7. Лабораторная работа № 1 «Пример разработки пользовательских интерфейсов приложения на платформе Android»

7.1. Цель лабораторной работы

Научиться базовым приемам разработки приложений на платформе Android, включающим в себя установку и настройку среды разработки, освоение программы с пользовательским интерфейсом (UI), проведение сеанса отладки.

7.2. Инструкция по выполнению лабораторной работы

7.2.1. Подготовка

Приложение, которое будет подробно рассмотрено, написано в IDE Eclipse Indigo, поэтому используйте именно эту версию или более позднюю. Также предполагается, что слушатели уже ознакомились с возможностью разрабатывать под Android в Eclipse и установили все требуемые плагины (см. Лекцию 3).

Пример имеет исключительно обучающий характер и нацелен на то, чтобы показать разработчику, процесс инициализации, описания внутреннего устройства компонентов UI и дизайна интерактивных приложений.

В лекциях были рассмотрены общие подходы к использованию основных компонентов UI и описаны вкратце внутреннее устройство каждого виджета. Изучаемое демонстрационное

приложение должно прояснить процесса создания и использования компонентов UI на практике.

7.2.2. Описание приложения

Общая идея. Обычно, если приложение не имеет сложную структуру, то в одном состоянии приложения используется только один стиль размещения. Мы рассмотрим приложение, которое содержит в себе много различных layouts. Это достигается путем создания в качестве главного layout, TabLayout, у которого каждый tab имеет свой способ размещения. Конечно, такая архитектура используется редко, так как пользователю приятнее видеть плавное переключение между табами в притивовес нашему варианту.

Структура.

TabLayout →

- Tab1: -> LinearLayout.
- Tab2: -> RelativeLayout.
- Tab3: -> GridLayout.

Архитектура Tab1.

Интерфейс для работы с датой в Андроиде.

Компоненты:

- TextView - простое текстовое поле, в котором будет отображаться текущая или вручную установленная дата.
- Button - кнопка, которая инициирует появление окошка DitaPicker, в котором отображается текущая дата.

В окне стандартно есть контролы, позволяющие изменить дату.

Архитектура Tab2.

В этом табе будут представлены некоторые виджеты UI и в качестве бонуса будет продемонстрирован Spinner и процесс взаимодействия с ним.

Компоненты:

- TextView - текстовое поле отображающее статическую информацию.
- EditText - текстовое поле с поддержкой редактирования.
- Button - кнопка OK.

- Button - кнопка Cancel.
- Spinner - выпадающий список с обработкой события выбора.

Архитектура Tab3.

В этом табе будут представлены графические объекты (картинки), размещенные в виде таблицы.

Компоненты:

- GridView — представление, отображающее элементы в двумерной сетке с поддержкой вертикального или горизонтального скрола. Элементы, ассоциированные с Grid представлением, должны быть представлены с помощью ListAdapter.

7.2.3. Модель «Дизайн-XML»

Two hands. Android SDK очень удобен, не только тем, что представляет понятный и прозрачный API, который позволяет неквалифицированному разработчику быстро освоить основные библиотеки Android и приступить к написанию собственных приложений, но и тем, что структура программы может быть разделена на две независимые части: Java-код, описывающий поведение программы и Xml-код, описывающий компоненты UI(кнопки, списки, окна, …).

Xml-подход. Такой способ был выбран по нескольким причинам:

- Разделение работы веб-дизайнера и программиста.
- Уменьшение массивности программы за счет использования фрагментов xml-файла в качестве виджетов UI.

7.2.4. Выполнение работы

Проделайте следующие шаги, чтобы посмотреть предварительно подготовленное приложение в действии.

Приложение может быть запущено с использованием виртуальной машины Davlik.

1). Импорт приложения.

Запустите Eclipse, выберите пункт меню «File», в списке выполните команду «Import», в появившемся окне во вкладке General, щелкните на «Existing project into workspace». Появиться диалоговое окно, в котором вы можете указать путь до проекта приложения либо в виде архива, либо в виде простой папки. Пример находится в каталоге lab01 файла labAtom21.rar. Затем нажимаете на кнопку «OK». В «Packageexplorer» у вас появится проект с исходным кодом.

