

Научная статья

Original article

DOI 10.55186/27131424_2023_5_6_1



**ДЕКЛАРАТИВНЫЙ ПОДХОД ПРИ СОЗДАНИИ
ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА МОБИЛЬНЫХ
ПРИЛОЖЕНИЙ ДЛЯ ОС ANDROID**

**DECLARATIVE APPROACH WHEN CREATING THE USER INTERFACE OF
MOBILE APPLICATIONS FOR ANDROID OS**

Узоров Кирилл Александрович, студент, федеральное государственное бюджетное образовательное учреждение высшего образования "МИРЭА - Российский технологический университет" РТУ МИРЭА (119454 Россия, г. Москва, ул. проспект Вернадского, д. 78, стр. 4), тел. +7(495)471-14-47, ORCID: <http://orcid.org/>, uzorov@mirea.ru

Михиенков Кирилл Сергеевич, студент, федеральное государственное бюджетное образовательное учреждение высшего образования "МИРЭА - Российский технологический университет" РТУ МИРЭА (119454 Россия, г. Москва, ул. проспект Вернадского, д. 78, стр. 4), тел. +7(495)999-51-61, ORCID: <http://orcid.org/>, mikheenkov@mirea.ru

Бузыкова Юлия Сергеевна, доцент, федеральное государственное бюджетное образовательное учреждение высшего образования "МИРЭА - Российский технологический университет" РТУ МИРЭА (119454 Россия, г. Москва, ул. проспект Вернадского, д. 78, стр. 4), тел. +7(495)681-87-90, ORCID: <http://orcid.org/>, juliaserg_buz@mail.ru

Международный журнал прикладных наук и технологий "Integral"

Kirill A. Uzorov, student, Federal State Budget Educational Institution of Higher Education «MIREA - Russian Technological University» (78 prospekt Vernadskogo st., p. 4, Moscow, 119454 Russia), tel. +7(495)471-14-47, ORCID: <http://orcid.org/uzorov@mirea.ru>

Kirill M. Sergeevich, student, Federal State Budget Educational Institution of Higher Education «MIREA - Russian Technological University» (78 prospekt Vernadskogo st., p. 4, Moscow, 119454 Russia), tel. +7(495)999-51-61, ORCID: <http://orcid.org/mikheenkov@mirea.ru>

Yulia S. Buzykova, docent, Federal State Budget Educational Institution of Higher Education «MIREA - Russian Technological University» (78 prospekt Vernadskogo st., p. 4, Moscow, 119454 Russia), tel. +7(495)681-87-90, ORCID: http://orcid.org/juliaserg_buz@mail.ru

Аннотация. В статье предпринимается попытка решения проблемы снижения высоких временных затрат при создании удобного пользовательского интерфейса мобильного приложения. Предлагается применить декларативный подход программирования через создания пользовательского интерфейса на языке Kotlin и набора библиотек Jetpack Compose. Описывает ключевые преимущества фреймворка и продемонстрирован процесс применения данной технологии в разработке.

Abstract. A large number of developers face the problem of high time costs when creating a user-friendly interface. A possible solution may be to use a set of Jetpack Compose libraries. This article describes the key advantages of the framework and demonstrates the process of using this technology in development.

Ключевые слова: *Jetpack Compose, декларативное программирование, создание пользовательского интерфейса, разработка Android приложений*

Keywords: *Jetpack Compose, declarative programming, user interface creation, Android Application Development*

Международный журнал прикладных наук и технологий "Integral"

Согласно ресурсу Statcounter GlobalStats, доля рынка операционной системы Android на 12 апреля 2023 года составляет 70.93% и стабильно удерживается на данной отметке последний год [2]. Такую популярность данной платформе обеспечивает множество факторов, среди которых постоянно растущий ассортимент приложений, поддерживаемых крупным сообществом программистов и дизайнеров.

Роль дизайнера в построении мобильного приложения колоссальна. Если приложение не будет обладать внешней привлекательностью и предоставлять пользователю удобный интерфейс, шансы его коммерческого успеха наряду с возможностью стать популярным значительно уменьшаются [3].

К сожалению, для начинающих разработчиков, стремящихся создать собственный продукт, реализовав его в виде мобильного приложения, не существует возможности общаться с дизайнерами на профессиональном уровне по причине ограниченного бюджета. Как следствие разработка может окончиться неудачей, даже если идея, положенная в основу мобильного приложения достаточно полезна и коммерциализируема.

