

```
1: import javax.xml.parsers.SAXParser;
2: import javax.xml.parsers.SAXParserFactory;
3: import org.xml.sax.Attributes;
4: import org.xml.sax.ContentHandler;
5: import org.xml.sax.SAXException;
6: import org.xml.sax.helpers.AttributesImpl;
7: import org.xml.sax.helpers.DefaultHandler;
8:
9:
10: /**
11:  * Lecture 2's demonstration of SAX 2.0.2
12:  * (that escapes whitespace characters).
13:  *
14:  * @author Computer Science E-259
15:  */
16:
17: public class SAXDemo2 extends DefaultHandler
18: {
19:     /**
20:      * Main driver. Expects one command-line argument:
21:      * the name of the file to parse.
22:      *
23:      * @param argv [0] - filename
24:      */
25:
26:     public static void main(String [] argv)
27:     {
28:         // ensure proper usage
29:         if (argv.length != 1)
30:         {
31:             System.out.println("Usage: java SAXDemo filename");
32:             System.exit(1);
33:         }
34:
35:         // grab filename
36:         String input = argv[0];
37:
38:         try
39:         {
40:             // instantiate a SAX parser
41:             SAXParserFactory factory = SAXParserFactory.newInstance();
42:             SAXParser parser = factory.newSAXParser();
43:
44:             // instantiate our little demo handler
45:             SAXDemo2 handler = new SAXDemo2();
46:
47:             // parse the file
48:             parser.parse(input, handler);
49:
50:         }
51:         catch (Exception e)
52:         {
53:             e.printStackTrace();
54:         }
55:     }
56:
57:
58:     /**
59:      * Report a startElement event.
60:      *
61:      * @param uri namespace
62:      * @param localName name of element, sans namespace
```

```
63:      * @param qName name of element, with namespace
64:      * @param attributes element's collection of attributes
65:      *
66:      * @throws SAXException general SAX error or warning
67:      */
68:
69:     public void startElement(String uri, String localName,
70:                             String qName, Attributes attrs)
71:         throws SAXException
72:     {
73:         System.out.print("startElement(\"" + qName + "\", {");
74:         for (int i = 0; i < attrs.getLength(); i++)
75:         {
76:             System.out.print("(" + attrs.getQName(i) + "\", \"" +
77:                               attrs.getValue(i) + "\");");
78:             if (i != attrs.getLength() - 1)
79:                 System.out.print(", ");
80:         }
81:         System.out.println("}");
82:     }
83:
84:
85:     /**
86:      * Report a characters event.
87:      *
88:      * @param ch characters
89:      * @param start start position in the character array
90:      * @param length number of characters to use from the character array
91:      *
92:      * @throws SAXException general SAX error or warning
93:      */
94:
95:     public void characters(char[] ch, int start, int length)
96:         throws SAXException
97:     {
98:         // store characters in a String object
99:         String s = new String(ch, start, length);
100:
101:         // escape whitespace characters
102:         s = s.replaceAll("\f", "\\f");
103:         s = s.replaceAll("\n", "\\n");
104:         s = s.replaceAll("\r", "\\r");
105:         s = s.replaceAll("\t", "\\t");
106:
107:         // output results
108:         System.out.println("characters(\"" + s + "\");");
109:     }
110:
111:
112:     /**
113:      * Report an endElement event.
114:      *
115:      * @param uri namespace
116:      * @param localName name of element, sans namespace
117:      * @param qName name of element, with namespace
118:      *
119:      * @throws SAXException general SAX error or warning
120:      */
121:
122:     public void endElement(String uri, String localName, String qName)
123:         throws SAXException
124:     {
```

```
125:         System.out.println("endElement(\"" + qName + "\");");
126:     }
127:
128:
129:     /**
130:      * Report a startDocument event.
131:      */
132:
133:     public void startDocument() throws SAXException
134:     {
135:         System.out.println("\nstartDocument();");
136:     }
137:
138:
139:     /**
140:      * Report an endDocument event.
