

## ЛАБОРАТОРНАЯ РАБОТА №2.

### РАБОТА С ДОКУМЕНТАМИ XML.

**ЦЕЛЬ:** изучить работу с XML-документами в среде NET.

#### Краткие теоретические сведения.

Во-первых, смотреть лекцию 14, посвященную данному вопросу. Надо научиться считывать и создавать xml-документы средствами языка C#. Следующий пример демонстрирует чтение документа xml.

```
using System;
using System.Xml;
namespace ReadXml1
{
    class Class1
    {
        static void Main(string[] args)
        {
            XmlTextReader textReader = new
            XmlTextReader("D:\\my.xml");
            textReader.Read();
            // If the node has value
            while (textReader.Read())
            {
                // Move to first element
                textReader.MoveToElement();
                Console.WriteLine("XmlTextReader Properties Test");
                Console.WriteLine("=====");
                // Read this element's properties and display them on
                console
                Console.WriteLine("Name:" + textReader.Name);
                Console.WriteLine("Base URI:" + textReader.BaseURI);
                Console.WriteLine("Local Name:" +
                textReader.LocalName);
                Console.WriteLine("Attribute Count:" +
                textReader.AttributeCount.ToString());
                Console.WriteLine("Depth:" +
                textReader.Depth.ToString());
                Console.WriteLine("Line Number:" +
                textReader.LineNumber.ToString());
                Console.WriteLine("Node Type:" +
                textReader.NodeType.ToString());
                Console.WriteLine("Attribute Count:" +
                textReader.Value.ToString());
            }
        }
    }
}
```

Ознакомьтесь и выполните данный пример. «Распаковка» тегов документа выполняется в цикле:

```
while (textReader.Read())
{
    ...
}
```

Следующий пример еще более подробный.

```
using System;
using System.Xml;
namespace ReadingXML2
{
    class Class1
    {
        static void Main(string[] args)
        {
            int ws = 0;
            int pi = 0;
            int dc = 0;
            int cc = 0;
            int ac = 0;
            int et = 0;
            int el = 0;
            int xd = 0;
            // Read a document
            XmlTextReader textReader = new
            XmlTextReader("D:\\my.xml");
            // Read until end of file
            while (textReader.Read())
            {
                XmlNodeType nType = textReader.NodeType;
                // If node type us a declaration
                if (nType == XmlNodeType.XmlDeclaration)
                {
                    Console.WriteLine("Declaration:" +
            textReader.Name.ToString());
                    xd = xd + 1;
                }
                // if node type is a comment
                if (nType == XmlNodeType.Comment)
                {
                    Console.WriteLine("Comment:" +
            textReader.Name.ToString());
                    cc = cc + 1;
                }
                // if node type us an attribute
                if (nType == XmlNodeType.Attribute)
                {
                    Console.WriteLine("Attribute:" +
            textReader.Name.ToString());
```

```

        ac = ac + 1;
    }
    // if node type is an element
    if (nType == XmlNodeType.Element)
    {
        Console.WriteLine("Element:" +
textReader.Name.ToString());
        el = el + 1;
    }
    // if node type is an entity\
    if (nType == XmlNodeType.Entity)
    {
        Console.WriteLine("Entity:" +
textReader.Name.ToString());
        et = et + 1;
    }
    // if node type is a Process Instruction
    if (nType == XmlNodeType.Entity)
    {
        Console.WriteLine("Entity:" +
textReader.Name.ToString());
        pi = pi + 1;
    }
    // if node type a document
    if (nType == XmlNodeType.DocumentType)
    {
        Console.WriteLine("Document:" +
textReader.Name.ToString());
        dc = dc + 1;
    }
    // if node type is white space
    if (nType == XmlNodeType.Whitespace)
    {
        Console.WriteLine("WhiteSpace:" +
textReader.Name.ToString());
        ws = ws + 1;
    }
}
// Write the summary
Console.WriteLine("Total Comments:" + cc.ToString());
Console.WriteLine("Total Attributes:" + ac.ToString());
Console.WriteLine("Total Elements:" + el.ToString());
Console.WriteLine("Total Entity:" + et.ToString());
Console.WriteLine("Total Process Instructions:" +
pi.ToString());
Console.WriteLine("Total Declaration:" + xd.ToString());
Console.WriteLine("Total DocumentType:" + dc.ToString());
Console.WriteLine("Total WhiteSpaces:" + ws.ToString());
    }
}
}

```

Создание xml-документа реализует следующий код. Изучите и выполните его.

```

using System;
using System.Xml;
namespace ReadingXML2
{
    class Class1
    {
        static void Main(string[] args)
        {
            // Create a new file in C:\ dir
            XmlTextWriter textWriter = new
XmlTextWriter("C:\\myXmFile.xml", null);
            // Opens the document
            textWriter.WriteStartDocument();
            // Write comments
            textWriter.WriteComment("First Comment XmlTextWriter
Sample Example");
            textWriter.WriteComment("myXmlFile.xml in root dir");
            // Write first element
            textWriter.WriteStartElement("Student");
            textWriter.WriteStartElement("r", "RECORD", "urn:record");
            // Write next element
            textWriter.WriteStartElement("Name", "");
            textWriter.WriteString("Student");
            textWriter.WriteEndElement();
            // Write one more element
            textWriter.WriteStartElement("Address", "");
textWriter.WriteString("Colony");
            textWriter.WriteEndElement();
            // WriteChars
            char[] ch = new char[3];
            ch[0] = 'a';
            ch[1] = 'r';
            ch[2] = 'c';
            textWriter.WriteStartElement("Char");
            textWriter.WriteChars(ch, 0, ch.Length);
            textWriter.WriteEndElement();
            // Ends the document.
            textWriter.WriteEndDocument();
            // close writer
            textWriter.Close();
        }
    }
}

```

Наконец, последний пример показывает. Как превратить файл базы данных в документ XML.

```

using System;
using System.Xml;
using System.Data;
using System.Data.OleDb;
namespace ReadingXML2
{

```

```

class Class1
{
    static void Main(string[] args)
    {
        // create a connection
        OleDbConnection con = new OleDbConnection();
        con.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\\Northwind.mdb";
        // create a data adapter
        OleDbDataAdapter da = new OleDbDataAdapter("Select * from
Customers", con);
        // create a new dataset
        DataSet ds = new DataSet();
        // fill dataset
        da.Fill(ds, "Customers");
        // write dataset contents to an xml file by calling
WriteXml method
        ds.WriteXml("C:\\OutputXML.xml");
    }
}

```

## ВАРИАНТЫ ЗАДАНИЙ

1. Написать программу для создания документа XML
2. Написать программу для поиска в документе XML
3. Написать программу для отображения документа XML в окне браузера
4. Написать программу для добавления в документ XML новых узлов
5. Написать программу, демонстрирующую работу с атрибутами в документе XML
6. Написать программу для перевода XML базу данных ACCESS
7. Написать программу для перевода базы данных ACCESS в XML
8. Написать программу демонстрирующую изменение тегов документа XML
9. Написать программу для удаления и замены узлов XML