

ЛЕКЦІЯ 9. ОСНОВИ РОЗРОБКИ БАГАТОВІКОННИХ ДОДАТКІВ

9.1 ДОДАТКИ З ДЕКІЛЬКОМА ВІКНАМИ

Для мобільних додатків головним обмеженням є розмір екрану пристрою. Дуже часто неможливо розмістити всі елементи повнофункціонального додатку так, щоб їх можна було побачити одночасно. Очевидним вирішенням цієї проблеми є поділ інтерфейсу на частини з якого-небудь принципом. Основні шляхи вирішення цієї проблеми:

- Використовувати різні повідомлення (діалогові вікна, повідомлення, спливаючі підказки). Цей спосіб найбільш простий і не вимагає редагування файлу маніфесту, проте очевидно, що так можна вирішити тільки частину завдань.
- Використовувати в одному додатку кілька активностей. Спосіб універсальний і підходить для будь-яких додатків, однак перш ніж його реалізовувати, необхідно дуже добре продумати структуру майбутнього програми. Тут потрібне редагувати маніфест і організувати перемикання між різними активностями зручним для користувача способом.
- Розмістити компоненти на активності таким чином, що в потрібний момент можна буде легко переключитися на роботу з іншою частиною інтерфейсу.

Кожен спосіб має свої нюанси використання. Розглянемо їх детальніше.

9.2 РОБОТА З ДІАЛОГОВИМИ ВІКНАМИ

9.2.1 ДІАЛОГОВІ ВІКНА

Діалог - це невелике вікно, що дозволяє користувачеві отримати або ввести додаткову інформацію. Діалогове вікно займає тільки частину екрану і зазвичай використовується в модальному режимі. Це означає, що робота додатка припиняється до моменту, поки користувач не закриє діалогове вікно. При цьому йому, можливо, буде потрібно ввести якісь дані або просто вибрати один з варіантів відповіді.

В ОС Android можна виділити три види діалогових вікон:

- Клас Dialog і його похідні. Крім традиційного набору діалогових вікон, він містить кілька додаткових варіантів, в яких використовуються можливості сенсорного інтерфейсу. Діалоги цього типу не створюють нових активностей і їх не

потрібно реєструвати у файлі маніфесту (див. Наступні розділи лекції), що істотно спрощує розробку. Однак вони працюють в модальному режимі і вимагають негайної відповіді користувача, тому для простого інформування рекомендується використовувати повідомлення наступних двох типів.

- **Повідомлення (notifications).** Це повідомлення, які відображаються у верхній панелі в області сповіщень. Для того щоб прочитати це повідомлення, необхідно на домашньому екрані потягнути вниз верхню шторку. Користувач може це зробити в будь-який момент часу, отже, повідомлення варто використовувати, коли повідомлення є важливим, проте не вимагає негайного прочитання і відповіді.
- **Спливаючі підказки (toasts).** Повідомлення, які з'являються прямо на екрані додатки, перекриваючи його інтерфейс, і через деякий час (зазвичай кілька секунд) автоматично пропадають. Їх рекомендується використовувати для простих повідомлень, які не потребують відповіді користувача, але важливих для продовження його роботи.

9.2.2 ВИКОРИСТАННЯ КЛАСУ `DIALOG`

Клас `Dialog` є базовим для діалогів і рідко використовується безпосередньо. Рекомендується застосовувати похідні від цього класу:

- `AlertDialog`. Діалогове вікно може містити заголовок, до трьох кнопок, список обраних значень або настраюється вміст.
- `DatePickerDialog` або `TimePickerDialog`. Діалогове вікно з наперед визначеним інтерфейсом, що дозволяє вибрати дату або час.
- `ProgressDialog`. Показує діалогове вікно, що містить лінійку процесу виконання якоїсь дії. У рекомендаціях по дизайну для Android радять використовувати замість нього компонент `ProgressBar`.

Існує можливість створювати власні діалогові вікна з використанням класу `DialogFragment` в якості контейнера. У такому випадку можна контролювати його поведінку. Зверніть увагу, що мінімальний версією, що підтримує `DialogFragment` безпосередньо, є Android 3.0 (API level 11). Якщо ви хочете використовувати можливості цього класу на більш ранніх версіях, необхідно додати бібліотеку `Support Library` в вашу програму.

Розглянемо створення діалогового вікна на прикладі класу AlertDialog. Існує безліч варіантів діалогових вікон цього класу, проте всі вони містять такі три частини:

1. Тема. Не є обов'язковим елементом і має бути використаний, тільки якщо змістовна частина зайнята деталізованим повідомленням, списком або чимось ще. Якщо потрібно зробити невелике повідомлення або запитання, не варто постачати його виділеним заголовком.
2. Змістовна частина. Тут може бути повідомлення, список або якийсь інший настраюється компонент.
3. Керуючі кнопки. Діалог може містити не більше трьох кнопок. Якщо елементи змістовної частини є розмітки, можна взагалі обійтися без кнопок.

9.2.3 ПОВІДОМЛЕННЯ

Повідомлення є невід'ємною частиною дизайну Android-додатків.

Існує два варіанти відображення сповіщень в розгорнутому вигляді - нормальний і розширений (доступний починаючи з Android 4.1).

Склад повідомлення в нормальному вигляді представлений на рис.9.1 . Висота повідомлення становить 64 dp. Повідомлення містить такі частини:

1. Тема.
2. Велика іконка.
3. Текст повідомлення.
4. Інформація про повідомленні.
5. Маленька іконка програми.
6. Час (або дата), коли було відправлено повідомлення.



