

Лабораторна робота №3

Тема: Структурні шаблони

Мета роботи: навчитися реалізовувати структурні шаблони проєктування Адаптер, Декоратор, Міст, Компонувальник, Проксі, Легковаговик

Хід роботи

Завдання 1: Адаптер.

1. Створіть клас `Logger`, який буде мати методи `Log()`, `Error()`, `Warn()`, які виводять повідомлення в консоль різними кольорами (зеленим, червоним і оранжевим відповідно).
2. Створіть клас `FileWriter` з методами `Write()`, `WriteLine()`.
3. За допомогою шаблону Адаптер створіть файловий логер.
4. Покажіть правильність роботи свого коду запусивши його в головному методі програми

Лістинг програми:

```
using System;  
using System.IO;
```

```
namespace LoggerAdapterExample  
{  
    // Клас Logger з методами Log(), Error(), Warn()  
    public class Logger  
    {  
        public void Log(string message)  
        {  
            Console.ForegroundColor = ConsoleColor.Green;  
            Console.WriteLine("LOG: " + message);  
            Console.ResetColor();  
        }  
  
        public void Error(string message)  
        {  
            Console.ForegroundColor = ConsoleColor.Red;  
            Console.WriteLine("ERROR: " + message);  
            Console.ResetColor();  
        }  
  
        public void Warn(string message)  
        {  
            Console.ForegroundColor = ConsoleColor.DarkYellow;  
            Console.WriteLine("WARN: " + message);  
            Console.ResetColor();  
        }  
    }  
  
    // Клас FileWriter з методами Write(), WriteLine()  
    public class FileWriter  
    {  
        private readonly string _filePath;  
  
        public FileWriter(string filePath)  
        {  
            _filePath = filePath;  
        }  
    }  
}
```

					ДУ «Житомирська політехніка».23.121.27.000 – Лр3			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Свистанюк Н.О.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Фант М.О.						1
Керівник								14
Н. контр.							ФІКТ Гр. ІПЗ-22-3[2]	
Зав. каф.								

```

    public void Write(string message)
    {
        File.AppendAllText(_filePath, message);
    }

    public void WriteLine(string message)
    {
        File.AppendAllText(_filePath, message + Environment.NewLine);
    }
}

// Інтерфейс IFileLogger
public interface IFileLogger
{
    void Log(string message);
    void Error(string message);
    void Warn(string message);
}

// Адаптер FileLoggerAdapter
public class FileLoggerAdapter : IFileLogger
{
    private readonly FileWriter _fileWriter;

    public FileLoggerAdapter(FileWriter fileWriter)
    {
        _fileWriter = fileWriter;
    }

    public void Log(string message)
    {
        _fileWriter.WriteLine("LOG: " + message);
    }

    public void Error(string message)
    {
        _fileWriter.WriteLine("ERROR: " + message);
    }

    public void Warn(string message)
    {
        _fileWriter.WriteLine("WARN: " + message);
    }
}

class Program
{
    static void Main(string[] args)
    {
        // Створення об'єкту Logger
        var logger = new Logger();
        logger.Log("This is a log message.");
        logger.Error("This is an error message.");
        logger.Warn("This is a warning message.");

        // Створення об'єкту FileWriter
        string filePath = "log.txt";
        var fileWriter = new FileWriter(filePath);

        // Створення об'єкту FileLoggerAdapter
        IFileLogger fileLogger = new FileLoggerAdapter(fileWriter);
        fileLogger.Log("This is a log message.");
        fileLogger.Error("This is an error message.");
        fileLogger.Warn("This is a warning message.");
    }
}

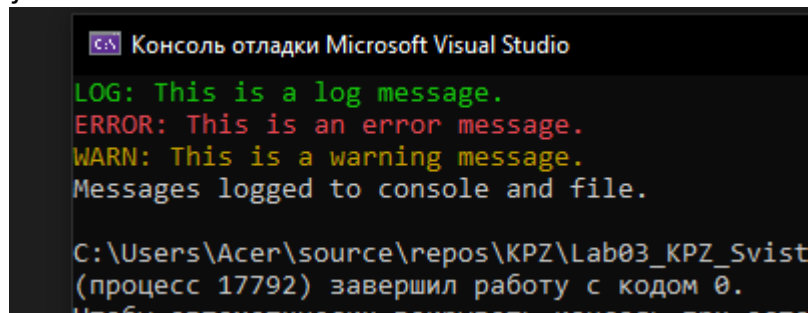
```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

