Технология компонентного программирования

Лабораторная работа № 2

Знакомство с XML-файлом

Цель занятия – знакомство с основными методами создания и чтения XML-файла в среде Delphi.

Немного о XML-файле. Он похож на HTML-файл, и изначально был придуман так же для Интернета – для обмена информацией между страницами. В нём так же используются тэги, но в отличие от HTML, они могут называться как угодно – это как поля в таблицах баз данных. И очень скоро XML-файлы стали применяться для обмена записями между базами данных. И сейчас это довольно распространённый способ обмена данными.

В XML-файле могут использоваться 2 способа записи информации: через элементы, через атрибуты.

Эту структуру в нашем примере можно сравнить с таблицей БД. Это простой пример. Имеется как бы таблица с именем INFOLIST. В ней записи PERSON, каждая состоит из полей FAM, IM, OT.

Рассмотрим 1-й вариант. INFOLIST – корневой элемент (узел – Node), он имеет дочерние элементы PERSON, в которых, кроме своих дочерних элементов (FAM, IM, OT), больше ничего нет. Элементы FAM, IM, OT имеют значения, дочерних элементов они уже не имеют. Всё это показывает следующий пример.

```
<?xml version="1.0" encoding="Windows-1251"?>
<INFOLIST>
   <PERSON>
      <FAM>Макарова</FAM>
      <IM>Ольга</IM>
      <OT>Владимировна</OT>
   </PERSON>
   <PERSON>
      <FAM>Петрова</FAM>
      <IM>Вера</IM>
      <OT>Павловна</OT>
   </PERSON>
</INFOLIST>
```

2-й вариант. Продолжим сравнивать XML с таблицей базы данных. Запись PERSON состоит из полей – атрибутов. Они имеют названия и значения. Элемент PERSON не имеет дочерних элементов и не имеет своего значения, но теперь состоит из атрибутов FAM, IM, OT со своими значениями. Оформляется это следующим образом:

```
<?xml version="1.0" encoding="Windows-1251"?>
<INFOLIST>
   <PERSON FAM="Макарова" IM="Ольга" OT="Владимировна"/>
   <PERSON FAM="Петрова" IM="Вера" OT="Павловна"/>
</INFOLIST>
```

В обоих случаях информация одна и та же, но записана по-разному. Это разные схемы XML-файла. И когда вы получаете новый XML-файл, надо знать его схему, чтобы правильно прочитать данные из файла.

XML получил широкое распространение из-за удобства передачи базы данных одним файлом. Дело в том, что можно составить схему так, что данных из нескольких связанных таблиц могут быть представлены в виде соответствующих узлов и веток в XML-файле.

Например, пусть информация выше – это родители (например, список матерей). У них могут быть дети. Тогда в элемент PERSON можно добавить элемент CHILD со своими атрибутами, причём можно добавлять сколько угодно этих элементов с тем же названием CHILD. Причём этого элемента может и не быть (нет ещё детей у женщины).

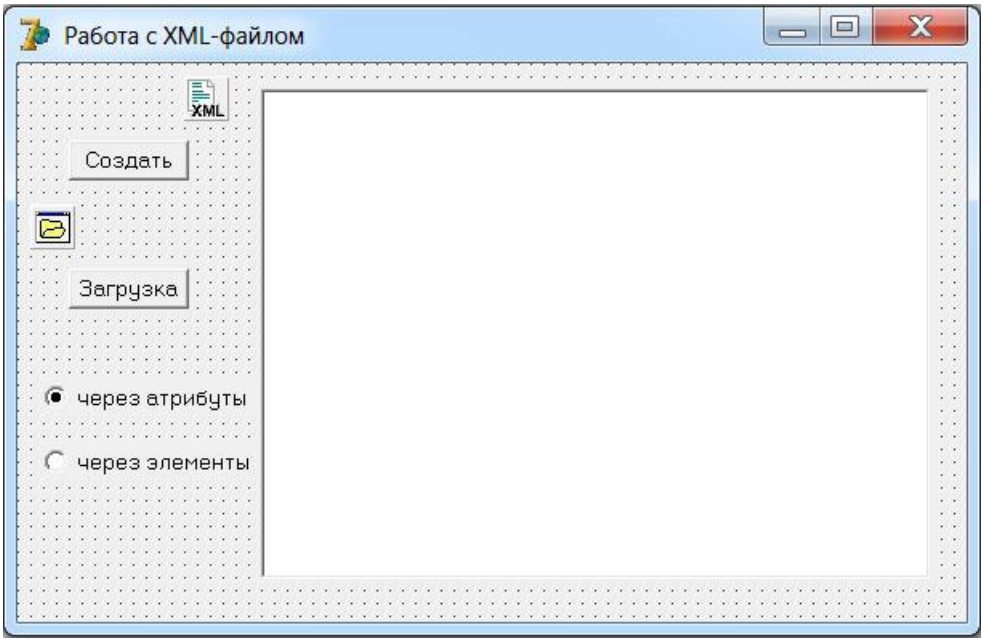
Пример:

```
<?xml version="1.0" encoding="Windows-1251"?>
<INFOLIST>
   <PERSON>
      <FAM>Макарова</FAM>
      <IM>Ольга</IM>
      <OT>Владимировна</OT>
      <CHILD FAM="Макарова" IM="Татьяна" OT="Даниловна" DR="01.02.2022"/>
      <CHILD FAM="Макаров" IM="Александр" OT="Данилович" DR="02.12.2020"/>
   </PERSON>
   <PERSON>
      <FAM>Петрова</FAM>
      <IM>Вера</IM>
      <OT>Павловна</OT>
   </PERSON>
</INFOLIST>
```

