### 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 в C#

Варіант №1

**Виконав**:  
студент 1-го курсу  
групи KNms1-B24  
Білик Я. Ю.

**Прийняв**:  
старший викладач,  
Слободянюк О. В.

### Кам’янець-Подільський – 2025

1. **Короткі теоретичні відомості.**

Серіалізація та десеріалізація в C#

Концепція серіалізації та десеріалізації використовується щоразу, коли дані, що стосуються об’єктів, потрібно надіслати з однієї програми в іншу. Серіалізація використовується для експорту даних програми у файл. Потім цільова програма використовує десеріалізацію, щоб витягти дані з програми для подальшого використання.

Серіалізація — це концепція, за якої об’єкти класу C# записуються або серіалізуються у файли. Припустімо, у вас є клас C# під назвою Tutorial. І клас має 2 властивості ID та назву Tutorials.

Серіалізацію можна використовувати для безпосереднього запису властивостей даних класу Tutorial у файл. Десеріалізація використовується для читання даних із файлу та повторного створення об’єкта Tutorial.

JSON (JavaScript Object Notation) є одним із найбільш популярних форматів для зберігання та передачі даних. І платформа .NET надає функціонал для роботи із JSON.

Основна функціональність роботи з JSON зосереджена у просторі імен System.Text.Json. Ключовим типом є клас JsonSerializer, який дозволяє серіалізувати об’єкт в json і, навпаки, десеріалізувати код json в об’єкт C#.

Для збереження об’єкта в json у класі JsonSerializer визначено статичний метод Serialize() та його асинхронний двійник SerializeAsyc(), які мають низку перевантажених версій.

1. **Повні умови завдань.**

Задача 1

Створіть клас Product з властивостями Name (string), Price (decimal) та Category (string). Створіть екземпляр цього класу та серіалізуйте його в JSONрядок, використовуючи JsonSerializer. Виведіть отриманий JSON-рядок на консоль.

Задача 2

Створіть XML-документ, що описує книгу (назва, автор, рік видання). Збережіть цей документ у файл з назвою book.xml. Прочитайте XML-документ book.xml, використовуючи класи з System.Xml. Виведіть на консоль назву та автора книги.

1. **Лістинги програм.**

Задача 1

using System;

using System.Text.Json;

public class Product

{

public string Name { get; set; }

public decimal Price { get; set; }

public string Category { get; set; }

}

class Program

{

static void Main()

{

Console.OutputEncoding = System.Text.Encoding.UTF8;

Product prod = new Product

{

Name = "Laptop",

Price = 19999.99m,

Category = "Electronics"

};

string json = JsonSerializer.Serialize(prod, new JsonSerializerOptions { WriteIndented = true });

Console.WriteLine("JSON-представлення об'єкта Product:");

Console.WriteLine(json);

}

}

Задача 2

using System;

using System.Xml;

class Program

{

static void Main()

{

Console.OutputEncoding = System.Text.Encoding.UTF8;

string xmlFilePath = "book.xml";

// Створення XML-документу

XmlDocument doc = new XmlDocument();

XmlElement root = doc.CreateElement("book");

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

title.InnerText = "Пригоди Шерлока Холмса";

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

author.InnerText = "Артур Конан Дойл";

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

year.InnerText = "1892";

root.AppendChild(title);

root.AppendChild(author);

root.AppendChild(year);

doc.AppendChild(root);

doc.Save(xmlFilePath);

Console.WriteLine($"XML-файл збережено у '{xmlFilePath}'");

// Зчитування з XML

XmlDocument readDoc = new XmlDocument();

readDoc.Load(xmlFilePath);

string readTitle = readDoc.DocumentElement.SelectSingleNode("title")?.InnerText;

string readAuthor = readDoc.DocumentElement.SelectSingleNode("author")?.InnerText;

Console.WriteLine("\nІнформація з XML-файлу:");

Console.WriteLine($"Назва книги: {readTitle}");

