Лабораторная работа №1. Обработка иерархических данных. Язык запросов для формирования выборок по иерархиям.

Цель: научиться представлять информацию в виде XML-документов, формировать их в приложениях C# и выполнять выборку интересующих данных с помощью языка XPath

Теоретические сведения.

Расширяемый язык разметки (eXtensible Markup Language — XML)

Первая технология, представляющая последовательный, результативный и расширяемый механизм для описания данных и метаданных в одном и том же документе. Эта технология позволяет получать самоописывающиеся данные, упрощает реализацию интероперабельности и предоставляет возможность хранения самых разнообразных данных, например, текстовые документы, реляционные данные, объектно-ориентированные структуры и иерархическую информацию.

В настоящее время XML может использоваться в любых приложениях, которым нужна структурированная информация — от сложных геоинформационных систем с гигантскими объемами передаваемой информации до простейших программ, использующих этот язык для описания служебной информации.

XML-документ представляет собой обычный текстовый файл, в котором при помощи специальных маркеров создаются элементы данных, последовательность и вложенность которых определяет структуру документа и его содержимое. Структура XML-документа (рис. 1) сходна со структурами из многих других иерархических, структурных и метаинформационных технологий.

```
<Корневой элемент>
      <Элемент_1 Атрибут_1=« »> </Элемент_1>
      <Элемент_2> </Элемент_2>
      <Элемент_3>
            <Элемент_4> </Элемент_4>
            <Элемент_5 Атрибут_1=« » Атрибут_2=« »> </Элемент_5>
      </Элемент_3>
</Корневой элемент>
```

Рисунок 1. - Структура XML-документа.

Элементом верхнего уровня XML-документа является корневой элемент, существующий в документе в единственном числе. Элементы, помещенные внутрь другого элемента, называются вложенными или дочерними элементами. Содержащий их элемент — родительским элементом.

Описание на языке XML представляет собой операторы, написанные с соблюдением строго определенного синтаксиса. Каждый элемент XML-

документа должен содержать начальный и конечный тег (либо специальный пустой тег). Любой вложенный элемент должен быть полностью определен внутри элемента, в состав которого он входит. Это обусловлено тем, что структура XML-документов должна быть понятной для программы, которая выполняет обработку и отображение информации, содержащейся в этих документах. Строгий синтаксис придает XML-документу предсказуемую форму и облегчает написание программы обработки.

При создании XML-документа часто используется возможность создания собственных элементов и присваивания им необходимых имен — именно поэтому язык XML является расширяемым (extensible). Например, на рис. 2 приведено описание перечня книг.

```
<?xml version=«1.0» encoding=«utf-8» ?>
<inventory>
      <book>
            <title>The Adventures of Huckleberry Finn</title>
            <author>Mark Twain</author>
            <binding>mass market paperback</binding>
            <pages>298</pages>
            <price>$5.49</price>
      </book>
      <book>
            <title>XML для проектировщиков</title>
            <author>Джеймс Бин</author>
            <binding>mass market paperback</binding>
            <pages>255</pages>
            <price>$5.00</price>
      </book>
      <book>
            <title>Создание корпоративных систем на основе JAVA 2 Enterprise
Edition</title>
            <author>Пол Дж. Перроун</author>
            <binding>trade paperback</binding>
            <pages>1184/pages>
            <price>$24.38</price>
      </book>
<inventory>
```

Рисунок 2. - Пример XML-документа с описанием перечня книг.

Основным достоинством XML-документов является то, что при относительно простом способе создания и обработки (обычный текст может редактироваться любым тестовым процессором и обрабатываться стандартными XML-анализаторами), они позволяют создавать структурированную информацию, которую удобно использовать при компьютерной обработке данных.

Чтобы обработать XML-документ, прикладная программа должна иметь возможность ориентироваться в его иерархической структуре и извлекать, при необходимости, содержимое контейнеров. Такую функциональность прикладному ПО предоставляют синтаксические анализаторы (парсеры). Упрощенно это выглядит так: прикладная программа вызывает парсер, который разбирает XML-документ, идентифицирует каждый контейнер и

передает значения данных, содержащихся в каждом контейнере, в прикладное ПО. Как правило, синтаксические анализаторы могут сравнивать исходный XML -документ с набором правил из некоторых метаданных и извещать прикладное ПО при обнаружении противоречивости или ошибки — верифицировать документ. Набор правил и ограничений для XML-документа определяет так называемая схема.