Т.е. можно создать и такую структуру, где часть информации представлена элементами, а часть (ветка CHILD) – атрибутами.

В задаче создаётся XML-файл, затем загружается для проверки. Изучим оба варианта XML-структуры: с атрибутами и с элементами. Сначала тестируется один вариант, затем – другой вариант. То есть, учимся правильно создавать и затем его пытаемся прочитать.

Подготовим форму следующего вида:



Кнопка «Создать» – для создания XML-файла, кнопка «Загрузить» – показать полученный результат для проверки и в этой части, естественно, учимся читать XML-файл. Работать с XML-файлом будем через компонент XMLDocument, который находится на

вкладке Internet. После того, как поместите этот компонент на форму, в наш модуль подключатся следующие модули: xmldom, XMLIntf, msxmldom, XMLDoc, которые и позволят нам работать с XML-форматом. Для просмотра результата после загрузки XML-файла используем компонент Memo.

Пусть главное дерево называется INFOLIST, а записи в нём – PERSON. Атрибуты этой записи (то же, что и поля в понятиях баз данных): FAM, IM, OT. Сделаем пока совсем простой пример из 2-3 таких записей и сохраним файл под каким-нибудь именем, например, Primer.xml.

Создание файла:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  Node1,Node2,Node3: IXMLNode;

  procedure AddPersonAttr(Fam,Im,Ot:string);
  begin
    Node2:= Node1.AddChild('PERSON');
    Node2.SetAttribute('FAM',Fam);
    Node2.SetAttribute('IM',Im);
    Node2.SetAttribute('OT',Ot);
  end;

  procedure AddPersonElem(Fam,Im,Ot:string);
  begin
    Node2:= Node1.AddChild('PERSON');
    Node3:= Node2.AddChild('FAM');
    Node3.NodeValue:= Fam;
    Node3:= Node2.AddChild('IM');
    Node3.NodeValue:= Im;
    Node3:= Node2.AddChild('OT');
    Node3.NodeValue:= Ot;
  end;

begin
  XMLDocument1.Options:= [doNodeAutoCreate, doNodeAutoIndent, doAttrNull,
      doAutoPrefix, doNamespaceDecl];
  XMLDocument1.NodeIndentStr:= '    ';
  XMLDocument1.Active:= true;
  XMLDocument1.Version:= '1.0';
  XMLDocument1.Encoding:= 'Windows-1251';
  Node1:= XMLDocument1.AddChild('INFOLIST');
  if RadioButton1.Checked then begin
    AddPersonAttr('Петров', 'Иван', 'Ильич');
    AddPersonAttr('Иванов', 'Пётр', 'Семёнович');
  end
  else begin
    AddPersonElem('Петров', 'Иван', 'Ильич');
    AddPersonElem('Иванов', 'Пётр', 'Семёнович');
  end;
  XMLDocument1.SaveToFile('Primer.xml');
  XMLDocument1.Active:= false;
  ShowMessage('OK');
end;
```

Здесь первые 2 строки – это настройки XMLDocument1, которые можно сделать и в Object Inspector.

Загрузка готового XML-файла:

