|  |
| --- |
| **Задание 1** |

|  |  |  |
| --- | --- | --- |
| Создание XML документа в текстовом редакторе и проверка структуры документа в программе-броузере Internet Explorer.    Задание:    Создать XML документ в файле 1.xml следующей структуры:    1    Информационная часть документа содержится в элементах **NAIM** и **PRICE** и равна слову **Бензин** – для элемента **NAIM** и числу **20** – для элемента **PRICE**. Кодировка документа должна быть **WINDOWS-1251**, версия языка XML – **1.0**, документ должен быть автономным.    Выполнение задания:    1. Разработка разметки документа, согласно приведенной структуры:    1.1. Указать начальный и конечный теги корневого элемента:    <TOVAR>  </TOVAR>    1.2. Указать начальные и конечные теги 2-х дочерних элементов (NAIM и PRICE):    <TOVAR>  **<NAIM>**  **</NAIM>**  **<PRICE>**  **</PRICE>**  </MAIN>    1.3. Указать информационную часть документа в элементах NAIM и PRICE:    <TOVAR>  <NAIM>  **Бензин**  </NAIM>  <PRICE>  **20**  </PRICE>  </TOVAR>    1.4. Указать пролог -  версию языка XML, кодировки документа и автономности.  1.4.1. Пролог указывается в первой строке документа  1.4.2. Содержимое пролога следующее:  <?xml version=”1.0” encoding=”WINDOWS-1251” standalone=”yes”?>    1.5. Результат работы - текстовое содержимое документа:    **<?xml version=”1.0” encoding=”WINDOWS-1251” standalone=”yes”?>**  <TOVAR>  <NAIM>   Бензин  </NAIM>  <PRICE>   20  </PRICE>  </TOVAR>     |  |  | | --- | --- | | 1. | Создание документа в виде текстового файла в текстовом редакторе |   2.1. Открыть редактор Блокнот (Notepad)  2.1.1. Перейти в Главное меню Windows (кнопка **Пуск**)  2.1.2. Выбрать пункт **Программы**  2.1.3. Выбрать пункт **Стандартные**  2.1.4. Выбрать пункт **Блокнот**  2.2. Ввести указанный на шаге 1.5 текстовое содержимое документа  2.3. Сохранить файл с именем **1.xml**  2.3.1. Значение параметра **Тип файла**должно быть **Все файлы**  2.3.2. Запомнить папку, в которой был сохранен файл    **Примечание**: Можно использовать другой редактор, например редактор файлового менеджера FAR. При применении редактора Microsoft Word  для создания XML документов при сохранении файла использовать тип сохраняемого файла **обычный  текст**.    3. Проверка структуры файла  3.1. Открыть программу-браузер Internet Explorer  3.2. Выбрать пункт меню **Файл**, пункт **Открыть**  3.3. Выбрать файл **1.xml** из папки (п.2.3.2), в которой он был сохранен.    Вид документа должен соответствовать рисунку:    15 |

|  |
| --- |
| **Задание 2** |

|  |
| --- |
| Создание XML-документ из задания 1 в программе XMLPad c проверкой структуры.    1. Открыть программу **XMLPad**  1.1. Перейти в Главное меню Windows (кнопка **Пуск**)  1.2. Выбрать пункт **Программы**  1.3. Выбрать пункт **WMHelp XMLPad**  1.4. Выбрать значок **XMLPad**    **Примечание**: Если такого пункта в Главном меню нет, можно попытаться найти указанную программу, она расположена в папке **Program Files\WMHelp Software\WMHelp XmlPad** в файле **XMLPad.exe**. После запуска программы появится окно, показанное на рис.2    16    Рис. 2    2. Выбрать пункт меню **File**  2.1. В меню **File** выбрать пункт **New**  2.2. выбрать **XML Document** в появившемся окне с шаблонами создаваемых файлов (рис. 3)    17    Рис. 3    После выбора указанного шаблона файлов, появится окно для указания дополнительных параметров файла, показанное на рис. 4    18    Рис. 4    3. Выбрать пункт **Not assigned** в разделе **DTD definition**  3.1. Указать **MAIN** в поле **Root Tag**  3.2. Нажать кнопку **ОК**.    Создаваемый документ будет иметь вид, показанный на рис. 5    19    Рис. 5    4. Создание пролога документа  4.1. Изменение кодировки  4.1.1. Выбрать пункт меню **File**  4.1.2. Выбрать пункт **Advanced Save Option**  4.1.3. Указать кодировку согласно заданию – **WINDOWS-1251**    Вид окна с указанием кодировки показан на рис. 6    20    Рис. 6    Документ приобретет вид, показанный на рис. 7    21    Рис. 7    4.2. Задание параметра standalone  4.2.1. Перейти в тексте документа на первую строку  4.2.2. Написать код: **standalone=”yes”**    Документ приобретет вид, показанный на рис. 8    22    Рис. 8    5. Удалить строку **02** с пустым элементом MAIN    6. Ввести содержимое документа (разметку и текстовое содержание):    <TOVAR>  <NAIM>Бензин</NAIM>  <PRICE>20</PRICE>  </TOVAR>    **Замечание:** Конечный тег элементов проставляется автоматически (дополнительно вводить не требуется)    Документ должен принять вид, показанный на рис. 9    23    Рис. 9    7. Выполнить проверку структуры документа  7.1. Выбрать пункт меню **XML**  7.2. Выбрать пункт **Validate**    В случае правильно оформленного документа в окне **Task List** должна быть отображена одна строка с надписью **No DTD for the document**. как на рис. 10.    24    Рис. 10    8. Сохранить документ  8.1. Выбрать пункт меню **File**  8.2. Выбрать пункт **Save**  8.3. Указать имя файла  8.4. Запомнить папку, в которую сохраняли документ    После сохранения имя файла появится в заголовке окна как на рис. 11    25 |

