Web-технологии 3

Программные модели доступа к XML

Введение

- XML — универсальный формат обмена данными
- XML — пассивный формат
- Необходимы правила создания и анализа XML документов
- Популярные API для работы с XML
  - Simple API for XML (простой API для XML, SAX) — событийная модель
  - Document Object Model (объектная модель документа, DOM) — объектная модель
- Использование в зависимости от условий

- SAX
  - В процессе анализа не сохраняет данные XML документа.
  - Нет поддержки модификации или записи данных XML документа.
  - Данные документа становятся доступны в процессе анализа.
- DOM
  - Создает в памяти копию дерева документа
  - Дерево документа в памяти может изменяться
  - Сначала анализ, потом доступ к дереву.

SAX. Введение

- Последовательный доступ к элементам документа
- Не хранит дерево XML
- Событийное программирование
  - Распознанные элементы передаются приложению
  - Связь осуществляется через функции обратного вызова (callback)
  - Использование для потоковых данных

Схема работы

- Открывающий корневой тег (создаем массив)

  …
- Открывающий тег (создаем элемент массива)
- Текстовое содержимое тега (записываем содержимое)
- Закрывающий тег (записываем элемент в массив)

  …
- Закрывающий корневой тег (сохраняем массив)

Проблемы

- События не сохраняют состояния (неизвестное положение)
- События не сохраняются (нельзя вернуться)
- Не является стандартом W3C

Пример XML

```
<?xml version="1.0" encoding="windows-1251"?>
<books>
    <book id="1">
        <title>Компьютерные сети</title>
        <autor>Олифер</autor>
    </book>
    <book id="2">
        <title>Вычислительные сети сети</title>
        <autor>Кто-то ещё</autor>
    </book>
</books>
```

SAX парсер

```
…
SAXParser parser;
try {
    xmlData = new FileInputStream("books.xml");
    parser = factory.newSAXParser();
    parser.parse(xmlData, new MyParser());
}
…
```

```
class MyParser extends DefaultHandler {
public void startElement(String uri, String localName, String qName, Attributes
attributes) throws SAXException {
    System.out.println("Тег: "+qName);
    if(qName.equals("book"))
        System.out.println("id книги " + attributes.getValue("id"));
    super.startElement(uri, localName, qName, attributes);
}
public void characters(char[] c, int start, int length) throws SAXException {
    super.characters(c, start,  length);
    for(int i=start;i< start+length;++i)
        System.err.print(c[i]);
}
public void endElement(String uri, String localName, String qName)
                                        throws SAXException {
    System.out.println("Тег разобран: "+qName);
    super.endElement(uri,localName, qName);
}
```

Тег: books
Тег: book
id книги 1
Тег: title
Тег разобран: title
    Компьютерные сети
Тег: autor
Тег разобран: autor
    Олифер
Тег разобран: book
Тег: book
id книги 2
Тег: title
Тег разобран: title
    Вычислительные сети сети
Тег: autor
Тег разобран: autor
    Кто-то ещё
Тег разобран: book
Тег разобран: books

DOM. Введение

- Полный доступ к XML
- Представление XML в виде дерева
- Уровни DOM
  - 0: Специфичные элементы (до стандарта)
  - 1 (core): Базовые функции
  - 2 (core): Пространство имен и события
  - 3: Дополнительные расширения DOM

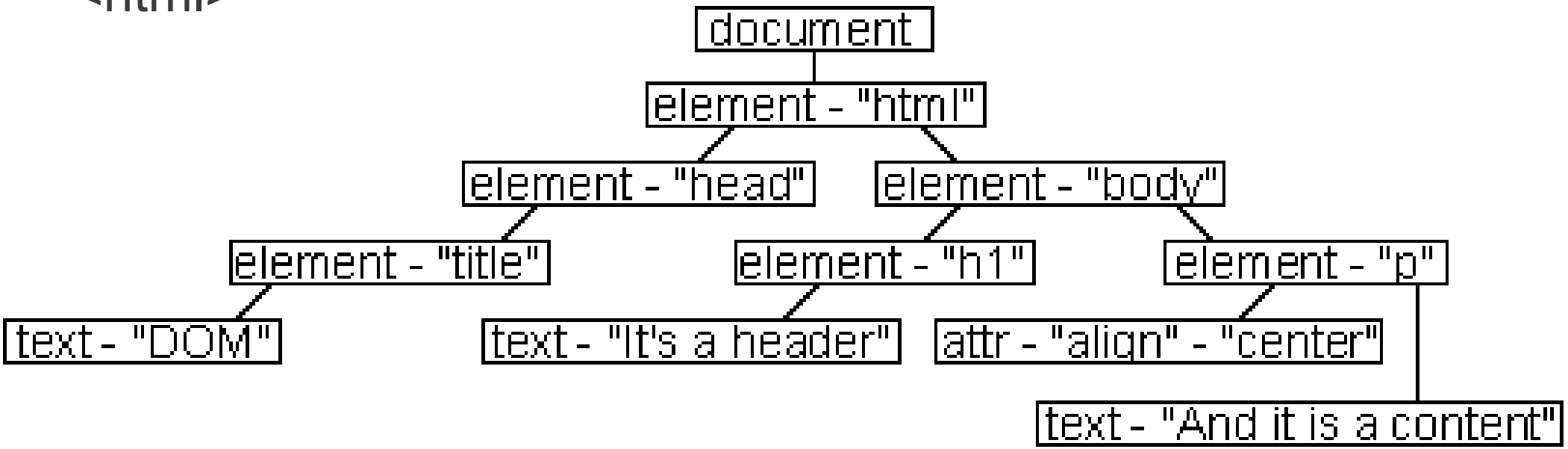
Проблемы

- DOM строит в памяти дерево всего документа.
- DOM создает объекты для всего что есть в документе
- Парсер DOM читает весь документ прежде чем предоставляет управление
- Нельзя обрабатывать потоковые данные

DOM level 1

- Узел — интерфейс Node
- Каждый тип узла имеет дополнительные методы
- Пример (interface Element):
  - DOMString tagName; Имя данного элемента
  - DOMString getAttribute(in DOMString name); Значение атрибута за его именем
  - setAttribute(in DOMString name, in DOMString value) Устанавливает значение атрибута
  - removeAttribute(in DOMString name) Удалает атрибут
  - NodeList getElementsByTagName(in DOMString name); Список элементов

```
<html>
<head><title>DOM</title></head>
<body>
<h1>It's a header<h1>
<p align="center">And it is a content</p>
</body>
<html>
```



DOM level 2

- Использование пространств имен для разделения доступа к узлам
- Element createElementNS(in DOMString namespaceURI, in DOMString qualifiedName);
- Attr createAttributeNS(in DOMString namespaceURI, in DOMString qualifiedName);
- NodeList getElementsByTagNameNS(in DOMString namespaceURI, in DOMString localName);

DOM level 3

- Работа со схемами DTD и XSD
- Поддержка XPath

```
// Initialize response object by parsing request...
String query = "/methodResponse/params/param/value/double";
if (impl.hasFeature("XPath", "3.0")) {
  XPathEvaluator evaluator = (XPathEvaluator) response;
  try {
    XPathResult index = evaluator.evaluate(query, response,
     null, XPathResult.ORDERED_NODE_ITERATOR_TYPE, null)
    // work with the result...
  }
…
}
```