

ТРАНСПИЛЯЦИЯ JETPACK COMPOSE (ANDROID) В SWIFTUI (IOS)

Курыбов Д.А. (Университет ИТМО)

Научный руководитель – PhD, доцент Аксенов В.Е.

(Университет ИТМО)

Введение. Jetpack Compose представляет из себя фреймворк для разработки UI в Android. Фреймворк предоставляет объекты для формирования пользовательского интерфейса, используя нативный язык разработки - Kotlin Native. Аналогичный фреймворк так же представлен для разработки пользовательских интерфейсов на языке Swift для платформы iOS.

Набирающий популярность KMM (Kotlin Multiplatform Mobile) предоставляет возможность использовать в iOS приложения код, написанный на Kotlin Native, преобразуя его в Objective-C. Таким образом, платформозависимой частью приложения разработанного с использованием KMM остается только часть кода, описывающая пользовательский интерфейс. В силу близости семантики Jetpack Compose и SwiftUI, логичным развитием будет транспилиция кода, описывающего UI в Android приложении, в Swift код описывающий подобный пользовательский интерфейс.

Основная часть. Первостепенной задачей является построение корректного дерева разбора кода на языке Kotlin. Для этого необходимо построить грамматику языка и сгенерировать по ней парсер языка. Для наибольшего покрытия кода, уже реализованного для Android платформы, важно поддерживать как можно больший объем синтаксических конструкций.

Одной из задач является создание гибкого решения, которое будет легко дорабатываемым при любых изменений фреймворков обеих платформ. Для решения этой задачи были созданы промежуточные структуры, сохраняющие все параметры UI элементов, описанных с использованием Jetpack Compose. Так, например, модификаторы размеров будут храниться отдельной сущностью, которая может быть создана из дерева разбора Kotlin кода и ее реализация может транспилироваться в корректный SwiftUI объект. Таким образом, при изменении объекта в SwiftUI или Jetpack Compose достаточно будет изменить реализацию преобразования лишь одной сущности, не затрагивая при том способ разбора изначального кода.

Завершающим этапом является построение по созданным сущностям корректного SwiftUI кода, описывающего аналогичные UI элементы. На этом этапе важно, что некоторые параметры, передаваемые в виде аргументов в UI функции Jetpack compose, в SwiftUI применяются специальным методом, вызываемым у SwiftUI объекта. Здесь так же помогает предварительное преобразование в промежуточные сущности.

Выводы. Реализованный транpiler можно применить в уже существующих Android проектах использующих Jetpack Compose для реализации iOS приложения. Транслятор легко расширяется и изменяется. На его основе возможно построить транpiler в другие UI фреймворки. Полученный код легко читается и оставляет возможность изменить части iOS UI независимо от Android платформы.

Список использованных источников:

1. [Электронный ресурс]. Режим доступа: <https://kotlinlang.org/docs/reference/>

grammar.html (дата обращения: 20.02.2023).

2. [Электронный ресурс]. Режим доступа: <https://developer.apple.com/documentation/swiftui> (дата обращения: 20.02.2023).

3. [Электронный ресурс]. Режим доступа: <https://developer.android.com/jetpack/compose/documentation> (дата обращения: 20.02.2023).

Курябов Д.А. (Автор)

Подпись

Аксенов В.Е. (Научный руководитель)

Подпись