GAM/IT/2022/F/0070

**Introduction to XML and Basic Operations**

Part 2: Creating Your First XML Document

<?xml version="1.0" encoding="UTF-8"?>

<!--

Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

Click nbfs://nbhost/SystemFileSystem/Templates/XML/XMLDocument.xml to edit this template

-->

<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>

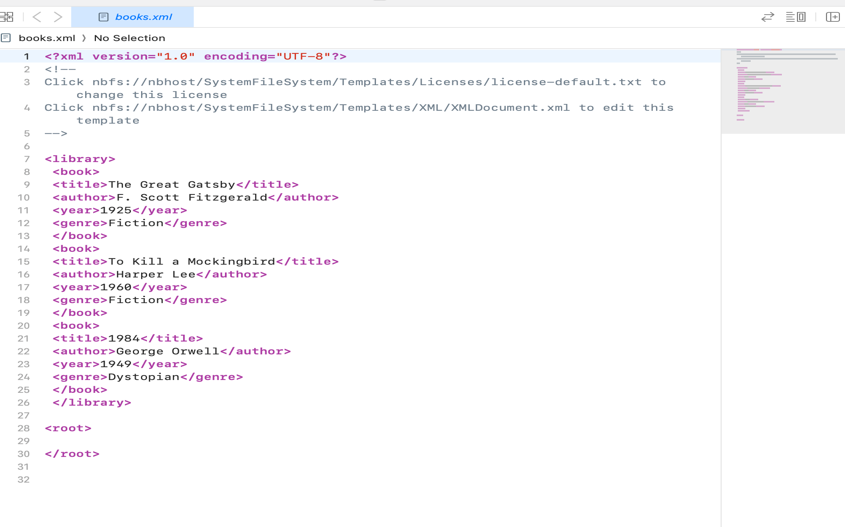
<author>George Orwell</author>

<year>1949</year>

<genre>Dystopian</genre>

</book>

</library>



Part 3: Parsing XML in Java

public class xmlPaeser {

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

File xmlFile = new File("ListOfBooks.xml");

Docu ment document = builder.parse(xmlFile);

// Normalize the document

document.getDocumentElement().normalize();

// Get the list of <book> elements

NodeList nodeList = document.getElementsByTagName("book");

// Loop through each <book> element

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 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("-----------");

}

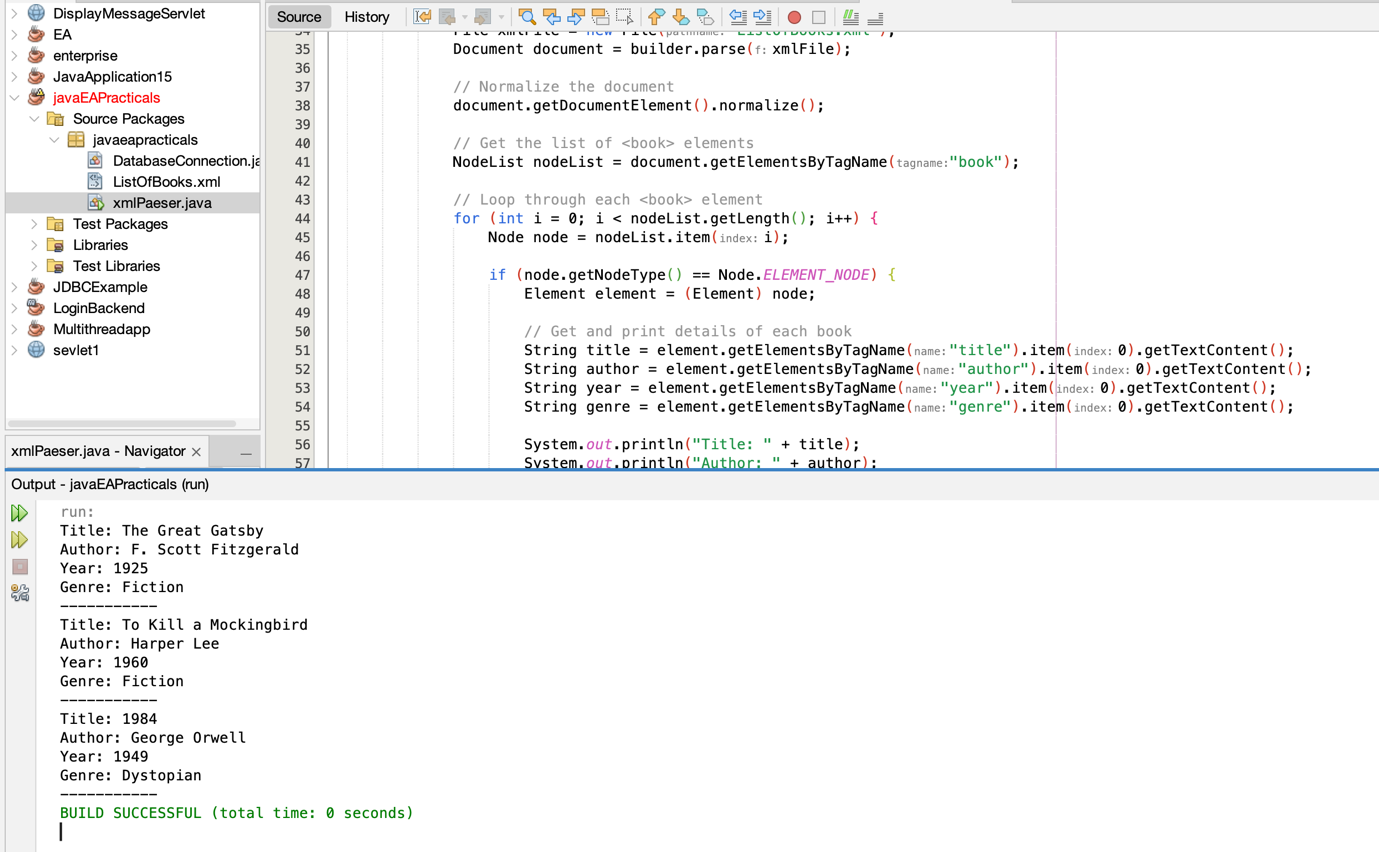
}

} catch (Exception e) {

e.printStackTrace();

}

}}

****

Part 4: 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 to updated\_books.xml

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

Counter counter=new Counter();

// Create and start multiple threads

Thread thread1 = new SynchronizedExample(counter);

Thread thread2 = new SynchronizedExample(counter);

thread1.start();

thread2.start();

// Wait for threads to finish

thread1.join();

thread2.join();

System.out.println("Final counter value: " + counter.getCount());