В данном случае решением проблемы может стать использование набора инструментов для построения пользовательского интерфейса – Jetpack Compose. Jetpack Compose - это разработанный и рекомендуемый компанией Google продукт, который позволяет упростить и ускорить разработку под Android, уменьшая количество кода и предоставляя мощные инструменты и интуитивно понятное API на языке Kotlin [1].

Описанный фреймворк обладает огромным множеством преимуществ, среди которых стоит отметить использование декларативного подхода программирования, означающее, что для создания пользовательского интерфейса достаточно описать его, используя язык Kotlin, а всю остальную работу выполнит Compose [4].

Но главным аргументом за использование Jetpack Compose для разработчиков, создающих собственные проекты с малым бюджетом, является возможность использования набора шаблонов проектирования, благодаря чему можно

создавать удобный и привлекательный пользовательский интерфейс, соответствующий распространённым и общепринятым стандартам и практикам, не привлекая прочих специалистов [6].

Для демонстрации практической пользы описанного набора инструментов приведём простой пример. При использовании Jetpack Compose не нужно создавать файлы разметки XML, вместо этого достаточно реализовать функцию Kotlin, пометив её аннотацией «Composable». Далее в тело функции необходимо поместить описание логики построения интерфейса, используя API фреймворка. Реализуем анимированный сплеш-экран, используя Jetpack Compose. Сплеш-экран представляет собой недолгую заставку перед запуском приложения во время которой подгружаются необходимые данные. В рассматриваемом случае мы поместим логотип приложения в центр экрана, после чего используем анимацию для его плавного увеличения.

Для того чтобы использовать функции Jetpack Compose необходимо установить последнюю версию интегрированной среды разработки Android Studio, после чего создать новый проект с использованием шаблона Empty Compose Activity. Для демонстрационного варианта предварительно был создан логотип. Данный логотип был импортирован в AndroidStudio в виде ImageVector (см. рисунок 1).



Рисунок 1 – Подготовленный логотип

Настроек проекта по умолчанию хватает, чтобы достичь поставленной цели, однако хорошим тоном считается добавление навигации, которая позволит перейти от сплеш-экрана к основному экрану приложения. Для навигации в Jetpack Compose существует отдельная библиотека, которую необходимо импортировать в проект, однако данный этап выходит за рамки рассмотрения данной статьи в целях сохранения краткости и модульности изложения.

Международный журнал прикладных наук и технологий "Integral"

Сплеш-экран будет представлен одной функцией по логике, описанной выше.

Код экрана с комментариями представлен в листинге 1.

Листинг 1 – Сплеш-экран

```
@Composable
fun SplashScreen() {
    //Задаём начальный размер нашего логотипа
    //При изменении данного значения, scale будет автоматически //анимироваться
    val scale = remember {
        Animatable(0f)
    }
    LaunchedEffect(key1 = true) {
        //Задаём конечный размер нашего логотипа, указываем //длительность
анимации
        //а также интерполятор, позволяющий поддерживать плавную //анимацию
```

Листинг 1 – Сплеш-экран (Продолжение)

```
        scale.animateTo(
            0.7f,
            animationSpec = tween(1000),
            block = {
                OvershootInterpolator(1.2f).getInterpolation(this.value)
            }
        )

        //Указываем задержку перед переход на другой //экран
        delay(1000L)
        //TODO: Navigation to Home Screen
    }
    //Контейнер для логотипа
    //Аналог LinearLayout
    Box(
        modifier = Modifier
            .fillMaxSize(),
        contentAlignment = Alignment.Center
    ) {
        //Сам логотип
        Image(
            painterResource(id = R.drawable.logo),
            contentDescription = null,
            modifier = Modifier.scale(scale.value)
        )
    }
}
```

Приведённый код позволяет оценить как сильно Jetpack Compose упрощает процесс разработки. Реализация аналогичного функционала с использованием классического императивного способа программирования требовала бы приложения больших усилий и значительных временных затрат.

Результат работы приведенной функции показан на рисунке 2.