        Console.WriteLine("Messages logged to console and file.");
    }
}

```



Мал.1. Результат виконання

Завдання 2: Декоратор.

1. Ви розробляєте РПГ гру. Створіть класи героїв Warrior, Mage, Palladin.
2. Для героїв створіть інвентар (одяг, зброю, артефакти), який може підходити будь-якому типу героїв, у вигляді декораторів.
3. Важливою вимогою є можливість використання декількох екземплярів інвентаря на герої одночасно.
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

Лістинг програми:

```

using System;

namespace RPGDecoratorExample
{
    // Базовий інтерфейс для героя
    public interface IHero
    {
        string Description { get; }
        int GetPower();
    }

    // Реалізація героїв
    public class Warrior : IHero
    {
        public string Description => "Warrior";
        public int GetPower() => 10;
    }

    public class Mage : IHero
    {
        public string Description => "Mage";
        public int GetPower() => 8;
    }

    public class Paladin : IHero
    {
        public string Description => "Paladin";
        public int GetPower() => 9;
    }

    // Базовий клас для декораторів інвентарю
    public abstract class HeroDecorator : IHero
    {
        protected IHero _hero;
    }
}

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – ЛрЗ	Арк.
		Фант М.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public HeroDecorator(IHero hero)
    {
        _hero = hero;
    }

    public virtual string Description => _hero.Description;
    public virtual int GetPower() => _hero.GetPower();
}

// Конкретні декоратори інвентарю
public class Armor : HeroDecorator
{
    public Armor(IHero hero) : base(hero) { }

    public override string Description => _hero.Description + ", Armor";
    public override int GetPower() => _hero.GetPower() + 5;
}

public class Sword : HeroDecorator
{
    public Sword(IHero hero) : base(hero) { }

    public override string Description => _hero.Description + ", Sword";
    public override int GetPower() => _hero.GetPower() + 7;
}

public class Amulet : HeroDecorator
{
    public Amulet(IHero hero) : base(hero) { }

    public override string Description => _hero.Description + ", Amulet";
    public override int GetPower() => _hero.GetPower() + 3;
}

class Program
{
    static void Main(string[] args)
    {
        // Створення героя-воїна
        IHero warrior = new Warrior();
        Console.WriteLine($"Description: {warrior.Description}, Power:
{warrior.GetPower()}");

        // Додавання інвентарю герою
        warrior = new Armor(warrior);
        warrior = new Sword(warrior);
        warrior = new Amulet(warrior);

        Console.WriteLine($"Description: {warrior.Description}, Power:
{warrior.GetPower()}");

        // Створення героя-мага
        IHero mage = new Mage();
        Console.WriteLine($"Description: {mage.Description}, Power:
{mage.GetPower()}");

        // Додавання інвентарю герою
        mage = new Amulet(mage);
        mage = new Armor(mage);

        Console.WriteLine($"Description: {mage.Description}, Power:
{mage.GetPower()}");

        // Створення героя-паладіна
        IHero paladin = new Paladin();

