

Основы разработки многооконных приложений

Для мобильных приложений главным ограничением является размер экрана устройства. Очень часто невозможно разместить все элементы полнофункционального приложения так, чтобы их можно было увидеть одновременно. Очевидным решением этой проблемы является разделение интерфейса на части по какому-либо принципу. Основные пути решения этой проблемы:

- Использовать различные сообщения (диалоговые окна, уведомления, всплывающие подсказки). Этот способ наиболее прост и не требует редактирования файла манифеста, однако очевидно, что так можно решить только часть задач.
- Использовать в одном приложении несколько активностей. Способ универсальный и подходит для любых приложений, однако прежде чем его реализовывать, необходимо очень хорошо продумать структуру будущего приложения. Здесь требуется редактировать манифест и организовать переключение между различными активностями удобным для пользователя способом.
- Разместить компоненты на активности таким образом, что в нужный момент можно будет легко переключиться на работу с другой частью интерфейса.

Диалоговые окна	Несколько активностей	Swipe
<ul style="list-style-type: none">• Взаимодействие с пользователем с помощью сообщений• Можно использовать уведомления и всплывающие подсказки• Не всегда подходит	<ul style="list-style-type: none">• Универсальный способ• Требуется редактирование файла манифеста• Необходимо продумать организацию перемещения между активностями	<ul style="list-style-type: none">• Одна активность, несколько экранов• Легкое переключение между экранами одним из двух способов (кнопка или слайдинг)

Диалоговые окна

Диалог - это небольшое окно, позволяющее пользователю получить или ввести дополнительную информацию. Диалоговое окно занимает только часть экрана и обычно используется в модальном режиме. Это означает, что работа приложения приостанавливается до момента, пока пользователь не закроет диалоговое окно. При

этом ему, возможно, потребуется ввести какие-то данные или просто выбрать один из вариантов ответа.

Примеры диалоговых окон

В ОС Android можно выделить три вида диалоговых окон.

1. **Класс Dialog и его производные.** Помимо традиционного набора диалоговых окон, он содержит несколько дополнительных вариантов, в которых используются возможности сенсорного интерфейса. Диалоги этого типа не создают новых активностей и их не нужно регистрировать в файле манифеста (см. следующие разделы лекции), что существенно упрощает разработку. Однако они работают в модалном режиме и требуют немедленного ответа пользователя, поэтому для простого информирования рекомендуется использовать сообщения следующих двух типов.
2. **Уведомления (notifications).** Это сообщения, которые отображаются в верхней панели в области уведомлений. Для того чтобы прочитать это сообщение, необходимо на домашнем экране потянуть вниз верхнюю шторку. Пользователь может это сделать в любой момент времени, следовательно, уведомления стоит использовать, когда сообщение является важным, однако не требует немедленного прочтения и ответа.
3. **Всплывающие подсказки (toasts).** Сообщения, которые появляются прямо на экране приложения, перекрывая его интерфейс, и через некоторое время (обычно несколько секунд) автоматически пропадают. Их рекомендуется использовать для простых уведомлений, не требующих ответа пользователя, но важных для продолжения его работы.

Использование класса Dialog

Класс Dialog является базовым для диалогов и редко используется напрямую. Рекомендуется применять производные от этого класса.

1. **AlertDialog.** Диалоговое окно может содержать заголовок, до трех кнопок, список выбираемых значений или настраиваемое содержимое.
2. **DatePickerDialog** или **TimePickerDialog.** Диалоговое окно с предопределенным интерфейсом, позволяющее выбрать дату или время.
3. **ProgressDialog.** Показывает диалоговое окно, содержащее линейку процесса выполнения какого-то действия. В рекомендациях по дизайну для Android советуют использовать вместо него компонент ProgressBar

Существует возможность создавать собственные диалоговые окна с использованием класса DialogFragment в качестве контейнера. В таком случае можно контролировать его поведение. Обратите внимание, что минимальной версией,

поддерживающей DialogFragment напрямую, является Android 3.0 (API level 11). Если вы хотите использовать возможности этого класса на более ранних версиях, необходимо добавить библиотеку Support Library в ваше приложение.

Существует множество вариантов диалоговых окон класса AlertDialog, однако все они содержат следующие три части:

Компоновка диалогового окна (класс AlertDialog)

1. Заголовок. Не является обязательным элементом и должен быть использован, только если содержательная часть занята детализированным сообщением, списком или чем-то еще. Если нужно сделать небольшое сообщение или вопрос, не стоит снабжать его выделенным заголовком.
2. Содержательная часть. Здесь может быть сообщение, список или какой-то другой настраиваемый компонент.
3. Управляющие кнопки. Диалог может содержать не больше трех кнопок. Если элементы содержательной части являются кликабельными, можно вообще обойтись без кнопок.