|  |
| --- |
| **Задание 3** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Задание    Исправить ошибки в документе со следующим содержимым:    <?xml version=”2.0” encoding=”Windows-1252”?>  <Tovars>  <tovar>   <naim>    Бензин   </price>   <price>    200   </naim>  </tovar>  </tovars>  </tovar0>  <tovar1>  </tovaR>    Выполнение задания:   |  |  | | --- | --- | | 1. | Открыть программу XMLPad | | 2. | Создать новый XML документ |  |  |  | | --- | --- | | 3. | Удалить все содержимое (разметку и информационное содержание) | | 4. | Вставить указанное в задание содержимое. |     Вид окна программы показан на рис. 12    26    Рис. 12    5. Проанализировать ошибки в документе (для этого можно сохранить документ).    **Примечание:** Ошибки в документе перечислены в окне **Task List** (в нижней части окна программы). Номер строки с ошибкой указан в колонке **Line** окна **Task List**.    **Список ошибок:**    6. Ошибка в строке 1 - неправильное оформление пролога.    **Устранение ошибки:**    Требуется заменить символы  псевдокавычек (І) на символ кавычки (“). Это стандартная ошибка, которая возникает при переносе текста из программы Microsoft Word. Сообщение об ошибке изменится  (рис. 13)    27    Рис. 13    7. Ошибка в строке 6 – нет конечного тега для элемента **naim**    **Устранение ошибки** –Поменять местами конечные теги </naim>, </price>. После взаимной замены этих тегов сообщение об ошибке изменилось (рис. 14)    28    Рис. 14    8. Ошибка в строке 11 – несоответствие начального тега (строка 2)  и конечного тега (строка 11).    **Устранение ошибки** – Изменить имя конечного тега в строке 11. После изменения, сообщение об ошибке изменилось и показано на рис. 15:    29    Рис. 15    9. Ошибка в строке 12 ‑ наличие фрагментов элементов после закрывающего тега корневого элемента (/tovar0, tovar1).    **Устранение ошибки** - Удаление всех строк после закрывающего тега корневого элемента.    После устранения ошибки документ стал правильно оформленным (сообщений об ошибках больше нет), что показано на рис. 16    30    Рис. 16    10. Некорректное отображение документа (вкладка Preview)    Неверно отображаются русские буквы (рис. 17)    31    Рис. 17    **Устранение ошибки:** Требуется изменить кодировку документа (значение атрибута **encoding**) в прологе документа. Процедура изменения описана в пукте 4.1 Задания 1    После задания указанных установок, содержимое документа отображается корректно, что показано на рис. 18    32    Рис. 18    Дополнительно требуется проверить правильное значение версии языка XML в прологе документа (правильное значение версии – **1.0**). Вид окна программы показан на рис. 19:    33    Рис. 19 |

|  |
| --- |
| **Задание 4** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Создание документа с применением атрибутов**    **Задание**    Создать документ со структурой, указанной на рис. 1    34    Рис. 1    Дополнительные требования: Имя файла: 1.xml    Требования к структуре документа: TOVARS и TOVAR– задать в виде элементов, PRICE и NAIM – в виде атрибута.    Документ должен содержать информацию о трех товарах, реквизиты которых описаны в таблице 1:    Таблица 1     |  |  | | --- | --- | | **TOVAR** | **PRICE** | | Керосин | 40 | | Помада | 300 | | Хлеб | 30 |     Документ должен быть правильно оформлен.    **Решение задания:**    Для решения задачи применяется программа XMLPad.    1. Открыть программу XMLPad и создать новый документ. Эта процедура подробно описана в задании 2 практической работы к уроку 1.    При задании имени корневого тега (Root tag) следует указать TOVARS (рис. 2)    35    Рис. 2    После указания корневого тега и нажатии кнопки OК, в программе создается документ, у которого требуется поменять кодировку. Процесс изменения кодировки описан в практическом задании к уроку 1, задание 2, п. 4.1    После изменения кодировки на WINDOWS-1251 содержимое документа показано на рис. 3    36    Рис. 3    Так как требуется создать документ только для 3-х товаров, можно создать 3 «заготовки» для элементов-товаров. Можно прости ввести с клавиатуры начальные и конечные теги. Результат показан на рис. 4    37    Рис. 4    На первом этапе можно создать атрибуты для первого элемента – товара с наименованием **Керосин**    Для этого можно создать атрибут **NAIM** и указать значение **Керосин**Это можно сделать, набрав заданный текст с клавиатуры, результат работы должен совпадать с видом документа на рис. 5.    38    Рис. 5    Аналогично можно ввести и атрибут **PRICE** и указать значение **40** Результат должен соответствовать рис. 6    39    Рис. 6    Аналогично создаются и заполняются атрибуты у двух других элементов. Результат должен соответствовать документу на рис. 7    40    Рис. 7    Для проверки правильности оформления документа требуется в пункте меню **XML** выбрать строку **Validate** или нажать клавишу **F7**.    В случае правильно оформленного документа, в окне Task List (нижнее окно программы) появится строка **No DTD for the document**  (рис. 8)    41    Рис. 8    Если документ оформлен неверно (например имя атрибута не соответствует правилам), выводится информация о найденной ошибке (рис. 9):    42    Рис. 9    Дополнительная проверка документа происходит при сохранении. Сохранение производится при помощи пункта **Сохранить** меню **Файл**. После сохранения имя файла отображается на заголовке вкладки (рис. 10)    43    Рис. 10 |

|  |
| --- |
| Задание 5 |

|  |
| --- |
| **Проверка документа на соответствие заданной структуре**    **Задание**    Структура документа показана на рис. 11    34    Рис. 11    Реквизиты TOVARS, TOVAR и PRICE должны быть оформлены как элементы, а реквизит NAIM– как атрибут. Документ должен быть правильно оформлен.    Содержимое документа:    <?xml version = "1.0" encoding = "WINDOWS-1251"?>  <TOVARS>  <TOVAR NA IM="Керосин" PRICE="40"></TOVAR>  <TOVAR NAIM="Помада" PRICE="300"></TOVAR>  <TOVAR NAI M="Хлеб" PRICE="30"></TOVAR>  </TOVARS>    **Решение задания:**    Задание предлагается решить в программе XMLPAD.    После открытия файла с документом XML, окно программы приобретает вид, показанный на рис. 12    44    Рис. 12    После преобразования атрибутов **PRICE** в элементы **PRICE** (выполняется простым набором текста) вид документа становится следующим (рис. 13)    45    Рис. 13    Выполненная проверка правильности оформления документа, отображает ошибку на строке 3 (рис. 14)    46    Рис. 14    После приведения имен атрибутов в соответствие правилам (удалены пробелы), документ становится правильно оформленным (рис. 15).    47    Рис. 15 |