2). Запуск приложения.

Нажмите на стрелку рядом с кнопкой «Run», в раскрывшемся списке выберите:

```
Run as->Androidapplication
```

В результате действий проект перестроится и запустится приложение. После загрузке эмулятора можно убедиться в работоспособности программы.

Файл графического интерфейса. Рассмотрим подробнее файл описания графического интерфейса приложения. Очень часто при написании приложений возникает необходимость разместить на экране несколько представлений, совершенно различных и не связанных друг с другом по логике. В этом случае программист сразу же представляет себе интерфейс расположения вкладок браузера. Такой интерфейс поддерживает и операционная система Android.

В папке layouts проекта, откройте с помощью xml редактора файл «tab_layout.xml», рассмотрим его содержимое.

Самый первый узел, «Tabhost»– представляет собой контейнер для вкладок. Он содержит всего два типа элементов: «TabWidget» и «FrameLayout». Элемент «TabWidget» используется для описания дизайна панели вкладок, содержит список вкладок. Менеджер размещения «FrameLayout» необходим при описании содержимого

одной вкладки. Для «TabHost» и «FrameLayout» заданы идентификаторы, так как в коде программы нам будет необходимо к ним обращаться. Менеджер размещения «LinearLayout» будет использоваться на первой вкладке, он содержит текстовое поле для отображения даты и кнопку, позволяющую установить дату. Следующий менеджер размещения «RelativeLayout» содержит несколько текстовых полей, поле с поддержкой редактирования, несколько кнопок и выпадающий список. И, наконец, последний менеджер размещения «LinearLayout» содержит «Grid» представление, для отображения картинок в таблице. Количество колонок жестко не задано, зато можно увидеть, что ширина каждой колонки 90 пикселей.

Код приложения. В файле «LabAndroidActivity.java» находится код, описывающий всю логику программы. Архитектура приложения основана на следующей идее: внутри одной activity обеспечить навигацию между represent-tab, используя «TabHost».

Прежде, чем переходить к подробному разбору кода, остановимся на вопросе «Жизненный цикл приложения».

В отличие от приложений в большинстве других систем приложения Android не имеют единой точки входа (например, отсутствует функция `main()`). Вместо этого существуют четыре типа основных компонентов, из которых строятся андроид-приложения и которые система может запускать по мере необходимости. Это Activity**,** Service**,** Broadcast receiver и Content provider. Как упоминалось раньше, Activity представляет собой визуальный интерфейс для одного действия, которое пользователь может совершить. Приложение обычно строится из нескольких Activity, каждый из которых выполняют свою определенную функцию. Обычно Activity не зависят друг от друга. Activity может находиться в одном из трех состояний:

- **Активно** (`active`) или **запущено** (`running`) – находится на переднем плане (на вершине стека activity текущей задачи) и имеет фокус для взаимодействия с пользователем.
- **Приостановлено** (`paused`) – потеряло фокус, но всё ещё видно пользователю. Сверху находится друго eactivity, которое или прозрачно или закрывает не весь экран. Приостановленное activity полностью <живое> (его

состояние сохранено и оно привязано к оконному менеджеру), но может быть уничтожено системой в случае нехватки памяти.

- **Остановлено** (`stopped`) – полностью перекрыто другим activity. Оно больше не видно пользователю и будет уничтожено системой, когда понадобится память.