Уведомления

Уведомления являются неотъемлемой частью дизайна Android-приложений.

Существует два варианта отображения уведомлений в развернутом виде - нормальный и расширенный (доступен начиная с Android 4.1).

Высота уведомления составляет 64 dp. Уведомление содержит следующие части:

1. Заголовок.
2. Большая иконка.
3. Текст сообщения.
4. Информация о сообщении.
5. Маленькая иконка приложения.
6. Время (или дата), когда было отправлено сообщение.

Уведомление появляется в расширенном виде, только если оно находится вверху списка уведомлений, или же когда пользователь сделал жест с целью его увеличения. Расширенное уведомление включает в себя те же пункты, что и обычное, но при этом дополнительно содержит детализированную область. Это может быть, например, картинка до 256 dp высотой, блок текстовой информации или что-то еще.

Всплывающие подсказки

Всплывающие подсказки помогают отобразить обратную связь с действиями пользователя. Они занимают минимум места на экране и быстро исчезают. Поэтому рекомендуется использовать их для простого уведомления пользователя, когда не требуется получить от него ответа. Всплывающие подсказки могут появляться в любом месте экрана, что позволяет делать их работу более эффективной.

Особенности разработки приложения, содержащего несколько активностей

Приложения, содержащие несколько активностей, используются в самых разных сферах. При проектировании такого приложения следует уделить большое внимание распределению его функционала по разным активностям. С одной стороны, не стоит перегружать экран информацией, а с другой - нужна ли активность, содержащая только одно поле для ввода? Может быть, стоит ее заменить диалоговым окном? Существует два основных способа переключения между активностями:

1. При помощи кнопок и других элементов управления. Не требует перестройки мышления у программистов, которые имеют большой опыт разработки десктопных приложений, а так же у пользователей, привыкших к действиям в стиле "нажал на кнопку, получил результат". Однако этот способ не является наиболее подходящим для сенсорных экранов и требует от опытного пользователя смартфона совершения лишних движений.
2. С использованием сенсорного экрана смартфона. Основная идея состоит в том, что весь экран мобильного устройства можно использовать в качестве управляющего элемента, и, нажимая на отдельные его участки, пользователь может инициировать те или иные действия.

Какой из способов выбрать, зависит от конкретной задачи.

Существует ряд правил расположения интерфейсных элементов в зависимости от их важности. Так, кнопку, выполняющую важное действие (например, отправку письма), не стоит располагать в том месте, где она может быть случайно нажата. В то же время управляющие элементы, используемые наиболее часто, должны быть расположены наиболее удобным для нажатия образом. Скорее всего, перемещение между активностями будет использоваться не очень часто, поэтому рекомендуется располагать кнопки, управляющие этими действиями, в верхней части экрана. Одновременно с этим неплохо продублировать нажатия кнопок перелистыванием между активностями.

В любом случае для вызова другой активности необходимо вручную править файл манифеста.

Для каждой новой активности необходимо занести информацию о ее имени и названии xml-файла, в котором она описана.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.myproject.screen"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.myproject.screen.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.myproject.screen.AboutActivity"
            android:label="@string/about_title" >
        </activity>
        <activity
            android:name="com.myproject.screen.SecondActivity"
            android:label="@string/title_activity_second" >
        </activity>
    </application>
</manifest>
```

Обратите внимание, что при загрузке приложения первой появляется активность, чье описание находится первым в манифесте! Если вы хотите изменить порядок загрузки активностей, необходимо поместить новую активность на первое место.

Перелистывание (Swipe)

Существует способ разместить на активности больше элементов, чем одновременно помещается на экран, иными словами, отображать по очереди несколько экранов, используя только одну активность. В этом случае не нужно править файл манифеста - активность только одна. Однако для каждого экрана необходимо сделать свой xml-файл с его описанием.

Такой способ размещения элементов удобен и программисту, и пользователю. Разработчик может организовать перемещение между частями активности и с помощью кнопок, и с помощью перелистывания. В качестве примера подобного интерфейса можно привести приложение Twitter. В нем можно переключаться между экранами, используя жест горизонтальной прокрутки или с помощью кнопок в верхней части.