

ЛАБОРАТОРНА РОБОТА № 2

Тема: Породжувальні шаблони

Мета роботи: навчитися реалізовувати породжувальні шаблони проектування

Хід роботи:

Завдання 1:

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 реалізовувати 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.

Таблиця 1 - Завдання

Лістинг програми:

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

abstract class Subscription
{
    public abstract float MonthlyFee { get; }
    public abstract int MinimumSubscriptionPeriod { get; }
    public abstract List<string> Channels { get; }
    public abstract List<string> Features { get; }

    public void GetDetails()
```

					ДУ «Житомирська політехніка».22.121.19.000 – Лр2			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Пасічник Б.Р			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Фант М.О.						1
Керівник							ФІКТ Гр. ВТ-22-1[2]	
Н. контр.								
Зав. каф.								

```

{
    Console.WriteLine($"Підписка типу {GetType().Name}:");
    Console.WriteLine($"Плата: {MonthlyFee} USD");
    Console.WriteLine($"Мінімальний період підписки: {MinimumSubscriptionPeriod}
місяць(ці)");
    Console.WriteLine("Канали: " + string.Join(", ", Channels));
    Console.WriteLine("Інші можливості: " + string.Join(", ", Features));
}
}

class DomesticSubscription : Subscription
{
    public override float MonthlyFee => 9.99f;
    public override int MinimumSubscriptionPeriod => 1;
    public override List<string> Channels => new List<string> { "Канал1", "Канал2",
"Канал3" };
    public override List<string> Features => new List<string> { "звичайна якість" };
}

class EducationalSubscription : Subscription
{
    public override float MonthlyFee => 14.99f;
    public override int MinimumSubscriptionPeriod => 1;
    public override List<string> Channels => new List<string> { "Навчальни1", "Навча-
льни2", "Навчальни3", "Навчальни4" };
    public override List<string> Features => new List<string> { "стандартна якість",
"навчальний контент" };
}

class PremiumSubscription : Subscription
{
    public override float MonthlyFee => 19.99f;
    public override int MinimumSubscriptionPeriod => 1;
    public override List<string> Channels => new List<string> { "Канал1", "Канал2",
"Канал3", "Канал4", "Канал5" };
    public override List<string> Features => new List<string> { "висока якість", "без
реклами" };
}

abstract class SubscriptionFactory
{
    public abstract Subscription CreateSubscription();
}

class DomesticSubscriptionFactory : SubscriptionFactory
{
    public override Subscription CreateSubscription()
    {
        return new DomesticSubscription();
    }
}

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

class EducationalSubscriptionFactory : SubscriptionFactory
{
    public override Subscription CreateSubscription()
    {
        return new EducationalSubscription();
    }
}

class PremiumSubscriptionFactory : SubscriptionFactory
{
    public override Subscription CreateSubscription()
    {
        return new PremiumSubscription();
    }
}

interface ISubscriptionCreator
{
    Subscription CreateSubscription(SubscriptionFactory factory);
}

class WebSite : ISubscriptionCreator
{
    public Subscription CreateSubscription(SubscriptionFactory factory)
    {
        Console.WriteLine("створення підписки через WebSite...");
        return factory.CreateSubscription();
    }
}

class MobileApp : ISubscriptionCreator
{
    public Subscription CreateSubscription(SubscriptionFactory factory)
    {
        Console.WriteLine("створення підписки через MobileApp...");
        return factory.CreateSubscription();
    }
}

class ManagerCall : ISubscriptionCreator
{
    public Subscription CreateSubscription(SubscriptionFactory factory)
    {
        Console.WriteLine("створення підписки через ManagerCall...");
        return factory.CreateSubscription();
    }
}

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class Program
{
    static void Main(string[] args)
    {
        // Використання WebSite для створення підписок
        ISubscriptionCreator webSiteCreator = new WebSite();
        SubscriptionFactory domesticFactory = new DomesticSubscriptionFactory();
        Subscription domesticSubscription =
webSiteCreator.CreateSubscription(domesticFactory);
        domesticSubscription.GetDetails();
        Console.WriteLine();

        // Використання MobileApp для створення підписок
        ISubscriptionCreator mobileAppCreator = new MobileApp();
        SubscriptionFactory educationalFactory = new
EducationalSubscriptionFactory();
        Subscription educationalSubscription =
mobileAppCreator.CreateSubscription(educationalFactory);
        educationalSubscription.GetDetails();
        Console.WriteLine();

        // Використання ManagerCall для створення підписок
        ISubscriptionCreator managerCallCreator = new ManagerCall();
        SubscriptionFactory premiumFactory = new PremiumSubscriptionFactory();
        Subscription premiumSubscription =
managerCallCreator.CreateSubscription(premiumFactory);
        premiumSubscription.GetDetails();
    }
}

