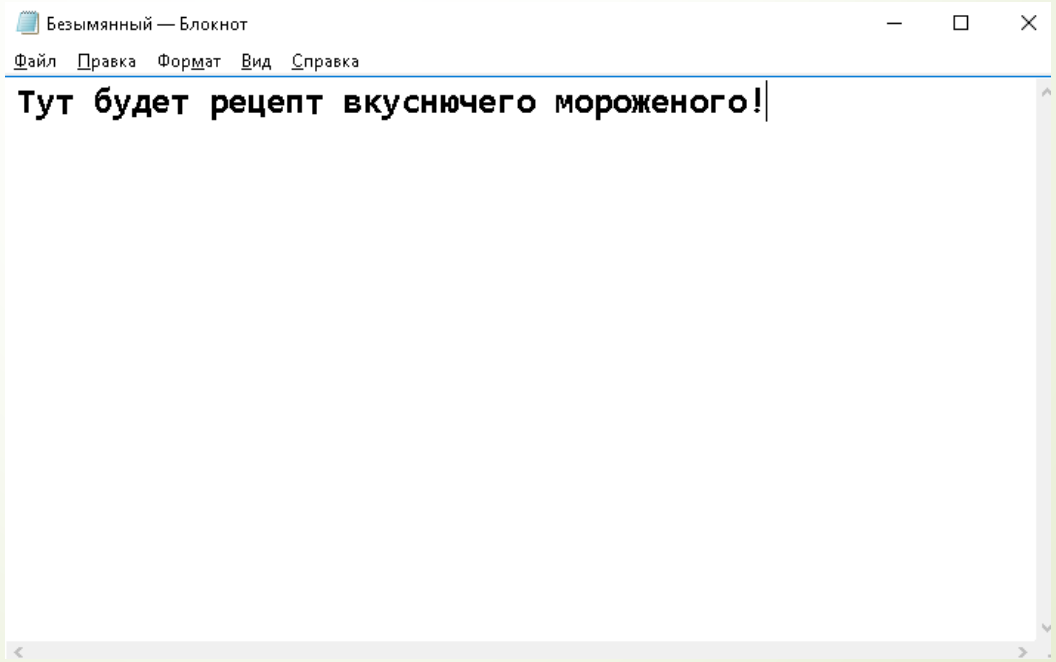
Документы XML. Схемы XML.

Информационное обеспечение предметно-ориентированных ИС.

Построение документа XML.

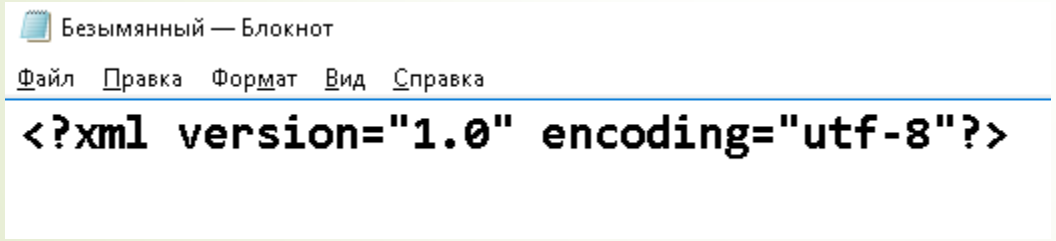
Предположим, мы в гостях попробовали вкуснейшее мороженое, которое мы хотим всенепременно записать в виде рецепта, чтобы потом поместить его в свою базу данных рецептов и периодически баловать себя прекрасным десертом. При себе у нас есть блокнот и сеть Интернет, чего вполне достаточно, чтобы справиться с задачей. Мы запишем рецепт в блокноте в формате XML, а потом выгрузим его в нашу базу данных.



Построение документа XML. Декларация XML.

Первой строкой документа XML может быть декларация XML. Эта необязательная часть файла определяет его как файл XML, что может помочь автоматическим инструментам и людям распознавать файл как XML, а не SGML или другой способ разметки.

С помощью декларации указываем только версию XML в документе.

