Кам’янець-Подільський національний університет імені Івана Огієнка

КАФЕДРА КОМП’ЮТЕРНИХ НАУК

Навчальна дисципліна «Об’єктно-орієнтоване програмування»

ЛАБОРАТОРНА РОБОТА #0112

Тема:

Робота з JSON та XML у C#

Варіант №3

**Виконала**:  
студентка 1-го курсу  
групи KNms1-B24  
Некрасова Ю.Д.

**Прийняв**:  
Слободянюк О.В.

Кам’янець-Подільський – 2025

1. **Короткі теоретичні відомості**

**Серіалізація в JSON. JsonSerializer**

Серіалізація — це процес перетворення об'єкта в формат, придатний для зберігання або передачі (наприклад, JSON). У .NET для цього використовується клас JsonSerializer з простору імен System.Text.Json. Методи Serialize() та SerializeAsync() дозволяють перетворювати об'єкти в JSON, а Deserialize() та DeserializeAsync() — з JSON в об'єкти. Для коректної десеріалізації об'єкт повинен мати публічний конструктор без параметрів або відповідність між назвами параметрів конструктора та властивостями JSON-об'єкта.

**XML-документи. Робота з XML за допомогою класів System.Xml**

У .NET для роботи з XML використовуються класи з простору імен System.Xml, зокрема:

* XmlDocument: представляє весь XML-документ.
* XmlNode: базовий клас для всіх типів вузлів XML.
* XmlElement: представляє елемент XML.
* XmlAttribute: представляє атрибут елемента.
* XmlText: представляє текстове значення елемента.

Ці класи дозволяють створювати, читати, змінювати та зберігати XML-документи.

**Модифікація XML-документа. XPath. LINQ to XML. Створення XML-документа**

Для модифікації XML-документа використовуються методи класу XmlNode, такі як AppendChild(), InsertAfter(), InsertBefore(), RemoveChild() тощо. Використовуючи ці методи, можна додавати, вставляти та видаляти елементи в XML-документі.

XPath — це мова запитів для вибору вузлів у XML-документі. Вона дозволяє знаходити елементи за їхніми іменами, атрибутами, значеннями тощо.

LINQ to XML — це технологія, що дозволяє працювати з XML-документами за допомогою LINQ-запитів. Вона забезпечує зручний та ефективний спосіб маніпулювання XML-даними.

Для створення нового XML-документа можна використовувати клас XDocument з простору імен System.Xml.Linq. Цей клас дозволяє створювати XML-документи програмно та зберігати їх у файл або потік.

**Вибірка елементів у LINQ to XML. Зміна XML-документа в LINQ to XML**

У LINQ to XML для вибірки елементів використовуються LINQ-запити. Ці запити дозволяють знаходити елементи за їхніми іменами, атрибутами, значеннями тощо. Зміна XML-документа в LINQ to XML здійснюється за допомогою методів, таких як Add(), Remove(), ReplaceWith() тощо.

**Серіалізація в XML. XmlSerializer**

Серіалізація в XML у .NET здійснюється за допомогою класу XmlSerializer. Цей клас дозволяє перетворювати об'єкти в XML та зберігати їх у файл або потік. Для серіалізації об'єкта необхідно створити екземпляр XmlSerializer, передавши йому тип об'єкта. Потім можна викликати методи Serialize() або Deserialize() для серіалізації або десеріалізації відповідно.

1. **Умова завдання 1 (Варіант 3)**

Створіть новий XML-документ, використовуючи класи з System.Xml. Додайте до нього інформацію про 10 студентів (ім'я, група (передбачити 2-3 назви груп), середній бал) та збережіть документ у файл students.xml.

1. **Послідовність виконання роботи**

Бібліотеки:



Основний клас Program. Створення документу:



Створення кореневого тегу «students»:



Масив з 10 імен студентів, які будуть записуються в XML:

Зображення, що містить текст, знімок екрана, Шрифт

Вміст, створений ШІ, може бути неправильним.