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				4
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання програми:

```
не debugger option 'Just My Code' is enabled...
створення підписки через WebSite...
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.
option 'Just My Code' is enabled.
Підписка типу DomesticSubscription:
Плата: 9,99 USD
Мінімальний період підписки: 1 місяць(ці)
Канали: Канал1, Канал2, Канал3
Інші можливості: звичайна якість

створення підписки через MobileApp...
Підписка типу EducationalSubscription:
Плата: 14,99 USD
Мінімальний період підписки: 1 місяць(ці)
Канали: Навчальни1, Навчальни2, Навчальни3, Навчальни4
Інші можливості: стандартна якість, навчальний контент

створення підписки через ManagerCall...
Підписка типу PremiumSubscription:
Плата: 19,99 USD
Мінімальний період підписки: 1 місяць(ці)
Канали: Канал1, Канал2, Канал3, Канал4, Канал5
Інші можливості: висока якість, без реклами
The program '[14932] Task1.exe' has exited with code 0 (0)
```

Рис. 1. Результат виконання програми

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				5
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2:

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;

namespace AbstractFactoryPattern
{
    public abstract class Laptop
    {
        public abstract void ShowDetails();
    }

    public abstract class Smartphone
    {
        public abstract void ShowDetails();
    }

    public class IPhoneLaptop : Laptop
    {
        public override void ShowDetails()
        {
            Console.WriteLine("IPhone Laptop");
        }
    }

    public class IPhoneSmartphone : Smartphone
    {
        public override void ShowDetails()
```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    {
        Console.WriteLine("IPhone Smartphone");
    }
}

public class KiaomiLaptop : Laptop
{
    public override void ShowDetails()
    {
        Console.WriteLine("Kiaomi Laptop");
    }
}

public class KiaomiSmartphone : Smartphone
{
    public override void ShowDetails()
    {
        Console.WriteLine("Kiaomi Smartphone");
    }
}

public class BalaxyLaptop : Laptop
{
    public override void ShowDetails()
    {
        Console.WriteLine("Balaxy Laptop");
    }
}

public class BalaxySmartphone : Smartphone
{
    public override void ShowDetails()
    {
        Console.WriteLine("Balaxy Smartphone");
    }
}

public abstract class TechFactory
{
    public abstract Laptop CreateLaptop();
    public abstract Smartphone CreateSmartphone();
}

public class IProneFactory : TechFactory
{
    public override Laptop CreateLaptop()
    {
        return new IProneLaptop();
    }
}

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        public override Smartphone CreateSmartphone()
        {
            return new IProneSmartphone();
        }
    }

    public class KiaomiFactory : TechFactory
    {
        public override Laptop CreateLaptop()
        {
            return new KiaomiLaptop();
        }

        public override Smartphone CreateSmartphone()
        {
            return new KiaomiSmartphone();
        }
    }

    public class BalaxyFactory : TechFactory
    {
        public override Laptop CreateLaptop()
        {
            return new BalaxyLaptop();
        }

        public override Smartphone CreateSmartphone()
        {
            return new BalaxySmartphone();
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            TechFactory iproneFactory = new IProneFactory();
            TechFactory kiaomiFactory = new KiaomiFactory();
            TechFactory balaxyFactory = new BalaxyFactory();

            Laptop iproneLaptop = iproneFactory.CreateLaptop();
            Smartphone iproneSmartphone = iproneFactory.CreateSmartphone();

            Laptop kiaomiLaptop = kiaomiFactory.CreateLaptop();
            Smartphone kiaomiSmartphone = kiaomiFactory.CreateSmartphone();

            Laptop balaxyLaptop = balaxyFactory.CreateLaptop();
            Smartphone balaxySmartphone = balaxyFactory.CreateSmartphone();

            iproneLaptop.ShowDetails();
        }
    }

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        iproneSmartphone.ShowDetails();

        kiaomiLaptop.ShowDetails();
        kiaomiSmartphone.ShowDetails();

        balaxyLaptop.ShowDetails();
        balaxySmartphone.ShowDetails();
    }
}
}

```

Результат виконання програми:

```

Loaded 'C:\Program Files\dotnet\dotnet.exe'
the debugger option 'Just My Code'
Loaded 'C:\Program Files\dotnet\dotnet.exe'
the debugger option 'Just My Code'
IPhone Laptop
IPhone Smartphone
Kiaomi Laptop
Kiaomi Smartphone
Balaxy Laptop
Balaxy Smartphone
The program '[17564] Task2.exe'

```

Рис. 2. Результат виконання програми

Завдання 3:

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;

namespace SingletonPattern
{
    public sealed class Authenticator
    {

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private static Authenticator instance = null;

private static readonly object lockObject = new object();

private Authenticator()
{
    Console.WriteLine("Створено екземпляр автентифікатора.");
}

public static Authenticator Instance
{
    get
    {
        if (instance == null)
        {
            lock (lockObject)
            {
                if (instance == null)
                {
                    instance = new Authenticator();
                }
            }
        }
        return instance;
    }
}

public void Authenticate(string username, string password)
{
    Console.WriteLine($"Автентифікація користувача {username}.");
}

class Program
{
    static void Main(string[] args)
    {
        Authenticator auth1 = Authenticator.Instance;
        Authenticator auth2 = Authenticator.Instance;

        if (auth1 == auth2)
        {
            Console.WriteLine("Обидві автентифікації однакові.");
        }
        else
        {
            Console.WriteLine("Автентифікації різні.");
        }

        auth1.Authenticate("user1", "password1");
    }
}

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        auth2.Authenticate("user2", "password2");
    }
}

```

Результат виконання програми:

```

The debugger option 'Just My Code' is en
Створено екземпляр автентифікатора.
Обидві автентифікації однакові.
Автентифікація користувача user1.
Автентифікація користувача user2.
The program '[18140] Task3.exe' has exi

```

Рис. 3. Результат виконання програми

Завдання 4:

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;

namespace PrototypePattern
{
    public class Virus : ICloneable
    {
        public double Weight { get; set; }
        public int Age { get; set; }
        public string Name { get; set; }
        public string Species { get; set; }
    }
}

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public List<Virus> Children { get; set; }

public Virus(string name, string species, double weight, int age)
{
    Name = name;
    Species = species;
    Weight = weight;
    Age = age;
    Children = new List<Virus>();
}

public void AddChild(Virus child)
{
    Children.Add(child);
}

public object Clone()
{
    Virus clone = (Virus)this.MemberwiseClone();
    clone.Children = new List<Virus>();

    foreach (var child in Children)
    {
        clone.Children.Add((Virus)child.Clone());
    }

    return clone;
}

public void Display(int generation = 0)
{
    string indent = new string(' ', generation * 2);
    Console.WriteLine($"{indent}Ім'я: {Name}, Вид: {Species}, Вага: {Weight},
    Вік: {Age}");
    foreach (var child in Children)
    {
        child.Display(generation + 1);
    }
}

class Program
{
    static void Main(string[] args)
    {
        Virus parentVirus = new Virus("Батько", "Вид1", 2.5, 5);

        Virus childVirus1 = new Virus("Дитина1", "Вид1", 1.2, 3);
        Virus childVirus2 = new Virus("Дитина2", "Вид1", 1.3, 4);
    }
}

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

parentVirus.AddChild(childVirus1);
parentVirus.AddChild(childVirus2);

Virus grandChildVirus1 = new Virus("Онук1", "Вид1", 0.6, 1);
Virus grandChildVirus2 = new Virus("Онук2", "Вид1", 0.7, 2);

childVirus1.AddChild(grandChildVirus1);
childVirus2.AddChild(grandChildVirus2);

Virus clonedParentVirus = (Virus)parentVirus.Clone();

Console.WriteLine("Оригінальна сім'я віруса:");
parentVirus.Display();

Console.WriteLine("\nклонована сім'я віруса:");
clonedParentVirus.Display();
    }
}

```

Результат виконання програми:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Оригінальна сім'я віруса:
Ім'я: Батько, Вид: Вид1, Вага: 2,5, Вік: 5
    Ім'я: Дитина1, Вид: Вид1, Вага: 1,2, Вік: 3
        Ім'я: Онук1, Вид: Вид1, Вага: 0,6, Вік: 1
    Ім'я: Дитина2, Вид: Вид1, Вага: 1,3, Вік: 4
        Ім'я: Онук2, Вид: Вид1, Вага: 0,7, Вік: 2

клонована сім'я віруса:
Ім'я: Батько, Вид: Вид1, Вага: 2,5, Вік: 5
    Ім'я: Дитина1, Вид: Вид1, Вага: 1,2, Вік: 3
        Ім'я: Онук1, Вид: Вид1, Вага: 0,6, Вік: 1
    Ім'я: Дитина2, Вид: Вид1, Вага: 1,3, Вік: 4
        Ім'я: Онук2, Вид: Вид1, Вага: 0,7, Вік: 2
The program '[3812] Task4.exe' has exited with code 0

```

Рис. 4. Результат виконання програми

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 5:

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;

namespace BuilderPattern
{
    public class Character
    {
        public string Name { get; set; }
        public double Height { get; set; }
        public string Build { get; set; }
        public string HairColor { get; set; }
        public string EyeColor { get; set; }
        public List<string> Inventory { get; set; } = new List<string>();
    }
}
```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public List<string> Deeds { get; set; } = new List<string>();

