University of Stirling, Computing Science and Mathematics

CSCU9T4/CSCU9TF - Managing Information.

XML Practical 3: Java and XML (SAX and DOM)

Creating and saving an XML document using DOM

- Open and explore in your preferred IDE or editor the java program: `DOMDemoCreate.java`
- Run the main program in `DOMDemoCreate` giving as a filename of your choice name with an `'.xml'` extension.
- You will notice that a small xml file is created containing data of a library with a single book. The only element of a book is its title, and the file is created without indentation.
- Your job is to modify the program and create a `DOMCreate` class to:
  - Add two new elements to `<book>` namely: `<category>` and `<year>`
  - Add at least 3 books to the library data
  - Save the created new DOM document as an XML file with the name `books.xml`. This time you should save the file with appropriate indentation (*Hint*: see lecture slides or the *stackoverflow* first answer in this link).
  - The saved file should `books.xml` be something like:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<library>
  <book>
    <title>Introduction to XML</title>
    <category>Computing</category>
    <year>2001</year>
  </book>
  <book>
    <title>The Blue Planet</title>
    <category>Science</category>
    <year>2009</year>
  </book>
  <book>
    <title>Climbing to the Stars</title>
    <category>Novel</category>
    <year>2014</year>
  </book>
</library>
```

Checkpoint 1: `DomDemo.Java, books.xml` , without errors and displaying the correct output.

Implementing a Java DOM Parser

- Open and explore the two provided XML documents: `staff.xml` and `cd_catalog.xml`
- Open and explore in your preferred IDE or editor the java program: `DOMDemo.java`
- Run the main program in `DOMDemo` using as input the two provided XML documents.
- You will notice that the last two lines of the `printNodes()` method are commented. This is because they only work for the `staff.xml` file. You can uncomment them and run the program for the `staff.xml` file.
- Your job is to modify this class and name it `DOMPrintInfoCD` with the following functionality for the `cd_catalog.xml` file:
    - Print out the name of the root element (first element in the document)
    - Using only DOM methods print all the text information (*hint:* use `getTextContent()`) of the first and the second CD in the catalog.
    - Print all the text information of the last CD in the catalog (*hint*: use a XPath query)
    - Print out the total number of CDs in the Catalog (*hint*: use a XPath query).
    - The output or your program should be something like:

```
-------------------
parsing cd_catalog.xml
Document root node is: CATALOG
  First CD Data:
              Empire Burlesque
              Bob Dylan
              USA
              Columbia
              10.90
              1985
  Second CD Data:
              Hide your heart
              Bonnie Tyler
              UK
              CBS Records
              9.90
              1988
  Last CD Data:
              Unchain my heart
              Joe Cocker
              USA
              EMI
              8.20
              1987
Total Number of CDs is: 26
```

Checkpoint 2: `DomPrintInfo.java`, without errors and displaying the correct screen output.

Implementing a Java SAX Parser

- Open and explore the two provided XML documents: `staff.xml` and `cd_catalog.xml`
- Open and explore in your preferred IDE or editor the two Java classes provided: `SAXDemo.java` and `BasicHandler.java`
- Run the main program in `SAXDemo` using as input files the two provided XML documents, and explore the output of the program.
- Your job is to create a new `SAXHandler` class: `SAXCountHandlerCD` with the following functionality for the `cd_catalog.xml` file:
    - Print a message indicating the start of processing the XML file.
    - While processing the XML file, print out a message in the screen indicating which CD number you are processing. (*Hint*: declare an integer variable to keep count of the processed elements and increment it where adequate)
    - The output or your program should be something like:

```
-------------------
parsing cd_catalog.xml

Reading CD Number =1
Reading CD Number =2
Reading CD Number =3
Reading CD Number =4

...
Reading CD Number =24
Reading CD Number =25
Reading CD Number =26
-------------------
```

Checkpoint 3:`SAXCountHandlerCD.java` , without errors and displaying the correct screen output