

2.2.2 部分的なハンドラの変更

部分的なハンドラの変更

- XMLFilter
 - イベントをフィルタリング
- XML文書内のまとまったブロックを別のハンドラで処理
 - 随時イベントハンドラを切り替える
 - XMLFilterを応用

XMLFilterを使用することによって、特定の名前空間でフィルタリングしたり、あるいはその他の条件で各イベントをフィルタリングすることができます。

一方XMLFilterをさらに応用して、XML文書内のあるまとまったブロックをまとめて別のハンドラで処理するような場合に、イベントハンドラを丸ごと切り替える方法もあります。

異なる2つのContentHandler

「maindoc_NS.xml」に対して、名前空間「urn:example:MAIN」部分処理するハンドラと、名前空間「urn:example:SUB」部分処理するハンドラの2つのContentHandlerを使用することにします。

はじめに、2つのContentHandlerを別々のプログラムで実行することによって、どのような不具合が生じるかを確認します。

【入力XML】

ファイル名：maindoc_NS.xml

```

01 : <?xml version="1.0" encoding="Shift_JIS"?>
02 : <maindoc xmlns="urn:example:MAIN">
03 :   <body>
04 :     <content>XMLコースの紹介</content>
05 :     <content>** ベーシック **</content>
06 :     <subdoc xmlns="urn:example:SUB">
07 :       <summary>XMLの基礎を学習する</summary>
08 :       <body section="1">XMLの概要</body>
09 :       <body section="2">XMLの構文</body>
10 :     </subdoc>
11 :   </body>
12 : </maindoc>

```

【リスト9a】 (部分)

本来、名前空間「urn:example:MAIN」部分を処理するハンドラです。要素の内容文字列を表示しているだけで、特に属性値の出力はありません。

ファイル名 : SAXSample09a.java

```
26 : // ContentHandler実装クラス
27 : class HandlerImpl09a extends DefaultHandler {
28 :
29 : // 文字列を保存するための変数を用意
30 : String str;
31 :
32 : public void startElement( String namespaceURI, String localName,
33 :                          String qName, Attributes atts ) throws SAXException {
34 : // strを初期化
35 : str = "";
36 : }
37 : public void endElement( String namespaceURI, String localName,
38 :                        String qName ) throws SAXException {
39 : // strを表示
40 : if (!str.trim().equals("")) System.out.println( str );
41 : // strを初期化
42 : str = "";
43 : }
44 : public void characters( char ch[], int start, int length )
45 :                        throws SAXException {
46 : // 文字データをstrに追加
47 : str += new String( ch, start, length );
48 : }
49 : }
```

【実行結果】

```
C:¥icec>java SAXSample09a
XMLコースの紹介
** ベーシック **
XMLの基礎を学習する
XMLの概要
XMLの構文

C:¥icec>
```

属性値を出力するコードがありませんので、本来表示したい、subdoc要素内のbody要素のsection属性値が表示されていません。

【リスト9b】 (部分)

本来、名前空間「urn:example:SUB」部分を処理するハンドラです。要素の内容文字列を表示し、そしてbody要素の開始タグでsection属性値を表示しています。

ファイル名：SAXSample09b.java

```

26 : // ContentHandler実装クラス
27 : class HandlerImpl09b extends DefaultHandler {
28 :
29 : // 文字列を保存するための変数を用意
30 : String str;
31 :
32 : public void startElement( String namespaceURI, String localName,
33 :     String qName, Attributes atts ) throws SAXException {
34 : // strを初期化
35 :     str = "";
36 : // section属性値の表示
37 :     if (qName.equals("body"))
38 :         System.out.println( "セクション" + atts.getValue("section") );
39 : }
40 : public void endElement( String namespaceURI, String localName,
41 :     String qName ) throws SAXException {
42 : // strを表示
43 :     if (!str.trim().equals("")) System.out.println( str );
44 : // strを初期化
45 :     str = "";
46 : }
47 : public void characters( char ch[], int start, int length )
48 :     throws SAXException {
49 : // 文字データをstrに追加
50 :     str += new String( ch, start, length );
51 : }
52 : }

```

【実行結果】

```

C:¥icec>java SAXSample09b
セクションnull
XMLコースの紹介
** ペーシック **
XMLの基礎を学習する
セクション1
XMLの概要
セクション2
XMLの構文

C:¥icec>

```

subdoc要素内のbody要素のsection属性値が表示されていますが、section属性を持たないbody要素で「セクションnull」が出力されています。