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

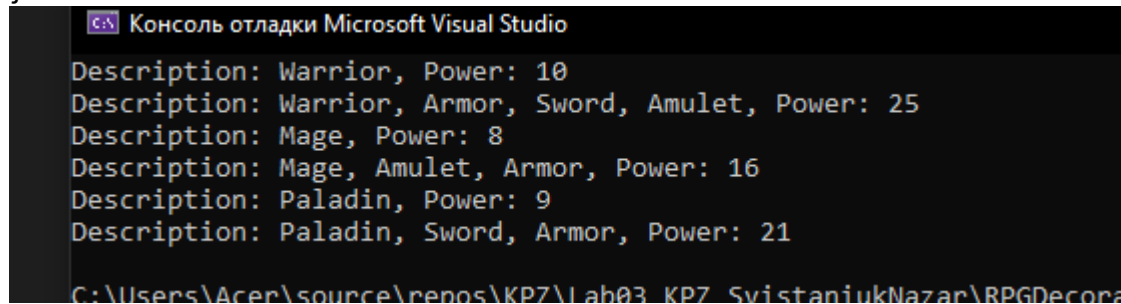
```

        Console.WriteLine($"Description: {paladin.Description}, Power:
{paladin.GetPower()}");

        // Додавання інвентарю герою
        paladin = new Sword(paladin);
        paladin = new Armor(paladin);

        Console.WriteLine($"Description: {paladin.Description}, Power:
{paladin.GetPower()}");
    }
}

```



```

Консоль отладки Microsoft Visual Studio
Description: Warrior, Power: 10
Description: Warrior, Armor, Sword, Amulet, Power: 25
Description: Mage, Power: 8
Description: Mage, Amulet, Armor, Power: 16
Description: Paladin, Power: 9
Description: Paladin, Sword, Armor, Power: 21
C:\Users\Acer\source\repos\KP7\Lab03_KP7_SvistaniukNazar\RP6Decor

```

Мал.2. Результат виконання

Завдання 3: Міст.

1. Ви працюєте над графічним редактором. Створіть базовий клас Shape.
2. Створіть дочірні до Shape класи, Circle, Square, Triangle.
3. За допомогою шаблону Міст додайте можливість рендерингу кожної з фігур як векторної або растрової графіки (вивівши відповідне повідомлення у консоль, наприклад "Drawing Triangle as pixels").
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