|  |
| --- |
| Задание 6 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Создание схемы DTD для XML документа**    **Задание**    Создать схему DTD для документа со структурой, указанной на рис. 1    48    Рис. 1    Дополнительные требования: Имя файла документа: 1.xml    Требования к структуре документа: TOVARS и TOVAR– задать в виде элементов, PRICE, NAIM, DESCRIPTION и VALUTA – в виде атрибутов.    Элементы TOVAR могут повторяться несколько раз.    Атрибуты NAIM, PRICE, VALUTA являются обязательными, атрибут DESCRIPTION – необязателен.    Документ должен содержать информацию о трех товарах, реквизиты которых описаны в таблице 1:    Таблица 1     |  |  |  |  | | --- | --- | --- | --- | | **NAIM** | **PRICE** | **DESCRIPTION** | **VALUTA** | | Керосин | 40 | Топливо | RUB | | Помада | 300 | Косметика | RUB | | Хлеб | 30 |  | RUB |     Схема должна быть расположена внешне, в файле с именем 1.dtd.    **Решение задания:**    Для решения задачи применяется программа XMLPad.    1. Открыть программу XMLPad и создать новый документ. Эта процедура подробно описана в задании 1 практической работы к уроку 2.    Содержимое документа (без ссылки на схему DTD):    <?xml version = "1.0" encoding = "WINDOWS-1251"?>  <TOVARS>  <TOVAR NAIM="Керосин" PRICE="40" DESCRIPTION=”Топливо” VALUTA=”RUB”></TOVAR>  <TOVAR NAIM="Помада" PRICE="300" DESCRIPTION=”Косметика” VALUTA=”RUB”></TOVAR>  <TOVAR NAIM="Хлеб" PRICE="30" VALUTA=”RUB”></TOVAR>  </TOVARS>    После выполнения проверки правильности оформления документа в программе XMLPad требуется убедиться в отсутствии ошибок (рис. 2)    49    Рис. 2    Для создания схемы DTD в программе XMLPad требуется создать новый документ, с типом **Document Type Difinition**(рис. 3)    50    Рис. 3    После нажатия на кнопку ОК создается пустая схема DTD с предопределенным заголовком и пустым корневым элементом, показанная на рис. 4    51    Рис. 4    Для корректной работы схемы требуется удалить всю информацию и ввести заново. Ввод информацию в схему начинается с указания корневого элемента и всех его дочерних элементов, при этом получается схема, показанная на рис. 5    52    Рис. 5    Было введено имя корневого тега **TOVARS** и имя дочернего элемента (**TOVAR**). Символ + после имени дочернего элемента означает, что он может повторяться несколько раз и будет включен в документ не менее одного раза.    Затем требуется задать атрибуты NAIM, PRICE, DESCRIPTION, VALUTA элемента TOVAR, причем атрибут DESCRIPTION являются необязательным, атрибут VALUTA является обязательным и может принимать одно из 3-х значений (**RUB**, **USD**, **EUR**). Вид схемы показан на рис. 6    53    Рис. 6    После указания всех элементов и атрибутов в схеме DTD, схема сохраняется в файле 1.dtd, который должен находится в той же директории (папке), что и XML-документ 1.xml    После этого потребуется перейти к вкладке с файлом 1.xml и добавить в его заголовке ссылку на схему DTD:    **<!DOCTYPE TOVARS SYSTEM "1.dtd">**    В результате XML - документ принимает вид, который показан на рис. 7    54    Рис. 7    Для проверки соответствия документа схеме DTD следует выбрать в меню **XML** пункт **Validate**. После успешной проверки должно появится окно, показанное на рис. 8    55    Рис. 8 |

|  |
| --- |
| Задание 7 |

|  |
| --- |
| **Проверка соответствия XML-документа на соответствие схеме DTD**    **Задание**    Проверить XML-документ на соответствие схеме DTD. В случае наличия несоответствия исправить документ.    **Содержимое схемы:**    <!ELEMENT TOVARS (TOVAR+)>  <!ELEMENT TOVAR (NAIM)>  <!ATTLIST TOVAR PRICE CDATA #REQUIRED>  <!ATTLIST TOVAR CUR CDATA #FIXED 'RUB'>  <!ATTLIST TOVAR DESCRIPTION CDATA #IMPLIED>  <!ELEMENT NAIM (#PCDATA)>    **Содержимое XML документа**    <?xml version="1.0" encoding="windows-1251" ?>  <TOVARS>  <TOVAR CUR=”RUB” PRICE=”20”>  <NAIM>Бензин</NAIM>  </TOVAR>  <TOVAR>  <NAIM>Керосин</NAIM>  <PRICE>40</PRICE>  </TOVAR>  </TOVARS>    После загрузки XML документа и схемы DTD в рекактор XMLPad, выполняется проверка соответствия схемы и структуры документа (меню **XML**, пункт **Validate**). Найденные несоответствия отображаются в нижнем окне редактора (рис. 9)    56    Рис. 9    Первая ошибка найдена на строке 7, строка выделена красным цветом. В элементе TOVAR отсутствует атрибут PRICE. Для устранения этой ошибки потребуется создать атрибут PRICE, значением которого будет значение дочернего элемента PRICE. Элемент PRICE подлежит удалению. После выполненных изменений документ примет вид, показанный на рис. 10.    57    Рис. 10    При проверке соответствия структуры документа и схемы DTD ошибок не обнаружено (рис. 11)    58    Рис. 11    Вид документа при просмотре в браузере показан на рис. 12.    59    Рис. 12    Как видно из рис. 12, при просмотре в браузере атрибут **CUR** у второго элемента **TOVAR** появился автоматически по причине объявления его в схеме **DTD** как **FIXED**. |

