# Лабораторная работа 6

# LINQ и XML

**6.1 Создание XML-документа методами пространств имен System.XML.Linq**

Кроме пространства имен System.xmi, содержащего классы для обработки XML-документов, в Visual С# имеем пространство имен System.xml.Linq, содержащее классы, которые позволяют легко и эффективно изменять документы XML, а также организовывать LINQ-запросы. В данном примере оформим сведения о наших повседневных телефонных контактах в XML-документ. Этот документ будет иметь интуитивно понятную структуру: имя контакта, домашний и мобильный телефоны. Создав такой XML-документ и получив соответствующий XML-файл, его очень удобно просмотреть в MS Excel в виде таблицы, содержащей три столбца: имя контакта, домашний телефон и мобильный.

Попутно обсудим структуру XML-документа.

1 Запустить среду Visual Studio

2 Выбрать проект шаблона Console Application

3 Указать имя Name — LinqCoздaтьXML-дoкyмeнт. Затем на вкладке

программного кода введем текст, представленный в листинге 1.

using System;

using System.Xml.Linq;

*// Другие директивы using удалены, поскольку они не используются в*

данной программе

namespace LinqСоздатьXML\_документ

{

class Program

{

static void Main(string[] args)

{

Console.Title = "Корневой элемент XML-документа";

*// Создаем новый XML-документ:*

XDocument XMLдокумент = new XDocument(

*// Комментарий в XML-документе:*

new XComment("Телефонная\_книга - это корневой элемент XMLдокумента:"),

new XElement("Телефонная\_книга", *// - имя корневого элемента*

new XComment("Элемент СТРОКА содержит атрибут Контакт и два

вложенных элемента"),

new XElement("СТРОКА", *// - имя (Name) элемента*

new XAttribute("Контакт", "Олег"),

new XElement("Домашний\_телефон", "236-23-67"), *// - имя элемента и*

его значение

new XElement("Мобильный\_телефон", "+7(495)625-31-43")),

new XComment("Атрибут Контакт имеет значение 'Прогноз погоды':"),

new XElement("СТРОКА",

new XAttribute("Контакт", "Прогноз погоды"), *// - атрибут элемента*

СТРОКА

new XElement("Домашний\_телефон", "001"),

new XElement("Мобильный\_телефон", "")), *// - имя элемента и его*

зачение (Value)

new XComment("Поскольку каждый элемент Контакт имеет атрибут и два

вложенных=>"),

new XElement("СТРОКА",

new XAttribute("Контакт", "Борис Григорьевич"), *// - имя атрибута -*

Контакт

new XElement("Домашний\_телефон", "402-12-45"),

new XElement("Мобильный\_телефон", "+7(495)536-79-94")),

new XComment("=> элемента, в MS Excel отобразится таблица с тремя

колонками"),

new XElement("СТРОКА",

new XAttribute("Контакт", "Света"), *// - значение атрибута - Света*

new XElement("Домашний\_телефон", ""),

new XElement("Мобильный\_телефон", "+7(495)615-24-41")))

);

*// Сохранить XML-документ:*

XMLдокумент.Save(@"C:\имя\_файла.XML");

Console.WriteLine(XMLдокумент);

Console.ReadKey();

}

}

}

Чтобы понять текст программы, рассмотрим структуру полученного XMLфайла, а для этого откроем этот файл с помощью браузера.

Здесь весь XML-документ вложен в так называемый корневой элемент между начальным тегом <Телефонная\_книга> И конечным тегом </Телефонная\_книга>. Четыре элемента СТРОКА вложены в корневой элемент. В соответствующей таблице MS Excel элементы СТРОКА будут представлять строку в таблице. В свою очередь элемент СТРОКА содержит в себе атрибут контакт и два вложенных в него элемента, имена (Name) которых — домашний\_телефон и мобильный\_телефон. Именно поэтому в MS Excel отобразится таблица с тремя колонками (один атрибут и два элемента): "Контакт", «Домашний\_телефон» и «Мобильный\_телефон».

Элемент может иметь один или несколько атрибутов (а может и не иметь, как, скажем, элемент домашний\_телефон), например, первый элемент СТРОКА имеет атрибут с именем (Name) контакт и со значением атрибута (value) — 001.

После запуска данной программы будет выведено на консоль содержимое XML-документа (без XML-объявления), а также будет создан XML-файл. Открыв этот файл с помощью MS Excel, получим таблицу телефонных контактов.

**Задания для самостоятельного выполнения:**

1. Создать ХМL файл, содержащий сведения о 5 студентах и их оценках

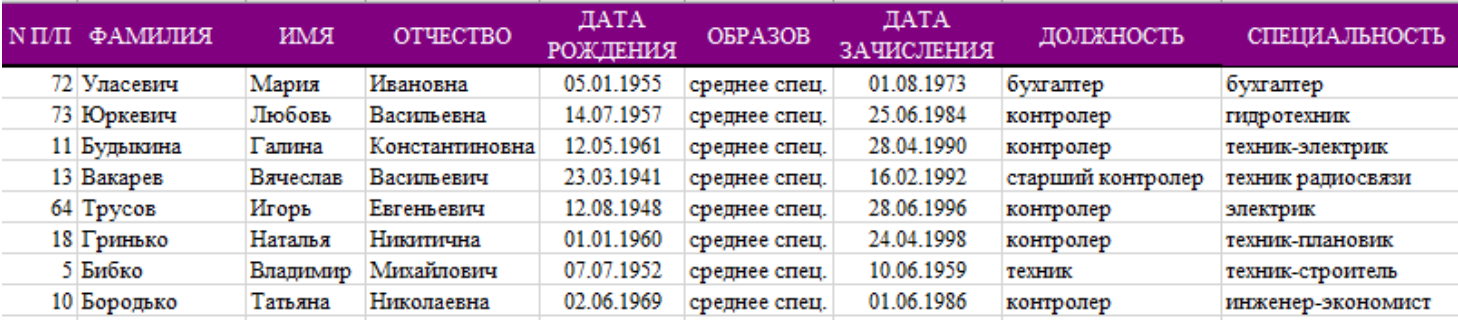
по 5 предметам.

