

ОБЗОР МЕТОДОВ ПРОЕКТИРОВАНИЯ ИНТЕРФЕЙСА НА SWIFT

Пасюта Максим Андреевич

студент, Алтайский государственный технический университет им. И. И. Ползунова, РФ, г. Барнаул

В сообществе Swift разгорелись жаркие дебаты по поводу использования раскадровки (Storyboard) и программного кода, когда дело доходит до создания интерфейса приложения [2]. Похоже, что Apple подталкивает сообщество к использованию раскадровки (Storyboard) и использования ее в качестве стандарта на будущее. Тем не менее, есть еще некоторые недостатки использования раскадровки (Storyboard). Рассмотрим преимущества использования каждого метода.

Раскадровка (Storyboard)

1. Простота использования. С помощью нескольких щелчков мыши и перетаскивания можно моментально создать простой макет для своего приложения. Используя для написания интерфейса программный код, создание простого макета может стать долгим процессом. В разы больше программного кода требуется только для создания простого представления. Теперь представьте, что вы делаете это для всех кнопок, меток, текстовых полей и т. д. Это может стать довольно утомительным. Важно включить `translatesAutoresizingMaskIntoConstraints` и `.active = true`, иначе наше представление не будет отображаться. Большое количество программного кода снижает мешает дальнейшему развитию приложения.

2. Визуализация интерфейса. Одним из основных преимуществ использования раскадровки (Storyboard) является то, что она дает очень хорошее визуальное представление вашего приложения [3]. Разработчик знает, как будет выглядеть каждая кнопка, метка, текстовое поле и представление. С помощью нескольких щелчков мыши и настройки можно легко создать целое представление. Кроме того, ограничения интерфейса также представлены визуально, любая корректировка ограничений будет отображаться другим цветом.

Создание элементов интерфейса программно

1. Контроль. Кодирование всех элементов пользовательского интерфейса дает ощущение полного контроля. Все, что можно сделать с помощью StoryBoard, возможно реализовать с помощью программного кода [1]. Разработчик точно знает, какие настройки назначили элементу пользовательского интерфейса. Изменить настройки в коде легко, потому что разработчик точно знает, где инициализировал все элементы пользовательского интерфейса.

2. Повторное использование. Создание простого представления в коде может быть долгим и трудным процессом, однако возможно повторно использовать эти программные коды в следующих проектах. При разработке нескольких типовых приложений одной тематике интерфейс конечного продукта будет крайне схож. Поэтому использование программного кода для создания интерфейса предпочтительней.

3. Беспорядок и навигация. Одна из основных минусов использования Storyboard, заключается в том, что он становится загроможденным [3]. На начальном этапе Storyboard это может показаться не проблемой, но, когда экран приложения заполнен элементами пользовательского интерфейса и контроллерами, ориентироваться в нем крайне сложно.

4. Конфликт слияния. Это основная проблема при использовании Storyboard. При работе над проектом с другими разработчиками, некоторые из них могут использовать Storyboard, а другие могут создавать элементы интерфейса программно в коде, сложности могут возникать, когда разработчики объединяют работу на GitHub [3]. Разрешить конфликт слияния крайне сложно.

В заключение хотелось бы добавить, что на данный момент полностью отказываться от их использования одного из методов практически невозможно. При правильном применении они приносят огромную пользу и помогают эффективно решать поставленные задачи. Нужно лишь научиться расставлять приоритеты при проектировании приложения.

Список литературы:

1. Кучеренко А. В. SWIFT – ЯЗЫК НОВОГО ПОКОЛЕНИЯ / А.В. Кучеренко // СУЧАСНА НАУКА ХХІ СТОЛІТТЯ(19-21.06.2014)[Электронный ресурс]. - Электрон. дан. – Заглавие с экрана. – Режим доступа: URL: <https://int-konf.org/ru/konf062014/814-kandidat-pedagogchnih-nauk-paschnik-n-o-podushna-podat-u-naukovy-konceptsi-opodatkuvannya-m-m-aleksyeyenka.html>
2. Фролов В.В. Swift - язык, который изменит мир программирования // Актуальные проблемы авиации и космонавтики. 2017. №13. URL: <https://cyberleninka.ru/article/n/swift-yazyk-kotoryy-izmenit-mir-programmirovaniya> (дата обращения: 27.01.2022).
3. Creating a Storyboard App [Электронный ресурс]. - Электрон. дан. – Заглавие с экрана. – Режим доступа: URL: <https://developer.apple.com/tutorials/app-dev-training/creating-a-storyboard-app>