

Fișiere XML în Windows Form în C#

Laborator 3

În cadrul acestui laborator vom învăța cum manipulăm fișierele XML în aplicații Windows Forms. Vom crea mici aplicații care să citească, să actualizeze sau să scrie fișiere XML

1. Crearea unui fișier XML în C#

Se dă o lista de studenți. Trebuie creat un document XML care să salveze lista respectivă într-un fișier XML de genul:

```
<studenti>
  <student>Nume_student</student>
  <!-- etc -->
</studenti>
```

Pentru a face acest lucru vom crea o aplicație Windows Form, în care utilizatorul are o listă de studenți, iar la apăsarea unui buton va crea fișierul XML cu studenții din lista respectivă. Pe formă vom adăuga un control Button și un control listBox pe care o vom popula cu elemente nume de studenți.

```
private void button1_Click(object sender, EventArgs e)
{
    XmlDocument xml = new XmlDocument();
    //crearea xmlDeclaration
    XmlDeclaration xml_declaration = xml.CreateXmlDeclaration("1.0", "utf-8", "");
    xml.AppendChild(xml_declaration);
    XmlNode root = xml.CreateNode(XmlNodeType.Element, "students", ""); //crearea nodului radacina
    xml.AppendChild(root);

    foreach (string student in listBox1.Items)
    {
        //pentru fiecare studenti din listBox
        XmlNode tmpNode = xml.CreateNode(XmlNodeType.Element, "student", "");
        tmpNode.InnerText = student;
        root.AppendChild(tmpNode); //adaugarea nodului ca si copil la nodul radacina
    }
    xml.Save("studenti.xml");

    MessageBox.Show("XML scris cu succes!");
}
```

Rezultatul la rulare este crearea fișierului XML, studenti.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<students>
  <student>Andreea</student>
  <student>Maria</student>
  <student>Oana</student>
  <student>Simona</student>
  <student>Andrei</student>
  <student>Paul</student>
</students>
```

În următorul exemplu mai adăugăm câteva date la câmpul unui student, astfel nodul student va avea următoarea formă:

```
<student an="3" varsta="21">
  <nume>Ana</nume>
  <specialitatea>Informatica</specialitatea>
</student>
```

Programul care generează acest Xml este următorul (de obicei datele vin dintr-o bază de date, dintr-un formular etc, însă în exemplu datele despre student vor fi date de forma: “Ana, 3, 21, Informatica”):

```
private void button2_Click(object sender, EventArgs e)
{
    string[] students = new string[] { "Ana, 3, 21, Informatica", "Ioana, 1, 20, Fizica", "Paul, 2, Istorie, 22", "Andrei, 4, 23, Calculatoare" };
    XmlDocument xml = new XmlDocument();
    XmlNode root = xml.CreateNode(XmlNodeType.Element, "students", "");
    xml.AppendChild(root);
    foreach (string student in students)
    {
        //pentru fiecare student din colectia de noduri
        string[] stInfo = student.Split(',');
        XmlNode studentNode = xml.CreateNode(XmlNodeType.Element, "student", "");
        XmlAttribute y = xml.CreateAttribute("an");
        y.Value = stInfo[1].Trim();
        XmlAttribute age = xml.CreateAttribute("varsta");
        age.Value = stInfo[2].Trim();
        studentNode.Attributes.Append(y); //adauga atributele
        studentNode.Attributes.Append(age);

        //adaugarea urmatoarelor 2 noduri
        XmlNode studentName = xml.CreateNode(XmlNodeType.Element, "nume", "");
        studentName.InnerText = stInfo[0].Trim();
        XmlNode studentSpec = xml.CreateNode(XmlNodeType.Element, "specialitatea", "");
        studentSpec.InnerText = stInfo[3].Trim();
        studentNode.AppendChild(studentName);
        studentNode.AppendChild(studentSpec);

        root.AppendChild(studentNode);
    }
    xml.Save("studenti2.xml");
    MessageBox.Show("XML scris!");
}
```

Rezultatul la rulare este crearea șișierului XML, studenti2.xml:

```
<students>
  <student an="3" varsta="21">
    <nume>Ana</nume>
    <specialitatea>Informatica</specialitatea>
  </student>
  <student an="1" varsta="20">
    <nume>Ioana</nume>
    <specialitatea>Fizica</specialitatea>
  </student>
  <student an="2" varsta="Istorie">
    <nume>Paul</nume>
    <specialitatea>22</specialitatea>
  </student>
  <student an="4" varsta="23">
    <nume>Andrei</nume>
    <specialitatea>Calculatoare</specialitatea>
  </student>
</students>
```

2. Crearea și citirea unei baze de date în XML cu C#

Vom crea o formă pe care adăugăm 2 butoane și un dataGridView.



Vom adăuga bibliotecile:

```
using System.Xml.Serialization;
using System.IO;
```

Vom adăuga o clasă Date.cs cu următorul cod:

```
public class Date
{
    public int ID { get; set; }
    public String Nume { get; set; }
    public String Prenume { get; set; }
}
```

Codul sursă pentru butonul de creare bază de date:

```
private void button1_Click(object sender, EventArgs e)
{
    List<Date> D1 = new List<Date>();
    XmlSerializer serial = new XmlSerializer(typeof(List<Date>));
    D1.Add(new Date() { ID = 1, Nume = "Chirila", Prenume = "Oana" });

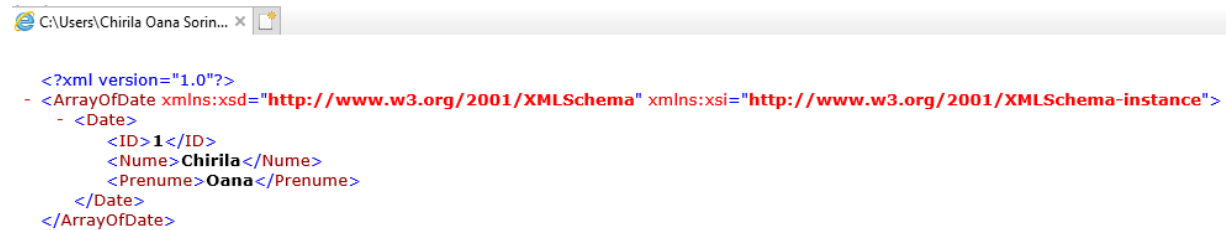
    using (FileStream fs = new FileStream(Environment.CurrentDirectory + "\\BD.xml", FileMode.Create,
    FileAccess.Write))
    {
        serial.Serialize(fs, D1);
        MessageBox.Show("Baza de date creata!");
    }
}
```

Codul sursă pentru citirea bazei de date:

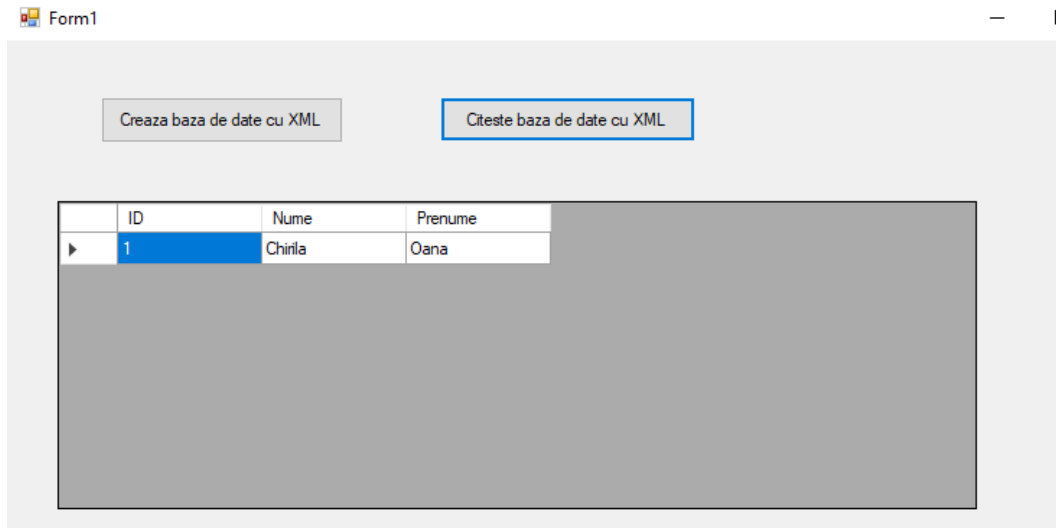
```
private void button2_Click(object sender, EventArgs e)
{
    List<Date> D1 = new List<Date>();
    XmlSerializer serial = new XmlSerializer(typeof(List<Date>));

    using (FileStream fs = new FileStream(Environment.CurrentDirectory + "\\BD.xml", FileMode.Open, FileAccess.Read))
    {
        D1 = serial.Deserialize(fs) as List<Date>;
    }
    dataGridView1.DataSource = D1;
}
```

Rezultatul la rulare:



```
<?xml version="1.0"?>
- <ArrayOfDate xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Date>
    <ID>1</ID>
    <Nume>Chirila</Nume>
    <Prenume>Oana</Prenume>
  </Date>
</ArrayOfDate>
```



ID	Nume	Prenume	
1	Chirila	Oana	

3. Crearea unui TreeView dintr-un fișier XML

Putem citi un fișier XML și să îl afișăm într-un control TreeView pentru o mai ușoară citire a fișierului XML sau pentru alte operații ce pot să intervină ulterior.

Pentru a realiza acest lucru vom crea o aplicație Windows Form în care vom adăuga pe formă un buton și un control TreeView:



Adăugăm următorul cod la click pe buton și mai creăm o funcție de adăugare nod (apelată în cadrul evenimentului click pe buton):

