

УДК 004.056.5:004.052

Н.Н. Безуглый, доцент, канд. техн. наук

Ю.Л. Явкун, ст. преподаватель

Севастопольский национальный технический университет

ул. Университетская 33, г. Севастополь, Украина, 99053

E-mail: kvt@sevgtu.sebastopol.ua

АВТОМАТИЗАЦИЯ ФОРМИРОВАНИЯ ФОРМАЛИЗОВАННЫХ XML ДОКУМЕНТОВ В СРЕДЕ VISUAL C#

Рассматриваются вопросы автоматизации разработки формализованных документов, включающие формирование бланка с помощью текстового редактора Word, сохраняемого с расширением (.xml), создание проекта в среде VS C#, шаблона формализованного документа, содержащего постоянную информацию и окна для ввода переменной информации и заполнения бланка.*

Ключевые слова: *текстовый редактор Word, XML документ, Templates (Шаблоны), формализованный документ, Visual Studio C#, пространство имен.*

Расширяемый язык разметки XML за последние несколько лет привлёк к себе огромное внимание разработчиков для создания широкого круга приложений: от описания конфигурации приложений до передачи информации между Web-службами. При разработке АРМов операторов должностных лиц органов управления для АСУ управления динамическими объектами возникает задача создания системы формализованных документов, которая могла бы удовлетворять следующим требованиям:

- оперативности и простоты создания формализованных документов;
- применения оператором широко известного текстового редактора Word для создания бланка формализованного XML документа, его просмотра и распечатки;
- возможности формирования на основе бланка формализованного XML документа, который содержит постоянную информацию бланка и необходимую переменную информацию, которая задается оператором;
- возможности оперативной обработки формализованных XML документов, включая быстрый поиск необходимых объектов, элементов, атрибутов и манипулирование ими;
- удобства передачи документа по NET сетям;
- поиска и переноса информации об объектах из XML документа в реляционную базу данных и наоборот.

Целью статьи является разработка технологии создания XML-шаблона, формирования нового документа путём манипулирования и ввода в него переменной информации, а также его записи и передачи по каналам связи.

Рассмотрим только технологию формирования XML документов в среде Visual Studio C#.

Пусть нам задан бланковый документ следующего вида (рисунок 1):

Бланк (образец)	
Документ №	
Фамилия	
Имя	
Отчество	

Рисунок 1 – Вид формализованного бланка

Данный документ содержит постоянную и переменную информацию.

Наберем постоянную информацию данного бланкового документа в текстовом редакторе Word с расширением (*.xml) и получим документ с именем Документ.xml, с которым будем дальше работать.

В VS C# создадим проект, выбирая в панели Templates (Шаблоны) Windows Forms Application. На форме проекта создадим шаблон формализованного документа содержащего постоянную информацию бланка и окна для ввода переменной информации с целью заполнения бланка. Вид формализованного документа на форме показан на рисунке 2:

Рисунок 2 – Вид формализованного документа на форме

Для манипуляції з XML документом перепишемо його в директорії форми і приведемо його текст:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?mso-application progid="Word.Document"?>
<w:wordDocument xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml"
  xmlns:v="urn:schemas-microsoft-com:vml"
  xmlns:w10="urn:schemas-microsoft-com:office:word"
  xmlns:sl="http://schemas.microsoft.com/schemaLibrary/2003/core"
  xmlns:aml="http://schemas.microsoft.com/aml/2001/core"
  xmlns:wx="http://schemas.microsoft.com/office/word/2003/auxHint"
  xmlns:o="urn:schemas-microsoft-com:office:office"
  xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882"
  w:macrosPresent="no" w:embeddedObjPresent="no" w:ocxPresent="no" xml:space="preserve">
  <o:DocumentProperties><o:Title>Документ №</o:Title><o:Author>Безуглый</o:Author>
  <o:LastAuthor>Безуглый</o:LastAuthor><o:Revision>1</o:Revision>
  <o:TotalTime>1</o:TotalTime>
  <o:Created>2010-02-02T08:07:00Z</o:Created><o:LastSaved>
  2010-02-02T08:08:00Z</o:LastSaved><o:Pages>1</o:Pages>
  <o:Words>4</o:Words><o:Characters>29</o:Characters><o:Lines>1</o:Lines>
  <o:Paragraphs>1</o:Paragraphs>
  <o:CharactersWithSpaces>32</o:CharactersWithSpaces><o:Version>11.5604</o:Version>
  </o:DocumentProperties>
  <w:fonts><w:defaultFonts w:ascii="Times New Roman" w:fareast="Times New Roman"
  w:h-ansi="Times New Roman" w:cs="Times New Roman"/></w:fonts>
  <w:styles><w:versionOfBuiltInStylenames w:val="4"/><w:latentStyles
  w:defLockedState="off" w:latentStyleCount="156"/><w:style w:type="paragraph"
  w:default="on" w:styleId="a">
  <w:name w:val="Normal"/><wx:uiName wx:val="Обычный"/>
  <w:rsid w:val="004F0A3B"/><w:rPr><wx:font wx:val="Times New Roman"/>
  <w:sz w:val="24"/><w:sz-cs w:val="24"/>
  <w:lang w:val="DE" w:fareast="RU" w:bidi="AR-SA"/></w:rPr></w:style><w:style
  w:type="character" w:default="on" w:styleId="a0">
  <w:name w:val="Default Paragraph Font"/><wx:uiName wx:val="Основной шрифт
  абзаца"/><w:semiHidden/></w:style>
  <w:style w:type="table" w:default="on" w:styleId="a1"><w:name w:val="Normal
  Table"/><wx:uiName wx:val="Обычная таблица"/><w:semiHidden/><w:rPr>
  <wx:font wx:val="Times New Roman"/></w:rPr><w:tblPr><w:tblInd
  w:w="0" w:type="dxa"/><w:tblCellMar><w:top w:w="0" w:type="dxa"/>
  <w:left w:w="108" w:type="dxa"/><w:bottom w:w="0" w:type="dxa"/>
  <w:right w:w="108" w:type="dxa"/></w:tblCellMar></w:tblPr></w:style>
```