2. Создать ХМL файл, содержащий сведения о доходах предприятия.

Столбцы представлены месяцами, а строки – статьей дохода (5 любых статей)

3. Создать ХМL файл, используемой для отображения следующей

таблицы:



**6.2 Извлечение значения элемента из XML-документа с использованием LINQ-запросов**

Имеем XML-данные, в которых содержится таблица с именами и телефонами, причем имена в этой телефонной табличке повторяются, например, строка с именем "Витя" содержит мобильный телефон, а потом по мере знакомства с этим Витей у нас появился уже и его домашний телефон. Задача состоит в том, чтобы в данной таблице телефонов (представленной в виде XML, см. листинг 1) найти все строчки с именем "Витя". Эта маленькая несерьезная, на первый взгляд, задача подразумевает, например, такую уже "серьезную" задачу. Имеем громадную базу данных, которую мы получили на каком-то этапе обработки в виде XML, и нам требуется "отфильтровать" записи в этой базе на предмет содержания в некотором поле определенной строки. Как видно, в редакторе MS Excel наши XML-данные представлены весьма наглядно. И очень понятно, что мы хотим получить, а именно, все номера телефонов напротив имени "Витя".

1 Запустим Visual Studio

2 Выбрать проект шаблона Windows Forms Application

3 Указать имя Name — LinqN.

4 В конструктор формы, из панели элементов Toolbox перетащить текстовое поле TextBox для вывода в него найденных строк из таблицы XML.

5 В свойствах текстового поля разрешить ввод множества строк (а не одной), для этого свойство Multiline переведем в состояние true. Затем на вкладке программного кода введем текст, представленный в листинге 2.

*// Имеем XML-данные, в которых содержатся таблица с именами и*

телефонами, причем

*// имена в этой телефонной табличке повторяются. Задача состоит в том,*

чтобы в

*// данной таблице телефонов (представленной в виде XML) найти все*

строчки c

*// именем "Витя" с помощью LINQ-запроса.*

using System.Linq;

using System.Windows.Forms;

*// Другие директивы using удалены, поскольку они не используются в*

данной программе

namespace Linq5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

this.Text = "LINQ-запрос к XML-данным"; textBox1.Multiline = true;

*// Инициализация XML-строки:*

string СтрокаXML =

@"<?xml version=""1.0""?>

<ТаблицаТелефонов>

<Строка>

<Имена>Витя</Имена>

<Номера\_телефонов>274 28 44</Номера\_телефонов>

</Строка>

<Строка>

<Имена>Андрей</Имена>

<Номера\_телефонов>8-085-456-2378</Номера\_телефонов>

</Строка>

<Строка>

<Имена>Карапузова Таня</Имена>

<Номера\_телефонов>445-56-47</Номера\_телефонов>

</Строка>

<Строка>

<Имена>Витя</Имена>

<Номера\_телефонов>099 72 161 52</Номера\_телефонов>

</Строка>

<Строка>

<Имена>Никипелов</Имена>

<Номера\_телефонов>236-77-76</Номера\_телефонов>

</Строка>

<Строка>

<Имена>Зиборов</Имена>

<Номера\_телефонов>254 67 97</Номера\_телефонов>

</Строка>

</ТаблицаТелефонов>";

var КорневойЭлемент =

System.Xml.Linq.XElement.Parse(СтрокаXML);

*// Запись строки, содержащей XML в файл:*

*// System.IO.File.WriteAllText("ТаблицаТелефонов.xml",*

СтрокаXML);

*// var КорневойЭлемент = System.Xml.Linq.*

*// XElement.Load("ТаблицаТелефонов.xml");*

var Записи =

from x in КорневойЭлемент.Elements("Строка")

where (string)x.Element("Имена") == "Витя"

select x.Element("Номера\_телефонов").Value;

textBox1.Text = textBox1.Text +

@"Строки, содержащие имя ""Витя"":" + "\r\n";

*// Вывод коллекции записей в текстовое поле textBox1:*

foreach (var x in Записи)

textBox1.Text = textBox1.Text + x + "\r\n";

*// Таких записей в этой коллекции - ровно одна*

}

private void Form1\_Load(object sender, System.EventArgs e)

{ }

}

}

Как видно, в начале программы мы инициализируем (т. е. присваиваем начальные значения) XML-строку. Далее извлекаем корневой элемент из XML документа, он, по сути, отличается от XML-документа отсутствием XML объявления (в этом можно убедиться в отладчике программы). В комментарии указано, как можно получить корневой элемент в том случае, если он представлен в виде XML-файла во внешней памяти. Затем организуем типовой, стандартный LINQ-запрос. Результат запроса попадает в коллекцию записей, которую выводим в текстовое поле, используя оператор цикла foreach.

**Задачи для самостоятельного выполнения:**

1. Используя созданный XML-файл в задании №3 из лабораторной работы №6.1, выбрать информацию о Трусове Игоре

2. Используя созданный XML-файл в задании №1 из лабораторной работы №6.1, выбрать информацию о каком-либо студенте

3. Используя созданный XML-файл в задании №3 из лабораторной работы №6.1, выбрать информацию о сотрудниках с должностью контроллер

4. Используя созданный XML-файл в задании №3 из лабораторной работы №6.1, выбрать информацию о сотрудниках, у которых должностью совпадает со специальностью