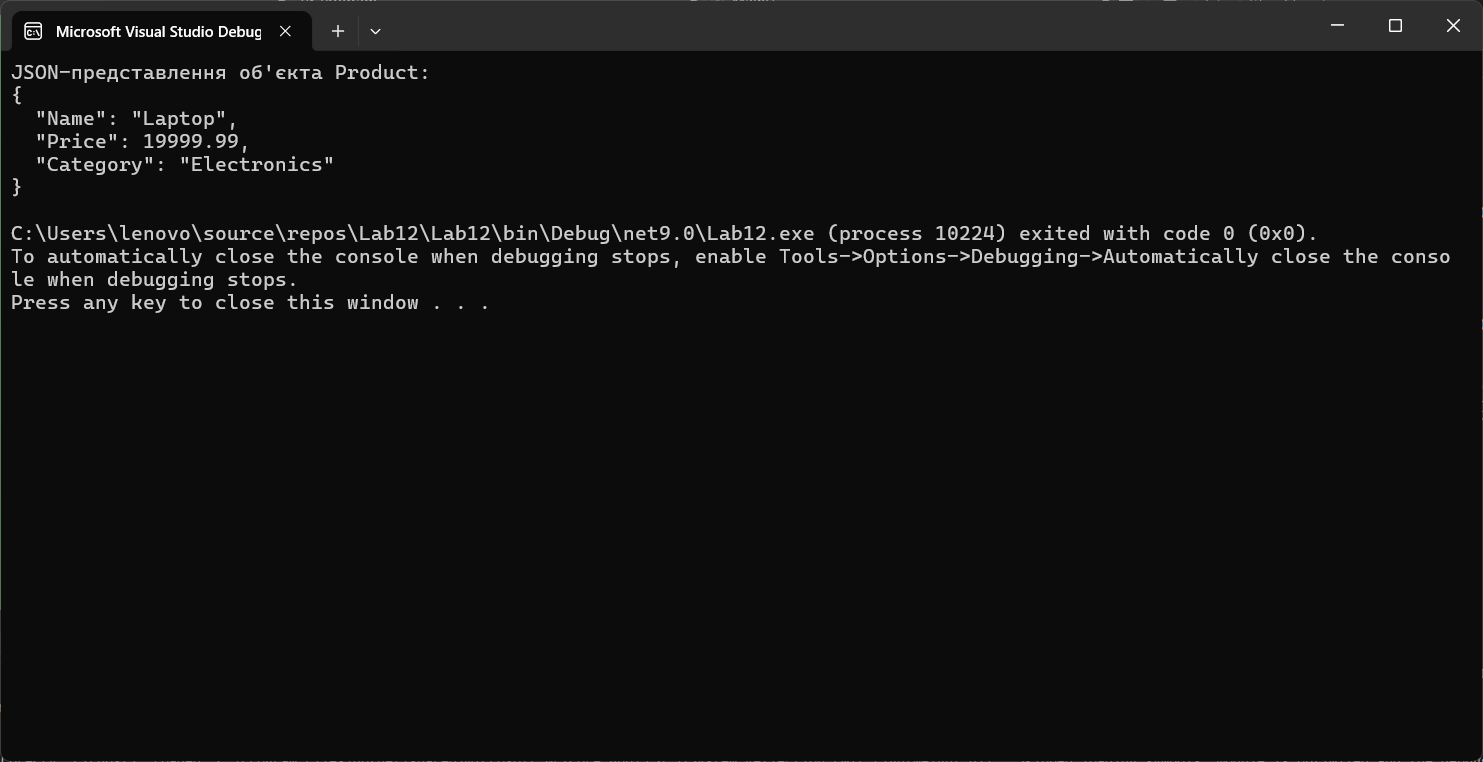
Console.WriteLine($"Автор: {readAuthor}");

}

}

1. **Результати роботи програм.**

Задача 1



Задача 2