|  |
| --- |
| Задание 8 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Создание документа с применением пространства имен**    **Задание**    Создать документ со структурой, указанной на рис. 1    60    Рис. 1    Дополнительные требования: Имя файла: 1.xml    Документ должен быть создан с использованием элементов.    В документе будет храниться информация о наличии товара на складе, каждый из товаров характеризуется наименованием (NAIM), ценой (PRICE) и упаковкой. Каждая упаковка характеризуется наименованием (NAIM) и количеством единиц товара в упаковке (KOL\_ED) и количеством упаковок в наличии (KOL\_VO). Также каждая упаковка характеризуется единицами измерения товара (ED\_IZM), в свою очередь единицы измерения товара (ED\_IZM) характеризуются наименованием (NAIM) и дополнительным описанием (DOP\_OPIS). Дополнительное описание является необязательным элементом.    Документ должен содержать информацию о трех товарах, реквизиты которых описаны в таблице 1:    Таблица 1     |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | TOVAR | PRICE | UPAK | | | ED\_IZM | | | NAIM | KOL\_VO | KOL\_ED | NAIM | DOP\_OPIS | | **Керосин** | **40** | **Ящик** | **10** | **20** | **Бутылка** | **0,5л** | | **Помада** | **300** | **Коробка** | **20** | **10** | **Тюбик** |  | | **Хлеб** | **30** | **Поддон** | **30** | **10** | **Буханка** | **0,5кг** |     Документ должен быть правильно оформлен.    **Решение задания:**    Для решения задачи применяется программа XMLPad.    1. Открыть программу XMLPad и создать новый документ. Эта процедура подробно описана в задании 2 практической работы к уроку 1.Создание документа описано в задании 1 практической работы к уроку 2.    Содержимое документа без применения пространства имен будет следующим:    <?xml version="1.0" encoding="windows-1251"?>  <TOVARS>  <TOVAR>   <NAIM>Керосин</NAIM>   <PRICE>40</PRICE>   <UPAK>    <NAIM>Ящик</NAIM>    <KOL\_VO>10</KOL\_VO>    <KOL\_ED>20</KOL\_ED>    <ED\_IZM>     <NAIM>Бутылка</NAIM>     <DOP\_OPIS>0,5л</DOP\_OPIS>    </ED\_IZM>   </UPAK>  </TOVAR>    <TOVAR>   <NAIM>Помада</NAIM>   <PRICE>300</PRICE>   <UPAK>    <NAIM>Коробка</NAIM>    <KOL\_VO>20</KOL\_VO>    <KOL\_ED>10</KOL\_ED>    <ED\_IZM>     <NAIM>Тюбик</NAIM>    </ED\_IZM>   </UPAK>  </TOVAR>    <TOVAR>   <NAIM>Хлеб</NAIM>   <PRICE>30</PRICE>   <UPAK>    <NAIM>Поддон</NAIM>    <KOL\_VO>30</KOL\_VO>    <KOL\_ED>10</KOL\_ED>    <ED\_IZM>     <NAIM>Буханка</NAIM>     <DOP\_OPIS>0,5кг</DOP\_OPIS>    </ED\_IZM>   </UPAK>  </TOVAR>  </TOVARS>    В этом документе отсутствует однозначность при определении элементов. Есть несколько элементов с одним названием – при автоматическом анализе документа могут возникнуть сложности.    Поэтому требуется задать пространство имен согласно таблице 2. В качестве базового URL адреса будет использован сайт организации: **http://www.sklad.ru**    Таблица 2     |  |  |  | | --- | --- | --- | | Элементы для пространства имен | Префикс | URL\_адрес | | **TOVAR** | **TOV** | <http://www.sklad.ru/tovar> | | **UPAK** | **UPK** | <http://www.sklad.ru/upack> | | **ED\_IZM** | **EDI** | <http://www.sklad.ru/edizm> |     После добавления пространств имен в корневой элемент документа (TOVARS), документ приобретает следующий вид:    <?xml version="1.0" encoding="windows-1251"?>  <TOV:TOVARS  xmlns:TOV="http://www.sklad.ru/tovar"  xmlns:UPK="http://www.sklad.ru/upack"  xmlns:EDI="http://www.sklad.ru/edizm"  >  <TOV:TOVAR>   <TOV:NAIM>Керосин</TOV:NAIM>   <TOV:PRICE>40</TOV:PRICE>   <UPK:UPAK>    <UPK:NAIM>Ящик</UPK:NAIM>    <UPK:KOL\_VO>10</UPK:KOL\_VO>    <UPK:KOL\_ED>20</UPK:KOL\_ED>    <EDI:ED\_IZM>     <EDI:NAIM>Бутылка</EDI:NAIM>     <EDI:DOP\_OPIS>0,5л</EDI:DOP\_OPIS>    </EDI:ED\_IZM>   </UPK:UPAK>  </TOV:TOVAR>    <TOV:TOVAR>   <TOV:NAIM>Помада</TOV:NAIM>   <TOV:PRICE>300</TOV:PRICE>   <UPK:UPAK>    <UPK:NAIM>Коробка</UPK:NAIM>    <UPK:KOL\_VO>20</UPK:KOL\_VO>    <UPK:KOL\_ED>10</UPK:KOL\_ED>    <EDI:ED\_IZM>     <EDI:NAIM>Тюбик</EDI:NAIM>    </EDI:ED\_IZM>   </UPK:UPAK>  </TOV:TOVAR>    <TOV:TOVAR>   <TOV:NAIM>Хлеб</TOV:NAIM>   <TOV:PRICE>30</TOV:PRICE>   <UPK:UPAK>    <UPK:NAIM>Поддон</UPK:NAIM>    <UPK:KOL\_VO>30</UPK:KOL\_VO>    <UPK:KOL\_ED>10</UPK:KOL\_ED>    <EDI:ED\_IZM>     <EDI:NAIM>Буханка</EDI:NAIM>     <EDI:DOP\_OPIS>0,5кг</EDI:DOP\_OPIS>    </EDI:ED\_IZM>   </UPK:UPAK>  </TOV:TOVAR>  </TOV:TOVARS>    После сохранения документа в файле с именем 2.xml, при просмотре документа в браузере Internet Explorer, вид документа будет соответствовать рис. 2:    61 |

|  |
| --- |
| Задание 9 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Создание документа XML на основе схемы XML**    **Задание**    Требуется создать для опубликования на сайте прайс-листа товаров документ XML на основе схемы XML. Схема представлена ниже:    <?xml version="1.0" encoding="windows-1251"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">  <xs:element name="price">   <xs:complexType>    <xs:sequence>     <xs:element name="date\_make" type="xs:date"/>     <xs:element name="tovars">      <xs:complexType>       <xs:sequence>        <xs:element name="tovar" minOccurs="1" maxOccurs="unbounded">         <xs:complexType>          <xs:attribute name="name" type="xs:string" use="required"/>          <xs:attribute name="description" type="xs:string" use="optional"/>          <xs:attribute name="price" type="xs:decimal" use="required"/>         </xs:complexType>        </xs:element>       </xs:sequence>      </xs:complexType>     </xs:element>    </xs:sequence>   </xs:complexType>  </xs:element>  </xs:schema>    Схема сохраняется в файле с именем **u5\_p1.xsd** и должна быть расположена в той же директории (на том же сайте) где расположен созданный документ XML. Имя файла с документом XML - **u5\_p1.xml**.    Для создания документа необходимо провести анализ схемы.    В схеме описаны 3 элемента  с именами: price, date\_make, tovars, tovar. Корневым элементом является price. Элемент date\_make имеет информационное содержимое типа даты, элемент tovars содержит дочерние элементы tovar (количество их неограниченно, но не менее одного). Элементы tovar имеют атрибуты с именами name, description и price. Информационное значение атрибута price – числовое, атрибутов description и name – строковое. Атрибут description не является обязательным, остальные атрибуты – обязательны.    После анализа схемы можно сделать вывод о том, что в элементе data\_make содержится дата формирования прайс-листа в формате гггг-мм-дд. Атрибуты элемента tovar  (name, description и price) содержат соответственно наименование товара, его описание и цену. Описание товара можно не указывать. Более наглядно схема показана на рис. 1    62    Рис. 1    При создании документа можно поместить в качестве информационного содержимого следующую информацию: Дата создания прайс-листа: 21.12.2008    Информация о товарах представлена в таблице 1:    Таблица 1     |  |  |  | | --- | --- | --- | | **Название** | **Описание** | **Цена** | | Керосин | Горючее | 10.20 | | Помада |  | 100.20 |     На основе схемы рис. 1 и информационного содержания схемы создан следующий документ:    <?xml version="1.0" encoding="windows-1251"?>  <price>  <date\_make>2009-12-20</date\_make>  <tovars>   <tovar name="Керосин" description="Горючее" price="10.20"/>   <tovar name="Помада" price="100.20"/>  </tovars>  </price>    Для подключения схемы к документу XML, требуется указать в корневом элементе информацию об имени файла со схемой и дополнительную служебную информацию.    Документ принимает следующий вид:    <?xml version="1.0" encoding="windows-1251"?>  <price xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="u5\_p1.xsd">  <date\_make>2009-12-21</date\_make>  <tovars>   <tovar name="Керосин" description="Горючее" price="10.20"/>   <tovar name="Помада" price="100.20"/>  </tovars>  </price>    Для контроля документа на соответствие схеме файл с документом и файл со схемой требуется открыть в редакторе XMLPad. Вид редактора с двумя открытыми файлами показан на рис. 2    63    Рис. 2    Для проверки соответствия требуется, находясь в окне с документом XML выбрать пункт меню **XML**/**Validate** или нажать клавишу **F7**.    В случае соответствия документа и схемы должно появится сообщение, показанное на рис. 3    64    Рис. 3    Для дополнительной проверки можно сознательно «испортить» документ – например поместить в атрибут price любой нецифровой символ – например букву и выполнить проверку. При этом в нижней части редактора в окне Task List появится сообщение об ошибке (рис. 4)    65    Рис. 4    После устранения ошибки (удаления символа из атрибута price) документ соответствует схеме (рис. 5)    66    Рис. 5 |

