

## Лабораторна робота №4

### Тема: Структурні шаблони

**Мета роботи:** навчитися реалізовувати структурні шаблони проектування **Адаптер, Декоратор, Фасад, Міст**

### Завдання на лабораторну роботу

#### 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 запустивши його в головному методі програми

Лістинг програми:

```
var Adapter = new Adapter();
Adapter.Log();
Adapter.WriteLine("1234");
Adapter.Error();
Adapter.Write("5678");
Adapter.Warn();
Adapter.Write("910");

class Logger
{
    public void Log()
    {
        Console.ForegroundColor = ConsoleColor.Green;
    }

    public void Error()
    {
        Console.ForegroundColor = ConsoleColor.Yellow;
    }
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.14.000 – Лр.4							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Іглінський І.Ю.			Звіт з лабораторної роботи №2				Літ.	Арк.	Аркуші	
Перевір.		Левківський В.Л.									1	9
Реценз.									ФІКТ, гр. ІПЗ-21-4			
Н. Контр.												
Зав.каф.												

```

    public void Warn()
    {
        Console.ForegroundColor = ConsoleColor.Red;
    }
}

class FileWriter:Logger
{
    public void Write(string text)
    {
        Console.Write(text);
    }

    public void WriteLine(string text)
    {
        Console.WriteLine(text);
    }
}

class Adapter : FileWriter
{
}

```

```

ter_Task_1.exe
1234
5678910
Process finish

```

Рис. 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.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.14.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

## Лістинг програми:

```
var hero = new Warrior("Sven");//hero Sven(DOTA 2)
var sword = new Item("Sword", 80);//Sword +80 damage
var boots = new Item("Boots", 45);//Boots +45 move speed
var blinkDagger = new Item("Blink", 5);//+5 for all and teleport to a target point
up to 1200 units away

var swordWeapon = new Weapon(sword);
var swordInWeaponAndArtifact = new Artifact(swordWeapon);
var bootsInClothing = new Clothing(boots);
var blinkDaggerInArtefact = new Artifact(blinkDagger);
var blinkDaggerInArtefactAndClothing = new Clothing(blinkDaggerInArtefact);

hero.AddToInventory(swordInWeaponAndArtifact);
hero.AddToInventory(bootsInClothing);
hero.AddToInventory(blinkDaggerInArtefactAndClothing);

Console.WriteLine(hero);
Console.WriteLine("Inventory:");
Console.WriteLine(hero.Inventory[0].Name);
Console.WriteLine(hero.Inventory[1].Name);
Console.WriteLine(hero.Inventory[2].Name);

public interface IItem
{
    string Name { get; }
    int Bonus { get; }
    Item Parent { get; }
}

public class Item : IItem
{
    public string Name { get; }
    public int Bonus { get; }
    public Item Parent { get; }

    public Item(string name, int bonus)
    {
        Name = name;
        Bonus = bonus;
    }
}

public class Clothing : Item
{
    public Clothing(IItem item) : base($"{item.Name} (Clothing)", item.Bonus)
    {
    }
}

public class Weapon : Item
{
    public Weapon(IItem item) : base($"{item.Name} (Weapon)", item.Bonus)
    {
    }
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.14.000 – Лр.4	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public class Artifact : Item
{
    public Artifact(IItem item) : base($"{item.Name} (Artifact)", item.Bonus)
    {
        Parent = item;
    }

    public IItem Parent { get; }
}

public abstract class Hero
{
    public string Name { get; set; }
    public int Health { get; set; } = 100;
    public List<Item> Inventory { get; set; } = new List<Item>();

    public Hero(string name)
    {
        Name = name;
    }

    public void AddToInventory(Item item)
    {
        Inventory.Add(item);

        // add parent items recursively
        if (item.Parent != null)
        {
            AddToInventory(item.Parent);
        }
    }
}

public class Warrior : Hero
{
    public Warrior(string name) : base(name)
    {
    }
}

public class Mage : Hero
{
    public Mage(string name) : base(name)
    {
    }
}

public class Paladin : Hero
{
    public Paladin(string name) : base(name)
    {
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.14.000 – Лр.4	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Dekorator_Task_2.exe
Warrior
Inventory:
Sword (Weapon) (Artifact)
Boots (Clothing)
Blink (Artifact) (Clothing)

