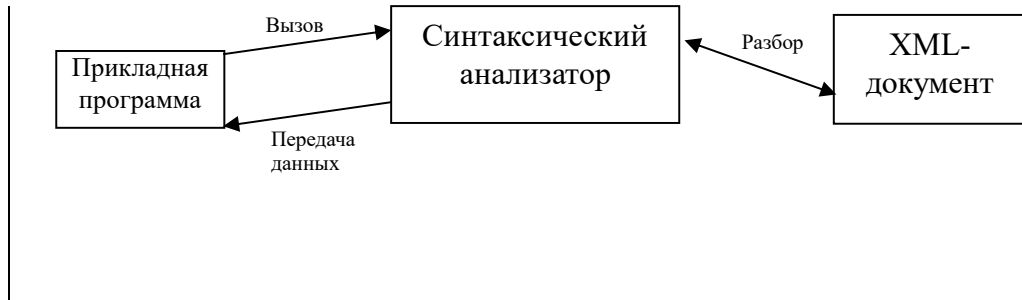


XML. Раздаточны й материал №1



Раздаточный материал № 2

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Напоминание</heading>
  <body>Не забудь обо мне в эти выходные!</body>
</note>
```

Раздаточный материал № 3

```
<корневой>
  <потомок>
    <подпотомок>.....</подпотомок>
  </потомок>
</корневой>
```

Раздаточный материал № 4

Дерево XML

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Раздаточный материал № 5

Синтаксис

<имя_элемента имя_атрибута= «значение»>

</ имя_элемента >

<имя_элемента/> - «стенографический» тег элемента допускает определение элемента единственным тегом. Элемент не имеет содержимого, т.е. нет данных, которые он должен определять, он называется пустым. Иногда появляется потребность в XML-элементах, в которых по смыслу никогда не будет содержимого.

Правила создания XML-документа:

1. в заголовке документа помещается XML-объявление, в котором указывается язык разметки документа, номер его версии и дополнительная информация;
2. каждый открывающий тег обязательно должен иметь закрывающий.
3. в XML учитывается регистр символов.
4. все значения атрибутов должны быть заключены в кавычки.
5. необходимо следить за порядком следования открывающих и закрывающих тегов.
6. вся информация, располагающаяся между начальными и конечными тегами, рассматривается в XML как данные и поэтому учитываются все символы форматирования, т.е. пробелы, переводы строк и т.д. не игнорируются как в HTML.

Раздаточный материал № 6

Старайтесь придумать описательные имена: <first_name>, <last_name>.

Имена следует составлять короткие и простые, вроде: <book_title>; а не: <the_title_of_the_book>.

Избегайте символа "-". Если вы напишете нечто вроде "first-name", то некоторые приложения могут решить, что вы вычитаете имя "name" из имени "first".

Избегайте символа ".". Если вы напишете нечто вроде "first.name", то некоторые приложения могут решить, что "name" это свойство объекта "first".

Избегайте символа ":". Двоеточие зарезервировано для механизма пространства имен.

Нелатинские символы, вроде, éòá вполне легальны в XML, однако, если некое приложение их не поддерживает, то возникнут проблемы.

Раздаточный материал № 7

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <body>Не забудь про меня в эти выходные!</body>
</note>
```

Приложение извлечет и сформирует следующее сообщение:

Кому: Tove
От: Jani
Не забудь про меня в эти выходные!

Если автор XML документа добавит некоторую дополнительную информацию:

```
<note>
  <date>2008-01-10</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Напоминание</heading>
  <body>Не забудь про меня в эти выходные!</body>
</note>
```

То приложение продолжит работу и отыщет новые элементы <to>, <from> и <body> и сформирует вывод.