```
procedure TForm1.Button2Click(Sender: TObject);
var
  XMLFile: string;
// сюда вставить процедуры ShowFileAttr и ShowFileElem

begin
        // Load xml-file
  if not OpenDialog1.Execute then exit;
  XMLFile:= OpenDialog1.FileName;
  XMLDocument1.LoadFromFile(XMLFile);
  XMLDocument1.Active:= true;
  if RadioButton1.Checked then
    ShowFileAttr(XMLDocument1.ChildNodes)
  else
    ShowFileElem(XMLDocument1.ChildNodes);
  XMLDocument1.XML.Clear;
  XMLDocument1.Active:= false;
end;
```

Для каждого варианта здесь вызывается своя внутренняя процедура: `ShowFileAttr` или `ShowFileElem`.

Их обе необходимо вставить в процедуру `Button2Click`.

```
procedure ShowFileAttr(XMLNodeList:IXMLNodeList);
var
  v:variant;
  s,w:string;
  i,N:integer;
  BufXMLNodeList:IXMLNodeList;    // ветка для <INFOLIST>
begin
  if XMLNodeList.Count <> 2 then exit;
  s:= AnsiUpperCase(XMLNodeList[1].NodeName);  // [0..1]:  xml, INFOLIST
  if s <> 'INFOLIST' then exit;  // что-то не то (не тот файл?)
  if not XMLNodeList[1].HasChildNodes then exit;  // ???
  Memo1.Clear;
  BufXMLNodeList:= XMLNodeList[1].ChildNodes;
  N:= BufXMLNodeList.Count;
  for i:= 0 to N - 1 do begin
    s:= AnsiUpperCase(BufXMLNodeList[i].NodeName);
    w:= '';
    if s = 'PERSON' then begin
      v:= BufXMLNodeList[i].Attributes['FAM'];
      if v = Null then w:='FAM??? ' else w:= VarToStr(v) + ' ';
      v:= BufXMLNodeList[i].Attributes['IM'];
      if v = Null then w:= w + 'IM??? ' else w:= w + VarToStr(v) + ' ';
      v:= BufXMLNodeList[i].Attributes['OT'];
      if v = Null then w:= w + 'OT??? ' else w:= w + VarToStr(v) + ' ';
      Memo1.Lines.Add(w);
    end
    else Memo1.Lines.Add(s + '!?');
  end;
end;
```

```
procedure ShowFileElem(XMLNodeList:IXMLNodeList);
var
  v:variant;
  s,w, Fam, Im, Ot:string;
  i,k,N:integer;
  BufXMLNodeList,    // ветка для <INFOLIST>
  PersXMLNodeList:IXMLNodeList; // ветка для <PERSON>
begin
  if XMLNodeList.Count <> 2 then exit;
  s:= AnsiUpperCase(XMLNodeList[1].NodeName);  // [0..1]:  xml, INFOLIST
  if s <> 'INFOLIST' then exit;  // что-то не то (не тот файл?)
  if not XMLNodeList[1].HasChildNodes then exit;  // ???
  Memo1.Clear;
  BufXMLNodeList:= XMLNodeList[1].ChildNodes;
  N:= BufXMLNodeList.Count;
  for i:= 0 to N - 1 do begin
    s:= AnsiUpperCase(BufXMLNodeList[i].NodeName);
    w:= '';
    if s = 'PERSON' then begin
      PersXMLNodeList:= BufXMLNodeList[i].ChildNodes;
      for k:= 0 to PersXMLNodeList.Count - 1 do begin  // перебор всех полей
        s:= AnsiUpperCase(PersXMLNodeList[k].NodeName);
        v:= PersXMLNodeList[k].NodeValue;
        if s = 'FAM' then begin
          if v = Null then Fam:='FAM???' else Fam:= VarToStr(v);
        end
        else if s = 'IM' then begin
          if v = Null then Fam:='IM???' else Im:= VarToStr(v);
        end
        else if s = 'OT' then begin
          if v = Null then Ot:='OT???' else Ot:= VarToStr(v);
        end;
        w:= Fam + ' ' + Im + ' ' + Ot;
      end;
      Memo1.Lines.Add(w);
    end
    else Memo1.Lines.Add(s + '!?');
  end;
end;
```