Process finished with exit code 0.

```

Рис. 2- результат виконання

### Завдання 3: Фасад.

1. Напишіть програмку для приготування "BigMac Menu".
2. В меню має входити щонайменше 3-4 окремі страви.
3. Кожній страві має відповідати окремий клас.
4. Також продумайте класи для упаковки, серветок і ціни.
5. Все меню має збиратися в окремому класі-фасаді.
6. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

Лістинг програми:

```

var Menu = new BigMacMenu();
Menu.PrintMenu();

class BigMacMenu
{
    private readonly Burger _burger;
    private readonly FrenchFries _frenchFries;
    private readonly CocaCola _cocaCola;
    private readonly Pack _pack;
    private readonly Napkins _napkins;
    private readonly Price _price;

    public BigMacMenu()
    {
        _burger = new Burger();
        _frenchFries = new FrenchFries();
        _cocaCola = new CocaCola();
        _pack = new Pack();
        _napkins = new Napkins();
        _price = new Price();
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.14.000 – Лр.4	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public void PrintMenu()
{
    Console.WriteLine("Your menu:");
    _burger.PrintBurger();
    _frenchFries.PrintFrenchFries();
    _cocaCola.PrintCocaCola();
    _pack.PrintPack();
    _napkins.PrintNapkins();
    _price.PrintPrice();
}
}

class Burger
{
    public void PrintBurger()
    {
        Console.WriteLine("Burger");
    }
}

class FrenchFries
{
    public void PrintFrenchFries()
    {
        Console.WriteLine("FrenchFries");
    }
}

class CocaCola
{
    public void PrintCocaCola()
    {
        Console.WriteLine("CocaCola");
    }
}

class Pack
{
    public void PrintPack()
    {
        Console.WriteLine("Pack");
    }
}

class Napkins
{
    public void PrintNapkins()
    {
        Console.WriteLine("Napkins");
    }
}

class Price
{
    public void PrintPrice()
    {
        Console.WriteLine("Price");
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.14.000 – Лр.4	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

```
Task_3.exe
Your menu:
Burger
FrenchFries
CocaCola
Pack
Napkins
Price
█
```

Рис. 3- результат виконання

**Завдання 4: 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:

```
IRenderer vectorRenderer = new VectorRenderer();
IRenderer rasterRenderer = new RasterRenderer();

Shape circle = new Circle(vectorRenderer);
Shape square = new Square(vectorRenderer);
Shape triangle = new Triangle(rasterRenderer);

circle.Draw();
square.Draw();
triangle.Draw();

Console.ReadKey();

interface IRenderer
{
    void RenderAsVector();
    void RenderAsRaster();
}
```

```

class VectorRenderer : IRenderer
{
    public void RenderAsVector()
    {
        Console.WriteLine("Rendering as vector");
    }

    public void RenderAsRaster()
    {
        Console.WriteLine("Cannot render as raster");
    }
}

class RasterRenderer : IRenderer
{
    public void RenderAsVector()
    {
        Console.WriteLine("Cannot render as vector");
    }

    public void RenderAsRaster()
    {
        Console.WriteLine("Rendering as raster");
    }
}

abstract class Shape
{
    protected IRenderer renderer;

    public Shape(IRenderer renderer)
    {
        this.renderer = renderer;
    }

    public abstract void Draw();
}

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

    public override void Draw()
    {
        Console.WriteLine("Drawing Circle ");
        renderer.RenderAsVector();
    }
}

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

    public override void Draw()
    {
        Console.WriteLine("Drawing Square ");
        renderer.RenderAsVector();
    }
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.14.000 – Лр.4	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

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

    public override void Draw()
    {
        Console.WriteLine("Drawing Triangle ");
        renderer.RenderAsRaster();
    }
}

```

```

SK_4.exe
Drawing Circle Rendering as vector
Drawing Square Rendering as vector
Drawing Triangle Rendering as raster

```

Рис. 4- результат виконання

**Висновок:** в ході виконання лабораторної роботи було розглянуто способи реалізовувати структурні шоблони проєктування Адаптер, Декоратор, Фасад, Міст.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.14.000 – Лр.4	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		