|  |
| --- |
| Задание 10 |

|  |
| --- |
| **Исправление ошибок в документе для соответствия его заданной схеме**    **Задание**    Схема XML соответствует схеме задания 9.    Документ имеет следующий вид:    <?xml version="1.0" encoding="windows-1251"?>  <price xmlns:xsi="http://www.w3.org/2001/XMLSchema-instanse" xsi:noNamespaceSchemaLocation="u5\_p1.xds">   <tovars>           <tovar name="Керосин" descrittion="Горючее" price="10-20"/>           <tovar price="100.20"/>   </tovars>   <date\_make>21.12.2009</date\_make>  </price>    После проверки содержимого в редакторе XMLPad отображаются сообщения об ошибке (рис. 6):    67    Рис. 6    Первая ошибка расположена на строке 2 – пространство имен указано неверно (должен быть указан **2001**, а не **1999**). После исправления количество ошибок не уменьшилось. Сравнив документы на рис. 6 и на рис. 5, требуется исправить ссылку на пространство имен, правильное название - **XMLSchema-instance** После исправления ошибки и проверки содержимого, вид документа показан на рис. 7. Видно, что количество ошибок уменьшилось.    68    Рис. 7    Следует обратить внимание на следующий момент – имя файла со схемой указано как u5\_p0.xds, в то время как правильное название файла – u5\_p1.xsd. Эта ошибка может не отображаться редактором по причине «кэширования» содержимого файла схемы. Для отображения этой ошибки требуется обновить вид документа (клавиша F5). После обновления отображается только одна ошибка – файл с указанным именем не существует (рис. 8)    69    Рис. 8    После исправления имени файла и повторного обновления документа (клавиша F5), сообщение об ошибке может не исчезнуть. В этом случае потребуется закрыть окно с документом и открыть его заново. После проверки документа количество ошибок изменилось (рис. 9)    70    Рис. 9    Одна из ошибок вызвана тем, что на строке 5 в атрибуте price информационное содержимое не соответствует числовому типу (разделитель целой и дробной части указан –, а не точка). После исправления этой ошибки можно исправить имя атрибута descritton на description . После исправления этих ошибок и дополнительной проверки документа количество ошибок изменилось (рис. 10)    71    Рис. 10    Следующая ошибка возникла на строке 10 – нет обязательного атрибута с именем name. Требуется добавить этот атрибут и заново проверить содержимое документа. В результате проверки осталось 2 ошибки на строке 8 (рис. 11)    72    Рис. 11    Одна из ошибок вызвана неправильным форматом даты – дата представлена в формате **дд.мм.гггг** в то время как требуется – **гггг-мм-дд** После изменения формата даты и проверки ошибок осталась одна ошибка (рис. 12)    73    Рис. 12    Ошибка вызвана тем, что элемент date\_make должен быть указан «выше» по тексту документа, чем элемент tovars. После изменения положения данного элемента документ соответствует схеме (рис. 13)    74    Рис. 13 |

|  |
| --- |
| Задание 11 |

|  |
| --- |
| **Поиск в документе с использованием XPATH**    **Задание**    Требуется найти все элементы с именем naim в документе 1:    **Документ 1**    <?xml version="1.0" encoding="windows-1251"?>  <price>  <naim>Прайс-лист нефтепродуктов</naim>  <firma>   <naim>Торговля-Нефть</naim>  </firma>  <tovars>   <tovar vid="Топливо">    <naim>Бензин</naim>    <price>20</price>   </tovar>   <tovar vid="Топливо">    <naim>Керосин</naim>    <price>40</price>   </tovar>  </tovars>  </price>    **Решение задания:**    Начинать решение задания требуется с составления XPATH-выражения. Поиск является абсолютным, то выражение начинается с символа «слэш» / Так как требуется найти элемент любого уровня подчиненности – дочерний, внучатый, правнучатый, то вторым символом выражения является также символ слэш / Затем в выражение надо добавить имя элемента. В результате получается выражение - //naim    Для контроля поиска можно выписать элементы, которые удовлетворяют результатам поиска. Это следующие элементы (с информационным содержимым):    <naim>Прайс-лист нефтепродуктов</naim>  <naim>Торговля-Нефть</naim>  <naim>Бензин</naim>  <naim>Керосин</naim>    Для осуществления поиска текст **Документа 1**требуется сохранить в виде XML файла и загрузить в редактор XMLPad. Вид редактора показан на рис. 1    75    Рис. 1    Для поиска в документе с применением XPATH выражения требуется выполнить пункт **Evaluate XPATH** меню **XML**. При этом откроется окно для ввода XPATH выражения, показанное на рис. 2    76    Рис. 2    Для поиска, в поле для ввода **XPath** требуется ввести выражение и нажать кнопку **Evaluate**. Результат поиска показан на рис. 3    77    Рис. 3    Полученный результат совпадает с ожидаемым, из чего можно сделать вывод о правильности решения задачи. |

