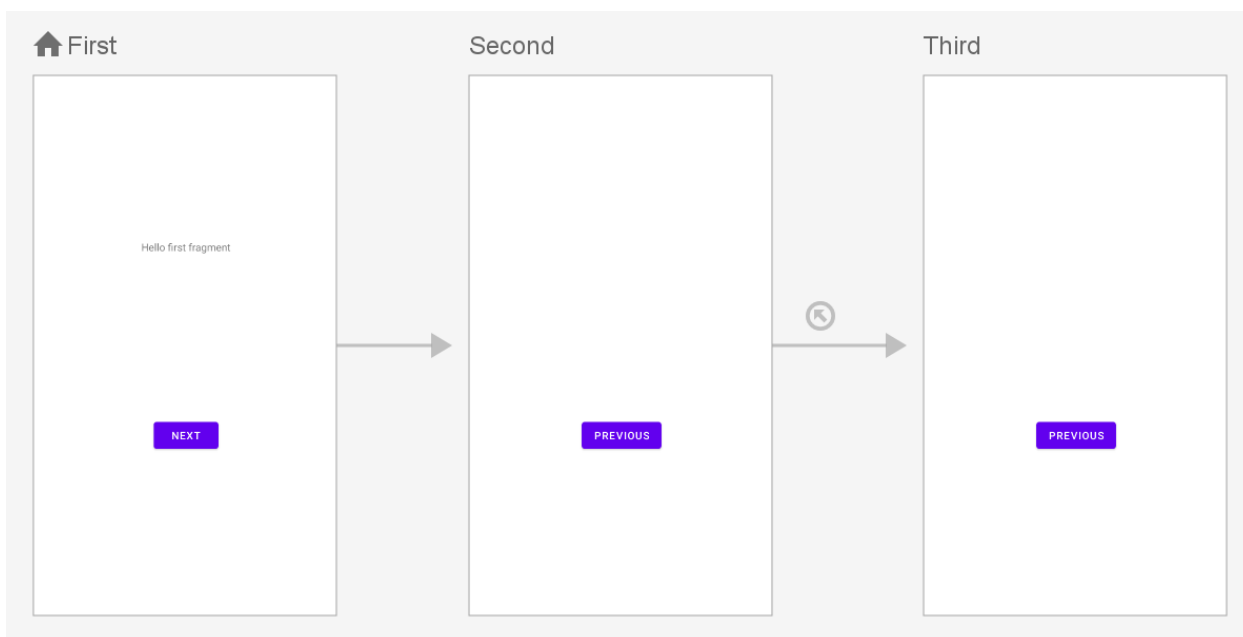


Рис. 2. – Схематичный вид «*NavigationGraph*»

Для того, чтобы отображать фрагменты, указанные в «*NavigationGraph*» необходим «*NavHostFragment*», который встраивается в разметку страницы приложения, на рис. 3 представлен данный контейнер, что можно понять по названию класса «...*NavHostFragment*». Сам по себе контейнер также является фрагментом. Среди атрибутов можно отметить «*navGraph*», который определяет навигационный граф, к которому прикреплен данный контейнер.

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$Scrolli...">

    <fragment
        android:id="@+id/nav_host_fragment_content_main"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        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/nav_graph" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Рис. 3. – Схематичный вид «*NavigationGraph*»

Чтобы управлять перемещением по графу, необходим «*NavController*», который можно получить с помощью специального метода «*findNavController(...)*». При вызове этого метода мы должны передавать идентификатор «*NavigationGraph*», за которым уже закреплен навигационный граф «*NavigationGraph*» [10]. Чтобы перейти к необходимому фрагменту в соответствии с указанным графом, мы должны вызвать метод контроллера «*navigate*» и передать ему идентификатор «*action*», который указан у соответствующего фрагмента, что показано на рис. 4.

```
val navController = findNavController(R.id.nav_host_fragment_content_main)  
navController.navigate(R.id.action_FirstFragment_to_Second)
```

Рис. 4. – Пример перехода по «*action*» фрагмента

Как можно отметить, библиотека «*The Navigation Component*», позволяет:

1. В удобной форме создавать сложные навигационные схемы, что значительно сокращает время на разработку приложения.
2. В дальнейшем легко вносить изменения в навигационную схему и расширять её.
3. Просматривать созданную схему в графическом виде, что упрощает её понимание, а также ускоряет процесс работы с ней.

