

Комп'ютерний практикум № 2. LINQ to XML

Мета:

- ознайомитися з обробкою XML документів з використанням технології LINQ to XML

Теоретичні основи

Запит LINQ (Language-Integrated Query) являє собою вираз, за допомогою якого можна отримувати дані з будь-яких джерел даних.

По типу джерела виділяють наступні різновиди LINQ:

- LINQ to Objects - дозволяє застосовувати запити до масивів і колекцій об'єктів;
- LINQ to DataSet - дозволяє застосовувати запити LINQ до об'єктів DataSet з ADO.NET;
- LINQ to Entities - дозволяє застосовувати запити LINQ всередині API-інтерфейсу ADO.NET Entity Framework (EF);
- LINQ to XML - дозволяє застосовувати запити LINQ до документів XML і маніпулювати XML-даними;
- Parallel LINQ (PLINQ) - дозволяє виконувати паралельну обробку даних, повернутих запитом LINQ.

Простір імен System.Xml.Linq містить класи, що представляють різні аспекти XML-документа (елементи, атрибути, тощо).

Основними класами є:

- XDocument - представляє цілком весь XML-документ;
- XElement - представляє певний елемент всередині XML-документа;
- XAttribute - представляє атрибут певного елемента;
- XDeclaration - представляє оголошення XML-документа;
- XComment - коментар XML; •
- XName - представляє ім'я елемента або атрибута XML;
- XNamespace - представляє простір імен XML;
- XmlNode - представляє вузол на дереві XML-документа (елемент, атрибут, коментар, тощо).

В класі XDocument властивість Root дозволяє отримувати кореневий елемент даного документа. Можна відзначити наступні загальні методи класу XDocument:

- Load (string fileName) - створює новий об'єкт XDocument з файлу з ім'ям fileName (статичний метод);
- Parse (string text) - створює новий об'єкт XDocument з рядка text (статичний метод);
- Save (string fileName) - перетворює об'єкт XDocument в XML-файл з ім'ям fileName;
- Element (XName name) - повертає перший (в порядку проходження) дочірній елемент з ім'ям name;
- Elements (XName name) - повертає колекцію дочірніх елементів для даного документа (до складу колекції входять тільки елементи з ім'ям name).

Клас XElement позначає елемент XML і є одним з основних класів в LINQ to XML. Цей клас можна використовувати для створення елементів, зміни вмісту елемента, додавання, зміни або видалення дочірніх елементів, додавання до елементів атрибутів або перетворення вмісту елемента в текстову форму.

Основні властивості класу XElement:

- Name - ім'я елемента;
- Value - текстовий вміст елемента;
- Parent - батьківський елемент для даного елемента.

Основні методи класу XElement:

- Element (XName name) - повертає перший дочірній елемент об'єкта XElement, що має вказане ім'я name;
- Attribute (XName name) - повертає атрибут XAttribute, що має ім'я name;
- Attribute () - повертає колекцію IEnumerable всіх атрибутів XAttribute.

Після завантаження XML-документа до нього можна застосувати запити LINQ для пошуку колекцій елементів і атрибутів, а також отримання їх значень. Колекції елементів і атрибутів XML отримують за допомогою методів, які називають осьовими методами або осями. Ці методи можна застосовувати безпосередньо до частин дерева XML або вузлів або використовувати їх для побудови більш складних запитів LINQ. Частина осей є методами класів XElement і XDocument. Інші осі є методами розширень в класі Extensions.

Основні осьові методи LINQ to XML:

- Elements (XName name) - повертає колекцію дочірніх елементів об'єкта XElement, що мають вказане ім'я name;

- Descendants (XName name) - повертає колекцію елементів-нащадків об'єкта XElement, що мають ім'я name;
- Ancestors () - повертає колекцію елементів-предків об'єкта XElement.

Приклад:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml;
using System.Xml.Linq;

namespace Lab2Example
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.OutputEncoding = Encoding.UTF8;
            // Створюємо файл
            IList<User> users = new List<User> {
                new User ("Bill Gates", "Microsoft", 48),
                new User ("Larry Page", "Google", 42)
            };

            XmlWriterSettings settings = new XmlWriterSettings();
            settings.Indent = true;

            using (XmlWriter writer = XmlWriter.Create("users.xml", settings))
            {
                writer.WriteStartElement("users");

                foreach (User user in users)
                {
                    writer.WriteStartElement("user");
                    writer.WriteElementString("name", user.Name);
                    writer.WriteElementString("company", user.Company);
                    writer.WriteElementString("age", user.Age.ToString());
                    writer.WriteEndElement();
                }
                writer.WriteEndElement();
            }
        }
    }
}
```