Существует два основных типа синтаксических анализаторов: парсеры, создающие объектную модель документа (Document Object Model — DOM), и парсеры, предоставляющие простой API для работы с XML (simple API for XML — SAX). Главным достоинством DOM-парсеров является то, что они передают прикладному ПО всю структуру XML-документа полностью. При использовании SAX-парсеров такая структура XML -документа не строится. SAX-парсеры обычно требуют меньше памяти в отличие от DOM-парсеров и являются достаточно производительными при обходе иерархических структур. Однако их использование утяжеляет логику прикладного ПО и усложняет навигацию по документу.

Введение в язык XPath

XPath - это набор синтаксических правил для адресации частей XML-документа

XPath - это синтаксис для адресации частей XML-документа

XPath использует пути для адресации элементов XML

XPath является важнейшей частью стандарта XSLT

XPath не является XML-форматом

XPath является стандартом W3C

Рассмотрим фрагмент XML-документа:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog>
  <cd country=”USA”>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd country=”UK”>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

Пусть имеется выражение XPath:

catalog/cd/price

Оно определяет набор элементов price, дочерних относительно элементов cd, подэлементов элемента catalog.

XPath используется в приложениях, в которых есть необходимость обрабатывать данные, представленные в формате XML, а также он является неотъемлемой частью стандарта XSLT, который используется при необходимости отображения информации в заданном формате.

Для данного примера XML-документа возможны следующие варианты извлечения данных:

Запрос	Действие
`/catalog`	Выбирает корневой элемент
`/catalog/cd`	Выбирает все элементы cd элемента catalog
`/catalog/cd/price`	Выбирает все элементы price элементов cd элемента catalog
`/catalog/cd[price>10.0]`	Выбирает все элементы cd элемента catalog с price>10
`Выражение, начинающееся с /`	Представляет абсолютный путь к элементу
`Выражение, начинающееся с //`	Выделяет все элементы документа, удовлетворяющие критерию
`//cd`	Выделяет все элементы cd в документе
`/catalog/cd/title | /catalog/cd/artist`	Выделяет все элементы title и artist элементов cd элемента catalog
`//title | //artist`	Выделяет все элементы title и artist в документе
`/catalog/cd/*`	Выделяет все дочерние элементы узла cd элемента catalog
`/catalog/*/price`	Выделяет всех «внуков» элемента catalog
`/*/*/price`	Выделяет все элементы price, имеющие двух предков
`//*`	Все элементы документа
`/catalog/cd[1]`	Первый дочерний элемент cd элемента catalog
`/catalog/cd[last()]`	Последний дочерний элемент cd элемента catalog
`/catalog/cd[price]`	Выделяет все элементы cd элемента catalog, которые имеют дочерний элемент price
`/catalog/cd[price=10.90]/price`	Выделяет все элементы price, дочерние относительно /catalog/cd со значением 10.90
`//@country`	Выделяет все атрибуты country
`//cd[@country]`	Выделяет все элементы cd в документе, имеющие атрибут country
`//cd[@*]`	Выделяет элементы cd с любыми атрибутами

Запрос	Действие
`//cd[@country='UK']`	Выделяет элементы cd, имеющие атрибут country со значением UK

Более подробно спецификацию XPath и все возможные варианты построения путей можно почерпнуть на http://www.w3.org/TR/xpath/

Порядок выполнения работы

При выполнении работы необходимо выполнить следующие требования:

1. Сформировать с помощью любого текстового редактора исходный XML-документ, содержащий не менее 10 записей, согласно индивидуальному заданию.
2. Считать документ в программе и выполнить обработку в соответствии с заданием. При обработке обязательно использование выражений XPath для разбора информации и/или отбора узлов, соответствующих заданию.
3. Сформировать на диске результирующий XML-документ.
4. Парсер, используемый в программе определяется вариантом:
- SAX Parser – для нечётных вариантов
- DOM Parser – для чётных вариантов

Индивидуальные задания

1. В расписании движения самолетов из аэропорта указаны следующие сведения: номер рейса (4 цифры), аэропорт назначения, расстояние в км, стоимость билета (взрослый билет, детский билет), время в часах и мин. (отправление, прибытие в аэропорт назначения). Сформировать документ со сведениями о трех рейсах, имеющих наибольшую продолжительность полета при расстоянии не большем заданного пользователем.

2. В каталоге студии звукозаписи имеются следующие данные: название группы, название альбома, год выпуска альбома, название студии, записавшей альбом. Необходимо сформировать XML-каталог групп, выпустивших альбомы в заданном году и на заданной студии.