XML. Раздаточный материал № 8

```
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

XML. Раздаточный материал № 9

<city>Hollivood</city> речь в этой части документа идет о городе,
<restaurant>Hollivood</restaurant> речь в этой части документа идет о ресторане

XML. Раздаточный материал № 10

```
<file type="gif">computer.gif</file>
```

XML. Раздаточный материал № 11

```
<gangster name='George "Shotgun" Ziegler'>
```

либо использовать символы сущностей:

```
<gangster name="George &quot;Shotgun&quot; Ziegler">
```

XML. Раздаточный материал № 12

Пример № 1

```
<person sex="female">  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

Пример № 2

```
<person>  
  <sex>female</sex>  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

В первом примере пол указан в атрибуте. Во втором, пол записан, как элемент. Оба примера предоставляют одну и ту же информацию.

XML. Раздаточный материал № 13

```
<messages>  
  <note id="501">  
    <to>Tove</to>  
    <from>Jani</from>  
    <heading>Напоминание</heading>  
    <body>Не забудь обо мне в эти выходные!</body>  
  </note>  
  <note id="502">  
    <to>Jani</to>  
    <from>Tove</from>  
    <heading>Re: Напоминание</heading>  
    <body>Не забуду</body>  
  </note>  
</messages>
```

XML. Раздаточный материал № 14

< > " или $(десятичная форма записи),
(шестнадцатеричная)

XML. Раздаточный материал № 15

```
<?xml version="1.0" encoding="UTF-8"?>
```

XML. Раздаточный материал № 16

Следующий код XML содержит информацию о HTML таблице:

```
<table>
  <tr>
    <td>Яблоки</td>
    <td>Бананы</td>
  </tr>
</table>
```

Следующий код XML содержит информацию о столе (предмет мебели), который по англ. тоже table:

```
<table>
  <name>Африканский кофейный столик</name>
  <width>80</width>
  <length>120</length>
</table>
```

XML. Раздаточный материал № 17

```
<h:table>
  <h:tr>
    <h:td>Яблоки</h:td>
    <h:td>Бананы</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>Африканский кофейный столик</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

XML. Раздаточный материал № 18

Декларация пространства имен имеет следующий синтаксис -
xmlns:префикс="URI".

XML. Раздаточный материал № 19

```
<root>
  <h:table xmlns:h="http://www.w3.org/TR/html4/">
    <h:tr>
      <h:td>Яблоки</h:td>
      <h:td>Бананы</h:td>
    </h:tr>
  </h:table>

  <f:table xmlns:f="http://www.w3schools.com/furniture">
    <f:name>Африканский кофейный столик</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>
</root>
```

В данном примере, атрибут xmlns в теге <table> определяет префиксам h: и f: квалифицирующее пространство имен.

XML. Раздаточный материал № 20

```
<root                                     xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="http://www.w3schools.com/furniture">
  <h:table>
    <h:tr>
      <h:td>Яблоки</h:td>
      <h:td>Бананы</h:td>
    </h:tr>
  </h:table>

  <f:table>
    <f:name>Африканский кофейный столик</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>
</root>
```

XML. Раздаточный материал № 21

xmlns="URI пространства имен"

XML. Раздаточный материал № 22

Следующий код XML содержит информацию о таблице HTML:

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Яблоки</td>
    <td>Бананы</td>
  </tr>
</table>
```

Следующий код XML содержит информацию о предмете мебели — столе:

```
<table xmlns="http://www.w3schools.com/furniture">
  <name>Африканский кофейный столик</name>
  <width>80</width>
  <length>120</length>
</table>
```

XML. Раздаточный материал № 23

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Напоминание</heading>
  <body>Не забудь обо мне в эти выходные!</body>
</note>
```

Посмотрите этот XML файл: note.xml.

XML. Раздаточный материал № 24

Попробуйте открыть следующий XML файл в браузере: note_error.xml.

Другие примеры XML

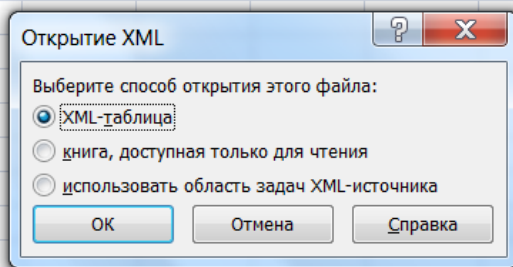
XML каталог компакт-дисков (cd_catalog.xml)
Коллекция CD, сохраненная в виде XML данных.

