

Bazy danych

Baza danych XML

sprawia, że jest to format wykorzystywany niemal wszędzie. Elastyczność tego formatu oraz fakt, że może zawierać informacje o swojej strukturze sprawiło, że bardzo szybko został przyjęty przez programistów oraz twórców stron internetowych. Warto zapoznać się ze strukturą budowy dokumentu XML, gdyż Microsoft bardzo silnie zintegrował swoje narzędzia z tym właśnie formatem. Pisząc programy w Visual Studio, nawet nieświadomie będziesz używał XML w swoich projektach.

Dokument XML składa się z:

- nagłówka,
- komentarzy,
- elementu głównego,
- elementów,
- atrybutów.

Nagłówek dokumentu XML nie jest wymagany, lecz warto go używać, ze względu na kompatybilność. Opcjonalnie, można dołączyć do nagłówka format kodowania znaków, co umożliwi ich poprawny odczyt.

Komentarze w XML, służą podobnie jak w HTML i innych językach programowania do opisywania określonych wierszy. Komentarze mogą zawierać dowolne treści.

Główny element – korzeń

Każdy dokument w formacie XML zawiera jeden element główny. W elemencie tym, są zawarte wszystkie inne zagnieżdżone elementy. Dlatego też, dokumenty XML mają strukturę zwaną drzewiastą lub hierarchiczną.

Elementy dokumentu

Elementy znajdują się wewnątrz głównego elementu (korzenia). Element składa się ze znacznika początkowego oraz końcowego, innych zagnieżdżonych dokumentów lub łańcucha znaków.

Atrybuty dokumentu

Każdy element w dokumencie XML może zawierać atrybuty opisane w znaczniku początkowym. Jest to po prostu jeszcze jeden sposób na przekazanie danych dotyczących elementu. Atrybut musi stosować się do szablonu nazwa = „wartość”.

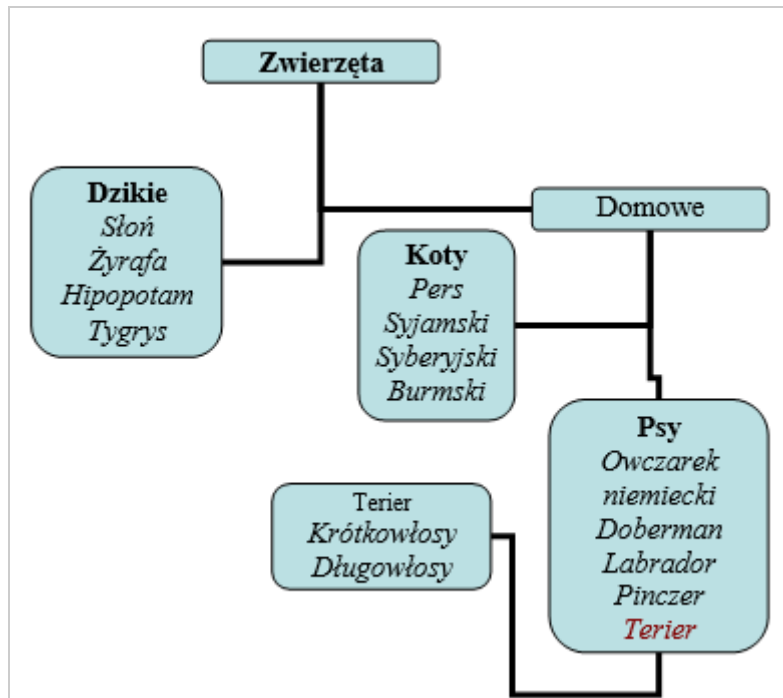
Elementem głównym (korzeniem) jest w naszym przypadku . Elementami są dwa „rekordy” zawierające atrybuty Imię, Nazwisko oraz Telefon. Warto nadmienić, że XML rozpoznaje wielkość liter. Tak więc wpisanie spowoduje błędy w odczycie dokumentu. W dokumentach XML należy także zadbać o zamykanie każdego znacznika. W przeciwnym wypadku dokument będzie niepoprawny składniowo. Wypełnianie elementu atrybutem jest opcjonalne. Warto jednak wiedzieć, że jeśli w jednym elemencie występuje znacznik , to jeśli nie zamierzamy wprowadzać go w kolejnym elemencie, należy umieścić znacznik pusty – , co widać w przykładowych dokumentach. Odpowiednikiem takiego zapisu jest . Dodatkowo, do elementu można dopisać atrybut przy czyli . Oczywiście wcale nie musi to być wartość numeryczna. Atrybuty mogą być pomocne na przykład podczas przeszukiwania bazy.