```

// Виводимо файл
XmlDocument doc = new XmlDocument();
doc.Load("users.xml");

foreach (XmlNode node in doc.DocumentElement)
{
    string name = node["name"].InnerText;
    string company = node["company"].InnerText;
    int age = Int32.Parse(node["age"].InnerText);

    Console.WriteLine(string.Format("Користувач={0} працює в {1}, вік {2}", name, company, age));
}
// XmlDocument, XElement
Console.WriteLine();
XmlDocument xmlDoc = XmlDocument.Load("users.xml");
foreach (XElement userElement in xmlDoc.Element("users").Elements("user"))
{
    XElement nameAttribute = userElement.Element("name");
    XElement companyElement = userElement.Element("company");
    XElement ageElement = userElement.Element("age");

    if (nameAttribute != null && companyElement != null && ageElement != null)
    {
        Console.WriteLine("Користувач: {0}", nameAttribute.Value);
        Console.WriteLine("Компанія: {0}", companyElement.Value);
        Console.WriteLine("Вік: {0}", ageElement.Value);
    }
    Console.WriteLine();
}
Console.WriteLine("Перелік компаній, в котрих працюють користувачі, відсортовані за зростанням");
var querySorted = xmlDoc.Descendants("user").Select(p => p.Element("company").Value).OrderBy(p => p.Trim());
foreach (var s in querySorted)
{
    Console.WriteLine(s);
}
Console.WriteLine();

```

```

Console.WriteLine("Фільтр за віком");
IEnumerable<XElement> queryAge =
    from b in xmlDoc.Root.Elements("user")
    where (int)b.Element("age") == 42
select b;
Console.WriteLine(queryAge.FirstOrDefault().Element("name").Value);

var items = from xe in xmlDoc.Element("users").Elements("user")
    where xe.Element("company").Value == "Google"
select new User
{
    Name = xe.Element("name").Value,
    Company = xe.Element("company").Value,
    Age = Int32.Parse(xe.Element("age").Value)
};
Console.WriteLine();
Console.WriteLine("Працюють в Google");
foreach (var item in items)
    Console.WriteLine("{0} - {1} - {2}", item.Name, item.Company,
item.Age);
Console.WriteLine();

Console.ReadKey();
}
}
class User
{
    /// <summary>
    ///   Ім'я
    /// </summary>
    public string Name { get; set; }
    /// <summary>
    /// <summary>
    ///   Компанія
    /// </summary>
    public string Company { get; set; }
    /// <summary>
    ///   Вік
    /// </summary>
    public int Age { get; set; }