|  |
| --- |
| Задание 12 |

|  |
| --- |
| Для **Документа 1** из [**задания 11**](https://sdo.niu.ranepa.ru/close/store/books/%7B45FB1951-3E99-49FD-9C54-AFAE74E8CD4E%7D/z11.htm) требуется найти информационное содержание всех элементов **naim**.    **Решение задания:**    Решение задания начинается с составления XPath выражения. Так как требуется найти информационное содержание, выражение должно закончится функцией text(). Начало выражения соответствует выражению из [**задания 11**](https://sdo.niu.ranepa.ru/close/store/books/%7B45FB1951-3E99-49FD-9C54-AFAE74E8CD4E%7D/z11.htm). Выражение принимает следующий вид: //naim/text()    Методика поиска описана в решении задания 1. После ввода выражения в поле XPath окна поиска XPATH, получается  результат, показанный на рис. 4    78    Рис. 4    Как видно, результат поиска очень похож на результат поиска задания 1, но содержит только информационную часть найденных элементов. |

|  |
| --- |
| Задание 13 |

|  |
| --- |
| Найти все элементы с именем tovar и положением с номером 2.    **Решение задания:**    Составление выражения – для нахождения элементов tovar используется следующее выражение: **//tovar** Для нахождения элемента с заданным положением используется номер в квадратных скобках. Следовательно, полное выражение - **//tovar[2]**    Предварительный контроль результата поиска. При анализе структуры документа видно, что количество элементов **tovar** в документе равно двум. Элемент tovar с положением номер 2 будет следующим:     <tovar vid="Топливо">    <naim>Керосин</naim>    <price>40</price>   </tovar>    Для поиска заданного элемента требуется ввести указанное выражение в поле XPATH окна поиска, получены результат показан на рис. 5    79    Рис. 5    Полученный результат соответствует предварительно полученному, значит выражение составлено верно. |

|  |
| --- |
| Задание 14 |

|  |
| --- |
| **Найти атрибуты**vid**для всех элементов**    **Решение задания:**    Перед решением задания требуется составить выражение. Так как осуществляется поиск атрибутов vid, выражение заканчивается именем атрибута, который начинается с символа @  - @vid. Для нахождения всех атрибутов используется стандартный синтаксис XPATH – два символа «слэш» - // полное выражение - **//@vid**    Перед решением требуется провести предварительный контроль результата – в документе содержатся 2 атрибута vid с одинаковым содержимым, но у разных элементов –    vid="Топливо"  vid="Топливо"    Для решения задания будет использовано окно поиска со сформированным выражением. Результат поиска показан на рис. 6    80    Рис. 6 |

|  |
| --- |
| Задание 15 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Преобразование XML-документа в HTML-документ с использованием XSL-файла**    **Задание**    Для документа следующего вида:    <?xml version="1.0" encoding="WINDOWS-1251"?>  <tovars>  <opis>Товары фирмы</opis>  <tovar>   <naim>Керосин</naim>   <price>1</price>  </tovar>  <tovar>   <naim>Бензин</naim>   <price>5</price>  </tovar>  <tovar>   <naim>Дизтопливо</naim>   <price>11</price>  </tovar>  </tovars>    создать XSL-файл для преобразования в html-файл, показанный на рис. 1:    81    Рис. 1    **Анализ требуемого преобразования**    До составления XSL-файла требуется проанализировать требуемые преобразования.   |  |  | | --- | --- | | • | Содержимое элемента <opis> должно быть представлено как заголовок первого уровня (H1) | | • | Подчиненные элементы tovar должны быть представлены в виде таблицы, причем содержимое элементов naim и price должно быть показано в отдельных колонках. |  |  |  | | --- | --- | | • | Таблица должна быть упорядочена по колонке Цена в убывающем порядке по числовому типу | | • | Таблица должна иметь заголовок |     **Реализация преобразования**    До проведения преобразования требуется создать XML-документ в редакторе XMLPad. Процесс создания XML документа рассматривался на уроке 1. Создать XSL файл также производится в редакторе XMLPad. Для этого требуется в меню **File** выбрать пункт **New** и в появившемся списке типов файлов выбрать XSL Styleshet (рис. 2)    82    Рис. 2    Откроется окно с шаблоном файла. Изменение файла требуется начать с кодировки – указать кодировку **windows-1251** (рис. 3)    83    Рис. 3    Далее потребуется сохранить XSL-файл с именем price\_pr1.xsl в тот же каталог, где находится исходный XML-файл и добавить в XML-файл ссылку на XSL-файл (второй строкой). Вид XML-файла показан на рис. 4    84    Рис. 4    Затем потребуется добавить правило соответствия для корневого элемента XPath - <xsl:template match=”/”>. Элемент должен иметь начальный и конечный теги. Вид документа показан на рис. 5    85    Рис. 5    На следующем шаге требуется добавить правило соответствия для элемента **opis**, который является дочерним элементом для элемента **tovars**. Также потребуется указать теги заголовка первого уровня **<H1>** и обозначить применение правил соответствия **<xsl:apply-templates>**. Вид полученного XSL-файла показан на рис. 6    86    Рис.6    Для контроля можно просмотреть XML-документ в браузере. Должно отобразиться информационное содержимое элемента opis. Вид документа показан на рис. 7    87    Рис. 7    Следующим этапом будет формирование таблицы. Для этого потребуется указать правило, которое будет отбирать элементы tovar и выводить информационное содержимое подчиненных элементов **naim** и **price**. Это правило будет выглядеть следующим образом:    <xsl:template  match="tovar">   <tr>    <td><xsl:value-of  select="naim"/></td>    <td align="right"><xsl:value-of  select="price"/></td>   </tr>  </xsl:template>    Затем потребуется после вывода содержимого элемента **opis** указать применение правил для вывода элементов tovar. Для этого применяется элемент **<xsl:apply-templates select="tovar"/>**    В результате этих изменений XSL-файл приобретает вид, показанный на рис. 8    88    Рис. 8    При просмотре XML-документа в броузере, информационное содержимое элементов отражается, но нет заголовка таблицы и табличного оформления (рис. 9).    89    Рис. 9    Для применения табличного оформления требуется возле элемента **<xsl:apply-templates select="tovar"/>**указать заголовок таблицы и табличные тэги (<table>). После проведения указанных изменений, вид XSL-файла показан на рис. 10    90    Рис. 10    В результате изменений вид XML-документа в броузере изменится – появится табличное оформление (рис. 11)    91    Рис. 11    Вид документа почти соответствует заданию, но требуется выполнить сортировку данных таблицы по цене. Для этого требуется применить элемент **<xsl:sort>**Этот элемент будет применен внутри элемента **<xsl:apply-templates select="tovar"/>** В результате изменений XSL-файл примет вид, показанный на рис. 12    92    Рис. 12    После проведения всех изменений, вид документа в броузере соответствует заданию (рис. 13)    93    Рис. 13 |

