

ЛАБОРАТОРНА РОБОТА № 2

Породжувальні шаблони

Мета: навчитися реалізовувати породжувальні шаблони проектування.

Хід роботи:

Завдання 1: Фабричний метод.

1. Напишіть систему класів для реалізації функціоналу створення різних типів підписок для відео провайдера.
2. Кожна з підписок повинна мати щомісячну плату, мінімальний період підписки та список каналів й інших можливостей.
3. Види підписок: DomesticSubscription, EducationalSubscription, PremiumSubscription.
4. Придбати (тобто створити) підписку можна за допомогою трьох різних класів: WebSite, MobileApp, ManagerCall, кожен з них має реалізувати свою логіку створення підписок.
5. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
6. Підготуйте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

Лістинг програми:

```
using System;
using System.Collections.Generic;

abstract class Subscription
{
    public float MonthlyFee { get; set; }
    public int MinPeriod { get; set; }
    public List<string> Channels { get; set; } = new List<string>();

    public abstract string GetDetails();
}
```

					ДУ «Житомирська політехніка».25.121.00.000 - Лр2			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Якусевич А.І.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Фант М.О.						1
Керівник								15
Н. контр.							ФІКТ Гр. ІПЗ-23-4[2]	
Зав. каф.								

```

class DomesticSubscription : Subscription
{
    public DomesticSubscription()
    {
        MonthlyFee = 9.99f;
        MinPeriod = 1;
        Channels.AddRange(new[] { "Новини", "Спорт", "Фільми" });
    }

    public override string GetDetails()
    {
        return $"Домашня підписка: {MonthlyFee}$ / місяць, Мінімальний період: {MinPeriod} міс., Канали: {string.Join(", ", Channels)}";
    }
}

class EducationSubscription : Subscription
{
    public EducationSubscription()
    {
        MonthlyFee = 4.99f;
        MinPeriod = 6;
        Channels.AddRange(new[] { "Наука", "Документальні", "Історія" });
    }

    public override string GetDetails()
    {
        return $"Освітня підписка: {MonthlyFee}$ / місяць, Мінімальний період: {MinPeriod} міс., Канали: {string.Join(", ", Channels)}";
    }
}

class PremiumSubscription : Subscription
{
    public PremiumSubscription()
    {
        MonthlyFee = 19.99f;
        MinPeriod = 3;
        Channels.AddRange(new[] { "Усі канали", "Підтримка 4K", "Без реклами" });
    }

    public void PurchaseParkingPass()
    {
        Console.WriteLine("Преміум підписку активовано!");
    }

    public override string GetDetails()
    {
        return $"Преміум підписка: {MonthlyFee}$ / місяць, Мінімальний період: {MinPeriod} міс., Канали: {string.Join(", ", Channels)}";
    }
}

interface SubscriptionFactory
{
    Subscription CreateSubscription(string type);
}

class WebSite : SubscriptionFactory
{
    public Subscription CreateSubscription(string type)

```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    Console.WriteLine("Створення підписки через вебсайт...");
    return Create(type);
}

private Subscription Create(string type)
{
    return type switch
    {
        "domestic" => new DomesticSubscription(),
        "education" => new EducationSubscription(),
        "premium" => new PremiumSubscription(),
        _ => throw new ArgumentException("Невірний тип підписки")
    };
}

class MobileApp : SubscriptionFactory
{
    public Subscription CreateSubscription(string type)
    {
        Console.WriteLine("Створення підписки через мобільний застосунок...");
        return Create(type);
    }

    private Subscription Create(string type)
    {
        return type switch
        {
            "domestic" => new DomesticSubscription(),
            "education" => new EducationSubscription(),
            "premium" => new PremiumSubscription(),
            _ => throw new ArgumentException("Невірний тип підписки")
        };
    }
}

class ManagerCall : SubscriptionFactory
{
    public Subscription CreateSubscription(string type)
    {
        Console.WriteLine("Створення підписки через дзвінок менеджера...");
        return Create(type);
    }