XML каталог растений (plant_catalog.xml)
Каталог растений из магазина растений, сохраненный в виде XML данных.

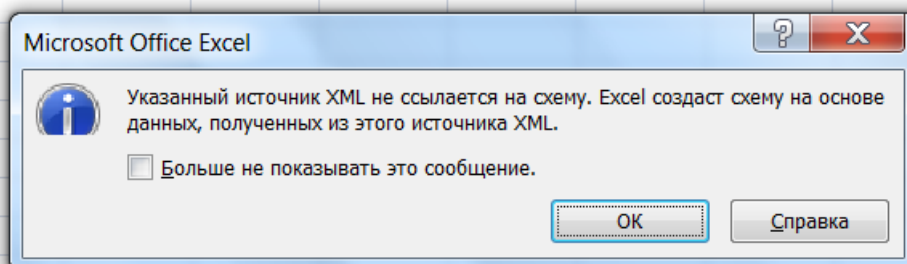
Простое меню (simple.xml)
Меню завтрака из ресторана, сохраненное в виде XML данных.

Упражнение № 1

1. В предложенном файле TASK1_error.xml исправить все ошибки, т.е. сделать его формально правильным.
2. Загрузить файл TASK1_error.xml в табличный процессор Excel. Для этого загрузить Excel, открыть файл TASK1_error.xml, предварительно выбрав пункт «Файлы XML». Появится диалоговое окно:



3. Нажать «ОК». Появится окно:



4. Нажать «ОК». Excel самостоятельно выбрал способ представления информации в виде электронной таблицы.

Свойства		Сервис	Данные из внешней таблицы		Параметры с
A1		f_x	TITLE		
	A	B	C	D	E
1	TITLE	AUTHOR	BINDING	PAGES	PRICE
2	Приключения Тома Сойера	Марк Твен	Массовый рынок Мягкая обложка	298	\$5.49
3	Война и мир	Лев Толстой	твердая обложка	462	\$7.75
4	Преступление и наказание	Федор Достоевский	Массовый рынок Мягкая обложка	98	\$2.95
5	Архипелаг Гулаг	Александр Солженицын	Мягкая обложка	473	\$10.95
6	Мастер и Маргарита	Михаил Булгаков	твердая обложка	724	\$9.95
7	Человек-амфибия	Александр Беляев	Массовый рынок Мягкая обложка	256	\$4.95
8	Капитанская дочь	Александр Пушкин	Мягкая обложка	253	\$4.25
9	Дворянское гнездо	Иван Тургенев	Мягкая обложка	384	\$3.35
10					

5. Внести в таблицу произвольные изменения (например, изменить стоимость нескольких книг) и сохранить файл в Excel в своей рабочей папке с именем booksExp.
6. Произвести экспорт всех данных из таблицы, включая измененные данные, в другой XML-документ, но с такой же структурой - book2.xml. Для этого на любой ячейке с данными щелкнуть правой кнопкой мыши → XML → Экспорт и выбрать файл.
7. Сравнить размеры полученных файлов booksExp.xls и booksExp.xml. Сделать соответствующие выводы.

XML. Раздаточный материал № 25

Посмотрите на этот XML файл: Каталог компакт-дисков ([cd_catalog.xml](#))

Теперь посмотрите на этот файл стилей: CSS файл ([cd_catalog.css](#))

И, наконец, посмотрите на: Каталог компакт-дисков отформатированный при помощи CSS ([cd_catalog_with_css.xml](#))

Ниже показана часть этого XML файла. Вторая строка подключает к XML файлу CSS файл:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="//msiter.ru/cd_catalog.css"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  ...
```

Упражнение № 2

1. Для файла TASK1_error.xml применить любое стилевой оформление элементов (тень, подчеркивание, цвет и размер шрифта и т.п.).

XML. Раздаточный материал № 26

Внешние файл DTD

```
<?xml version="1.0"?>
<!DOCTYPE journal SYSTEM "journal.dtd">
```

Внутреннее DTD-описание

