LẬP TRINH TREN ANDROID

Part 2

Ví dụ 3. Lớp trình phân tích điểm tin cơ bản

```
public abstract class BaseFeedParser implements
FeedParser {

    // names of the XML tags
    static final String PUB_DATE = "pubDate";
    static final  String DESCRIPTION = "description";
    static final  String LINK = "link";
    static final  String TITLE = "title";
    static final  String ITEM = "item";

    final URL feedUrl;

    protected BaseFeedParser(String feedUrl){
        try {
            this.feedUrl = new URL(feedUrl);
        } catch (MalformedURLException e) {
            throw new RuntimeException(e);
        }
    }

    protected InputStream getInputStream() {
        try {
            return
feedUrl.openConnection().getInputStream();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Lớp cơ sở lưu trữ `feedUrl` và sử dụng nó để mở `java.io.InputStream`. Nếu có bất kỳ sai sót nào, đơn giản nó thả một `RuntimeException`, sao cho

ứng dụng dừng hoạt động một cách nhanh chóng. Lớp cơ sở cũng xác định một vài hằng số đơn giản cho tên các thẻ. Ví dụ 4 trình bày một số nội dung mẫu từ điểm tin, qua đó bạn có thể thấy được ý nghĩa của các thẻ này.

Ví dụ 4. Điểm tin XML mẫu

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- generator="FeedCreator 1.7.2" -->

<rss version="2.0">

  <channel>

    <title>android_news</title>

    <description>android_news</description>


<link>http://www.androidster.com/android_news.php</link
>

    <lastBuildDate>Sun,    19    Apr    2009    19:43:45
+0100</lastBuildDate>

    <generator>FeedCreator 1.7.2</generator>

    <item>

      <title>Samsung S8000 to Run Android, Play DivX,
Take Over the
World</title>


<link>http://www.androidster.com/android_news/samsung-
s8000-to-run-android-
play-divx-take-over-the-world</link>

      <description>More details have emerged on the
```

```
first Samsung handset
to run Android. A yet-to-be announced phone called the
S8000 is being
reported ...</description>

      <pubDate>Thu,     16     Apr     2009     07:18:51
+0100</pubDate>

    </item>

    <item>

      <title>Android    Cupcake    Update    on    the
Horizon</title>


<link>http://www.androidster.com/android_news/android-
cupcake-update-
on-the-horizon</link>

      <description>After   months   of   discovery   and
hearsay, the Android
build that we have all been waiting for is about to
finally make it
out ...</description>

      <pubDate>Tue,     14     Apr     2009     04:13:21
+0100</pubDate>

    </item>

  </channel>

</rss>
```

Như bạn có thể thấy từ mẫu trong Ví dụ 4, một `ITEM` tương đương với một thể hiện `Message`. Các nút con của mục chọn (`TITLE`, `LINK` và v.v..) tương đương các đặc tính của thể hiện `Message`. Vì bạn biết điểm tin trông như thế nào rồi và

có sẵn tất cả các phần phổ biến, hãy xem làm thế nào để phân tách điểm tin này sử dụng các công nghệ khác nhau có sẵn trên Android. Bạn sẽ bắt đầu với SAX.

Sử dụng SAX

Trong môi trường Java, bạn có thể thường xuyên sử dụng SAX API khi bạn muốn có một trình phân tích nhanh và muốn hạn chế tối đa việc sử dụng (footprint) bộ nhớ ứng dụng của bạn. Điều đó khiến cho cho nó rất phù hợp cho thiết bị di động chạy Android. Bạn có thể sử dụng SAX API như là từ môi trường Java, mà không cần đến những thay đổi đặc biệt cần thiết để chạy trên Android.Ví dụ 5 trình bày một thực thi SAX của giao diện `FeedParser`.

Ví dụ 5. Thực thi SAX

```
public class SaxFeedParser extends BaseFeedParser {

    protected SaxFeedParser(String feedUrl){
        super(feedUrl);
    }

    public List<Message> parse() {
        SAXParserFactory factory =
SAXParserFactory.newInstance();
        try {
            SAXParser parser = factory.newSAXParser();
            RssHandler handler = new RssHandler();
            parser.parse(this.getInputStream(),
handler);
            return handler.getMessages();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

Nếu trước đây bạn đã sử dụng SAX, thì cái này trông cũng khá quen thuộc. Như với bất kỳ thực thi SAX nào, phần lớn các chi tiết đều nằm trong trình xử lý SAX.

Trình xử lý nhận các sự kiện từ trình phân tích SAX khi nó chạy nhanh qua tài liệu XML. Trong trường hợp này, bạn vừa tạo ra một lớp mới gọi là `RssHandler` và đăng ký nó như là một trình xử lý cho trình phân tích, như trong Ví dụ 6.