W dokumencie XML bardzo ważną zasadą poprawności formatowania jest kolejność otwierania i zamykania znaczników. Węzły muszą być otwierane i zamykane w tej samej kolejności.

Powyższy zapis jest błędny, gdyż nie zachowano kolejności zamknięcia znaczników. Warto zwrócić na to uwagę, podczas tworzenia dokumentów XML.

Węzły w dokumencie XML można dowolnie zagnieżdżać. Oczywiście, jak już wspomnieliśmy, wszelkie węzły muszą znajdować się wewnątrz znaczników korzenia. Spróbujmy dla przykładu opisać dokumentem XML zwierzęta.

Podział będzie wyglądał tak:



Zapiszmy więc dokument w formacie XML o strukturze, jak na Rysunek 15 Struktura dokumentu zwierzęta. Pamiętajmy o zasadach formatowania dokumentów, zamykaniu i kolejności znaczników.

Zapisz dokument XML na dysku pod dowolną nazwą i uruchom go za pomocą przeglądarki internetowej.

Kod:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

// dołączenie niezbędnych bibliotek
using System.Xml;
using System.Xml.XPath;
using System.IO;
```

```

namespace WindowsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            //wspólne dane - nazwa pliku bazy
            const string FILE_NAME = "przyklad.xml"; //sciezka do bazy
            public void czytajDane()
            {
                try
                {
                    DataSet _dataSet = new DataSet();
                    _dataSet.ReadXml(FILE_NAME);
                    this.dataGrid1.DataSource = _dataSet;
                    this.dataGrid1.DataMember = "Dane";
                }
                catch
                {
                    //tworzymy nową bazę XML
                    MessageBox.Show("Brak danych do odczytu", "Informacja");
                    string nazwaPliku = FILE_NAME;
                    XmlTextWriter _xmlTextWriter = new XmlTextWriter(nazwaPliku,
                    System.Text.Encoding.UTF8);
                    _xmlTextWriter.WriteStartDocument();
                    _xmlTextWriter.WriteStartElement("dataSet");
                    _xmlTextWriter.WriteStartElement("Dane");
                    _xmlTextWriter.WriteElementString("Imie", "zmień dane");
                    _xmlTextWriter.WriteElementString("Nazwisko", "zmień dane");
                    _xmlTextWriter.WriteElementString("Adres", "zmień dane");
                    _xmlTextWriter.WriteElementString("KodPocztowy", "zmień
dane");
                    _xmlTextWriter.WriteElementString("Miasto", "zmień
dane");
                    _xmlTextWriter.WriteElementString("Telefon", "zmień
dane");
                    _xmlTextWriter.WriteElementString("Email", "zmień dane");
                    _xmlTextWriter.WriteEndElement();
                    _xmlTextWriter.WriteEndDocument();
                    _xmlTextWriter.Flush();
                    _xmlTextWriter.Close();
                    czytajDane();
                }
            }
        }
    }
}

```

```

        MessageBox.Show("Brak bazy XML. Utworzono nową bazę.",
"Informacja");
    }
} //koniec czytaj dane

private void Form1_Load(object sender, EventArgs e)
{
    czytajDane();
}

private void button5_Click_1(object sender, EventArgs e)
{
    StreamReader sr = File.OpenText(FILE_NAME);
    String input;
    input = sr.ReadToEnd();
    sr.Close();
    textBox6.Text = input;
    XmlTextReader reader = new XmlTextReader(FILE_NAME);
    XmlDocument doc = new XmlDocument();
    doc.Load(reader);
    reader.Close();

    XmlNode currNode;
    XmlDocumentFragment docFrag = doc.CreateDocumentFragment();
    if (textBox1.Text == "") //imie
    {
        MessageBox.Show("Uzupełnij pole: Imię", "Informacja");
    }
    else if (textBox2.Text == "") //nazwisko
    {
        MessageBox.Show("Uzupełnij pole: Nazwisko",
"Informacja");
    }
    else
    {
        docFrag.InnerXml = "<Dane>" +
            "<Imie>" + textBox1.Text + "</Imie>" +
            "<Nazwisko>" + textBox2.Text +
            "</Nazwisko>" +
            "<Adres>" + textBox3.Text + "</Adres>"
            +
            "<KodPocztowy>" + textBox4.Text +
            "</KodPocztowy>" +
            "<Miasto>" + textBox5.Text +
            "</Miasto>" +

```