Лістинг програми:

```

using System;

namespace BridgePatternExample
{
    // Інтерфейс IRenderer
    public interface IRenderer
    {
        void Render(string shape);
    }

    // Конкретні реалізації інтерфейсу IRenderer
    public class VectorRenderer : IRenderer
    {
        public void Render(string shape)
        {
            Console.WriteLine($"Drawing {shape} as vector graphics.");
        }
    }

    public class RasterRenderer : IRenderer
    {
        public void Render(string shape)
        {
            Console.WriteLine($"Drawing {shape} as pixels.");
        }
    }

    // Базовий клас Shape

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public abstract class Shape
{
    protected IRenderer _renderer;

    protected Shape(IRenderer renderer)
    {
        _renderer = renderer;
    }

    public abstract void Draw();
}

// Конкретні класи фігур
public class Circle : Shape
{
    public Circle(IRenderer renderer) : base(renderer) { }

    public override void Draw()
    {
        _renderer.Render("Circle");
    }
}

public class Square : Shape
{
    public Square(IRenderer renderer) : base(renderer) { }

    public override void Draw()
    {
        _renderer.Render("Square");
    }
}

public class Triangle : Shape
{
    public Triangle(IRenderer renderer) : base(renderer) { }

    public override void Draw()
    {
        _renderer.Render("Triangle");
    }
}

class Program
{
    static void Main(string[] args)
    {
        // Використання VectorRenderer
        IRenderer vectorRenderer = new VectorRenderer();
        Shape vectorCircle = new Circle(vectorRenderer);
        Shape vectorSquare = new Square(vectorRenderer);
        Shape vectorTriangle = new Triangle(vectorRenderer);

        vectorCircle.Draw();
        vectorSquare.Draw();
        vectorTriangle.Draw();

        // Використання RasterRenderer
        IRenderer rasterRenderer = new RasterRenderer();
        Shape rasterCircle = new Circle(rasterRenderer);
        Shape rasterSquare = new Square(rasterRenderer);
        Shape rasterTriangle = new Triangle(rasterRenderer);

        rasterCircle.Draw();
        rasterSquare.Draw();
    }
}

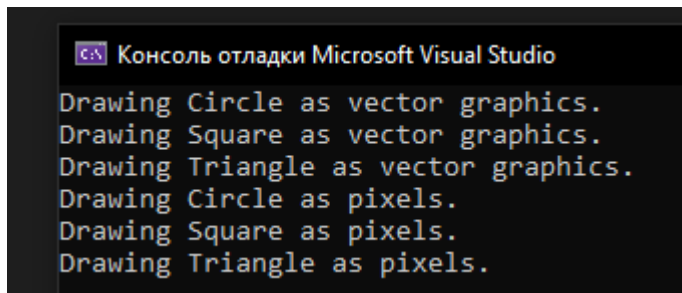
```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        rasterTriangle.Draw();
    }
}

```



Мал.3. Результат виконання

Завдання 4: Проксі.

1. Створіть клас SmartTextReader, який вміє читати вміст текстового файлу і перетворювати його на двовірний масив якому зовнішній масив відповідає рядкам тексту, а вкладені масиви відповідають символам у відповідному рядку.
2. Створіть проксі для SmartTextReader з логуванням SmartTextChecker, який буде виводити інформацію про успішне відкриття, прочитання і закриття файлу, а також буде виводити загальну кількість рядків і символів у прочитаному тексті.
3. Створіть проксі для SmartTextReader з обмеженням доступу до певних файлів SmartTextReaderLocker. Цей клас в конструкторі приймає регулярний вираз, по якому лімітується доступ до певної групи файлів. Якщо клієнт викликати метод для прочитання такого лімітованого файлу, замість прочитання файлу в консоль має виводитися повідомлення “Access denied!”.
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

Лістинг програми:

```

using System;
using System.IO;
using System.Text.RegularExpressions;

namespace ProxyPatternExample
{
    // Клас SmartTextReader, який читає файл і перетворює його на двовірний масив
    public class SmartTextReader
    {
        private readonly string _filePath;

        public SmartTextReader(string filePath)
        {
            _filePath = filePath;
        }

        public virtual char[][] ReadFile()
        {
            string[] lines = File.ReadAllLines(_filePath);
            char[][] result = new char[lines.Length][];
            for (int i = 0; i < lines.Length; i++)
            {
                result[i] = lines[i].ToCharArray();
            }
            return result;
        }
    }
}

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        public string GetFilePath()
        {
            return _filePath;
        }
    }

    // Проксі з логуванням SmartTextChecker
    public class SmartTextChecker : SmartTextReader
    {
        private readonly SmartTextReader _reader;

        public SmartTextChecker(SmartTextReader reader) : base(reader.GetFilePath())
        {
            _reader = reader;
        }

        public override char[][] ReadFile()
        {
            Console.WriteLine("Opening file...");
            char[][] content = _reader.ReadFile();
            Console.WriteLine("File read successfully!");
            Console.WriteLine($"Total lines: {content.Length}");
            int totalChars = 0;
            foreach (var line in content)
            {
                totalChars += line.Length;
            }
            Console.WriteLine($"Total characters: {totalChars}");
            Console.WriteLine("Closing file...");
            return content;
        }
    }

    // Проксі з обмеженням доступу SmartTextReaderLocker
    public class SmartTextReaderLocker : SmartTextReader
    {
        private readonly SmartTextReader _reader;
        private readonly Regex _regex;

        public SmartTextReaderLocker(SmartTextReader reader, string pattern) :
        base(reader.GetFilePath())
        {
            _reader = reader;
            _regex = new Regex(pattern, RegexOptions.IgnoreCase);
        }

        public override char[][] ReadFile()
        {
            if (_regex.IsMatch(_reader.GetFilePath()))
            {
                Console.WriteLine("Access denied!");
                return null;
            }
            return _reader.ReadFile();
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            string filePath = "sample.txt";
            // Створення реального читача файлів
            SmartTextReader reader = new SmartTextReader(filePath);

