

Лабораторна робота № 4

Тема: LINQ TO XML

Мета: отримати навички при роботі з XML-файлами та LINQ

Практична частина

Створити XML-файл, що містить інформацію про продукти в інтернет-магазині (назва продукту, ціна, категорія, наявність на складі). Зчитати дані з XML-файлу та виконати наступні завдання з використанням Linq to XML:

1. Вивести на екран список продуктів з категорії «Електроніка» та ціною менше 1000 грн.
2. Змінити ціну продукту з назвою «Samsung Galaxy S20» на 15000 грн.
3. Видалити продукт з назвою «Xiaomi Redmi Note 8» з XML-файлу.
4. Додати новий продукт з назвою «Apple iPhone 14» з ціною 25000 грн та категорією «Мобільні телефони» та позначити його як в наявності.
5. В звіті відобразити скріншоти коду та заповненого XML-файлу до змін та після змін.

Завдання:

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

class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = System.Text.Encoding.Default;

        // Створення XML-файлу
        XElement productsXml = new XElement("products",
            new XElement("product",
                new XElement("name", "Samsung Galaxy S20"),
                new XElement("price", 12000),
                new XElement("category", "Мобільні телефони"),
                new XElement("inStock", true)
            ),
            new XElement("product",
                new XElement("name", "Xiaomi Redmi Note 8"),
                new XElement("price", 8000),
                new XElement("category", "Мобільні телефони"),
                new XElement("inStock", true)
            ),
            new XElement("product",
                new XElement("name", "Sony Headphones"),
                new XElement("price", 700),
                new XElement("category", "Електроніка"),
```

```

        new XElement("inStock", true)
    ),
    new XElement("product",
        new XElement("name", "Dell Laptop"),
        new XElement("price", 25000),
        new XElement("category", "Електроніка"),
        new XElement("inStock", false)
    )
);

// Збереження XML у файл
string filePathOriginal = "products.xml";
productsXml.Save(filePathOriginal);

// Збереження XML у файл
string filePathChange = "products2.xml";
productsXml.Save(filePathChange);

// Завдання 1: Вивести список продуктів з категорії «Електроніка» та ціною
менше 1000 грн
Console.WriteLine("Продукти з категорії «Електроніка» та ціною менше 1000
грн:");
var electronicsUnder1000 = productsXml.Elements("product")
    .Where(p => (string?)p.Element("category") == "Електроніка" &&
(int?)p.Element("price") < 1000);

foreach (var product in electronicsUnder1000)
{
    Console.WriteLine(product.Element("name")?.Value);
}

// Завдання 2: Змінити ціну продукту «Samsung Galaxy S20» на 15000 грн
var samsungProduct = productsXml.Elements("product")
    .FirstOrDefault(p => (string?)p.Element("name") == "Samsung Galaxy
S20");
if (samsungProduct != null)
{
    samsungProduct.Element("price")?.SetValue(15000);
}

// Завдання 3: Видалити продукт «Xiaomi Redmi Note 8»
var xiaomiProduct = productsXml.Elements("product")
    .FirstOrDefault(p => (string?)p.Element("name") == "Xiaomi Redmi Note
8");
xiaomiProduct?.Remove();

// Завдання 4: Додати новий продукт «Apple iPhone 14»
productsXml.Add(new XElement("product",
    new XElement("name", "Apple iPhone 14"),
    new XElement("price", 25000),
    new XElement("category", "Мобільні телефони"),
    new XElement("inStock", true)
));

// Збереження змін у файл
productsXml.Save(filePathChange);

Console.WriteLine("\nЗміни збережено у файл.");
}
}

```

На рисунку 4.1 зображений результат виконання програми.

```
Продукти з категорії «Електроніка» та ціною менше 1000 грн:  
Sony Headphones  
  
