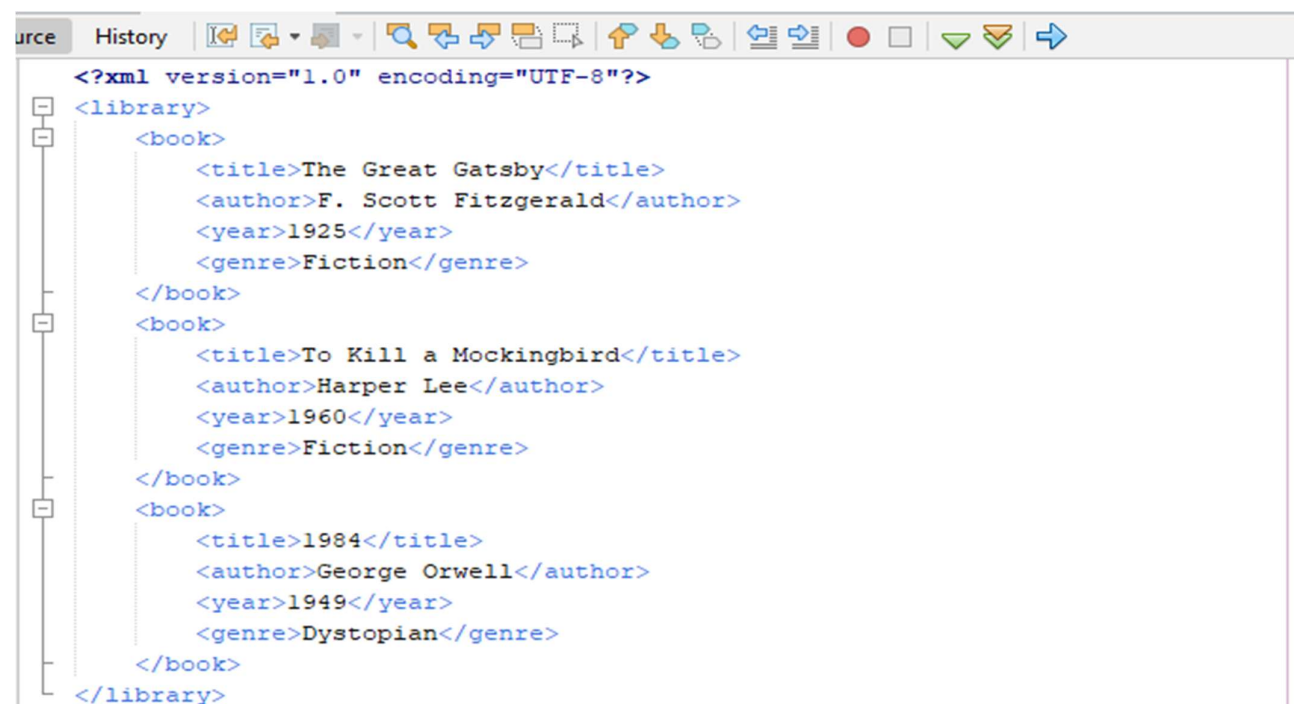


1. Creating Your First XML Document:

```
<?xml version="1.0" encoding="UTF-8"?>
<library> //root Element
  <book> // child element
    <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>
```

Code:



The image shows a screenshot of an XML editor window. The window has a title bar with 'urce' and 'History' tabs. Below the title bar is a toolbar with various icons for editing XML documents. The main area of the window displays the following XML code:

```
<?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>
```

2. Parsing XML in 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 all book elements
            NodeList nodeList = document.getElementsByTagName("book");

            // 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();
        }
    }
}
```

Output:



The screenshot shows an IDE's output window titled "Output - JavaApplication8 (run)". The window contains the following text:

```
run:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Year: 1925
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)
```

The output displays book information for three titles: "The Great Gatsby", "To Kill a Mockingbird", and "1984". Each entry includes the title, author, year, and genre, separated by a horizontal line. The final line indicates a successful build with a total time of 0 seconds.

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 to a new file
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);
```

Code:

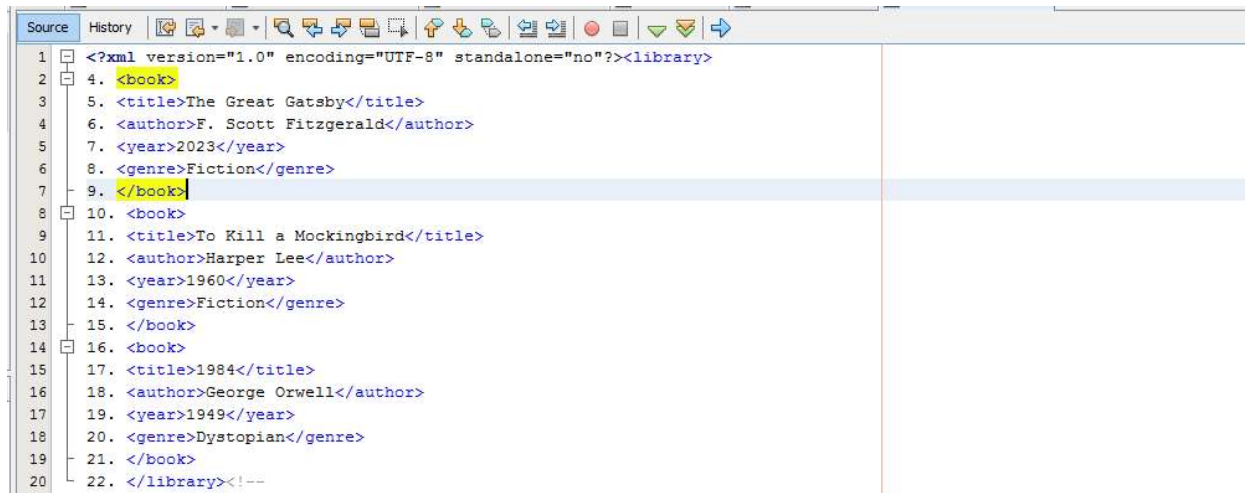
```
// Modify the year of the first book
Element firstBook = (Element) nodeList.item( index: 0);
firstBook.getElementsByTagName( name: "year").item( index: 0).setTextContent( textContent: "2023");

// Save the modified document to a new file
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
DOMSource source = new DOMSource( n: document);
StreamResult result = new StreamResult(new File( pathname: "updated_books.xml"));
transformer.transform( xmlSource: source, outputTarget: result);

System.out.println( n: "XML file has been updated and saved as updated_books.xml.");

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

updated XML file (updated_books.xml):



```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?><library>
2 4. <book>
3 5. <title>The Great Gatsby</title>
4 6. <author>F. Scott Fitzgerald</author>
5 7. <year>2023</year>
6 8. <genre>Fiction</genre>
7 9. </book>
8 10. <book>
9 11. <title>To Kill a Mockingbird</title>
10 12. <author>Harper Lee</author>
11 13. <year>1960</year>
12 14. <genre>Fiction</genre>
13 15. </book>
14 16. <book>
15 17. <title>1984</title>
16 18. <author>George Orwell</author>
17 19. <year>1949</year>
18 20. <genre>Dystopian</genre>
19 21. </book>
20 22. </library><!--
```