

**Конкурсное задание**

**по компетенции**

**«Разработка мобильных приложений»**

Сессия 4

Необходимо спроектировать и реализовать компонент для отображения списка карточек.

В работе необходимо использовать систему контроля версий Git. Ссылка на Git: . Для входа используйте учетную запись вида wsruserX, где X – это номер участника. Необходимо загрузить каждую сессию в отдельную ветку с именем “Session-X”, где Х – это номер сессии. Для каждого проекта необходим отдельный репозиторий.

Необходимо строго следовать предложенному дизайну. Макеты приложения доступны по ссылке: https://www.figma.com/file/aBHYbZgtg0eUIDwMbb8P04/WorldCinema-Session4-%D0%9C%D0%B5%D0%B6%D0%B2%D1%83%D0%B7?node-id=0%3A1

Во время работы не будет доступа в Интернет, кроме документации и API. Описание протокола API доступно по ссылке:

https://app.swaggerhub.com/apis-docs/WorldSkills-MAD/WorldCinema/1.0.0

Требования к компоненту:

1. Компонент должен быть оформлен в отдельном модуле, быть наследником класса View (UIView) (или его потомков).
2. При проектировании компонента следует соблюдать основные принципы объектно-ориентированного программирования. Для всех публичных методов и свойств необходимо добавить описания в виде комментариев.
3. Компонент пользовательского интерфейса должен корректно отображаться на смартфонах в портретной ориентации.
4. Входные параметры компонента (устанавливаются с помощью публичных сеттеров):
   * Список url изображений для отображения.
   * Иконка, отображаемая поверх карточки при свайпе влево (optional).
   * Иконка, отображаемая поверх карточки при свайпе вправо (optional).
   * Placeholder для отображения при пустом списке изображений (optional).
   * Радиус скругления углов карточек.
5. Публичные методы компонента:
   * Вызов свайпа влево.
   * Вызов свайпа вправо.
   * Получение информации о текущей отображаемой карточке.
6. События, получаемые от компонента:
   * Произошел свайп влево.
   * Произошел свайп вправо.
   * Список изображений пуст.
7. Поведение компонента должно соответствовать сценарию:

**Сценарий варианта использования “Обработка жеста пользователя “свайп влево”:**

Предусловие: пользователь перешел на экран с компонентом.

|  |  |  |
| --- | --- | --- |
| Пользователь | Мобильное приложение | Компонент |
| 1. Пользователь делает свайп влево. |  |  |
|  |  | 2. Поверх карточки отображается иконка, соответствующая свайпу влево. |
|  |  | 3. Отображается анимация движения текущей карточки и иконки влево с наклоном на 5°. |
|  |  | 4. Компонент сообщает мобильному приложению о событии “Произошел свайп влево”. |
|  | 5. Приложение обрабатывает событие от компонента “Произошел свайп влево”. |  |
|  |  | 6. Текущее изображение удаляется из списка. |
| Главный успешный сценарий: | | |
|  |  | 7. Список изображений не пуст. Отображается следующее изображение. |
| Расширение: | | |
|  |  | 7а. Список изображений пуст. Отображается placeholder. |
|  |  | 7а8. Компонент сообщает мобильному приложению о событии “Список изображений пуст”. |
|  | 7а9. Приложение обрабатывает событие от компонента “Список изображений пуст”. |  |

Для свайпа вправо поведение аналогичное.

При вызове метода “Вызов свайпа влево/вправо” поведение аналогичное, за исключением шага 1: инициатором жеста является не Пользователь, а Мобильное приложение.

В качестве источника данных используйте любой запрос API, возвращающий список фильмов.

Необходимо реализовать автоматические тесты для проверки работы компонента. Спроектируйте, опишите тест в виде сценария варианта использования и реализуйте его. Необходимо наличие 1 unit-теста и 1 UI-теста.

Подготовьте презентацию, описывающую реализованный компонент. Опишите структуру и поведение компонента с помощью диаграмм. Обоснуйте реализованную архитектуру. Расскажите о реализации отправки событий от компонента к мобильному приложению.

Целевая аудитория выступления - разработчики мобильных приложений. Продолжительность выступления - 5 минут.

Презентацию необходимо сохранить в репозитории с проектом.