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

// Створення проксі з логуванням
SmartTextChecker checker = new SmartTextChecker(reader);
checker.ReadFile();

// Створення проксі з обмеженням доступу
string restrictedPattern = @"^restricted.*";
SmartTextReaderLocker locker = new SmartTextReaderLocker(reader,
restrictedPattern);

Console.WriteLine("\nTrying to read a non-restricted file:");
locker.ReadFile();

// Зміна шляху файлу для перевірки обмеження доступу
SmartTextReader restrictedReader = new
SmartTextReader("restricted_sample.txt");
SmartTextReaderLocker restrictedLocker = new
SmartTextReaderLocker(restrictedReader, restrictedPattern);

Console.WriteLine("\nTrying to read a restricted file:");
restrictedLocker.ReadFile();
    }
}
}

```

Мал.4. Результат виконання

Завдання 5: Компонувальник. 1. Вам потрібно створити власну мову розмітки LightHTML. 2. Кожен елемент розмітки має наслідувати клас LightNode. 3. Створіть два дочірніх класи від LightNode: LightElementNode, LightTextNode. 4. LightTextNode може містити лише текст. 5. LightElementNode може містити будь-які LightNode. LightElementNode повинен мати інформацію про назву тега, його тип відображення (блочний чи рядковий), тип закриття (одиночний тег, як **Помилка! Не вказано ім'я файлу.** чи з закриваючим тегом) список CSS класів, кількість дочірніх елементів, а також має бути можливість виводити на екран його outerHTML і innerHTML. 6. За допомогою своєї мови розмітки виведіть в консоль елемент сторінки на Ваш вибір (наприклад якусь таблицю, список тощо). 7. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

Лістинг програми:

```

using System;
using System.Collections.Generic;
using System.Text;

namespace LightHTML
{
    // Базовий клас для всіх елементів розмітки
    public abstract class LightNode
    {
        public abstract string OuterHTML { get; }
        public abstract string InnerHTML { get; }
    }

    // Клас для текстових вузлів
    public class LightTextNode : LightNode
    {
        private readonly string _text;

        public LightTextNode(string text)

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – ЛрЗ	Арк.
		Фант М.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    {
        _text = text;
    }

    public override string OuterHTML => _text;
    public override string InnerHTML => _text;
}

// Типи відображення елементів
public enum DisplayType
{
    Block,
    Inline
}

// Типи закриття тегів
public enum ClosingType
{
    SelfClosing,
    Normal
}

// Клас для елементів розмітки
public class LightElementNode : LightNode
{
    private readonly string _tagName;
    private readonly DisplayType _displayType;
    private readonly ClosingType _closingType;
    private readonly List<string> _cssClasses = new List<string>();
    private readonly List<LightNode> _children = new List<LightNode>();

    public LightElementNode(string tagName, DisplayType displayType, ClosingType
closingType)
    {
        _tagName = tagName;
        _displayType = displayType;
        _closingType = closingType;
    }

    public void AddClass(string className)
    {
        _cssClasses.Add(className);
    }

    public void AddChild(LightNode child)
    {
        _children.Add(child);
    }

    public override string OuterHTML
    {
        get
        {
            var sb = new StringBuilder();
            sb.Append($"<{_tagName}>");

            if (_cssClasses.Count > 0)
            {
                sb.Append($" class="{string.Join(" ", _cssClasses)}\"");
            }

            if (_closingType == ClosingType.SelfClosing)
            {
                sb.Append(" />");
            }
        }
    }
}

