

Лабораторная работа №10

Создание методами пространства имен `System.Xml.Linq` и поиск строк с помощью LINQ-запроса в XML-файлах

Кроме пространства имен `System.Xml`, содержащего классы для обработки XML-документов, в Visual C# 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 студентах в XML-документ. Этот документ будет иметь интуитивно понятную структуру: фамилия студента, группа 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: студент, группа и мобильный телефон.

Запустим среду **Visual Studio**, выберем проект шаблона **Console Application**, укажем любое имя. Затем на вкладке программного кода введем текст, следуя шагам:

1. Объявим директиву

```
using System.Xml.Linq;
```

2. Создаем новый XML-документ

```
XDocument XMLдокумент = new XDocument(структура XML-документа: п.3+п.4)
```

3. Задаем имя корневого элемента

```
new XElement("Телефонная_книга",элементы и вложенные элементы: п.4)
```

4. Зададим вложенные элементы, имя элемента и его значение, а также атрибуты элемента, их имена и значения

```
new XElement("СТРОКА", // - имя (Name) элемента  
new XAttribute("Студент", "Олег"), // - атрибут элемента Строка  
new XElement("Группа", "ИТ-31"), // - имя элемента и его значение  
new XElement("Мобильный_телефон", "+7(902)120-20-20")),
```

5. Повторим код пункта 4 для каждого студента

6. Сохраняем и выводим на экран полученный документ

```
XMLдокумент.Save(@"C:\имя_файла.XML");
Console.WriteLine(XMLдокумент);
Console.ReadKey();
```

Чтобы понять текст программы, рассмотрим структуру полученного XML- файла, 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>`. Элементы *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 студент и два вложенных в него элемента, имена (Name) которых — группа и мобильный_телефон. Именно поэтому в MS Excel отобразится таблица с тремя колонками (один атрибут и два элемента): «Студент», «Группа» и «Мобильный_телефон».

Элемент может иметь один или несколько атрибутов (а может и не иметь, как, скажем, элемент мобильный_телефон).

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 студентов.

Задание для самостоятельного выполнения (блок 1):

1. Создать XML-файл сотрудников, формирующий следующую таблицу:

№ П/П	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ДАТА РОЖДЕНИЯ	ОБРАЗОВ	ДАТА ЗАЧИСЛЕНИЯ	ДОЛЖНОСТЬ	СПЕЦИАЛЬНОСТЬ
72	Уласевич	Мария	Ивановна	05.01.1955	среднее спец.	01.08.1973	бухгалтер	бухгалтер
73	Юркевич	Любовь	Васильевна	14.07.1957	среднее спец.	25.06.1984	контролер	гидротехник
11	Будькина	Галина	Константиновна	12.05.1961	среднее спец.	28.04.1990	контролер	техник-электрик
13	Вакарев	Вячеслав	Васильевич	23.03.1941	среднее спец.	16.02.1992	старший контролер	техник радиосвязи
64	Трусов	Игорь	Евгеньевич	12.08.1948	среднее спец.	28.06.1996	контролер	электрик
18	Гринько	Наталья	Никитична	01.01.1960	среднее спец.	24.04.1998	контролер	техник-плановик
5	Бибко	Владимир	Михайлович	07.07.1952	среднее спец.	10.06.1959	техник	техник-строитель
10	Бородько	Татьяна	Николаевна	02.06.1969	среднее спец.	01.06.1986	контролер	инженер-экономист

Имеем 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, например, отобразить список телефон студентов необходимой группы.

Для решения этой задачи запустим **Visual Studio** и выберем проект шаблона **Windows Forms Application**, укажем любое имя. Далее, попав в конструктор формы, из панели элементов **Toolbox** перетащим текстовое поле **TextBox** для вывода в него найденных строк из таблицы XML. В свойствах текстового поля разрешим ввод множества строк (а не одной), для этого свойство **Multiline** переведем в состояние true. Затем на вкладке программного кода введем текст, следуя шагам:

1. Извлекаем информацию из XML-файла

```
var КорневойЭлемент = System.Xml.Linq.  
    XElement.Load("C:\имя_файла.XML");
```

2. Используем LINQ-запрос для выборки данных

```
var Записи =  
    from x in КорневойЭлемент.Elements("СТРОКА")  
    where (string)x.Element("Группа") == "ИТ-31"  
    select x.Element("Мобильный_телефон").Value;
```

Для выборки записей по значению атрибута конструкцию **where** необходимо заменить на:

```
where (string)x.Attribute("Студент").Value=="Иванов"
```

3. Вывести коллекцию записей в текстовое поле

Извлекаем корневой элемент из XML-документа. 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 (блок 2):

1. Вывести сотрудников, у которых в этом году юбилей
2. Выбрать сотрудников, рожденных после 01.01.1960 года.
3. Создать интерфейсную форму, позволяющую осуществлять гибкий поиск по XML-файлу: по фамилии, дате рождения (конкретная дата или диапазон), образованию, дате зачисления (конкретная дата или диапазон), должности. Пользователь может выбрать как один критерий поиска, так и несколько различных.