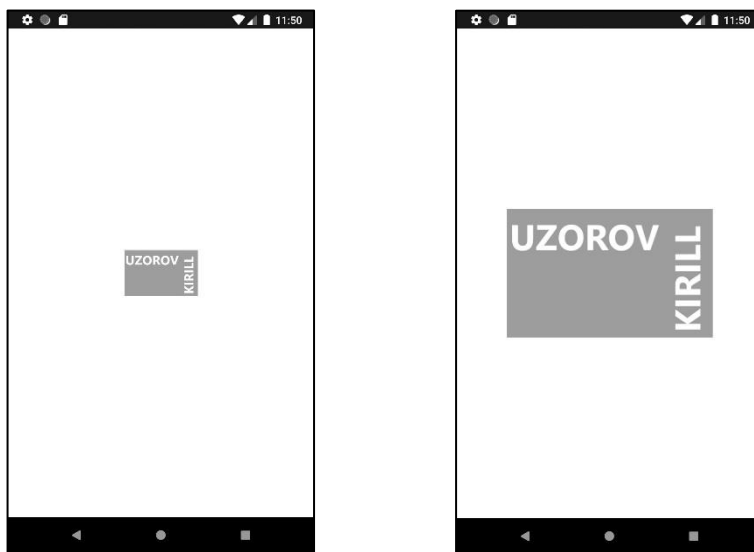


Рисунок 2 – Полученный сплеш-экран

Таким образом на практическом примере было показано насколько эффективным является использование Jetpack Compose при создании пользовательского интерфейса. Помимо набора инструментов описанный фреймворк предоставляет шаблоны проектирования, что позволяет любому разработчику создавать мобильные приложения, соответствующие современным стандартам проектирования пользовательских интерфейсов и рекомендациям компании Google.

Литература

1. Доля рынка мобильных операционных систем // Statcounter Global Stats
URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (дата обращения: 13.04.2023).

2. Зачем использовать Compose // Android Developers URL: <https://developer.android.com/jetpack/compose/why-adopt#accelerate-development> (дата обращения: 13.04.2023).
3. Интерполятор // Сайт Александра Климова URL: <https://developer.alexanderklimov.ru/android/animation/interpolator.php> (дата обращения: 13.04.2023).
4. Клифтон Я. Проектирование пользовательского интерфейса в Android/ пер. с англ. А.Н. Киселёва. 2-е изд. М: ДМК ПРЕСС. 2017.
5. Клоков С. А. Рассмотрение декларативного подхода к разработке интерфейсов мобильных приложений для Android / С. А. Клоков. Текст: непосредственный // Молодой ученый. 2022. № 4 (399). С. 10-12. URL: <https://moluch.ru/archive/399/88225/> (дата обращения: 13.04.2023).
6. Материальные компоненты и компоновка // Android Developers URL: <https://developer.android.com/jetpack/compose/layouts/material> (дата обращения: 13.04.2023).

References

1. Mobile Operating System Market Share Worldwide // Statcounter Global Stats URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (date of access: 04/13/2023).
2. Why adopt Compose // Android Developers URL: <https://developer.android.com/jetpack/compose/why-adopt#accelerate-development> (date of access: 04/13/2023).
3. Interpolator // Alexander Klimov`s site URL: <https://developer.alexanderklimov.ru/android/animation/interpolator.php> (date of access: 04/13/2023).
4. Ian G. Clifton. User interface design in Android / translation into russian A.N. Kiselev 2nd ed. - M: DMK PRESS, 2017
5. Klovov S.A. Consideration of a declarative approach to the development of mobile application interfaces for Android / S. A. Klovov. Text: direct // Young

scientist. 2022. No. 4 (399). S. 10-12. URL: <https://moluch.ru/archive/399/88225/>
(date of access: 04/13/2023).

6. Material Components and layouts // Android Developers URL: <https://developer.android.com/jetpack/compose/layouts/material> (date of access: 04/13/2023).

© Узоров К.А., Михиенков К.С., Бузыкова Ю.С., 2023 *Международный журнал прикладных наук и технологий "Integral" №6/2023*

Для цитирования: Узоров К.А., Михиенков К.С., Бузыкова Ю.С. **ДЕКЛАРАТИВНЫЙ ПОДХОД ПРИ СОЗДАНИИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ДЛЯ ОС ANDROID // Международный журнал прикладных наук и технологий "Integral" №6/2023**