Рис. 9.1. Стандартне повідомлення

Повідомлення з'являється в розширеному вигляді, тільки якщо воно знаходиться вгорі списку повідомлень, або ж коли користувач зробив жест з метою його збільшення. Розширене оповіщення (див. рис. 9.2) включає в себе ті ж пункти, що і звичайне, але при цьому додатково містить деталізовану область (на малюнку відзначено номером 7). Це може бути, наприклад, картинка до 256 dp висотою, блок текстової інформації або щось ще.

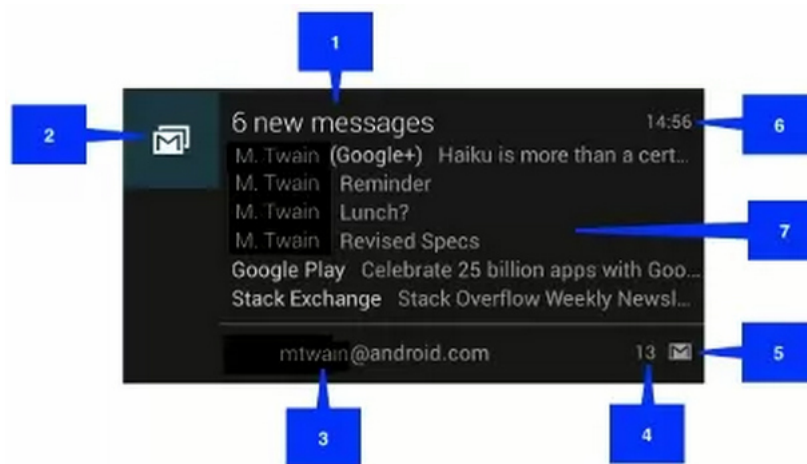


Рис. 9.2. Розширене оповіщення

9.2.4 ПІДКАЗКИ

Спливаючі підказки допомагають відобразити зворотний зв'язок з діями користувача. Вони займають мінімум місця на екрані і швидко зникають. Тому рекомендується використовувати їх для простого повідомлення користувача, коли не потрібно отримати від нього відповіді. Спливаючі підказки можуть з'являтися в будь-якому місці екрану, що дозволяє робити їх роботу більш ефективною.

9.3 ОСОБЛИВОСТІ РОЗРОБКИ ПРОГРАМИ, ЩО МІСТИТЬ КІЛЬКА АКТИВНОСТЕЙ

Додатки, що містять кілька активностей, використовуються в самих різних сферах. При проектуванні такого додатка слід приділити велику увагу розподілу його функціоналу за різними активностями. З одного боку, не варто перевантажувати екран інформацією, а з іншого - чи потрібна активність, яка містить тільки одне поле для введення? Можливо, варто її замінити діалоговим вікном?

Існує два основних способи перемикання між активностями:

- За допомогою кнопок та інших елементів управління. Не вимагає перебудови мислення у програмістів, які мають великий досвід розробки десктопних додатків, а так само у користувачів, звиклих до дій в стилі "натиснув на кнопку, отримав результат". Однак цей спосіб не є найбільш підходящим для сенсорних екранів і вимагає від досвідченого користувача смартфона вчинення зайвих рухів.

- З використанням сенсорного екрану смартфона. Основна ідея полягає в тому, що весь екран мобільного пристрою можна використовувати в якості керуючого елемента, і, натискаючи на окремі його ділянки, користувач може ініціювати ті чи інші дії. Більш докладно можливості жестової інтерфейсу будуть розглянуті далі, в цій та наступній темах.

Який із способів вибрати, залежить від конкретного завдання.

Існує ряд правил розташування інтерфейсних елементів залежно від їх важливості. Так, кнопку, що виконує важливе дію (наприклад, відправку листа), не варто розташовувати в тому місці, де вона може бути випадково натиснута. У той же час керуючі елементи, використовувані найбільш часто, повинні бути розташовані найбільш зручним для натискання чином. Швидше за все, переміщення між активностями буде використовуватися не дуже часто, тому рекомендується розташовувати кнопки, керуючі цими діями, у верхній частині екрана. Одночасно з цим непогано продублювати натискання кнопок перегортанням між активностями.

У будь-якому випадку для виклику іншої активності необхідно вручну правити файл маніфесту. Для кожної нової активності необхідно занести інформацію про її імені та назві xml-файлу, в якому вона описана (див. лістинг 9.1). Зверніть увагу, що при завантаженні програми першим з'являється активність, чиє опис знаходиться першим у маніфесті! Якщо ви хочете змінити порядок завантаження активностей, необхідно помістити нову активність на перший місце.

```
<? 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>

```

Лістинг 9.1. Виправлений файл маніфесту

9.4 ПЕРЕГОРТАННЯ (SWIPE)

Існує спосіб розмістити на активності більше елементів, ніж одночасно поміщається на екран, іншими словами, відображати по черзі кілька екранів, використовуючи тільки одну активність. У цьому випадку не потрібно правити файл маніфесту - активність тільки одна. Однак для кожного екрану необхідно зробити свій xml- файл з його описом.

Такий спосіб розміщення елементів зручний і програмісту, і користувачеві. Розробник може організувати переміщення між частинами активності і за допомогою кнопок, і за допомогою перегортання. Як приклад подібного інтерфейсу можна привести додаток Twitter.