    private Subscription Create(string type)
    {
        return type switch
        {
            "domestic" => new DomesticSubscription(),
            "education" => new EducationSubscription(),
            "premium" => new PremiumSubscription(),
            _ => throw new ArgumentException("Невірний тип підписки")
        };
    }
}

class Program
{
    static void Main()
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;
    }
}

```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

SubscriptionFactory website = new WebSite();
Subscription domestic = website.CreateSubscription("domestic");
Console.WriteLine(domestic.GetDetails());

SubscriptionFactory app = new MobileApp();
Subscription education = app.CreateSubscription("education");
Console.WriteLine(education.GetDetails());

SubscriptionFactory manager = new ManagerCall();
Subscription premium = manager.CreateSubscription("premium");
Console.WriteLine(premium.GetDetails());

if (premium is PremiumSubscription premiumSub)
{
    premiumSub.PurchaseParkingPass();
}

Console.WriteLine("\n=== Усі підписки успішно створено! ===");
}

```

Результат виконання програми:

```

Створення підписки через вебсайт...
Домашня підписка: 9,99$ / місяць, Мінімальний період: 1 міс., Канали: Новини, Спорт, Фільми
Створення підписки через мобільний застосунок...
Освітня підписка: 4,99$ / місяць, Мінімальний період: 6 міс., Канали: Наука, Документальні, Історія
Створення підписки через дзвінок менеджера...
Преміум підписка: 19,99$ / місяць, Мінімальний період: 3 міс., Канали: Усі канали, Підтримка 4K, Без реклами
Преміум підписку активовано!

=== Усі підписки успішно створено! ===