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        else
        {
            sb.Append(">");
            foreach (var child in _children)
            {
                sb.Append(child.OuterHTML);
            }
            sb.Append($"</{_tagName}>");
        }

        return sb.ToString();
    }
}

public override string InnerHTML
{
    get
    {
        var sb = new StringBuilder();
        foreach (var child in _children)
        {
            sb.Append(child.OuterHTML);
        }
        return sb.ToString();
    }
}

class Program
{
    static void Main(string[] args)
    {
        // Створення елементів розмітки
        LightElementNode div = new LightElementNode("div", DisplayType.Block,
ClosingType.Normal);
        div.AddClass("container");

        LightElementNode header = new LightElementNode("h1", DisplayType.Block,
ClosingType.Normal);
        header.AddChild(new LightTextNode("Welcome to LightHTML!"));

        LightElementNode paragraph = new LightElementNode("p", DisplayType.Block,
ClosingType.Normal);
        paragraph.AddChild(new LightTextNode("This is an example paragraph."));

        LightElementNode link = new LightElementNode("a", DisplayType.Inline,
ClosingType.Normal);
        link.AddClass("link");
        link.AddChild(new LightTextNode("Click here"));

        LightElementNode image = new LightElementNode("img", DisplayType.Inline,
ClosingType.SelfClosing);
        image.AddClass("image");

        LightElementNode ul = new LightElementNode("ul", DisplayType.Block,
ClosingType.Normal);
        ul.AddClass("item-list");

        LightElementNode li1 = new LightElementNode("li", DisplayType.Block,
ClosingType.Normal);
        li1.AddClass("item");
        li1.AddChild(new LightTextNode("Item 1"));

        LightElementNode li2 = new LightElementNode("li", DisplayType.Block,
ClosingType.Normal);

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

li2.AddClass("item");
li2.AddChild(new LightTextNode("Item 2"));

ul.AddChild(li1);
ul.AddChild(li2);

div.AddChild(header);
div.AddChild(paragraph);
div.AddChild(link);
div.AddChild(image);
div.AddChild(ul);

// Виведення розмітки в консоль
Console.WriteLine(div.OuterHTML);
Console.WriteLine("\nInner HTML:");
Console.WriteLine(div.InnerHTML);

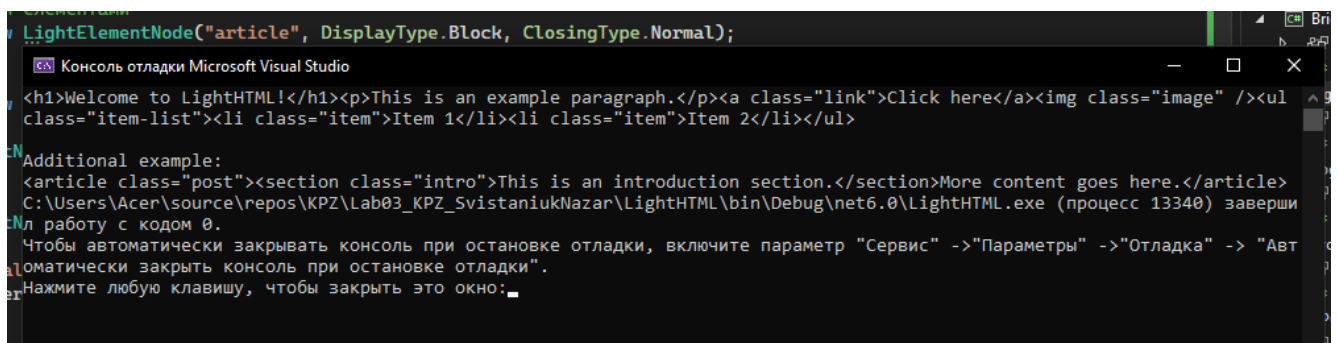
// Додатковий приклад з іншими елементами
LightElementNode article = new LightElementNode("article", DisplayType.Block,
ClosingType.Normal);
article.AddClass("post");