Масив з назв груп, які призначатимуться студентам:



Генерація рандомного середнього балу за 100-бальною шкалою (від 60 до 100 балів):

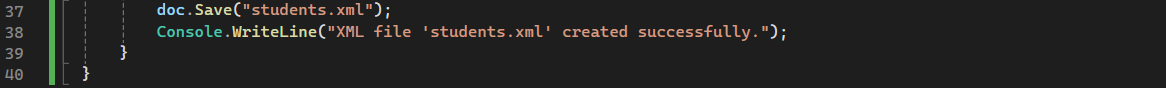


Цикл для створення нового запису про студента, його ім’я, групу та середній бал:

Зображення, що містить текст, програмне забезпечення, Мультимедійне програмне забезпечення, Шрифт

Вміст, створений ШІ, може бути неправильним.

Збереження документу і виведення відповідного повідомлення:



1. **Код програми**

using System;

using System.Xml;

class Program

{

static void Main()

{

XmlDocument doc = new XmlDocument();

doc.AppendChild(doc.CreateElement("students"));

string[] names = {

"Andrii Shevchenko",

"Olena Ivanenko",

"Taras Petrenko",

"Iryna Bondarenko",

"Maksym Koval",

"Kateryna Zhuk",

"Dmytro Tkachenko",

"Oksana Melnyk",

"Yurii Polishchuk",

"Natalia Kravets"

};

string[] groups = { "KN1-B24", "M1-B24", "F1-B24" };

Random rnd = new Random();

for (int i = 0; i < 10; i++)

{

XmlElement student = doc.CreateElement("student");

XmlElement name = doc.CreateElement("name");

name.InnerText = names[i];

student.AppendChild(name);

XmlElement group = doc.CreateElement("group");

group.InnerText = groups[i % groups.Length];

student.AppendChild(group);

XmlElement average = doc.CreateElement("averageScore");

average.InnerText = rnd.Next(60, 101).ToString();

student.AppendChild(average);

doc.DocumentElement.AppendChild(student);

}

doc.Save("students.xml");

Console.WriteLine("XML file 'students.xml' created successfully.");

}

}

1. **Приклад виконання програми**

Зображення, що містить знімок екрана, програмне забезпечення, текст, Мультимедійне програмне забезпечення

Вміст, створений ШІ, може бути неправильним.

**Зображення, що містить текст, знімок екрана, програмне забезпечення

Вміст, створений ШІ, може бути неправильним.**

1. **Умова завдання 2 (Варіант 3)**

Використовуючи XPath, знайдіть усіх студентів, що навчаються разом у заданій групі та виведіть їхні прізвища.

1. **Послідовність виконання роботи**

Бібліотека:



Основний клас Program. Створення порожнього об’єкту XmlDocument і завантаження XML-файлу students.xml:

Зображення, що містить знімок екрана, Мультимедійне програмне забезпечення

Вміст, створений ШІ, може бути неправильним.

Введення і зчитування назви групи:



XPath-запит, який знаходить всіх студентів заданої групи:



Виведення прізвищ відповідних студентів:

Зображення, що містить текст, Мультимедійне програмне забезпечення, програмне забезпечення, Графічний редактор

Вміст, створений ШІ, може бути неправильним.

1. **Код програми**

using System;

using System.Xml;

class Program

{

static void Main()

{

XmlDocument doc = new XmlDocument();

doc.Load("students.xml");

Console.Write("Enter group name (KN1-B24 | M1-B24 | F1-B24) ");

string groupName = Console.ReadLine();

string xpath = $"/students/student[group='{groupName}']";

XmlNodeList studentNodes = doc.SelectNodes(xpath);

Console.WriteLine($"\nStudents in group {groupName}:");

foreach (XmlNode student in studentNodes)

{

string fullName = student.SelectSingleNode("name").InnerText;

string lastName = fullName.Split(' ')[1];

Console.WriteLine(lastName);

}

}

}

1. **Приклад виконання програми**

**Зображення, що містить знімок екрана, текст, програмне забезпечення, Мультимедійне програмне забезпечення

Вміст, створений ШІ, може бути неправильним.**