Зміни збережено у файл.
```

Рисунок 4.1 – Виконання завдання 1

На рисунку 4.2 зображене порівняння оригінального xml списку та видозміненого під час виконання завдань.

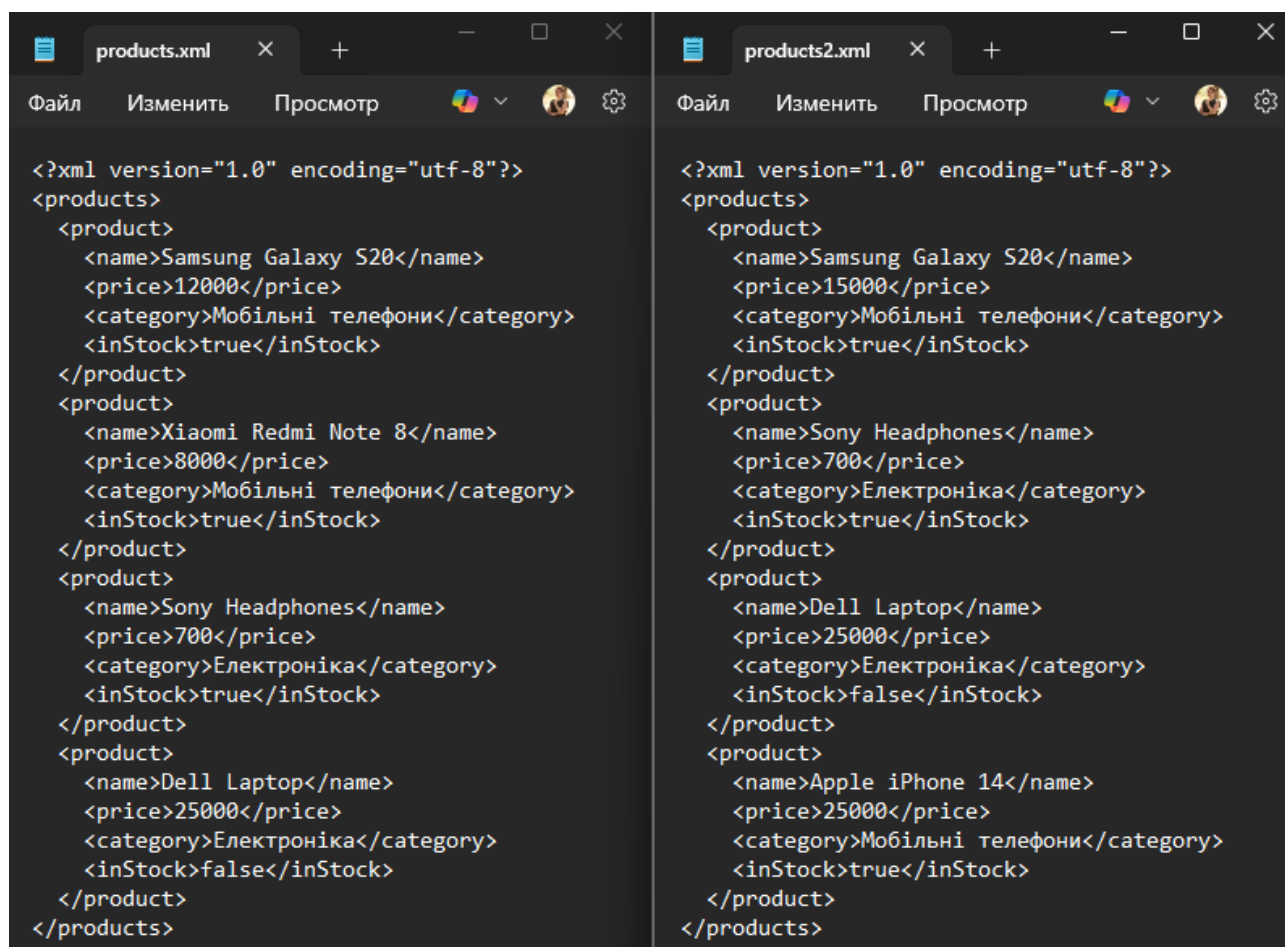


Рисунок 4.1 – Порівняння xml файлів

Контрольні питання:

1. Що таке XML?

XML (eXtensible Markup Language) — це розширювана мова розмітки, призначена для зберігання та передачі структурованих даних у текстовому форматі, що легко читається як людиною, так і машиною.

2. Які основні компоненти XML-документу?

- Елементи (elements) — основні блоки даних (<name>Value</name>).
- Атрибути (attributes) — додаткові дані в тегах (<user id="1" />).
- Кореневий елемент (root) — один головний елемент, що містить усі інші.
- Теги (tags) — відкриваючі і закриваючі (<tag> і </tag>).
- Текстовий вміст (text content) — дані всередині елементів.

3. Які основні методи LINQ to XML?

- XDocument.Load() — завантаження XML.
- XElement.Element() — доступ до одного вкладеного елемента.
- XElement.Elements() — доступ до колекції елементів.
- XElement.Attribute() — доступ до атрибутів.
- XDocument.Save() — збереження документа.

4. Як завантажити XML-документ з файлу за допомогою LINQ to XML?

```
XDocument doc = XDocument.Load("data.xml");
```

5. Які методи LINQ to XML використовуються для вибірки даних з XML-документу?

- Elements() — отримати всі вкладені елементи певного типу.
- Where() — фільтрація елементів за умовою.
- Select() — проєкція (витяг значень, атрибутів).
- Value — отримати текстове значення елемента.
- Attribute("name").Value — отримати значення атрибута.

6. Як змінити дані в XML-документі за допомогою LINQ to XML?

```
var doc = XDocument.Load("data.xml");  
var user = doc.Root.Element("User");  
user.Element("Name").Value = "Новий Ім'я";  
doc.Save("data.xml");
```