Таким образом, создание подобных библиотек позволяет эффективно работать с довольно сложными и громоздкими задачами, что может значительно упростить процесс работы над любым приложением, а это, в свою очередь, уменьшает время, которое затрачивается на сам процесс разработки, а также позволяет быстро освоиться в проекте новым разработчикам.

Литература

1. Joseph D. Blackburn, Gary D. Scudder, Luk Van Wassenhove. Improving Speed and Productivity of Software Development: A Global Survey of Software
-

Developers // IEEE Transactions on Software Engineering, 1997. URL: researchgate.net/publication/3187839_Improving_Speed_and_Productivity_of_Software_Development_A_Global_Survey_of_Software_Developers.

2. Александровский В.Г. Мобильные технологии в строительстве. Программное обеспечение на платформе Android. Часть 1 // Инженерный вестник Дона, 2019, №4. URL: ivdon.ru/ru/magazine/archive/n4y2019/5874.

3. Mentsiev A.U., Alams M.T. Mobile forensic tools and techniques: Android data security // Инженерный вестник Дона, 2019, №2. URL: ivdon.ru/ru/magazine/archive/n2y2019/5766.

4. Paula Natoli. Upwardly mobile—mobile technologies span the supply chain // Supply & Demand Chain Executive, 2015. URL: sdexec.com/warehousing/article/12091968/upwardly-mobilemobile-technologies-span-the-supply-chain.

5. Robert Destefano. Four steps to successfully deploy android in your supply chain // Supply & Demand Chain Executive, 2018. URL: sdexec.com/software-technology/blog/20998353/ivanti-supply-chain-4-steps-to-successfully-deploy-android-in-your-supply-chain.

6. Rachel Harrison, Derek Flood, David Duce. Usability of mobile applications: literature review and rationale for a new usability model // Supply & Journal of Interaction Science, 2013. URL: journalofinteractionspringeropen.com/articles/10.1186/2194-0827-1-1.

7. Navigation [Интернет-ресурс]. URL: developer.android.com/guide/navigation.

8. Global actions [Интернет-ресурс]. URL: developer.android.com/guide/navigation/navigation-global-action.

9. Animate transitions between destinations. [Интернет-ресурс]. URL: developer.android.com/guide/navigation/navigation-animate-transitions.

10. Navigate to a destination [Интернет-ресурс]. URL: developer.android.com/guide/navigation/navigation-navigate.

References

1. Joseph D. Blackburn, Gary D. Scudder, Luk Van Wassenhove. IEEE Transactions on Software Engineering, 1997. URL: researchgate.net/publication/3187839_Improving_Speed_and_Productivity_of_Software_Development_A_Global_Survey_of_Software_Developers.

2. Alexandrovsky V.G. Inzhenernyj vestnik Dona, 2019, №4. URL: ivdon.ru/ru/magazine/archive/n4y2019/5874.

3. Mentsiev A.U., Alams M.T. Inzhenernyj vestnik Dona, 2019, №2. URL: ivdon.ru/ru/magazine/archive/n2y2019/5766.

4. Paula Natoli. Supply & Demand Chain Executive, 2015. URL: sdcexec.com/warehousing/article/12091968/upwardly-mobilemobile-technologies-span-the-supply-chain.

5. Robert Destefano. Supply & Demand Chain Executive, 2018. URL: sdcexec.com/software-technology/blog/20998353/ivanti-supply-chain-4-steps-to-successfully-deploy-android-in-your-supply-chain.

6. Rachel Harrison, Derek Flood, David Duce. Supply & Journal of Interaction Science, 2013. URL: journalofinteractionsience.springeropen.com/articles/10.1186/2194-0827-1-1.

7. Navigation [Internet resource]. URL: developer.android.com/guide/navigation.

8. Global actions [Internet resource]. URL: developer.android.com/guide/navigation/navigation-global-action.

9. Animate transitions between destinations. [Internet resource]. URL: developer.android.com/guide/navigation/navigation-animate-transitions.

10. Navigate to a destination [Internet resource]. URL: developer.android.com/guide/navigation/navigation-navigate.