

```

@XmlRegistry
public class ObjectFactory {
    public ObjectFactory() {
    }
    public Course createCourse() {
        return new Course();
    }
    public University createUniversity() {
        return new University();
    }
}

```

XML-анализаторы

XML как набор байт в памяти, запись в базе или текстовый файл представляет собой данные, которые еще предстоит обработать. То есть из набора строк необходимо получить данные, пригодные для использования в программе. Поскольку XML представляет собой универсальный формат для передачи данных, существуют универсальные средства его обработки – XML-анализаторы (парсеры).

Парсер – это библиотека (в языке Java: класс), которая читает XML-документ, а затем предоставляет набор методов для обработки информации этого документа.

Валидирующие и невалидирующие анализаторы

Как было выше упомянуто, существует два вида корректности XML-документа: синтаксическая (well-formed) – документ сформирован в соответствии с синтаксическими правилами построения, и действительная (valid) – документ синтаксически корректен и соответствует требованиям, заявленным в DTD.

Соответственно есть невалидирующие и валидирующие анализаторы. И те, и другие проверяют XML-документ на соответствие синтаксическим правилам. Но только валидирующие анализаторы знают, как проверить XML-документ на соответствие структуре, описанной в XSD или DTD.

Никакой связи между видом анализатора и видом XML-документа нет. Валидирующий анализатор может разобрать XML-документ, для которого нет DTD, и, наоборот, невалидирующий анализатор может разобрать XML-документ, для которого есть DTD. При этом он просто не будет учитывать описание структуры документа.

Древовидная и событийная модели

Существует три подхода (API) к обработке XML-документов:

- DOM (Document Object Model – объектная модель документов) – платформенно-независимый программный интерфейс, позволяющий программам и скриптам управлять содержимым документов HTML и XML, а также изменять их структуру и оформление. Модель DOM не накладывает ограничений на структуру документа. Любой документ известной структуры с помощью DOM может быть представлен в виде дерева узлов, каждый узел которого содержит элемент, атрибут, текстовый, графический или любой другой объект. Узлы связаны между собой отношениями родитель-потомок.
- SAX (Simple API for XML) базируется на модели последовательной одноразовой обработки и не создает внутренних деревьев. При прохожде-

нии по XML вызывает соответствующие методы у классов, реализующих интерфейсы, предоставляемые SAX-парсером.

- StAX (Streaming API for XML) не создает дерево объектов в памяти, но, в отличие от SAX-парсера, за переход от одной вершины XML к другой отвечает приложение, которое запускает разбор документа.

Анализаторы, которые строят древовидную модель, – это DOM-анализаторы. Анализаторы, которые генерируют события, – это SAX-анализаторы.

Анализаторы, которые ждут команды от приложения для перехода к следующему элементу XML – StAX-анализаторы.

В первом случае анализатор строит в памяти дерево объектов, соответствующее XML-документу. Далее вся работа ведется именно с этим деревом.

Во втором случае анализатор работает следующим образом: когда происходит анализ документа, анализатор вызывает методы, связанные с различными участками XML-файла, а программа, использующая анализатор, решает, как реагировать на тот или иной элемент XML-документа. Так, анализатор будет генерировать событие о том, что он встретил начало документа либо его конец, начало элемента либо его конец, символьную информацию внутри элемента и т.д.

StAX работает как **Iterator**, который указывает на наличие элемента с помощью метода **hasNext()** и для перехода к следующей вершине использует метод **next()**.

Когда следует использовать DOM-, а когда – SAX, StAX -анализаторы?

DOM-анализаторы следует использовать тогда, когда нужно знать структуру документа и может понадобиться изменить эту структуру либо использовать информацию из XML-файла несколько раз.

SAX/StAX-анализаторы используются тогда, когда нужно извлечь информацию о нескольких элементах из XML-файла либо когда информация из документа нужна только один раз.

Событийная модель

Как уже отмечалось, SAX-анализатор не строит дерево элементов по содержанию XML-файла. Вместо этого анализатор читает файл и генерирует события, когда находит элементы, атрибуты или текст. На первый взгляд, такой подход менее естествен для приложения, использующего анализатор, так как он не строит дерево, а приложение само должно догадаться, какое дерево элементов описывается в XML-файле.

Однако нужно учитывать, для каких целей используются данные из XML-файла. Очевидно, что нет смысла строить дерево объектов, содержащее десятки тысячи элементов в памяти, если всё, что необходимо, – это просто посчитать точное количество элементов в файле.

SAX-анализаторы

SAX API определяет ряд методов, используемых при разборе документа:

void startDocument() – вызывается на старте обработки документа;

void endDocument() – вызывается при завершении разбора документа;

void startElement(String uri, String localName, String qName, Attributes attrs) – будет вызван, когда анализатор полностью обработает содержимое открывающего тега, включая его имя и все содержащиеся атрибуты;