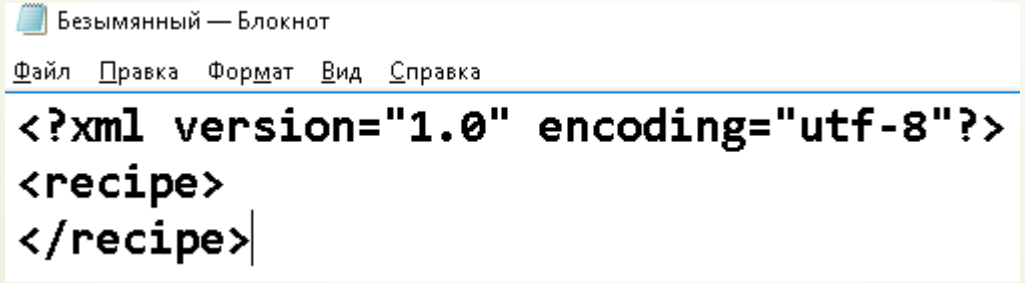


Добавляем к декларации описание применяемой в документе кодировки. В данном случае это Unicode.

Построение документа XML. Корневой элемент.

Начальный и замыкающий теги корневого элемента окружают весь текст XML-документа. В файле должен присутствовать только один корневой элемент, и это необходимая "обложка" для него.
По сути, корневой элемент – это то, что будет описываться в конкретном данном XML документе. Так как мы хотим записать рецепт мороженого, то корневым элементом будет РЕЦЕПТ.
Для того, чтобы его ввести нужен начальный тэг: <recipe>.
Далее вводится замыкающий тэг: </recipe>.

В самом документе это будет выглядеть так:



Построение документа XML. Создание обычных элементов.

После определения в корневом элементе “темы” документа настало время внести в документ обычные элементы, которые характеризуют отдельно взятый экземпляр из определенной темы.
Элементы также, как и корневой элемент имею начальный и замыкающий тэги.

Также, имеется ряд правил, стандартизующий имена этих элементов:
1. Пробелы в именах элементов не допускаются.
2. Имена должны начинаться с буквы, а не с цифры или знака. (После этой первой буквы можно использовать любую комбинацию из букв, цифр и допустимых символов.)
3. Регистр не имеет значения, но во избежание путаницы соблюдайте его.
4. При создании XML регистры начального и конечного тегов должны совпадать.

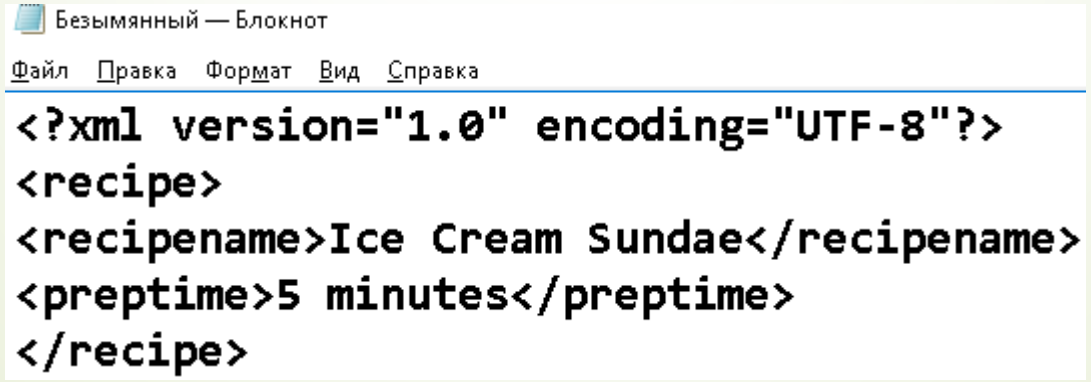
Между именами элемента с начальным и замыкающим тэгом помещается текущее значение.

Построение документа XML. Создание обычных элементов.

Для нашего примера с рецептом создадим два обычных элемента:

- название рецепта (<recipename> и </recipename>)
- время, затрачиваемое на приготовление (<preptime> и </preptime>)

В качестве значения recipename используем “Десерт из мороженого” - Ice Cream Sundae. В качестве значения preptime установим 5 минут – 5 minutes. Поместим обычные элементы внутри предварительно созданного корневого элемента. Вот так:



Построение документа XML. Вложение элементов.

Вложение - это размещение элементов внутри других элементов. Эти новые элементы называются *дочерними* элементами, а элементы, которые их окружают, - их *родительскими* элементами.

Слайдом выше мы можем сказать, что для корневого элемента *Рецепт* мы вложили элементы *Название рецепта* и *Время приготовления*.