```
private void button1_Click(object sender, EventArgs e)
{
    XmlDocument xmldoc = new XmlDocument();
    XmlNode xmlnode;
```

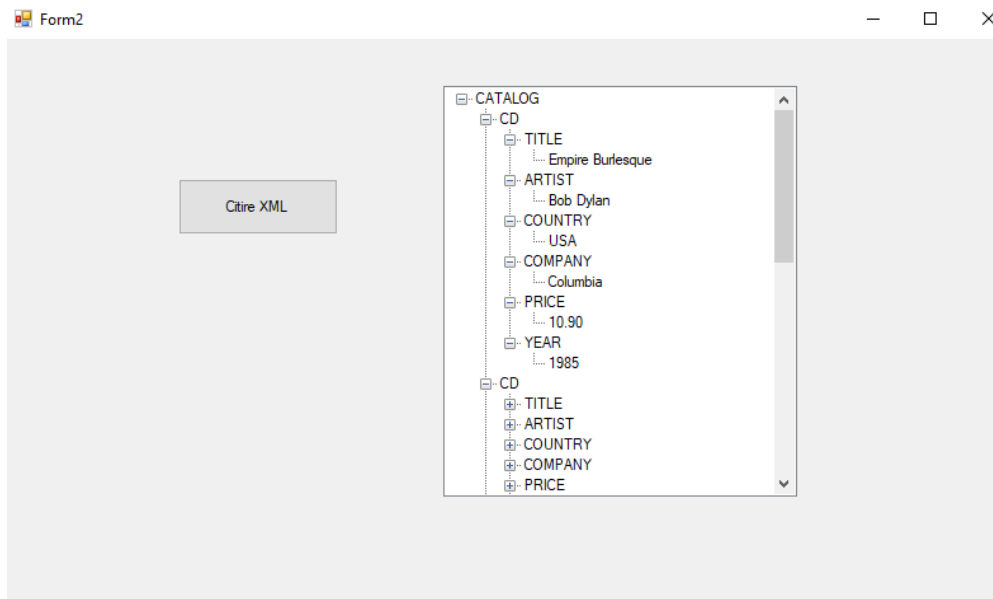
```

FileStream fs = new FileStream("D:\\tree.xml", FileMode.Open, FileAccess.Read);
xmldoc.Load(fs);
xmlnode = xmldoc.ChildNodes[1];
treeView1.Nodes.Clear();
treeView1.Nodes.Add(new TreeNode(xmldoc.DocumentElement.Name));
TreeNode tNode;
tNode = treeView1.Nodes[0];
AddNode(xmlnode, tNode);
}

private void AddNode(XmlNode inXmlNode, TreeNode inTreeNode)
{
    XmlNode xNode;
    TreeNode tNode;
    XmlNodeList nodeList;
    int i = 0;
    if (inXmlNode.HasChildNodes)
    {
        nodeList = inXmlNode.ChildNodes;
        for (i = 0; i <= nodeList.Count - 1; i++)
        {
            xNode = inXmlNode.ChildNodes[i];
            inTreeNode.Nodes.Add(new TreeNode(xNode.Name));
            tNode = inTreeNode.Nodes[i];
            AddNode(xNode, tNode);
        }
    }
    else
    {
        inTreeNode.Text = inXmlNode.InnerText.ToString();
    }
}

```

Rezultatul la rulare este următorul:



4. Probleme propuse

1. Să se creeze o aplicație Windows Form care citește un fișier xml. Fișierul xml conține denumirile și proprietățile mai multor controale. Să se genereze pe formă controalele respective cu toate proprietățile aferente:

```
<?xml version="1.0"?>
<Forma>
  <textBox nume="Nume" detalii="Nume angajat"/>
  <textBox nume="Prenume" detalii="Prenume angajat"/>
  <textBox CNP="CNP" detalii="Cnp angajat"/>
  <radioButton nume="gen" detalii="Gen:">
    <val>masculin</val>
    <val>feminin</val>
  </radioButton>
  <checkBox nume="LP" detalii="Limbaje de programare cunoscute">
    <val>C</val>
    <val>C++</val>
    <val>C#</val>
    <val>Java</val>
    <val>Python</val>
  </checkBox>
  <listBox judet="Judet" detalii="Judet Domiciliu">
    <val>Alba</val>
    <val>Arad</val>
    <val>Hunedoara</val>
    <val>Timis</val>
    <val>Bihor</val>
    <val>Olt</val>
    <val>Gorj</val>
  </listBox>
</Forma>
```

2. Sa se creeze un fisier xml in care se va afisa numarul de click-uri date pe butoanele de pe o formă (forma dispune de 10 butoane). Numele fiecarui tag xml este dat de numele butonului care are următorii copii: culoare buton și număr click-uri.
3. După fișierul xml generat la problema precedentă să se creeze o aplicație care generează automat pe formă butoane cu proprietățile acestora din fișierul XML. Fiecare buton va avea ca și text numărul de click-uri scris în fișierul XML.
4. Să se afișeze pe o formă, într-o aplicație Windows Form, într-un control DataGridView datele de la următorul link:
[https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ms762271\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ms762271(v=vs.85))
 Fișierul XML se va salva pe disc în folderul în care se va crea fișierul executabil al aplicației pe care o veți crea.
5. Să se creeze o aplicație care citește datele din xml-ul de la adresa [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ms762271\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ms762271(v=vs.85)), salvat pe disc în directorul de start al aplicației, modifică prețurile produselor prin adăugarea unui adaos de 10% la fiecare pret și le salvează din nou în documentul XML.