XML

Part 1: Creating Your First XML Document

```xml
<?xml version="1.0" encoding="UTF-8"?>
<library>
    <book>
        <title>The Great Gatsby</title>
        <author>F. Scott Fitzgerald</author>
        <year>1925</year>
        <genre>Fiction</genre>
    </book>
    <book>
        <title>To Kill a Mockingbird</title>
        <author>Harper Lee</author>
        <year>1960</year>
        <genre>Fiction</genre>
    </book>
    <book>
        <title>1984</title>
        <author>George Orwell</author>
        <year>1949</year>
        <genre>Dystopian</genre>
    </book>
</library>
```
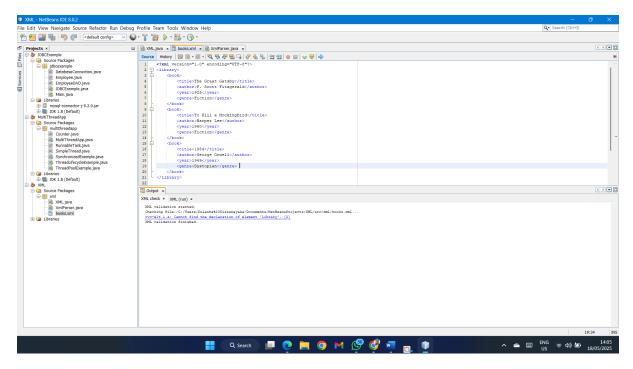
Part 2: Parsing XML in Java

```java
import org.w3c.dom. *;

import javax.xml.parsers.*;

public class XmlParser {

public static void main(String[] args) {

  try {

  // Create a new DocumentBuilderFactory and DocumentBuilder

    DocumentBuilderFactory factory =  DocumentBuilderFactory.newInstance();

    DocumentBuilder builder = factory.newDocumentBuilder();


  // Parse the XML file

    Document document = builder.parse("books.xml");


  // Normalize the document

    document.getDocumentElement().normalize();


  // Get the root element (library)

    NodeList nodeList = document.getElementsByTagName("book");
```
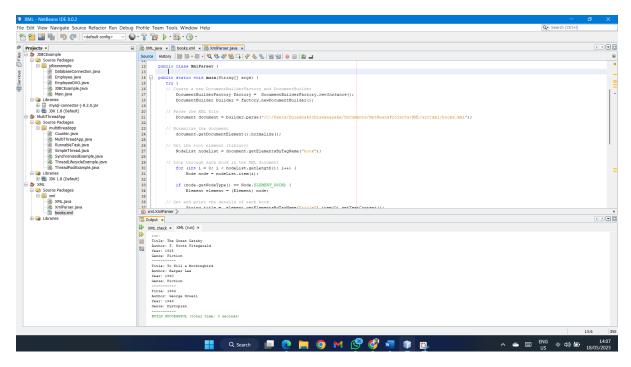
```java
// Loop through each book in the XML document
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);


    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;


// Get and print the details of each book
        String title =  element.getElementsByTagName("title").item(0).getTextContent();
        String author = element.getElementsByTagName("author").item(0).getTextContent();
        String year =  element.getElementsByTagName("year").item(0).getTextContent();
        String genre =  element.getElementsByTagName("genre").item(0).getTextContent();
            System.out.println("Title: " + title);
            System.out.println("Author: " + author);
            System.out.println("Year: " + year);
            System.out.println("Genre: " + genre);
            System.out.println("-----------");
    }
    }


    } catch (Exception e) {
        e.printStackTrace();
}
}
}
```

Part 3: Modifying XML Data

```
// Modify the year of the first book

    Element firstBook = (Element) nodeList.item(0);

    firstBook.getElementsByTagName("year").item(0).setTextContent("2023");


// Save the modified document

    TransformerFactory transformerFactory =  TransformerFactory.newInstance();

    Transformer transformer = transformerFactory.newTransformer();  DOMSource
source = new DOMSource(document);

    StreamResult result = new StreamResult(new File("updated_books.xml"));

    transformer.transform(source, result);
```

Projects

- JDBCExample
  - Source Packages
    - Jdbcexample
      - DatabaseConnection.java
      - Employee.java
      - EmployeeDAO.java
      - JDBCExample.java
      - Main.java
  - Libraries
    - mysql-connector-j-9.2.0.jar
    - JDK 1.8 (Default)
- MultiThreadApp
  - Source Packages
    - multithreadapp
      - Counter.java
      - MultiThreadApp.java
      - RunnableTask.java
      - SimpleThread.java
      - SynchronizedExample.java
      - ThreadLifecycleExample.java
      - ThreadPoolExample.java
  - Libraries
    - JDK 1.8 (Default)
- XML
  - Source Packages
    - xml
      - XML.java
      - XmlParser.java
      - books.xml
  - Libraries

XML.java | books.xml | XmlParser.java

Source | History

```java
19
20      // Parse the XML file
21          Document document = builder.parse("/C:/Users/Dulasha%20Dissanayaka/Documents/NetBeansProjects/XML/src/xml/books.xml");
22
23      // Normalize the document
24          document.getDocumentElement().normalize();
25
26      // Get the root element (library)
27          NodeList nodeList = document.getElementsByTagName("book");
28
29      // Modify the year of the first book
30          Element firstBook = (Element) nodeList.item(0);
31          firstBook.getElementsByTagName("year").item(0).setTextContent("2023");
32
33      // Save the modified document
34          TransformerFactory transformerFactory =  TransformerFactory.newInstance();
35          Transformer transformer = transformerFactory.newTransformer();  DOMSource source = new DOMSource(document);
36          StreamResult result = new StreamResult(new File("updated_books.xml"));
37          transformer.transform(source, result);
38
39      // Loop through each book in the XML document
40          for (int i = 0; i < nodeList.getLength(); i++) {
41              Node node = nodeList.item(i);
42
43              if (node.getNodeType() == Node.ELEMENT_NODE) {
```

xml.XmlParser > main > try > for (int i = 0; i < nodeList.getLength(); i++) >

Output

XML check | XML (run)

```
run:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Year: 2023
Genre: Fiction
-----------
Title: To Kill a Mockingbird
Author: Harper Lee
Year: 1960
Genre: Fiction
-----------
Title: 1984
Author: George Orwell
Year: 1949
Genre: Dystopian
-----------
BUILD SUCCESSFUL (total time: 0 seconds)
```

40:58 INS

ENG US

14:43
18/05/2025