Вкладывать элементы можно не только в корневой элемент, но и в обычные элементы. Для демонстрации, создадим еще один обычный элемент – *Список ингредиентов* <ingredlist> и </ingredlist> соответственно. В него поместим несколько вложенных элементов *Ингредиент из списка:* <listitem> и </listitem>.

Сделаем структуру вложения еще более сложной и в каждый элемент listitem Вложим еще по два элемента: *Количество:* <quantity> и </quantity>; *Описание Ингредиента:* <itemdescription> и </itemdescription>

При организации документа будем внимательно следить “лесенкой”, чтобы все было вложено в свое место.

Построение документа XML. Вложение элементов.

```
<?xml version="1.0" encoding="UTF-8"?>
<recipe>
        <recipename>Ice Cream Sundae</recipename>
                <ingredlist>
                        <listitem>
                                <quantity>3</quantity>
                                <itemdescription>chocolate syrup or chocolate fudge</itemdescription>
                        </listitem>
                        <listitem>
                                <quantity>1</quantity>
                                <itemdescription>nuts</itemdescription>
                        </listitem>
                        <listitem>
                                <quantity>1</quantity>
                                <itemdescription>cherry</itemdescription>
                        </listitem>
                </ingredlist>
        <preptime>5 minutes</preptime>
</recipe>
```

Построение документа XML. Атрибуты.

К элементам иногда добавляются *Атрибуты*. Атрибуты состоят из пары имя-значение, где значение берется в двойные кавычки ("), например: type="dessert" (тип нашего рецепта – десерт, или блюдо из курочки, или компот…) Атрибуты позволяют сохранять вместе с элементом дополнительные параметры, меняя значения этих параметров от элемента к элементу в одном и том же документе.

Атрибут, или несколько атрибутов указываются внутри начального тега элемента: <recipe type ="dessert">. При добавлении нескольких атрибутов они разделяются пробелами: <recipename cuisine="american" servings ="1">. К нашему рецепту с помощью атрибутов добавили информацию о том, что это блюду американской кухни, и что рецепт этот для одной порции. Приведем начальную часть документа XML с выбранными атрибутами.

Файл Правка Формат Вид Справка

```
<?xml version="1.0" encoding="UTF-8"?>
<recipe type="dessert">
<recipename cuisine="american" servings="1">Ice Cream Sundae</recipename>
<preptime>5 minutes</preptime>
</recipe>
```

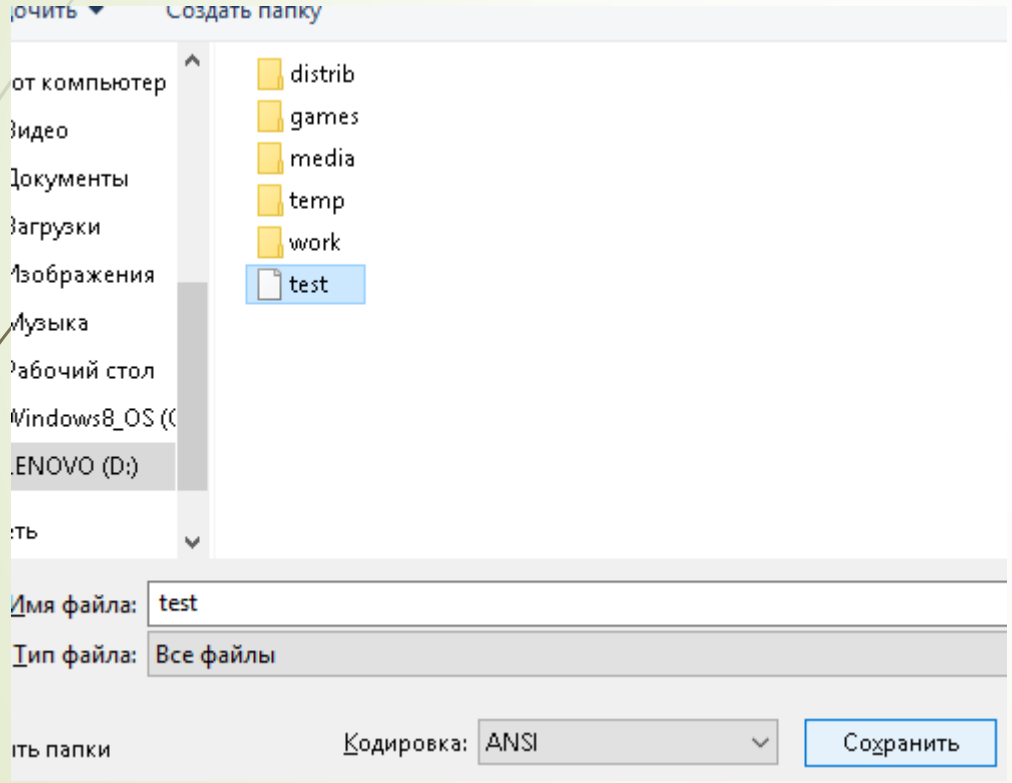
Правильно построенный документ XML.