3. В журнале успеваемости академгруппы по программированию имеются следующие данные: фамилия студента, оценки по пяти лабораторным работам, количество пропусков занятий. Определить трех студентов, имеющих наибольшее количество пропусков (студентов, сдавших все работы не включать). Записи в XML-документе расположить в алфавитном порядке.

4. В магазине имеются следующие данные о товарах: название, единица измерения, цена, норма отпуска в одни руки. Составить документ, включающий элемент, содержащий список товаров, норма отпуска которых не более двух единиц, а также элемент, содержащий список товаров, стоимость которых превышает указанную пользователем. Списки упорядочить по названию товаров в алфавитном порядке.

5. На заводе радиоэлектроники выпускают звуковоспроизводящую технику и имеются следующие данные: название прибора, назначение (магнитофон, магнитола, проигрыватель), год создания, стоимость, гарантийный срок эксплуатации. Составить XML-документ, включающий список магнитофонов, разработанных в заданном году, а также список проигрывателей, гарантийный срок эксплуатации которых более 3-х лет.

6. В каталоге программного обеспечения имеются следующие данные: имя файла, расширение, размер файла, дата создания. Составить каталог текстовых файлов (расширение TXT, DOC), а также таблицу файлов, размером более 64 Кбайт. Каталоги отсортировать по имени файла в алфавитном порядке.

7. В больнице ведется учет больных по следующим данным: фамилия больного, номер палаты, дата поступления, дата выписки (может отсутствовать), диагноз (название болезни). Необходимо выдать список больных, лежавших в больнице на заданную дату. Список сортировать по номеру палаты.

8. На станции технического обслуживания автомобилей (СТО) ведется учет автомобилей, прошедших капитальный ремонт, по следующим данным: марка машины, серийный номер, пробег (в км) после предыдущего ремонте, год выпуска автомобиля. Необходимо составить документ, содержащий список машин, имеющих пробег более 100 000 км, а также список автомобилей, выпущенных после заданного года и прошедших ремонт. Списки сортировать по году выпуска машины.

9. В библиотеке имеются следующие данные о книгах: название, фамилия автора, год издания, издательство, количество экземпляров книг в библиотеке. Необходимо сформировать XML-документ, содержащий список книг, изданных в заданном году и список книг, имеющихся в библиотеке в одном экземпляре. Список упорядочить по названию книг в алфавитном порядке.

11. В военкомате ведется учет юношей допризывного и призывного возраста. Имеются следующие данные: фамилия, год рождения, номер личного дела, годность к службе («годен» или «не годен»). Необходимо вывести список юношей, призываемых на службу в заданном году (по достижении 18 лет). Список упорядочить по году рождения.

11. В аптеке ведется учет лекарственных средств. Имеются следующие данные: название лекарства, цена одной упаковки, количество упаковок в аптеке, год выпуска, срок хранения (в годах). Необходимо вывести документ, содержащий список лекарств, не годных к употреблению на заданный год, и и список лекарств, стоимость которых выше заданной пользователем. Упорядочить по названию лекарства в алфавитном порядке.

12. В заводском цеху ведется журнал расхода материалов по следующим данным: название материала, ГОСТ, расход в сутки, количество имеющихся в цеху. Необходимо вывести документ, содержащий список материалов, которые

закончатся через заданное количество дней, а также список 5-ти наименее расходуемых материалов. Списки упорядочить по названию материала в алфавитном порядке.

13. За материально ответственным лицом числятся материальные ценности, записанные в журнале: название предмета, количество, дата приобретения (год), срок службы (в годах). Необходимо вывести документ, содержащий список предметов, подлежащих списанию в заданный год, а также список предметов, срок службы которых превышает указанный пользователем. Списки упорядочить по дате приобретения.

14. В заводском цеху ведется учет электроэнергии, расходуемой машинами и приборами. Имеются следующие данные: название машины или прибора, инвентарный номер, потребляемая мощность, количество таких приборов в цеху. Вывести список десяти наиболее энергоемких приборов, в котором атрибутом корневого узла указать суммарную мощность, потребляемую цехом при всех включенных приборах. Список упорядочить по инвентарному номеру.

15. На АТС ведется учет междугородних разговоров абонентов по следующим данным: фамилия абонента, домашний адрес, номер телефона, сумма междугородних телефонных разговоров за месяц (в рублях). Необходимо вывести список абонентов, тратящих на междугородние звонки больше указанной суммы. Списки упорядочить по номеру телефона.