```

Рис. 1. Результат виконання програми

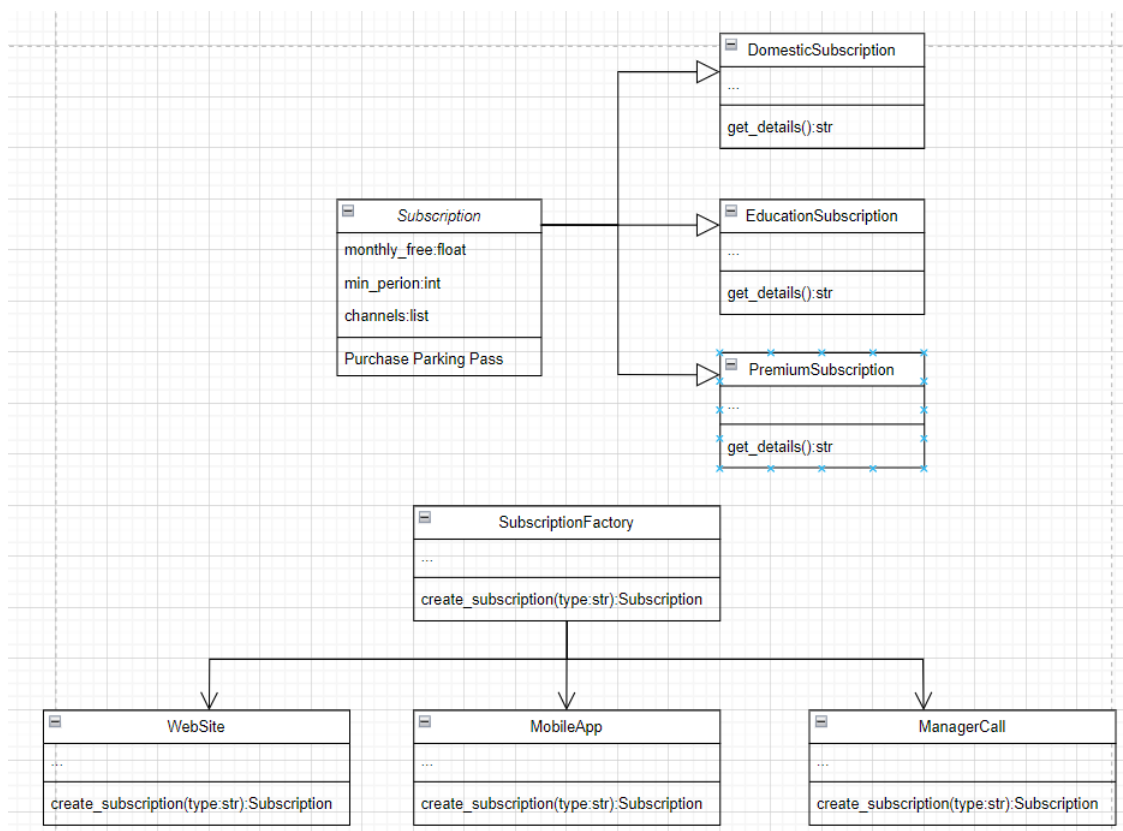


Рис. 2. Діаграма класів для реалізації функціоналу створення різних типів підписок для відео провайдера

Завдання 2: Абстрактна фабрика.

1. Створіть фабрику виробництва техніки.
2. На фабриці мають створюватися різні девайси (наприклад, Laptop, Netbook, EBook, Smartphone) для різних брендів (IPhone, Kiaomi, Balaxy).
3. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
4. Підготуйте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

Лістинг програми:

```
using System;

abstract class Device
{
    public abstract void GetInfo();
}

class Laptop : Device
{
    private string brand;
    public Laptop(string brand) => this.brand = brand;

    public override void GetInfo()
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;
        Console.WriteLine($"Ноутбук бренду {brand}");
    }
}

class Netbook : Device
{
    private string brand;
    public Netbook(string brand) => this.brand = brand;

    public override void GetInfo()
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;
        Console.WriteLine($"Нетбук бренду {brand}");
    }
}

class EBook : Device
{
    private string brand;
    public EBook(string brand) => this.brand = brand;

    public override void GetInfo()
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;
```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Console.WriteLine($"Електронна книга бренду {brand}");
    }
}

class Smartphone : Device
{
    private string brand;
    public Smartphone(string brand) => this.brand = brand;

    public override void GetInfo()
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;
        Console.WriteLine($"Смартфон бренду {brand}");
    }
}

abstract class DeviceFactory
{
    public abstract Laptop CreateLaptop();
    public abstract Netbook CreateNetbook();
    public abstract EBook CreateEBook();
    public abstract Smartphone CreateSmartphone();
}

class IProneFactory : DeviceFactory
{
    private string brand = "IProne";

    public override Laptop CreateLaptop() => new Laptop(brand);
    public override Netbook CreateNetbook() => new Netbook(brand);
    public override EBook CreateEBook() => new EBook(brand);
    public override Smartphone CreateSmartphone() => new Smartphone(brand);
}

class KiaomiFactory : DeviceFactory
{
    private string brand = "Kiaomi";

    public override Laptop CreateLaptop() => new Laptop(brand);
    public override Netbook CreateNetbook() => new Netbook(brand);
    public override EBook CreateEBook() => new EBook(brand);
    public override Smartphone CreateSmartphone() => new Smartphone(brand);
}

class BalaxyFactory : DeviceFactory
{
    private string brand = "Balaxy";

    public override Laptop CreateLaptop() => new Laptop(brand);
    public override Netbook CreateNetbook() => new Netbook(brand);
    public override EBook CreateEBook() => new EBook(brand);
    public override Smartphone CreateSmartphone() => new Smartphone(brand);
}

class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;

        DeviceFactory iprone = new IProneFactory();
        DeviceFactory kiaomi = new KiaomiFactory();
    }
}

```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

DeviceFactory balaxy = new BalaxyFactory();

Console.WriteLine("Пристрої IProne:");
iprone.CreateLaptop().GetInfo();
iprone.CreateNetbook().GetInfo();
iprone.CreateEBook().GetInfo();
iprone.CreateSmartphone().GetInfo();

Console.WriteLine("\nПристрої Kiaomi:");
kiaomi.CreateLaptop().GetInfo();
kiaomi.CreateNetbook().GetInfo();
kiaomi.CreateEBook().GetInfo();
kiaomi.CreateSmartphone().GetInfo();

Console.WriteLine("\nПристрої Balaxy:");
balaxy.CreateLaptop().GetInfo();
balaxy.CreateNetbook().GetInfo();
balaxy.CreateEBook().GetInfo();
balaxy.CreateSmartphone().GetInfo();
}
}