Файл Правка Формат Вид Справка

```
<?xml version="1.0" encoding="UTF-8"?>
<recipe type="dessert">
<recipename cuisine="american" servings="1">Ice Cream Sundae</recipename>
<ingredlist>
<listitem><quantity units="cups">0.5</quantity>
<itemdescription>vanilla ice cream</itemdescription></listitem>
<listitem><quantity units="tablespoons">3</quantity>
<itemdescription>chocolate syrup or chocolate fudge</itemdescription></listitem>
<listitem><quantity units="tablespoons">1</quantity>
<itemdescription>nuts</itemdescription></listitem>
<listitem><quantity units="each">1</quantity>
<itemdescription>cherry</itemdescription></listitem>
</ingredlist>
<utensils>
<listitem><quantity units="each">1</quantity>
<utensilname>bowl</utensilname></listitem>
<listitem><quantity units="each">1</quantity>
<utensilname>spoons</utensilname></listitem>
<listitem><quantity units="each">1</quantity>
<utensilname>ice cream scoop</utensilname></listitem>
</utensils>
<directions>
<step>Using ice cream scoop, place vanilla ice cream into bowl.</step>
<step>Drizzle chocolate syrup or chocolate fudge over the ice cream.</step>
<step>Sprinkle nuts over the mound of chocolate and ice cream.</step>
<step>Place cherry on top of mound with stem pointing upward.</step>
<step>Serve.</step>
</directions>
<variations>
<option>Replace nuts with raisins.</option>
<option>Use chocolate ice cream instead of vanilla ice cream.</option>
</variations>
<preptime>5 minutes</preptime>
</recipe>
```

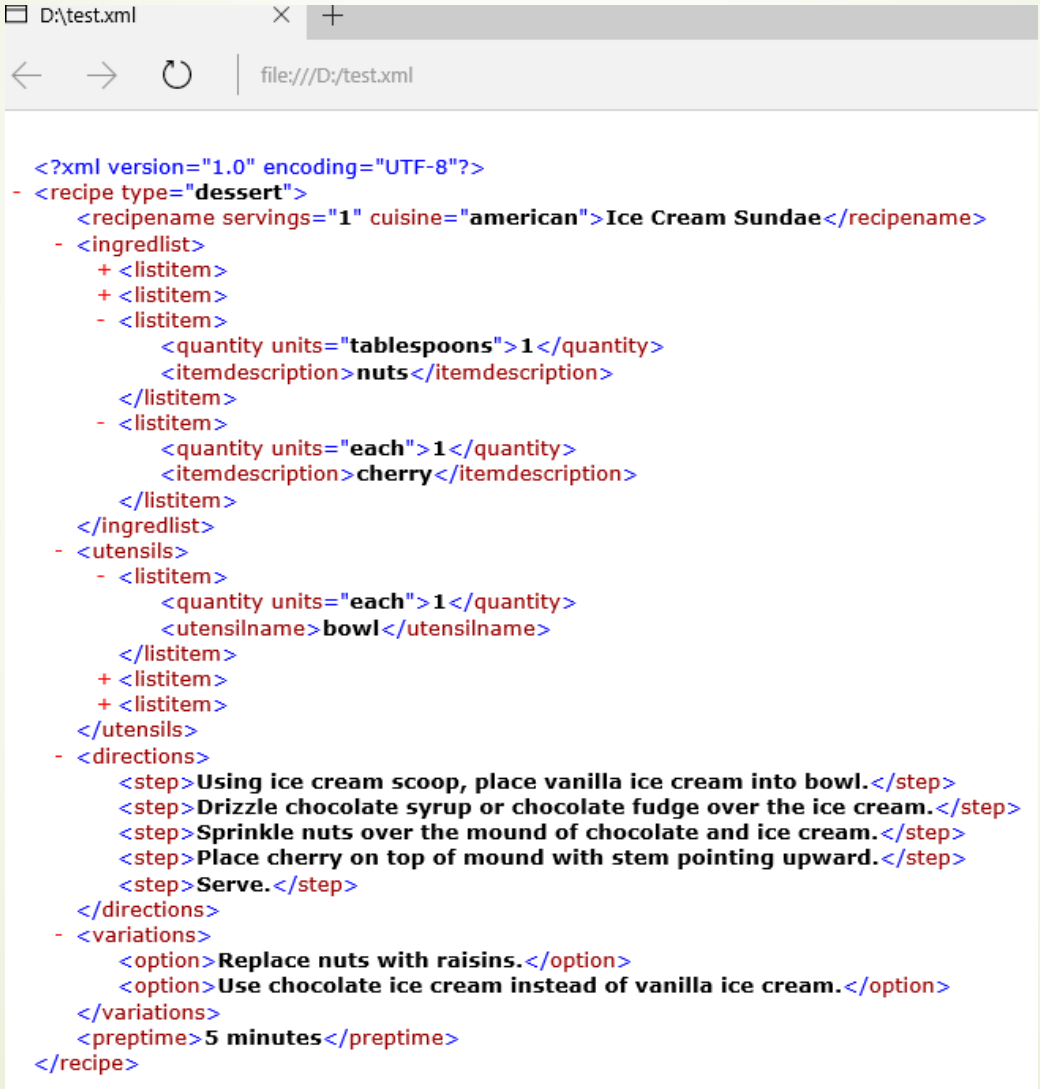
Дальнейшие действия с XML документом.