Жизненный цикл activity состоит из трёх вложенных циклов:

- Жизненный цикл activity начинается с вызова метода `onCreate()`, в котором производится первоначальная настройка глобального состояния, и завершается вызовом метода `onDestroy()`, в котором оно освобождает занятые ресурсы. Например, в onCreate() можно создать поток, загружающий данные из сети в фоновом режиме, и затем остановить его в `onDestroy()`.
- Видимая часть жизненного цикла происходит между вызовами `onStart()` и `onStop()`. В течение этого времени пользователь может видеть activity на экране, хотя оно может быть не на переднем плане и не взаимодействовать с пользователем. Между этими двумя методами вы можете выделять ресурсы, необходимые для отображения activity пользователю. Методы `onStart()` и `onStop()` могут вызываться столько раз, сколько activity становится видимым или скрытым для пользователя.
- На переднем плане activity находится между вызовами `onResume()` и `onPause()`. В течение этого времени activity находится поверх других и взаимодействует с пользователем. Activity может часто переходить в состояние паузы и выходить из него. Например, метод `onPause()` может быть вызван, когда устройство переходит в спящий режим или когда запускается другое activity, а метод `onResume()` — при получении результата от закрывающегося activity. Таким образом, код в этих двух методах должен быть довольно легким.

Рассмотрев основные детали построения приложения, перейдем к обзору кода. В главном методе onCreate(), зададим в качестве содержимого Activity, созданный нами интерфейс, с помощью

вызова метода `setContentView` («Путь к файлу через класс ресурсов»). Очень часто при раздельном описании интерфейса и логики приложения, необходимо получить доступ к конкретному элементу интерфейса (кнопка), чтобы назначить ему определенную функцию. В таком случае, лучшим способом является использование метода `findViewById` (идентификатор). Параметр идентификатор – это имя компонента, определенное в файле интерфейса. Использование метода `setup()`, перед добавлением вкладок необходимо, если объект «TabHost», загружается через файл, как в нашем примере.

Далее используется TabSpec для создания вкладки, устанавливания индикатора и содержимого вкладки. В классе определена закрытая переменная mDateDisplay, в которой храниться время, и переменная mPickDate – кнопка, вызывающая диалог DatePicker. У кнопки установлен фон с авто-триггером, подключаемым из файла «Button.xml». Кнопке назначен слушатель на нажатие, внутри определения которого, реализован метод `onClick(View)`, вызываемый всякий раз при возникновении события. В методе `onClick()` вызывается функция `ShowDialog()`, которая влечет функцию `onCreateDialog()`. Обратите внимание, что внутри этой функции инициализируется объект DatePickerDialog, причем вторым параметром конструктору передается объект слушателя для события «изменение даты». В переменные mYear, mMonth, mDay заносится текущая дата и вызывается метод `UpdateDisplay()`, который обновляет содержимое текстового блока.

Аналогично, создается вторая вкладка и заполняется содержимым. Единственное, что стоит здесь отметить, это класс Spinner. Объект класса Spinner – выпадающий вниз список, с возможностью выбора элементов. В файле strings.xml определен тег string-array, содержащий элементы item. По сути, там содержатся данные, которые будут отображены в Спиннере. Метод createFromResource() создает новый ArrayAdapter, который связывает каждый элемент из массива «planets_array» с начальным представлением Спиннера. Иначе говоря, как каждый элемент будет отображаться в тот момент, когда пользователь надумает раскрыть Спиннер. Метод setDropDownViewResource, устанавливает

настройку «раскрытие-вниз» для Спиннера. И наконец, все элементы ArrayAdapter сопоставляются со Спиннером при помощи метода `setAdapter()`. Для Спиннера регистрируется слушатель на событие «Выбор элемента списка». Реализация объекта слушателя описана в отдельном классе с целью проведения аналогии читателем с предыдущим способом. Класс `MyOnItemSelectedListener` реализует интерфейс `OnItemSelectedListener`. В методе `onItemSelected()` пользователю показывается всплывающее окошко с информацией о выбранном элементе.

Последняя вкладка отображает элементы с помощью табличного менеджера компоновки, описанного в файле интерфейса. С помощью метода `setAdapter()`, назначаются данные для отображения. `ImageAdapter` - дочерний класс `BaseAdapter`. Массив `mThumbs` содержит ссылки на картинки через класс ресурсов. Аналогично, как и Спиннеру, `GridView` устанавливается слушатель на событие «Выбор элемента». Заканчивают метод `onCreate()` три строки кода, в которых контейнеру табов добавляются созданные вкладки, по очереди.