```

Результат виконання програми:

```

Пристрої IProne:
Ноутбук бренду IProne
Нетбук бренду IProne
Електронна книга бренду IProne
Смартфон бренду IProne

Пристрої Kiaomi:
Ноутбук бренду Kiaomi
Нетбук бренду Kiaomi
Електронна книга бренду Kiaomi
Смартфон бренду Kiaomi

Пристрої Balaxy:
Ноутбук бренду Balaxy
Нетбук бренду Balaxy
Електронна книга бренду Balaxy
Смартфон бренду Balaxy

```

Рис. 3. Результат виконання програми

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

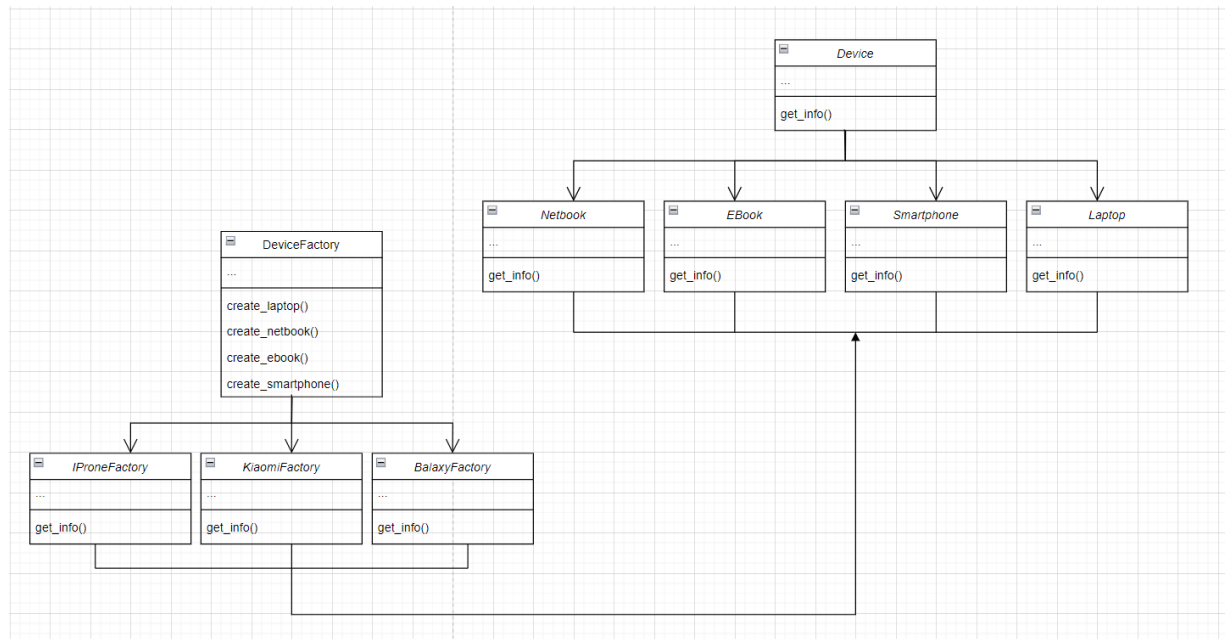


Рис. 4. Діаграма класів фабрики виробництва техніки

Завдання 3: Одинак.

1. Створіть клас Authenticator таким чином, щоб бути впевненим, що цей клас може створити лише один екземпляр, незалежно від кількості потоків і класів, що його наслідують.

2. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

Лістинг програми:

```

using System;
using System.Threading;

public sealed class Authenticator
{
    private static volatile Authenticator instance;
    private static readonly object lockObject = new object();
    private Authenticator()
    {
        if (instance != null)
        {
            throw new InvalidOperationException("Use GetInstance() method to get the single instance of this class.");
        }
    }

    public static Authenticator GetInstance()
    {
        if (instance == null)
        {
            lock (lockObject)
            {
                if (instance == null)
                {
                    instance = new Authenticator();
                }
            }
        }
    }
}
  
```



```

    }
}
return instance;
}

public void Authenticate(string username, string password)
{
    Console.WriteLine($"Authenticating user: {username}");
}

}

public class Program
{
    public static void Main(string[] args)
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;

        Authenticator auth1 = Authenticator.GetInstance();
        Authenticator auth2 = Authenticator.GetInstance();

        Console.WriteLine($"auth1 і auth2 це один і той же об'єкт? {object.ReferenceEquals(auth1, auth2)}");

        auth1.Authenticate("admin", "password123");

        Thread thread1 = new Thread(() => {
            Authenticator auth = Authenticator.GetInstance();
            Console.WriteLine($"Authenticator instance in thread {Thread.CurrentThread.ManagedThreadId}: {auth.GetHashCode()}");
        });

        Thread thread2 = new Thread(() => {
            Authenticator auth = Authenticator.GetInstance();
            Console.WriteLine($"Authenticator instance in thread {Thread.CurrentThread.ManagedThreadId}: {auth.GetHashCode()}");
        });

        thread1.Start();
        thread2.Start();

        thread1.Join();
        thread2.Join();
    }
}

```

Результат виконання програми:

```

auth1 і auth2 це один і той же об'єкт? True
Authenticating user: admin
Authenticator instance in thread 11: 39086322
Authenticator instance in thread 12: 39086322

```

Рис. 5. Результат виконання програми

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

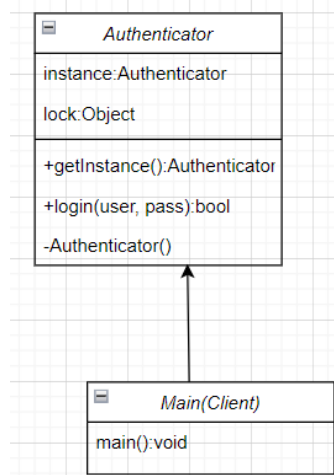


Рис. 6. Діаграма класа Authenticator

Завдання 4: Прототип.

1. Створіть клас Virus. Він повинен містити вагу, вік, ім'я, вид і масив дітей, екземплярів Virus.

2. Створіть екземпляри для цілого “сімейства” вірусів (мінімум три покоління).

3. За допомогою шаблону Прототип реалізуйте можливість клонування наявних вірусів.

4. При клонуванні віруса-батька повинні клонуватися всі його діти.

5. Покажіть правильність роботи свого коду запустивши його в головному методі програми

Лістинг програми:

```

using System;
using System.Collections.Generic;

class Virus
{
    private string name;
    private double weight;
    private int age;
    private string species;
    private List<Virus> children;

    public Virus(string name, double weight, int age, string species)
    {
        this.name = name;
        this.weight = weight;
        this.age = age;
        this.species = species;
        this.children = new List<Virus>();
    }
}
  
```

```

public void AddChild(Virus v)
{
    children.Add(v);
}

public Virus Clone()
{
    Virus clone = new Virus(name, weight, age, species);

    foreach (Virus child in children)
    {
        clone.AddChild(child.Clone());
    }

    return clone;
}

public void Print(int level = 0)
{
    Console.WriteLine($"{new string('-', level * 2)} {name} (age: {age}, weight: {weight}, species: {species})");
    foreach (Virus child in children)
    {
        child.Print(level + 1);
    }
}

class VirusSimulation
{
    static void Main()
    {
        Virus grandparent = new Virus("Alpha", 1.2, 5, "Corona");
        Virus parent1 = new Virus("Beta", 1.0, 3, "Corona");
        Virus parent2 = new Virus("Gamma", 0.9, 3, "Corona");

        Virus child1 = new Virus("Delta", 0.5, 1, "Corona");
        Virus child2 = new Virus("Epsilon", 0.4, 1, "Corona");

        parent1.AddChild(child1);
        parent2.AddChild(child2);

        grandparent.AddChild(parent1);
        grandparent.AddChild(parent2);

        Console.WriteLine("Оригінал:");
        grandparent.Print();

        Virus cloned = grandparent.Clone();
        Console.WriteLine("\nКлон:");
        cloned.Print();
    }
}