|  |
| --- |
| Задание 16 |

|  |
| --- |
| **Формирование XML документа на основе файла Excel**    Файл Excel - [**прайс\_лист.xls**](https://sdo.niu.ranepa.ru/close/store/books/%7B45FB1951-3E99-49FD-9C54-AFAE74E8CD4E%7D/price_list.xls) и документ XML – [**price\_list.xml**](https://sdo.niu.ranepa.ru/close/store/books/%7B45FB1951-3E99-49FD-9C54-AFAE74E8CD4E%7D/price_list.xml) прилагаются к Практической работе.    **Задание**    Информация по товарам фирмы экспортирована из программы 1С в виде файла Excel. Требуется сформировать XML документ со структурой, показанной на рис. 1    94    Рис. 1    Вид файла Excel показан на рис. 2    95    Рис. 2    **Анализ файла Excel**    В файле представлена таблица с указанием наименований товаров, отпускной цены, количества товаров и единиц измерения. Также в таблице указаны категории товаров (выделены жирным шрифтом). Из всей информации в указанной таблице, для формирования XML документа потребуется наименование товаров (колонка **A**), единица измерения (колонка **C**), отпускная цена (колонка **D**). Наименование категории товара расположено также в колонке **A**, поэтому потребуется определить, какая именно информация расположена в колонке **A** – наименование товара или категория. Для этого можно проверить колонку **D** - если она заполнена, то в строке находится информация о товаре, если нет – то о категории. При этом можно сделать предположение, что цена не может быть пустой.    Согласно структуре документа, показанной на рис. 1, у элементов category и tovar есть атрибуты **id**. Для элемента **tovar** в этом атрибуте сохраняется уникальный номер товара, а для элемента **category** – наименование категории.    **Формирование текста XML-документа**    Формирование текста документа можно начать с заполнения имен элементов. Это можно сделать в любых незаполненных ячейках. Например, в ячейках **A8, A9, B8, B9, C9, D9**. Для наглядности эти ячейки «залиты» желтым цветом. Вид таблицы показан на рис. 3    96    Рис. 3    Текст XML документа будет сформирован в колонке **G**. Формирование текста можно начать с анализа строки – требуется определить, какая информация в ней расположена – о товаре или о категории. Если в строке расположена информация о товаре, то будет сформирован элемент tovar (начальный тег, атрибут id, дочерние элементы, конечный теги), если информация о категории, то будет сформирован конечный тег (для предыдущей категории), начальный тег для текущей категории, атрибут id c наименованием категории. Перейдя на язык XML, можно сказать, что в первом случае будут сформированы теги <tovar></tovar>, а во втором случае – теги </category><category>. Так как имена элементов записаны в ячейках A9 и A8, можно в первом случае записать такую формулу: **=”</” & A8 & “><” & A8 & “>”** а во втором случае **=”<” & A9 & “></” & A9 & “>”**Для проверки заполнения колонки С для определения товар это или категория можно использовать функцию **ЕПУСТО()**, а для проверки условия – функцию **ЕСЛИ**. Таким образом, предварительная формула в ячейке G12 будет иметь вид:    **=ЕСЛИ(ЕПУСТО(D12); "</" & A8 & "><" & A8 & ">";"<" & A9 & "></" & A9 & ">")**    Для «размножения этой формулы на все ячейки таблицы, можно ее просто скопировать. Но при этом изменятся адреса ячеек с наименованием элементов. Поэтому для их обозначения лучше применить абсолютную адресацию (со знаком **$**). Формула примет следующий вид:    **=ЕСЛИ(ЕПУСТО(D12); "</" & $A$8 & "><" & $A$8 & ">";"<" & $A$9 & "></" & $A$9 & ">")**    Для проверки результата применения формулы можно скопировать ее в несколько нижестоящих ячеек в той же колонке. Для наглядности, на рис. 4 эти ячейки выделены красным шрифтом.    97    Рис. 4    Следующим шагом потребуется создать подчиненные элементы для элемента tovar. Можно воспользоваться теми же приемами – наименования элементов брать из ячеек B9, C9, D9 c абсолютной адресацией, а информационное содержание – из соответствующей строки таблицы (адресация относительная). Формула примет следующий вид (изменения выделены красным цветом.    **=ЕСЛИ(ЕПУСТО(D12);"</"&$A$8&"><"&$A$8&">";"<"&$A$9&">" & "<" & $B$9 & ">" & A12 & "</"&$B$9&">" & "<" & $C$9 & ">" & C12 & "</"& $C$9 &">" & "<" & $D$9 & ">" & D12 & "</"&$D$9&">"&"</"&$A$9&">")**    Результат применения формулы показан на рис. 5    98    Рис. 5    Следующим шагом будет обработка атрибутов. В структуре XML-документа показаны 2 атрибута, имена у них одинаковы – **id**. Имя атрибута сохранено в ячейке **B8**. Значение атрибута **id** у элемента **category** будет равно имени категории и расположено в колонке **A**, значением атрибута **id** у элемента **tovar** будет уникальное число. Для формирования уникального числа можно воспользоваться функцией **СТРОКА()**, которая возвращает номер текущей строки. Формула примет следующий вид (изменения выделены красным цветом):    **=ЕСЛИ(ЕПУСТО(D12);"</"&$A$8&"><"&$A$8 & " " & $B$8 & "=""" & A12 &""">";"<"&$A$9 & " " & $B$8 & "=""" & СТРОКА() &""">" & "<" & $B$9 & ">" & A12 & "</"&$B$9&">" & "<" & $C$9 & ">" & C12 & "</"& $C$9 &">" & "<" & $D$9 & ">" & D12 & "</"&$D$9&">"&"</"&$A$9&">")**    Результат применения формулы показан на рис. 6    99    Рис. 6    На следующем шаге потребуется заменить наименовании товара и наименовании категории символы, которые не могут применяться в информационной части XML-документа на соответствующие сущности. Эти символы и сущности будут сохранены в ячейках G7…K8. Для замены будет использована функция **ПОДСТАВИТЬ** и абсолютная адресация ячеек. Формула примет следующий вид:    **=ЕСЛИ(ЕПУСТО(D12);"</"&$A$8&"><"&$A$8 & " " & $B$8 & "=""" & ПОДСТАВИТЬ(ПОДСТАВИТЬ(ПОДСТАВИТЬ(ПОДСТАВИТЬ(ПОДСТАВИТЬ(A12;$G$7;$G$8);$H$7;$H$8);$I$7;$I$8);$J$7;$J$8);$K$7;$K$8) &""">";"<"&$A$9 & " " & $B$8 & "=""" & СТРОКА() &""">" & "<" & $B$9 & ">" & ПОДСТАВИТЬ(ПОДСТАВИТЬ(ПОДСТАВИТЬ(ПОДСТАВИТЬ(ПОДСТАВИТЬ(A12;$G$7;$G$8);$H$7;$H$8);$I$7;$I$8);$J$7;$J$8);$K$7;$K$8) & "</"&$B$9&">" & "<" & $C$9 & ">" & C12 & "</"& $C$9 &">" & "<" & $D$9 & ">" & D12 & "</"&$D$9&">"&"</"&$A$9&">")**    Формулу требуется скопировать во все ячейки строк с информацией.    Результат применения формулы показан на рис. 7    100    Рис. 7    На последнем шаге потребуется создать шаблон XML-документа в редакторе XMLPad с указанием начальных и конечных тегов для элементов **price\_list**, **tovars**, конечного тэга для элемента **category** и заголовка документа. Шаблон будет иметь следующий вид:    <?xml version="1.0" encoding="windows-1251"?>  <price\_list>  <tovars>    </category>  </tovars>  </price\_list>    Последним шагом в создании документа будет перенос основной части документа из файла Excel методом копирования и вставки. После этого потребуется убрать «лишний» конечный тег для первого элемента category.    Вид документа в редакторе XMLPad показан на рис. 8    101    Рис. 8    Вид документа в броузере Internet Explorer показан на рис. 9    102    Рис. 9 |

