\*ВКР - буду произносить полностью

Слайд 0.

Доброго времени суток коллеги. Темой моей ВКР является "Разработка библиотеки для реализации анимаций представлений в рамках модели MVC"

Слайд 1.

План рассказа будет следующий: расскажу с чем пришлось столкнуться, какая возникла цель. Далее опишу сформулированные задачи. Расскажу, что было спроектировано и как было разработано. Подведу выводы и с удовольствие отвечу на Ваши вопросы.

Слайд 2.

На сегодняшний день мобильная разработка представляет собой перспективное направление и развивается большими темпами. Исходя из данных темпов развития, а также требований промышленности необходимо постоянно совершенствовать и ускорять процессы разработки вместе с используемыми подходами.

Слайд 3.

Поэтому главной целью ВКР - оптимизация времени разработки и уменьшение сложности разработки анимаций перехода между видами контроллеров в рамках модели MVC.

С данной проблемой столкнулся лично, когда была задача разработать необычную систему переходов между видами контроллеров в одном из коммерческих приложений.

Слайд 4.

В соответствии с целью были сформулированы следующие задачи: анализ существующих решений среди библиотек с открытым исходным кодом, выявление достоинств и недостатков выбранных библиотек, выработка требований к проекту создаваемой библиотеки, анализ недостатков, реализация проекта библиотеки, подведение выводов и рассмотрение перспектив развития библиотеки. ВКР строится из последовательных глав, каждая из которых описывает выполнение соответствующей задачи.

Слайд 5.

Глава первая в которой производится обзор существующих решений. В качестве входных данных были выбраны следующие библиотеки: "JHChainableAnimations", "Presentr", "PresenterKit", "HYBControllerTransitions". В результате проведенного анализа было выявлено, чем отличается каждая библиотека, а также в каких моментах ее функциональность либо нераскрыта, либо реализована не до конца. Вкратце расскажу по каждой из них.

Слайд 6.

Первая библиотека - "JHChainableAnimations" отличается удобством по работе с анимациями, но никак не облегчает работу по анимированным переходам между видами контроллеров. "Presentr" добавляет удобство по работе с модальными анимациями, но ограничена лишь функционалом стандартной библиотеки.

Слайд 7.

"PresenterKit" в некоторой степени очень схожа с предыдущей, но отличается работой не только с модальным показом видов контроллеров. Последняя, "HYBControllerTransitions" - представляется единственной библиотекой, которая обеспечивает достойную, но не очень описанную и удобную работу с анимациями по переходу между видами контроллеров. Итог - понимание того, что разработка новой библиотеки является необходимым и возможным достижением поставленной цели.

Слайд 8.

Вторая глава рассказывает о принципе работе анимаций, какие инструменты предоставляет компания Apple, как ими грамотно пользоваться. После чего подробно описывается архитектура проекта библиотеки с аргументацией представленных решений, к которым относится выделение уровней абстракции и использование паттернов проектирования.

Слайд 8.

Архитектура - самый важный момент в процессе разработки любого программного обеспечения. Библиотека для анимаций не является исключением. Известно, что процесс анимирования должен быть построен на контроллерах анимации, поэтому конфигурация данных объектов должна происходить неявным образом. В данном случае логика скрыта за приватным свойством transitioningFactory категории UIViewController.

Слайд 9.

Ответственным за решение предоставления нужного анимационного контроллера для определенного типа анимации должен быть сторонний объект. В этом случае был применен наиболее подходящий шаблон проектирования - фабрика, который возвращает объекты, реализующие интерфейс UIViewControllerAnimatedTransitioning.

Данные объекты, принимающие конфигурационные аспекты анимации непосредственно от объекта, скрытого за протоколом BCTTransitioning, будут определять и конфигурировать анимацию по переходу между представлениями. Это является оптимальным решением для анимаций вида выскакивания, переворотов и выпадения.

Слайд 10.

Для оставшихся двух видов анимации: выталкивания и выхода, необходимо было унифицировать решение задачи из-за ее однотипности. Конфигурация анимации была идентична, отличия заключались в направлении анимации и способа воздействия на вид текущего контроллера. Для наиболее оптимального решения необходимо было скрыть особенности показа за объектами, которые будут определять положения объектов вида UIView до анимации и после нее. Иными словами, данные объекты будут полностью описывать пред- и пост-состояния анимации.

Слайд 11.

В таком случае анимационный контроллер типа BCTTransitionController получает зависимость на объекты, реализующие протокол BCTViewPerformer, которые занимаются определением местоположения объектов вида на протяжении анимации. Руководствуясь принципом инверсии зависимостей, был выделен протокол BCTViewPerformer, что привело к образованию цепи наследования и все объекты, реализующие протокол BCTViewPerformer, также наследуют базовый класс BCTBasicViewPerformer, который занимается получением значений вида и базовым сохранением.

Слайд 12.

Наследники базового класса ответственные только за подсчет новых координат. Такое решение приводит к тому, что вводятся уровни абстракции и общая ответственность разделяется на классы с меньшими зонами ответственности.

Слайд 13.

Третья глава описывает сложности разработки анимаций, с которыми пришлось столкнуться и способы, которыми можно решить данные задачи. Например, добавление свойств к неизменяемым объектам и модификация синтаксиса библиотеки. Проблемы и способы их решения были рассказаны мной на VI Конгрессе Молодых Ученых. К другим особенностям и задачам работы библиотеки можно отнести работу с низкоуровневыми данными для отображения, а также модификацию якорей элементов - так называемых точек опоры во время совершения анимации.

Слайд 14.

Отдельным моментом в главе выделен этап публикации библиотеки в менеджере зависимостей CocoaPods. Он состоит из следующих этапов:

1. Подготовка исходного кода библиотеки; Задача данного этапа состоит в упорядочивании и агрегировании необходимых файлов для сборки библиотеки на другом компьютере
2. Настройка специального конфигурационного файла; Данный файл занимается описанием всех свойств и параметров библиотеки
3. Выполнение локальной проверки библиотеки; Локальная проверка - один из последних моментов, чтобы убедиться в том, что библиотека пригодная к использованию.

Слайд 15.

1. Предоставление документации, инструкции к проекту, его визуализация; Необязательный этап, но рекомендуется выполнять, чтобы другим разработчикам было легче взаимодействовать с программным интерфейсом библиотеки.
2. Процесс загрузки библиотеки в менеджер зависимостей
3. Далее стоит еще раз проверить работоспособность библиотеки, на случай непредвиденных сбоев. Взять существующий проект и интегрировать библиотеку в него.

Слайд 16. Примеры работы библиотеки

Слайд 17.

В ходе выполнения ВКР на тему " Разработки библиотеки для реализации анимаций представлений в рамках модели MVC " были проанализированы существующие решения данной задачи, были определены достоинства и недостатки каждого, на основании чего были выдвинуты предпосылки для разработки новой библиотеки.

На основании разработанного проекта была реализована библиотека для реализации анимации представлений. В ходе работы были решены проблемы, связанные с синтаксисом библиотеки, а также осуществлено добавление свойств для неизменяемых объектов.

В данный момент библиотека находится в открытом доступе, готова к использованию, прошла проверку и опубликована в менеджере зависимостей CocoaPods. Среди направлений развития стоит отметить расширение текущих возможностей библиотеки. План развития подробно раскрыт в описании и включает следующие этапы: оптимизация исходного кода и поддержка интерактивных переходов.

Важно отметить, что в результате выполнения данной работы были получены практические навыки разработки программного обеспечения, а также закреплены знания в области проектирования информационных систем.

Слайд 18. Секция вопросов.