

جاءا في استخدا

إليك. **بيانات تعديل أو إنشاء تحليل،** تحقيقيه تريدها على اعتمدا طرق عدة على يتم أن يمكن في **اسخدام** الشائعة: الطرق بعض مع جيدة بداية على للحصول مختصر دليل

1. باسخدام $\frac{1}{x^2}$ تحليل

جيد. خياله هو فإن الذكرة، في شجرة كهيكل وتعديل قراءة تريديكنت إذا

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;

public class XMLParser {

    public static void main(String[] args) {

        try {

            // XML

            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document doc = builder.parse("example.xml");

            // XML

            doc.getDocumentElement().normalize();

            // ) "item"

            NodeList list = doc.getElementsByTagName("item");
            for (int i = 0; i < list.getLength(); i++) {

                Element element = (Element) list.item(i);
                System.out.println("Element: " + element.getNodeName());
                System.out.println("Value: " + element.getTextContent());

            }

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

مثال: □□□ □□□□□□□□.□□□□:

```
<root>
  <item>Item 1</item>
  <item>Item 2</item>
</root>
```

الذاكرة هيكل في والتعديل والتنقل سهولة: المزايا
الكبيرة. للملفات الذاكرة كثافة: العيوب

2. تحليل XML باستخدام XMLParser

الذاكرة. إلى بأكمله الملف يحمل ولا الأحداث موجه هو XML. فإن مهمة، الذاكرة كفاءة تكون حيث الذاكرة XML للملفات

```
import org.xml.sax.Attributes;
import org.xml.sax.helpers.DefaultHandler;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

public class SAXExample {
    public static void main(String[] args) {
        try {
            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser saxParser = factory.newSAXParser();

            DefaultHandler handler = new DefaultHandler() {
                boolean inItem = false;

                @Override
                public void startElement(String uri, String localName, String qName, Attributes attributes) {
                    if (qName.equalsIgnoreCase("item")) {
                        inItem = true;
                    }
                }

                @Override
                public void characters(char[] ch, int start, int length) {
                    if (inItem) {
                        System.out.println("Item: " + new String(ch, start, length));
                        inItem = false;
                    }
                }
            };
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
};

    saxParser.parse("example.xml", handler);
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}

```

الكمبيوتر. للملفات وسريّة الوزن خفيّة: المزايا
تسلسلياً. الملف مع الـ جة يتم عشوائياً؛ وصول يوجد لـ: العيوب

3. باستخدام XML لإنشاء

الصفحة: من XML لإنشاء

```

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

public class XMLCreator {
    public static void main(String[] args) {
        try {
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document doc = builder.newDocument();

            //
            Element root = doc.createElement("root");
            doc.appendChild(root);

            //
            Element item = doc.createElement("item");

```