```

Результат виконання програми:

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Microsoft Visual Studio Debug Console
Оригінал:
Alpha (age: 5, weight: 1,2, species: Corona)
-- Beta (age: 3, weight: 1, species: Corona)
---- Delta (age: 1, weight: 0,5, species: Corona)
-- Gamma (age: 3, weight: 0,9, species: Corona)
---- Epsilon (age: 1, weight: 0,4, species: Corona)

Клон:
Alpha (age: 5, weight: 1,2, species: Corona)
-- Beta (age: 3, weight: 1, species: Corona)
---- Delta (age: 1, weight: 0,5, species: Corona)
-- Gamma (age: 3, weight: 0,9, species: Corona)
---- Epsilon (age: 1, weight: 0,4, species: Corona)

```

Рис. 7. Результат виконання програми

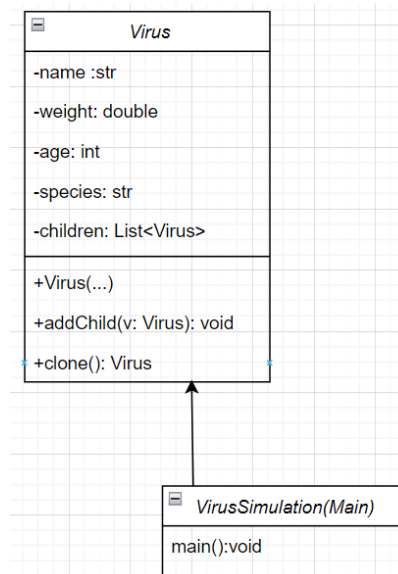


Рис. 8. Діаграма класа Virus

Завдання 5: Будівельник.

1. Створіть клас HeroBuilder, який буде створювати персонажа гри, поступово додаючи до нього різні ознаки, наприклад зріст, статуру, колір волосся, очей, одяг, інвентар тощо (можете включити фантазію).
2. Створіть клас EnemyBuilder, який буде реалізовувати єдиний інтерфейс з HeroBuilder. Відмінністю між ними можуть бути спеціальні методи для творення добра або зла, а також списки добрих і злих справ відповідно.
3. За допомогою свого білдера і класу-директора створіть героя (або героїню) своєї мрії 😊, а також свого найзапеклішого ворога.
4. Зверніть увагу, що Ваші білдери повинні реалізовувати текучий інтерфейс (fluent interface).

5. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

6. Підготуйте діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>, експортуйте та завантажте її до репозиторія.

Лістинг програми:

```
using System;
using System.Collections.Generic;

public interface ICharacterBuilder
{
    ICharacterBuilder SetHeight(int height);
    ICharacterBuilder SetBodyType(string bodyType);
    ICharacterBuilder SetHairColor(string hairColor);
    ICharacterBuilder SetEyeColor(string eyeColor);
    ICharacterBuilder SetClothes(string clothes);
    ICharacterBuilder SetInventory(List<string> inventory);
    ICharacterBuilder AddGoodDeed(string deed);
    ICharacterBuilder AddEvilDeed(string deed);
    GameCharacter Build();
}

public class GameCharacter
{
    public int Height { get; set; }
    public string BodyType { get; set; } = string.Empty;
    public string HairColor { get; set; } = string.Empty;
    public string EyeColor { get; set; } = string.Empty;
    public string Clothes { get; set; } = string.Empty;
    public List<string> Inventory { get; set; } = new();
    public List<string> GoodDeeds { get; set; } = new();
    public List<string> EvilDeeds { get; set; } = new();

