

УДК 004.4

АНАЛИЗ ЭЛЕМЕНТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ ФРЕЙМВОРКА ЭЛЕКТРОННОГО МУЛЬТИМЕДИЙНОГО ИЗДАНИЯ

Чеканов И.О., Projects manager and Software developer, WPS Hong Kong

Марушевская А.А., студентка, кафедра МСТ ХНУРЭ

***Аннотация.** Рассмотрены элементы пользовательского интерфейса для разработки фреймворка электронного мультимедийного издания. Описаны основные группы элементов, такие как основные контейнеры, вспомогательные контейнеры и объекты различных типов.*

***Ключевые слова:** ИНТЕРФЕЙС, ФРЕЙМВОРК, ЭЛЕКТРОННОЕ МУЛЬТИМЕДИЙНОЕ ИЗДАНИЕ, КОНТЕЙНЕРЫ, ОБЪЕКТЫ.*

При разработке фреймворка, на начальном этапе необходимо определить структурные элементы, из которых будут состоять интерфейсы приложений, созданных с его помощью. Элементы интерфейса, реализованные в разрабатываемом фреймворке для создания электронного мультимедийного издания можно разделить на 3 основные группы:

- основные контейнеры;
- вспомогательные контейнеры;
- объекты разных типов.

Основные контейнеры имеют строгий уровень вложенности и определяют основу структуры приложения, они являются контейнерами, которые вмещают в себя элементы остальных двух групп.

К ним относятся:

- application (корневой элемент, вмещающий в себя все приложение). Этот контейнер содержит дочерние элементы, которые объявляют каждый из компонентов, входящих в состав приложения [1];

- pull-out_app_menu (выезжающее меню уровня «приложение»);
- page (представляет собой страницу электронного издания; контейнеров такого типа может быть несколько, в зависимости от общего количества страниц в разрабатываемом приложении);

- pull-out_page_menu (выезжающее меню уровня «страница»).

Вспомогательные контейнеры имеют не строгий уровень вложенности и могут вкладываться друг в друга. Ниже перечислены вспомогательные контейнеры, которые реализованы во фреймворке:

- layout_line (контейнер, который упорядочивает все дочерние элементы в одном направлении: по горизонтали или по вертикали. Все элементы расположены один за другим. Таким образом, если требуется расположить элементы в столбик, используется вертикальная ориентация, а если необходимо расположить элементы в строку – горизонтальная ориентация);

– `layout_grid` (контейнер, который позволяет создавать табличные представления. Состоит из коллекции строк и столбцов);

– `layout_wrap` (контейнер, который прижимает все элементы к одному углу в зависимости от того, какое значение имеет свойство «orientation») [2];

– `layout_constraint` (контейнер, который позволяет создавать гибкий интерфейс. Позиционирование может производиться относительно границ самого контейнера, в этом случае ограничение имеет значение "parent". Также позиционирование может осуществляться относительно любого другого элемента внутри контейнера, тогда в качестве значения ограничения указывается id данного элемента) [3];

– `scrollView` (контейнер, которые создаёт полосу прокрутки для такого интерфейса, все элементы которого одновременно не могут поместиться на экране устройства).

Третья группа элементов интерфейса – объекты разных типов, которые реализуют функционал отдельной страницы.

Ниже рассмотрен список элементов на примере страницы с тестовыми заданиями в электронном мультимедийном издании. Исходя из потребности в организации блока с тестами разных видов, могут быть выбраны следующие объекты:

– `ex_text` (объект, который позволяет создать предложение с несколькими текстовыми полями для проверки написанного текста в одном из полей);

– `ex_check` (объект, который создаёт текст с элементами «checkbox» для выбора нескольких правильных вариантов);

– `ex_radio` (объект, создающий текст с элементами «checkbox» для выбора одного правильного варианта);

– `ex_select` (объект, позволяющий создать предложение, где правильный ответ выбирается из выпадающего списка);

– `ex_pازل` (объект, который позволяет создать задание, где следует собрать пазл из деталей);

– `ex_link` (объект, организовывающий задание, в котором следует поставить соответствие между элементами из двух и более списков).

Таким образом, исходя из анализа элементов интерфейса для фреймворка, можно сделать вывод, что главной задачей на этапе проектирования является точное определение всех необходимых контейнеров и объектов для дальнейшего описания пользовательского интерфейса.

Литература

1. Структура элемента <application>. – Режим доступа: <http://androidfanclub.ru/programming/структура-элемента>. – 25.04.2017. – Загл. с экрана.

2. Контейнер `WrapPannel`. – Режим доступа: <https://metanit.com/sharp/wpf/4.6.php>. – 25.04.2017. – Загл. с экрана.

3. Контейнер `ConstraintLayout`. – Режим доступа: <https://metanit.com/java/android/3.8.php>. – 25.04.2017. – Загл. с экрана.

4. Дурняк Б. В. Стандарти в поліграфії та видавничій справі: довідник / Б. В. Дурняк, В. П. Ткаченко, І. Б. Чеботарьова // Львів: Українська академія друкарства, 2011. – 320 с.