```

<w:style w:type="list" w:default="on" w:styleId="a2"><w:name w:val="No List"/>
<wx:uiName wx:val="Hem cнucka"/>
<w:semiHidden/></w:style></w:styles><w:docPr><w:view w:val="print"/>
<w:zoom w:percent="100"/><w:dontDisplayPageBoundaries/>
<w:doNotEmbedSystemFonts/><w:proofState w:spelling="clean"
    w:grammar="clean"/><w:attachedTemplate
w:val=""/><w:defaultTabStop w:val="708"/><w:punctuationKerning/><w:characterSpacingControl
w:val="DontCompress"/><w:optimizeForBrowser/>
<w:validateAgainstSchema/><w:saveInvalidXML w:val="off"/><w:ignoreMixedContent
    w:val="off"/><w:alwaysShowPlaceholderText w:val="off"/><w:compat>
<w:breakWrappedTables/><w:snapToGridInCell/><w:wrapTextWithPunct/>
<w:useAsianBreakRules/><w:dontGrowAutofit/></w:compat></w:docPr><w:body><wx:sect><w:p><w:pPr>
<w:jc w:val="center"/><w:rPr><w:sz w:val="28"/><w:sz-cs w:val="28"/>
<w:lang w:val="RU"/></w:rPr></w:pPr><w:r><w:rPr><w:sz w:val="28"/>
<w:sz-cs w:val="28"/><w:lang w:val="RU"/></w:rPr><w:t>Документ №</w:t>
</w:r></w:p><w:p><w:pPr><w:ind w:first-line="708"/><w:rPr><w:sz w:val="28"/>
<w:sz-cs w:val="28"/><w:lang w:val="RU"/></w:rPr></w:pPr><w:r><w:rPr>
<w:sz w:val="28"/><w:sz-cs w:val="28"/>
<w:lang w:val="RU"/></w:rPr><w:t>Фамилия</w:t>
</w:r></w:p><w:p><w:pPr><w:ind w:first-line="708"/><w:rPr>
<w:sz w:val="28"/><w:sz-cs w:val="28"/><w:lang w:val="RU"/></w:rPr></w:pPr><w:r>
<w:rPr><w:sz
                                w:val="28"/><w:sz-cs
                                w:val="28"/><w:lang
w:val="RU"/></w:rPr><w:t>Имя</w:t></w:r></w:p><w:p>
<w:pPr><w:ind w:first-line="708"/><w:rPr><w:sz w:val="28"/><w:sz-cs w:val="28"/>
<w:lang w:val="RU"/></w:rPr></w:pPr><w:r><w:rPr><w:sz w:val="28"/>
<w:sz-cs w:val="28"/><w:lang w:val="RU"/></w:rPr><w:t>Омечество</w:t>
</w:r></w:p><w:p><w:sectPr><w:pgSz w:w="11906" w:h="16838"/>
<w:pgMar w:top="1134" w:right="850" w:bottom="1134" w:left="1701" w:header="708"
    w:footer="708" w:gutter="0"/><w:cols w:space="708"/><w:docGrid w:line-
pitch="360"/></w:sectPr></wx:sect></w:body></w:wordDocument>

```

Для обеспечения манипулирования данными документа будем использовать классы из пространства имен *System.Xml* и *System.Xml.Path*.

Напишем код, используя C#, для формирования формализованного XML документа. Для этого введем два приватных поля для хранения документа и текущего узла, а также инициализируем их в конструкторе:

```

public partial class Form1 : Form

private XmlDocument mDocument;

private XmlNode mCurrentNode;
public Form1()
{
    InitializeComponent();
    mDocument = new XmlDocument();
    mDocument.Load(@"Документ.xml");
    mCurrentNode = mDocument.DocumentElement;
}
}

```

Найдем все текстовые элементы в XML документе и определим к какому пространству имен они относятся. Все они в тексте формализованного документа выделены жирным шрифтом. Это элементы **<w:t>Документ №</w:t>**, **<w:t>Фамилия</w:t>**, **<w:t>Имя</w:t>**, **<w:t>Омечество</w:t>** имеют префикс "w". Нужно идентифицировать префикс "w" с определенным пространством имен. Для этого используем атрибут *xmlns:prefix* внутри XML документа. Нам нужно найти *xmlns:w*. По атрибуту *xmlns:w* находим код, который и идентифицирует пространство имен. Обратившись к рисунку 2 находим, что префикс "w" описан схемой *xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml"*.

Для применения XPath запросов с целью поиска элементов в XML документе содержащих префикс расширим пространство имен написав в C# следующий код, который позволит нам осуществлять корректную навигацию по XML документу.

```
XmlNamespaceManager mgr = new XmlNamespaceManager(mDocument.NameTable);  
mgr.AddNamespace("w", "http://schemas.microsoft.com/office/word/2003/wordml");
```

Используя метод выбора узлов *SelectSingleNode* и XPath запросы для XML документа, напомним код в C# для каждого элемента документа.

```
mCurrentNode = mDocument.DocumentElement.SelectSingleNode("//w:r[w:t=  
    'Документ №']", mgr);  
proba = " Документ №" + textBox1.Text;  
mCurrentNode.LastChild.InnerText = proba;  
  
mCurrentNode = mDocument.DocumentElement.SelectSingleNode("//w:r[w:t=  
    'Фамилия']", mgr);  
proba = " Фамилия" + textBox2.Text;  
mCurrentNode.LastChild.InnerText = proba;  
  
mCurrentNode = mDocument.DocumentElement.SelectSingleNode("//w:r[w:t='Имя']", mgr);  
proba = " Имя" + textBox3.Text;  
mCurrentNode.LastChild.InnerText = proba;  
  
mCurrentNode = mDocument.DocumentElement.SelectSingleNode("//w:r[w:t=  
    'Отчество']", mgr);  
proba = " Отчество" + textBox4.Text;  
mCurrentNode.LastChild.InnerText = proba;
```

Откомпилируем набранный в C# код и введем переменную информацию в формализованный документ, как показано на рисунке 3:

Рисунок 3 – Ввод информации в формализованный документ

При нажатии кнопки "Сформировать" осуществляется формирование нового документа путем добавления в XML документ переменной информации с дальнейшей записью сформированного документа как нового.

Запись сформированного документа осуществляется следующим кодом:

```
mDocument.Save(@"Документ1.xml").
```

При нажатии кнопки "Распечатать" на принтер выдается сформированный документ (рисунок 4):

Документ №21/8	
Фамилия	Безуглый
Имя	Николай
Отчество	Николаевич

Рисунок 4 – Вид сформированного формализованного документа

При нажатии кнопки "Передать" осуществляется передача формализованного донесения адресату указанному в коде программы.

В результате работы программы получаем новый XML документ "Документ1.xml", содержащий постоянную и переменную информацию, который можно передать по каналам связи и распечатать.

Таким образом, разработана технология формирования формализованных документов с применением современных информационных технологий, позволяющих очень легко и оперативно создавать, обрабатывать и передавать XML документы.

Задачами дальнейшего исследования в этом направлении могут быть: разработка технологии передачи формализованных документов по каналам связи, записи информации в базу данных и считывания из нее.

Библиографический список

1. Фёдоров А.Г. Visual Studio 2008. Краткий обзор ключевых новинок: Монография / А.Г.Фёдоров. — К.: Издательская группа БНУ, 2008. — 175с.

2. Шилдт Г. Полный справочник по C# / Г.Шилдт. — М. : Издательский дом "Вильямс", 2004. — 752 с.

Поступила в редакцию 18.05.2010 г.