```

        "<Telefon>" + textBox7.Text +
"</Telefon>" +
        "<Email>" + textBox8.Text + "</Email>"
+
        "</Dane>";

        //czyszcimy pola
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";
        textBox5.Text = "";
        textBox7.Text = "";
        textBox8.Text = "";
    }

    currNode = doc.DocumentElement;
    currNode.InsertAfter(docFrag, currNode.LastChild);
    doc.Save(FILE_NAME);
    czytajDane();
} //dodaj koniec

private void button9_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("Skasować rekord?",
"Wyberz opcję", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button3);
    if (result == DialogResult.Yes)
    {
        try
        {
            int z = dataGrid1.CurrentRowIndex; //zwraca aktualna
pozycje kursora
            int v = z + 1;
            XmlTextReader reader = new XmlTextReader(FILE_NAME);
            XmlDocument doc = new XmlDocument();
            doc.Load(reader);
            reader.Close();
            XmlNode cd;
            XmlElement root = doc.DocumentElement;
            //XPath
            cd = root.SelectSingleNode("/dataSet/Dane[position()=" +
v + "]);
            root.RemoveChild(cd);
            doc.Save(FILE_NAME);
            czytajDane();

```

```

        }
    catch
    {
        MessageBox.Show("Brak rekordów do skasowania",
"Informacja");
    }
}
} //koniec kasuj

private void button2_Click(object sender, EventArgs e)
{
    StreamReader sr = File.OpenText(FILE_NAME);
    String input;
    input = sr.ReadToEnd();
    sr.Close();
    textBox6.Text = input;
} //koniec odswiez

private void button3_Click_1(object sender, EventArgs e)
{
    try
    {
        XmlTextReader reader = new XmlTextReader(FILE_NAME);
        XmlDocument doc = new XmlDocument();
        doc.Load(reader);
        reader.Close();
        int z = dataGridView1.CurrentRowIndex;
        int v = z + 1;
        XmlNode oldCd;
        XmlElement root = doc.DocumentElement;
        oldCd = root.SelectSingleNode("/dataSet/Dane[position()="
+ v + "]); //stary wezel

        XmlElement newCd = doc.CreateElement("Dane"); //nowy
wezel
        newCd.InnerXml = "<Imie>" + textBox1.Text + "</Imie>" +
            "<Nazwisko>" + textBox2.Text +
"</Nazwisko>" +
            "<Adres>" + textBox3.Text + "</Adres>"
+
            "<KodPocztowy>" + textBox4.Text +
"</KodPocztowy>" +
            "<Miasto>" + textBox5.Text +
"</Miasto>" +

```

```

        "<Telefon>" + textBox7.Text +
"</Telefon>" +
        "<Email>" + textBox8.Text + "</Email>";
        root.ReplaceChild(newCd, oldCd);
        doc.Save(FILE_NAME);
        czytajDane();
    }
    catch
    {
        this.DialogResult = DialogResult.Cancel;
    }
} //koniec edycja

public void wczytajDane()
{
    XPathDocument doc;
    XPathNavigator nav;
    XPathExpression expr;
    XPathNodeIterator iterator;
    int z = dataGrid1.CurrentRowIndex;
    int v = z + 1;
    doc = new XPathDocument(FILE_NAME);
    nav = doc.CreateNavigator();
    expr = nav.Compile("/dataSet/Dane[position()=" + v + "]");
    iterator = nav.Select(expr);
    while (iterator.MoveNext())
    {
        XPathNavigator nav2 = iterator.Current.Clone();
        nav2.MoveToFirstChild(); //pierwszy rekord
        textBox1.Text = nav2.Value;
        nav2.MoveToNext(); // ..i następny
        textBox2.Text = nav2.Value;
        nav2.MoveToNext(); // ..i następny
        textBox3.Text = nav2.Value;
        nav2.MoveToNext(); // ..i następny
        textBox4.Text = nav2.Value;
        nav2.MoveToNext();
        textBox5.Text = nav2.Value;
        nav2.MoveToNext();
        textBox7.Text = nav2.Value;
        nav2.MoveToNext();
        textBox8.Text = nav2.Value;
    }
} //koniec wczytajDane

```



```
private void dataGrid1_Click(object sender, EventArgs e)
{
    //dodana akcja. Kliknięcie powoduje odswiezenie danych w
    polach tekstowych
    wczytajDane();
}
}
```