```
...
<!DOCTYPE journal [
  <!ELEMENT journal (contacts, issues, authors)>
  ...
]>
...
```

XML. Раздаточный материал № 27

```
<!ELEMENT pharmaceuticals ANY >

<!ELEMENT contacts (address, tel+, email?)>
```

XML. Раздаточный материал № 28

Например, для XML файла из РМ № 2 DTD-определение будет таким:

```
<!DOCTYPE note [  
<!ELEMENT note (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>  

```

Упражнение 3

Для xml-документа из упражнения №1 создать внутреннее DTD-определение.

XML. Раздаточный материал № 29

```
<!ELEMENT contacts (address, tel, email)
```

XML. Раздаточный материал № 30

```
<!ELEMENT note (to,from,header,(message|body))>
```

XML. Раздаточный материал № 31

```
<!ELEMENT pharmaceuticals (#PCDATA)>
```

для элемента pharmaceuticals допускаются только символьные данные.

Следующие два элемента в соответствии с данной декларацией являются корректными:

```
<pharmaceuticals>Новые медицинские препараты</pharmaceuticals>  
<pharmaceuticals></pharmaceuticals>
```

Элемент, который в соответствии с объявлением должен содержать символьные данные, может и не иметь никаких символов, т.е. быть пустым.

XML. Раздаточный материал № 32

```
<!ELEMENT pharmaceuticals (#PCDATA | FORMA)*>
```

В соответствии с этим объявлением следующие элементы pharmaceuticals являются допустимыми:

```
< pharmaceuticals >  
Новые медицинские препараты  
< FORMA>Таблетки</FORMA >  
</ pharmaceuticals>
```

```
< pharmaceuticals >  
Новые медицинские препараты  
< FORMA>Таблетки</FORMA >  
< FORMA>Капсулы</FORMA >  
</ pharmaceuticals>
```

XML. Раздаточный материал № 33

Содержимое файла node.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://msiter.ru" xmlns="http://msiter.ru" element-
FormDefault="qualified">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XML. Раздаточный материал № 34

Подключение XML схемы к XML-файлу

```
<?xml version="1.0"?>
<note xmlns="http://msiter.ru"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://msiter.ru note.xsd">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Напоминание</heading>
  <body>Не забудь обо мне в эти выходные!</body>
</note>
```

XML. Раздаточный материал № 35

```
<?xml version="1.0"?>
<xs:schema>
...
</xs:schema>
```

XML. Раздаточный материал № 36

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://msiter.ru"
xmlns="http://msiter.ru"
elementFormDefault="qualified">
...
</xs:schema>
```

XML. Раздаточный материал № 37

```
<?xml version="1.0"?>
<note xmlns="http://msiter.ru"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://msiter.ru note.xsd">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Напоминание</heading>
  <body>Не забудь обо мне в эти выходные!</body>
</note>
```

XML. Раздаточный материал № 38

```
<xs:element name="xxx" type="yyy"/>
```

где xxx — имя элемента, а ууу — тип данных элемента.

XML схемы имеют множество встроенных типов данных. Наиболее часто используемыми являются следующие типы:

XML. Раздаточный материал № 39

```
xs:string  
xs:decimal  
xs:integer  
xs:boolean  
xs:date  
xs:time
```

XML. Раздаточный материал № 40

Возьмем некоторые XML элементы:

```
<lastname>Refsnes</lastname>  
<age>36</age>  
<dateborn>1970-03-27</dateborn>
```

Так будут выглядеть определения этих элементов в XML схеме:

```
<xs:element name="lastname" type="xs:string"/>  
<xs:element name="age" type="xs:integer"/>  
<xs:element name="dateborn" type="xs:date"/>
```

XML. Раздаточный материал № 41

```
<xs:element name="color" type="xs:string" default="red"/>
```

XML. Раздаточный материал № 42

```
<xs:element name="color" type="xs:string" fixed="red"/>
```

XML. Раздаточный материал № 43

```
<xs:attribute name="xxx" type="yyy"/>
```

где xxx — имя атрибута, а ууу — тип данных атрибута.