    /// <summary>

```

```

    /// Конструктор за замовчанням
    /// </summary>
    public User()
    {
    }

    /// <summary>
    /// Конструктор
    /// </summary>
    /// <param name="name">Им'я</param>
    /// <param name="company">Компанія</param>
    /// <param name="age">Вік</param>
    public User(string name, string company, int age)
    {
        Name = name;
        Company = company;
        Age = age;
    }
}

```

Постановка задачі комп'ютерного практикуму № 2

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

- 1) Розробити структуру XML для зберігання даних згідно варіантів, наведених нижче.
- 2) Створити XML-файл з використанням XmlWriter. Дані необхідно вводити з консолі, зберегти його. Завантажити файл з використанням XmlDocument.
- 3) LINQ to XML:
 - а. Вивести зміст файлу, створеного в п.2
 - б. Для файлу, створеного в п.2 розробити як мінімум 15 різних запитів, використовуючи різні дії над отриманими даними. Запити не повинні повторюватись.
- 4) Створити програмне забезпечення, котре реалізує обробку даних з використання бібліотеки LINQ to XML.
- 5) Програмне забезпечення необхідно розробити у вигляді консольного застосування на мові C#.
- 6) Коротко описати архітектуру проекту проекту та створити звіт, котрий завантажити в moodle

Варіанти індивідуальних завдань:

- 1) Розробити структуру даних для зберігання інформації про студентів-дипломників та їх керівників. Про студентів необхідно зберігати щонайменше наступну інформацію: ПІБ, група, дата народження, середній бал. Про керівників: ПІБ, посада. У одного керівника може бути декілька студентів-дипломників.
- 2) Розробити структуру даних для зберігання інформації про літаки. Для об'єктів зберігається наступна інформація: літак - тип літака, вантажопідйомність, максимальна дальність, розмах крил, довжина розбігу, шифр компанії; вертоліт - тип вертольоту, вантажопідйомність, максимальна висота, дальність польоту, шифр компанії; авіакомпанія - назва, місце розташування офісу, дата утворення фірми, шифр компанії.
- 3) Розробити структуру даних для зберігання інформації для відстеження фінансових показників роботи пункту прокату автомобілів. В автопарк компанії входить кілька автомобілів різних марок, вартостей і типів. Кожен автомобіль має свою ціну прокату. В пункт прокату звертаються клієнти. Всі клієнти проходять обов'язкову реєстрацію - про них збирається стандартна інформація (ПІБ, адреса, телефон). Кожен клієнт може звертатися в пункт прокату кілька разів. Всі звернення клієнтів фіксуються, при цьому по кожній угоді запам'ятовуються дата видачі та очікувана дата повернення. Перед отриманням автомобіля клієнт залишає деяку заставну суму, яка йому повністю повертається після успішного повернення автомобіля. Вартість прокату автомобіля залежить не тільки від самого автомобіля, але і від терміну його прокату, а також від року випуску.
- 4) Розробити структуру даних для зберігання інформації про книги в бібліотеці. Книга характеризується: назвою, прізвищем автора, вартістю, датою видання, видавництвом, списком інвентарних номерів (книга в кількох примірниках). У одного автора може бути декілька книг.
- 5) Розробити структуру даних для зберігання інформації про обчислювальну техніку на підприємстві. Обчислювальна техніка характеризується: назвою, вартістю, обчислювальною потужністю, списком осіб, які експлуатують обчислювальну техніку.
- 6) Розробити структуру даних для зберігання інформації про будинки. Для об'єктів зберігається наступна інформація: житловий будинок - тип проекту, число поверхів, число під'їздів, дата побудови, шифр; район міста - назва, адреса районної адміністрації, кількість жителів, площа, шифр; список будинків - шифр району, шифр будинку.
- 7) Розробити структуру даних для зберігання інформації про реєстрацію транспортних засобів. Для кожного транспортного засобу зберігається як мінімум марка авто, виробник, модель, тип кузова, рік випуску, номер шасі (VIN-код), колір, номерний знак, технічний стан, власник

автомобіля, перелік водіїв, котрі мають право керувати транспортним засобом, тощо. Для власників та тих персон, котрі мають право керувати транспортним засобом, - номер прав водія, прізвище, ім'я, по батькові, дата народження, адреса реєстрації. Необхідно врахувати, що транспортний засіб може мати декілька власників (тобто бути зареєстрованим декілька разів).

- 8) Розробити структуру даних для зберігання інформації про товари на складі. Товар характеризується: назвою, вартістю, кількістю (в штуках), списком дат надходження на склад, виробником (найменування фірми).
- 9) Розробити структуру даних для зберігання інформації про квартири, котрі продаються різними агентствами нерухомості. По квартирам необхідно зберігати щонайменше наступну інформацію: адреса, поверх, площа, ціна. По агентствам – назва, адреса, ріелтери, контактні телефони кожного з ріелтерів. Передбачити, що одну й ту саму квартиру можуть продавати різні агентства нерухомості і ціна може бути різною.
- 10) Розробити структуру даних для зберігання інформації про проекти, що виконуються на підприємстві. По кожному проекту зберігається інформація: шифр проекту, найменування проекту, вартість робіт для виконання проекту, дата початку проекту і дата закінчення проекту, список осіб, які беруть участь в проекті.
- 11) Розробити структуру даних для зберігання інформації про статті авторів, котрі опубліковані в різних журналах. Для кожного журналу зберігається наступна інформація: назва журналу, інформація про періодичність виходу журналу, дата виходу журналу, тираж журналу. Для статті зберігається інформація про її назву, авторів, журнал, в якому вона опублікована та дата надходження статті в редакцію. По кожному з авторів зберігається інформація щодо його ПІБ, організації, в якій він працює.
- 12) Розробити структуру даних для зберігання інформації про кінотеатри міста. Для кінотеатру зберігається інформація: найменування кінотеатру, місткість (кількість місць), рік побудови, ранг кінотеатру (для перегляду відеофільмів, для перегляду широкоформатних фільмів, наявність стереоформатного обладнання, тощо).