141:      *
142:      * @throws SAXException    general SAX error or warning
143:      */
144:
145:     public void endDocument() throws SAXException
146:     {
147:         System.out.println("endDocument();\n");
148:     }
149: }
```

```
1: import javax.xml.parsers.SAXParser;
2: import javax.xml.parsers.SAXParserFactory;
3: import org.xml.sax.Attributes;
4: import org.xml.sax.ContentHandler;
5: import org.xml.sax.SAXException;
6: import org.xml.sax.helpers.AttributesImpl;
7: import org.xml.sax.helpers.DefaultHandler;
8:
9:
10: /**
11:  * Lecture 2's demonstration of SAX 2.0.2.
12:  *
13:  * @author Computer Science E-259
14:  */
15:
16: public class SAXDemo extends DefaultHandler
17: {
18:     /**
19:      * Main driver. Expects one command-line argument:
20:      * the name of the file to parse.
21:      *
22:      * @param argv [0] - filename
23:      */
24:
25:     public static void main(String [] argv)
26:     {
27:         // ensure proper usage
28:         if (argv.length != 1)
29:         {
30:             System.out.println("Usage: java SAXDemo filename");
31:             System.exit(1);
32:         }
33:
34:         // grab filename
35:         String input = argv[0];
36:
37:         try
38:         {
39:             // instantiate a SAX parser
40:             SAXParserFactory factory = SAXParserFactory.newInstance();
41:             SAXParser parser = factory.newSAXParser();
42:
43:             // instantiate our little demo handler
44:             SAXDemo handler = new SAXDemo();
45:
46:             // parse the file
47:             parser.parse(input, handler);
48:
49:         }
50:         catch (Exception e)
51:         {
52:             e.printStackTrace();
53:         }
54:     }
55:
56:
57:     /**
58:      * Report a startElement event.
59:      *
60:      * @param uri namespace
61:      * @param localName name of element, sans namespace
62:      * @param qName name of element, with namespace
```

```
63:      * @param attributes element's collection of attributes
64:      *
65:      * @throws SAXException general SAX error or warning
66:      */
67:
68:     public void startElement(String uri, String localName,
69:                             String qName, Attributes atts)
70:         throws SAXException
71:     {
72:         System.out.print("startElement(\"" + qName + "\", {");
73:         for (int i = 0; i < atts.getLength(); i++)
74:         {
75:             System.out.print("(" + atts.getQName(i) + "\", \"" +
76:                               atts.getValue(i) + "\");");
77:             if (i != atts.getLength() - 1)
78:                 System.out.print(", ");
79:         }
80:         System.out.println("}");
81:     }
82:
83:
84:     /**
85:      * Report a characters event.
86:      *
87:      * @param ch characters
88:      * @param start start position in the character array
89:      * @param length number of characters to use from the character array
90:      *
91:      * @throws SAXException general SAX error or warning
92:      */
93:
94:     public void characters(char[] ch, int start, int length)
95:         throws SAXException
96:     {
97:         System.out.println("characters(\"" + new String(ch, start, length) +
98:                             "\");");
99:     }
100:
101:
102:     /**
103:      * Report an endElement event.
104:      *
105:      * @param uri namespace
106:      * @param localName name of element, sans namespace
107:      * @param qName name of element, with namespace
108:      *
109:      * @throws SAXException general SAX error or warning
110:      */
111:
112:     public void endElement(String uri, String localName, String qName)
113:         throws SAXException
114:     {
115:         System.out.println("endElement(\"" + qName + "\");");
116:     }
117:
118:
119:     /**
120:      * Report a startDocument event.
121:      */
122:
123:     public void startDocument() throws SAXException
124:     {
```

```
125:         System.out.println("\nstartDocument();");
126:     }
127:
128:
129:     /**
130:      * Report an endDocument event.
131:      *
132:      * @throws SAXException    general SAX error or warning
133:      */
134:
135:     public void endDocument() throws SAXException
136:     {
137:         System.out.println("endDocument();\n");
138:     }
139: }
```