XML. Раздаточный материал № 44

xs:string
xs:decimal
xs:integer
xs:boolean
xs:date
xs:time

```
<lastname lang="EN">Smith</lastname>
```

Соответствующая декларация атрибута:

```
<xs:attribute name="lang" type="xs:string"/>
```

XML. Раздаточный материал № 45

```
<xs:attribute name="lang" type="xs:string" default="EN"/>
```

XML. Раздаточный материал № 46

```
<xs:attribute name="lang" type="xs:string" fixed="EN"/>
```

XML. Раздаточный материал № 47

```
<xs:attribute name="lang" type="xs:string" use="required"/>
```

XML. Раздаточный материал № 48

Существует четыре вида составных элементов:

пустые элементы

элементы, которые содержат только другие элементы

элементы, которые содержат только текст

элементы, которые содержат как другие элементы, так и текст

При этом все эти виды составных элементов могут содержать еще и атрибуты.

XML. Раздаточный материал № 49

```
<xs:complexType>  
  <xs:sequence>  
    <xs:element name="firstname" type="xs:string"/>  
    <xs:element name="lastname" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>  
</xs:element>
```

XML. Раздаточный материал № 50

```
<xs:element name="employee" type="personinfo"/>
<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

При таком способе описания элемента, один и тот же составной тип могут использовать и другие элементы:

```
<xs:element name="employee" type="personinfo"/>
<xs:element name="student" type="personinfo"/>
<xs:element name="member" type="personinfo"/>
<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Также можно, основываясь на каком-либо существующем составном элементе, добавить ему новые элементы:

```
<xs:element name="employee" type="fullpersoninfo"/>
<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
    <xs:extension base="personinfo">
      <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

XML. Раздаточный материал № 51

Составной XML элемент "product", который является пустым:

```
<product pid="1345"/>
```

Одна из возможных схем будет иметь вид:

```
<xs:element name="product" type="prodtype"/>
<xs:complexType name="prodtype">
  <xs:attribute name="prodid" type="xs:positiveInteger"/>
</xs:complexType>
```

XML. Раздаточный материал № 52

XML элемент "person", который содержит только элементы:

```
<person>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</person>
```

Одна из возможных схем будет иметь вид:

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XML. Раздаточный материал № 53

```
<xs:element name="person" type="persontype"/>
<xs:complexType name="persontype">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

XML. Раздаточный материал № 54

```
<xs:element name="somename">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="basetype">
        ...
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

ИЛИ

```
<xs:element name="somename">
  <xs:complexType>
    <xs:simpleContent>
      <xs:restriction base="basetype">
        ...
      </xs:restriction>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

XML. Раздаточный материал № 55

```
<shoesize country="france">35</shoesize>
```

Следующий пример декларирует тип при помощи тега `complexType` для элемента "shoesize". Контент определен, как целочисленное значение (`integer`), кроме этого элемент "shoesize" содержит атрибут с именем "country":

```
<xs:element name="shoesize">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="country" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

XML. Раздаточный материал № 56

```
<xs:element name="shoesize" type="shoetype"/>
<xs:complexType name="shoetype">
  <xs:simpleContent>
    <xs:extension base="xs:integer">
      <xs:attribute name="country" type="xs:string" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

XML. Раздаточный материал № 57

```
<letter>
  Уважаемый господин <name>Джон Смит</name>.
  Ваш заказ <orderid>1032</orderid>
  будет выполнен <shipdate>2001-07-13</shipdate>.
</letter>
```

Следующая XML схема декларирует элемент "letter":

```
<xs:element name="letter">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="orderid" type="xs:positiveInteger"/>
      <xs:element name="shipdate" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```


XML. Раздаточный материал № 58

```
<xs:element name="letter" type="lettertype"/>
<xs:complexType name="lettertype" mixed="true">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="orderid" type="xs:positiveInteger"/>
    <xs:element name="shipdate" type="xs:date"/>
  </xs:sequence>
</xs:complexType>
```