- 13) Розробити структуру даних для зберігання інформації про абонементи бібліотеки. Бібліотека заробляє гроші, видаючи напрокат деякі книги, які наявні в невеликій кількості примірників. У кожній книзі, яка видається в прокат, є назва, автор, жанр. В залежності від цінності книги для кожної з них визначена заставна вартість (сума, яку вносить клієнт при взятті книги напрокат) і вартість прокату (сума, яку клієнт платить при поверненні книги, отримуючи назад заставу). Вартість прокату книги залежить не тільки від самої книги, а й від терміну її прокату. У бібліотеку звертаються читачі. Всі читачі реєструються в

картотеці, яка містить стандартні анкетні дані (ПІБ, адреса, телефон, категорія). Кожен читач може звертатися в бібліотеку кілька разів. Всі звернення читачів фіксуються, при цьому по кожному факту видачі книги (при наявності примірника) фіксується дата видачі та очікувана дата повернення.

- 14) Розробити структуру даних для зберігання інформації про тролейбусні маршрути міста. Для кожного маршруту зберігається інформація: найменування початкової і кінцевої зупинки, кількість тролейбусів на маршруті, час проїзду від початку маршруту до кінця, список номерів тролейбусів на маршруті.
- 15) Розробити структуру даних для зберігання інформації про бронювання номерів готелю. Номер має наступні характеристики: клас, кількість місць розміщення, додаткові опції: кондиціонер, дитяче ліжко, тощо. Клієнти бронюють номери визначеного типу на період (з дати по дату). Передбачити, що один той самий клієнт може забронювати декілька номерів на один й той самий період, або різні номери на різні періоди.
- 16) Розробити структуру даних для зберігання інформації про автомобілі. Для об'єктів зберігається наступна інформація: вантажний автомобіль - марка автомобіля, вантажопідйомність, дата випуску, дата капітального ремонту, державний номер, шифр автопарку; таксі - марка автомобіля, кількість посадкових місць, дата випуску, державний номер, шифр автопарку; автопарк - назва, адреса розміщення, площа для розміщення автомобілів, шифр
- 17) Розробити структуру даних для формування навантаження викладачів кафедри. Навчальна дисципліна повинна мати наступні атрибути: назву, ПІБ викладача, форму контролю, кількість годин, код спеціальності, курс викладання. Спеціальність характеризується назвою та кодом. Передбачити, що один викладач може викладати різні дисципліни. Дисципліна з одною і тою ж назвою може бути на різних спеціальностях.
- 18) Розробити структуру даних для зберігання інформації про робітників підприємства та їх заробітну платню по місяцях. По кожному з робітників зберігається наступна інформація: прізвище, ім'я, по-батькові, дата народження, табельний номер, реєстраційний номер облікової картки платника податків, освіта, спеціальність, дата початку роботи на підприємстві. В об'єкті «Заробітна платня по місяцях» зберігається інформація про заробітну плату співробітника по місяцях з початку його роботи на підприємстві. Обидва об'єкти зв'язані по реєстраційному номеру облікової картки платника податків.
- 19) Розробити структуру даних для зберігання інформації про успішність студентів по різних дисциплінам в розрізі семестрів та курсів.

Передбачити, що дисципліни можуть мати різні фіксовані форми контролю (екзамен, залік).

- 20) Розробити структуру даних для зберігання інформації про депозити населення в національній, так і в іноземній валюті в банку. Кожен вклад має свій строк, валюту, відсоткову ставку, крім того, початковий внесок по різних типах депозитів – різний. Крім того, банк може надавати кредити населенню в національній та іноземній валютах. Для кожного клієнта повинна зберігатися наступна інформація: ПІБ, код клієнту, реєстраційний номер облікової картки платника податків, контактні дані. Для депозитів – код депозиту, код клієнта, номер рахунка, сума вкладу, дата початку та завершення. Для кредитів – код кредиту, код клієнта, валюта, сума кредиту, дата видачі та повернення.
- 21) Розробити структуру даних для зберігання інформації про меню ресторану. Для об'єктів зберігається наступна інформація: продукт – назва, кількість калорій; блюдо – назва, продукти та їх кількість; меню – дата, блюдо, вартість.
- 22) Розробити структуру даних для зберігання інформації про акторів та їх фільмографію. Крім фільмів, актори можуть грати у театрі. Кінофільми характеризуються назвою, роком випуску, жанром та режисером. Спектаклі – назвою та жанром. По кожному актору необхідно зберігати інформацію: ПІБ, амплуа, рік народження. Врахувати той факт, що в деяких фільмах актори можуть виступати і у якості режисеру.
- 23) Розробити структуру даних для зберігання інформації про країни. Для об'єктів зберігається наступна інформація: країни – назва, континент, загальна площа, загальна кількість населення, столиця, валюта, мова, тощо; область – назва, площа, обласний центр; місто – назва, площа, населення; район міста – назва, адреса районної адміністрації, кількість жителів, площа.
- 24) Розробити структуру даних для зберігання інформації про морські перевезення. Для об'єктів зберігається наступна інформація: кораблі – назва, пасажиромісткість, пароплавство, рік побудови, тип (річкове, морське); пароплавство – назва, місце розташування (річка/море); річка – назва, довжина, максимальна ширина, кількість притоків; море – назва, площа. Врахувати наступне, що кожен корабель може експлуатуватися тільки на річці або тільки на морі. На річці і на морі можуть експлуатуватися кілька кораблів.
- 25) Розробити структуру даних для зберігання інформації про розклад пасажирських залізничних потягів. Для об'єктів зберігається наступна інформація: потяг – інвентарний номер потягу, ПІБ головного по потягу, номер потягу, кількість вагонів; вагон – потяг, тип вагону (плацкарт/купейний/спальний), кількість місць; розклад – місто звідки, місто куди, дата, потяг, час прибуття, час відправлення