LightElementNode section = new LightElementNode("section", DisplayType.Block,
ClosingType.Normal);
section.AddClass("intro");
section.AddChild(new LightTextNode("This is an introduction section.));

article.AddChild(section);
article.AddChild(new LightTextNode("More content goes here.));

Console.WriteLine("\nAdditional example:");
Console.WriteLine(article.OuterHTML);
}
}
}

```



Мал.5. Результат виконання

Завдання 6: Легковаговик. 1. За допомогою свого LightHTML з завдання 1 перетворіть текст книги в HTML верстку за такими правилами: а. Перший рядок має бути елементом

б. Якщо в рядку менше 20 символів - це має бути елемент

с. Якщо рядок починається з пробільного символу - це має бути

д. В будь-якому іншому випадку - елемент

2. Покажіть скільки займає все дерево Вашої верстки, коли воно повністю утримується в пам'яті процесу.

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Використайте Легковаговик на Ваших класах HTML елементів, щоб зменшити споживання пам'яті.

4. Покажіть правильність роботи свого коду запусивши його в головному методі програми.

Лістинг програми:

```
using System;
using System.IO;
using System.Text;
using System.Runtime.InteropServices;

class Program
{
    class HtmlDocument
    {
        public HtmlElement Body { get; set; }
    }

    class HtmlElement
    {
        public string TagName { get; set; }
        public string InnerText { get; set; }
        public HtmlElement Parent { get; set; }
        public HtmlElement NextSibling { get; set; }
        public HtmlElement PreviousSibling { get; set; }
        public HtmlElement FirstChild { get; set; }
        public HtmlElement LastChild { get; set; }

        public void AppendChild(HtmlElement element)
        {
            if (FirstChild == null)
            {
                FirstChild = element;
                LastChild = element;
            }
            else
            {
                LastChild.NextSibling = element;
                element.PreviousSibling = LastChild;
                LastChild = element;
            }
            element.Parent = this;
        }
    }

    static void Main()
    {
        string text = File.ReadAllText("input.txt");
        var document = new HtmlDocument();
        document.Body = new HtmlElement { TagName = "body" };

        int h1Counter = 0;
        foreach (var line in text.Split('\n'))
        {
            HtmlElement element;
            if (h1Counter == 0)
            {
                element = new HtmlElement { TagName = "h1" };
                h1Counter++;
            }
            else if (line.Length < 20)
            {
                element = new HtmlElement { TagName = "h2" };
            }
        }
    }
}
```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        else if (char.IsWhiteSpace(line[0]))
        {
            element = new HtmlElement { TagName = "blockquote" };
        }
        else
        {
            element = new HtmlElement { TagName = "p" };
        }
        element.InnerText = line;
        document.Body.AppendChild(element);
    }

    Console.WriteLine($"Використання пам'яті до оптимізації:
{GC.GetTotalMemory(false)} байт");

    // Оптимізація з використанням GarbageCollector
    GC.Collect();

    Console.WriteLine($"Використання пам'яті після оптимізації:
{GC.GetTotalMemory(false)} байт");

    Console.WriteLine("HTML-верстка:");
    Console.WriteLine(GetHtml(document));
}

static string GetHtml(HtmlDocument document)
{
    var sb = new StringBuilder();
    sb.AppendLine("<html>");
    sb.AppendLine("  <body>");
    GetHtml(document.Body, sb);
    sb.AppendLine("  </body>");
    sb.AppendLine("</html>");
    return sb.ToString();
}

static void GetHtml(HtmlElement element, StringBuilder sb)
{
    sb.Append($"    <{element.TagName}>");
    sb.AppendLine(element.InnerText);
    sb.Append($"    </{element.TagName}>");
    if (element.NextSibling != null)
    {
        GetHtml(element.NextSibling, sb);
    }
}
}

```

		Свистанюк Н.О.			ДУ «Житомирська політехніка».23.121.27.000 – Лр3	Арк.
		Фант М.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		