|  |
| --- |
| Задание 17 |

|  |
| --- |
| **Формирование XML документа на основе выборки (запроса) из базы данных MySQL**    **Задание**    Требуется на используя выборку (запроса) из базы данных MySQL создать файл в Excel, который будет являться основой для документа XML. Структура XML-документа показана на рис. 1    103    Рис. 1    **Анализ структуры документа**    Информационная часть документа состоит из атрибута **naim** для элемента **category** и дочернего элемента **tovar** c атрибутом **id**, а также дочерних для элемента **tovar** элементов **naim**, **price**, **edizm**. Можно сделать вывод, что потребуется 2 SQL предложения с вертикальным объединением через UNION. Первое SQL предложение должно делать выборку из таблицы-источника для получения наименования категории товара, второе SQL предложение делает выборку из таблицы для получения наименования, цены и единицы измерения товара. Значением атрибута id элемента tovar будет являться номер строки листа в Excel. SQL-предложения будут содержать 2 колонки – первая колонка служит для упорядочивания, вторая – содержит содержимое html-документа, который будет являться основой для файла Excel.    **Структура таблицы-источника данных**    В рассматриваемом примере данные о товарах и категориях хранятся в таблице tovars базы данных test. Для подключения к серверу и базе данных и выполнения SQL-предложения используется поставляемая с сервером mySQL программа mysql, хотя можно воспользоваться любой другой программой, например Mysql-Front.    Для подключения к серверу требуется знать имя пользователя и пароль. Пользователь должен иметь право выборки из таблицы, в которой хранится информация о товаре.    Для подключения к серверу требуется выполнить следующую команду операционной системы:    **mysql –uИМЯ\_ПОЛЬЗОВАТЕЛЯ –pПАРОЛЬ**    После успешного подключения требуется выбрать базу данных. Для этого используется команда mysql **use ИМЯ\_БАЗЫ**.    После выбора базы можно просмотреть структуру таблицы. tovar. Для этого можно использовать команду desc ИМЯ\_ТАБЛИЦЫ.    Пример использования команд показан на рис. 2    104    Рис. 2    Проанализировав структуру таблицы видно, что название товара или категории хранятся в колонке naim, единица измерения – в колонке edizm, цена – в колонке price. Каждая строка таблицы характеризуется уникальным номером, который хранится в колонке id, а id категории для товаров хранится в колонке id\_par. Для категории значение в этой колонке =0. Для контроля можно сделать выборку из этой таблицы при помощи простого SQL-предложения. Результат выборки показан на рис. 3    105    Рис. 3    Из полученной выборки видно, что в таблице есть 3 категории товара и 14 видов товара.    Программа mysql некорректно отображает русские буквы, поэтому для анализа результатов выборки можно сохранять их в текстовый файл, который можно просматривать в текстовом редакторе. Для сохранения результатов в текстовый файл требуется использовать команду mysql **\T ИМЯ\_ФАЙЛА**.    SQL –предложение для выборки категорий товара будет следующим:    select id\*1000, concat('<tr><td>', id\_par,'</td><td colspan=3>', naim, '</td></tr>' ) as tbl from tovars  where id\_par=0    SQL-предложение для товаров:    select t.id\_par\*1000+t.id, concat('<tr><td>', t.id\_par,'</td><td>', t.naim, '</td><td>',t.edizm,'</td><td>',t.price, '</td></tr>' ) as tbl  from tovars t  inner join tovars tp  on t.id\_par = tp.id  where tp.id\_par=0    При вертикальном объединении этих предложений с использованием UNION и добавления 2-х SQL-предложений для получения заголовка и «подвала» html-файла с указанием сортировки,  итоговое SQL-предложение будет следующим:    select distinct 0 as id,'<html><body><table>' as tbl  from tovars  union  select id\*1000, concat('<tr><td>', id\_par,'</td><td colspan=3>', naim, '</td></tr>' ) as tbl from tovars  where id\_par=0  union  select t.id\_par\*1000+t.id, concat('<tr><td>', t.id\_par,'</td><td>', t.naim, '</td><td>',t.edizm,'</td><td>',t.price, '</td></tr>' ) as tbl  from tovars t  inner join tovars tp  on t.id\_par = tp.id  where tp.id\_par=0  union  select distinct 9999,'</table></body></html>' as tbl  from tovars  order by 1    Чтобы не вводить это SQL-предложение в программу mysql вручную можно сохранить его в текстовом файле и потом вызвать этот текстовый файл из программы mysql. Для вызова файла выполняется следующая команда mysql - **source ИМЯ\_ФАЙЛА**.    Предварительно требуется вызвать команду для вывода результата выборки в файл. Вид полученного файла показан на рис. 4    106    Рис. 4    Следующим шагом будет «вырезка» из файла текста второй колонки и формирование на ее основе html-файла. «Вырезку» можно сделать при помощи Excel. Требуется открыть текстовый файл в Excel и выполнить пункт «Текст по столбцам» в меню «Данные». В качестве разделителя указать символ |. Вид окна мастера разбора текста показан на рис. 5    107    Рис. 5    Вид книги Excel после завершения работы Мастера показан на рис. 6    108    Рис. 6    Выделив и скопировав текст из ячеек С5..С23, в текстовом редакторе создается html-файл путем вставки скопированного текста. Файл сохраняется и открывается в Excel. Вид файла показан на рис. 7    109    Рис. 7    Используя данные этого файл и методику, описанную в [уроке 16](https://sdo.niu.ranepa.ru/close/store/books/%7B45FB1951-3E99-49FD-9C54-AFAE74E8CD4E%7D/z16.htm) можно создать XML-документ. |