

DOMTest.java

```
/* Code adapted from Chris Mair */

import java.io.File;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DOMTest {

    static private Document doc;
    static String xmlFile = "/Users/nutt/local/FUB-MeineLehre/"
        + "XML-Data Management/Daten/Books/books.xml";

    public static void main(String[] args) throws Exception {

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

        DocumentBuilder builder = factory.newDocumentBuilder();

        doc = builder.parse(new File(xmlFile));

        System.out.println("<?xml version=" + doc.getXmlVersion() + "?>");
    }
}
```

```
    traverse(doc.getDocumentElement(),0);
}

private static final int OPEN = 1;
private static final int CLOSE = 2;

private static void traverse(Node n, int level) {
    int i;

    // find out what type of node this is

    short t = n.getNodeType();

    if (t == Document.ELEMENT_NODE) {
        print_element(level, n, OPEN);
    } else if (t == Document.TEXT_NODE) {
        if (!is_empty(n)) {
            print_text(level, n);
        }
    }

    NodeList nl = n.getChildNodes();
    if (nl.getLength() == 0) {
        return;
    }

    for (i = 0; i < nl.getLength(); i++) {
```

DOMTest.java

```
        traverse(nl.item(i), level + 1);
    }

    if (t == Document.ELEMENT_NODE) {
        print_element(level, n, CLOSE);
    }
}

static void print_element(int level, Node n, int mode) {
    String slash = "";
    if (mode == CLOSE) {
        slash = "/";
    }
    System.out.println(get_indent(level) + "<" + slash + n.getNodeName()
        + ">");
}

static void print_text(int level, Node n) {
    String txt = n.getNodeValue().trim();
    txt = txt.replaceAll("\\s+", " ");
    System.out.println(get_indent(level) + txt);
}

static String get_indent(int level) {
    int i;
    StringBuffer buf = new StringBuffer();
    for (i = 0; i < level; i++) {
```

DOMTest.java

```
        buf.append(" ");
    }
    return buf.toString();
}

static boolean is_empty(Node n) {
    String val = n.getNodeValue();
    val = val.replaceAll("\\s+", "");
    val = val.replaceAll("\n+", "");
    if (val.equals("")) {
        return true;
    }
    return false;
}
}
```