    public override string ToString()
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;
        return $"Ріст: {Height}, Статура: {BodyType}, Колір волосся: {HairColor}, Колір очей: {EyeColor}, Одяг: {Clothes},\n" +
            $"Інвентар: {string.Join(", ", Inventory)},\n" +
            $"Добрі справи: {string.Join(", ", GoodDeeds)},\n" +
            $"Погані справи : {string.Join(", ", EvilDeeds)}\n";
    }
}

public abstract class BaseCharacterBuilder : ICharacterBuilder
{
    protected GameCharacter character = new();

    public ICharacterBuilder SetHeight(int height) { character.Height = height; return this; }
    public ICharacterBuilder SetBodyType(string bodyType) { character.BodyType = bodyType; return this; }
    public ICharacterBuilder SetHairColor(string hairColor) { character.HairColor = hairColor; return this; }
    public ICharacterBuilder SetEyeColor(string eyeColor) { character.EyeColor = eyeColor; return this; }
    public ICharacterBuilder SetClothes(string clothes) { character.Clothes = clothes; return this; }
    public ICharacterBuilder SetInventory(List<string> inventory) { character.Inventory = inventory; return this; }

    public abstract ICharacterBuilder AddGoodDeed(string deed);
    public abstract ICharacterBuilder AddEvilDeed(string deed);
}
```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public GameCharacter Build() => character;
}

public class HeroBuilder : BaseCharacterBuilder
{
    public override ICharacterBuilder AddGoodDeed(string deed)
    {
        character.GoodDeeds.Add(deed);
        return this;
    }

    public override ICharacterBuilder AddEvilDeed(string deed)
    {
        Console.WriteLine(" Герой не може чинити злі справи!");
        return this;
    }
}

public class EnemyBuilder : BaseCharacterBuilder
{
    public override ICharacterBuilder AddEvilDeed(string deed)
    {
        character.EvilDeeds.Add(deed);
        return this;
    }

    public override ICharacterBuilder AddGoodDeed(string deed)
    {
        Console.WriteLine(" Ворог не здатен на добрі вчинки!");
        return this;
    }
}

public class Director
{
    public void ConstructHero(HeroBuilder builder)
    {
        builder
            .SetHeight(180)
            .SetBodyType("Стрункий")
            .SetHairColor("Блондин")
            .SetEyeColor("Блакитні")
            .SetClothes("Броня")
            .SetInventory(new List<string> { "Меч", "Щит" })
            .AddGoodDeed("Врятував кота");
    }

    public void ConstructEnemy(EnemyBuilder builder)
    {
        builder
            .SetHeight(200)
            .SetBodyType("Мускулистий")
            .SetHairColor("Брюнет")
            .SetEyeColor("Червоні")
            .SetClothes("Темний плащ")
            .SetInventory(new List<string> { "Отрута", "Кинджал" })
            .AddEvilDeed("Спалив місто")
            .AddEvilDeed("Вкрав стародавній артефакт");
    }
}

class MainClass

```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
public static void Main(string[] args)
{
    Console.OutputEncoding = System.Text.Encoding.UTF8;

    Director director = new();

    HeroBuilder heroBuilder = new();
    director.ConstructHero(heroBuilder);
    GameCharacter hero = heroBuilder.Build();

    EnemyBuilder enemyBuilder = new();
    director.ConstructEnemy(enemyBuilder);
    GameCharacter enemy = enemyBuilder.Build();

    Console.WriteLine("Герой:");
    Console.WriteLine(hero);
    Console.WriteLine("Ворог:");
    Console.WriteLine(enemy);

    heroBuilder.AddEvilDeed("Вкрав торт");
    enemyBuilder.AddGoodDeed("Подарував квіти");
}
}

```

Результат виконання програми:

```

Герой:
Ріст: 180, Статура: Стрункий, Колір волосся: Блондин, Колір очей: Блакитні, Одяг: Броня,
Інвентар: Меч, Щит,
Добрі справи: Врятував кота,
Погані справи :
Ворог:
Ріст: 200, Статура: Мускулистий, Колір волосся: Брюнет, Колір очей: Червоні, Одяг: Темний плащ,
Інвентар: Отрута, Кинджал,
Добрі справи:
Погані справи : Спалив місто, Вкрав стародавній артефакт

Герой не може чинити злі справи!
Ворог не здатен на добрі вчинки!

```

Рис. 9. Результат виконання програми

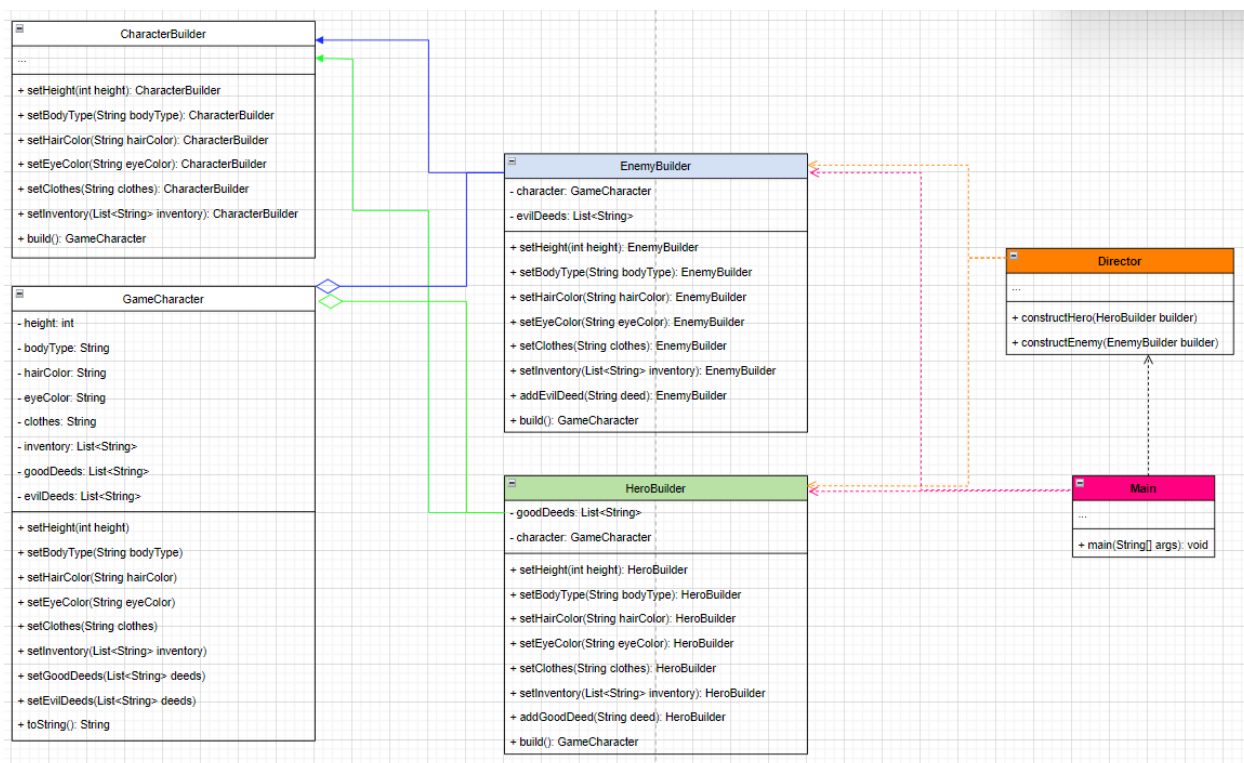


Рис. 10. Діаграма класів для створення свого героя та свого найзапеклішого
ворога

Висновки: після виконання цієї лабораторної роботи я навчилася реалізовувати породжувальні шаблони проектування.

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.00.000 - Лр2	Арк.
		Фант М.О.				16
Змн.	Арк.	№ докум.	Підпис	Дата		