Дальше можно созданный документ сохранить в формате *.xml.



Удобнее всего его будет открыть для чтения в вашем браузере.

Демонстрация XML документа.



Смысл применения XML схем.

XML Schema - язык описания структуры XML-документа.
XML Schema была задумана для определения правил, которым должен подчиняться документ., при этом, XML Schema была разработана так, чтобы её можно было использовать в создании программного обеспечения для обработки документов XML.

После проверки документа на соответствие XML Schema, читающая программа может создать модель данных документа, которая включает:
- словарь (названия элементов и атрибутов);
- модель содержания (отношения между элементами и атрибутами и их структура);
- типы данных.

Также, через формат XML схемы данные из XML-документа можно выгрузить в СУБД.

Построение схемы XML. Декларация XML-схемы.

Для декларации XML схемы будем использовать самое простое выражение:

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

Документ XML схемы сохраняется в расширении *.xsd

Построение схемы XML. Создание обычных элементов.

Для определения обычного элемента, который есть в XML документе, в XML схеме необходимо указать имя элемента и один из встроенных в XML типов данных, которые может содержать этот элемент.
Наиболее часто используемыми являются следующие типы данных:

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

Напишем XML схему для наших обычных элементов из XML документа - recipename и preptime.

Файл Правка Формат Вид Справка

```
<recipename>Ice Cream Sundae</recipename>
<preptime>5 minutes</preptime>

<xs:element name="recipename" type="xs:string"/>
<xs:element name="preptime" type="xs:integer"/>
```

Описание из XML документа

Описание из XML схемы

Построение схемы XML. Создание вложенных элементов.

Вложенные элементы определяются в XML схеме с помощью группы операторов:
xs:complexType
xs:sequence
На примере созданного XML документа посмотрим, как можно решить задачу определения вложенных элементов в XML-схеме.

XML-документ

```
<?xml version="1.0" encoding="UTF-8"?>
<recipe>
<recipename>Ice Cream Sundae</recipename>
<ingredlist>
<listitem>
<quantity>3</quantity>
<itemdescription>chocolate syrup or chocolate fudge</itemdescription>
</listitem>
<listitem>
<quantity>1</quantity>
<itemdescription>nuts</itemdescription>
</listitem>
<listitem>
<quantity>1</quantity>
<itemdescription>cherry</itemdescription>
</listitem>
</ingredlist>
<preptime>5 minutes</preptime>
</recipe>
```

XML-схема

```
<xs:element name="recipename">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="ingredlist" type="xs:string"/>
       <xs:element name="listitem" type="xs:string"/>
       <xs:element name="quantity" type="xs:integer"/>
     </xs:sequence>
   </xs:complexType>
</xs:element>
```