    public override string ToString()
    {
        return $"Ім'я: {Name}, Ріст: {Height}, Тіло: {Build}, Колір волосся:
{HairColor}, Колір очей: {EyeColor}, Інвентар: {string.Join(", ", Inventory)}, Заслу-
ги: {string.Join(", ", Deeds)}";
    }
}

public interface ICharacterBuilder
{
    ICharacterBuilder SetName(string name);
    ICharacterBuilder SetHeight(double height);
    ICharacterBuilder SetBuild(string build);
    ICharacterBuilder SetHairColor(string hairColor);
    ICharacterBuilder SetEyeColor(string eyeColor);
    ICharacterBuilder AddInventoryItem(string item);
    ICharacterBuilder AddDeed(string deed);
    Character Build();
}

public class HeroBuilder : ICharacterBuilder
{
    private Character character;

    public HeroBuilder()
    {
        character = new Character();
    }

    public ICharacterBuilder SetName(string name)
    {
        character.Name = name;
        return this;
    }

    public ICharacterBuilder SetHeight(double height)
    {
        character.Height = height;
        return this;
    }

    public ICharacterBuilder SetBuild(string build)
    {
        character.Build = build;
        return this;
    }

    public ICharacterBuilder SetHairColor(string hairColor)
    {

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				15
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        character.HairColor = hairColor;
        return this;
    }

    public ICharacterBuilder SetEyeColor(string eyeColor)
    {
        character.EyeColor = eyeColor;
        return this;
    }

    public ICharacterBuilder AddInventoryItem(string item)
    {
        character.Inventory.Add(item);
        return this;
    }

    public ICharacterBuilder AddDeed(string deed)
    {
        character.Deeds.Add(deed);
        return this;
    }

    public Character Build()
    {
        return character;
    }
}

public class EnemyBuilder : ICharacterBuilder
{
    private Character character;

    public EnemyBuilder()
    {
        character = new Character();
    }

    public ICharacterBuilder SetName(string name)
    {
        character.Name = name;
        return this;
    }

    public ICharacterBuilder SetHeight(double height)
    {
        character.Height = height;
        return this;
    }

    public ICharacterBuilder SetBuild(string build)
    {

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				16
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        character.Build = build;
        return this;
    }

    public ICharacterBuilder SetHairColor(string hairColor)
    {
        character.HairColor = hairColor;
        return this;
    }

    public ICharacterBuilder SetEyeColor(string eyeColor)
    {
        character.EyeColor = eyeColor;
        return this;
    }

    public ICharacterBuilder AddInventoryItem(string item)
    {
        character.Inventory.Add(item);
        return this;
    }

    public ICharacterBuilder AddDeed(string deed)
    {
        character.Deeds.Add(deed);
        return this;
    }

    public Character Build()
    {
        return character;
    }
}

public class Director
{
    public Character ConstructHero(ICharacterBuilder builder)
    {
        return builder
            .SetName("Герой Архілес")
            .SetHeight(180)
            .SetBuild("Атлет")
            .SetHairColor("Чорний")
            .SetEyeColor("Blue")
            .AddInventoryItem("Меч")
            .AddInventoryItem("Щит")
            .AddDeed("Врятував село")
            .Build();
    }
}

```

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				17
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public Character ConstructEnemy(ICharacterBuilder builder)
{
    return builder
        .SetName("Негідник Арлонг")
        .SetHeight(190)
        .SetBuild("Качок")
        .SetHairColor("Чорний")
        .SetEyeColor("Голубий")
        .AddInventoryItem("Бойові крабики")
        .AddInventoryItem("Луфі")
        .AddDeed("Намагався вбити Намі")
        .Build();
}

}

class Program
{
    static void Main(string[] args)
    {
        Director director = new Director();

        HeroBuilder heroBuilder = new HeroBuilder();
        Character hero = director.ConstructHero(heroBuilder);
        Console.WriteLine("Hero:");
        Console.WriteLine(hero);

        EnemyBuilder enemyBuilder = new EnemyBuilder();
        Character enemy = director.ConstructEnemy(enemyBuilder);
        Console.WriteLine("\nEnemy:");
        Console.WriteLine(enemy);
    }
}

```

Результат виконання програми:

```

On 'Just My Code' is enabled.
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.4\System.Collections.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.4\System.Threading.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.4\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.4\System.Runtime.InteropServices.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
Hero:
Ім'я: Герой Архілес, Ріст: 180, Тіло: Атлет, Колір волосся: Чорний, Колір очей: Blue, Інвентар: Меч, Щит, Заслуги: Врятував село

Енему:
Ім'я: Негідник Арлонг, Ріст: 190, Тіло: Качок, Колір волосся: Чорний, Колір очей: Голубий, Інвентар: Бойові крабики, Луфі, Заслуги: Намагався вбити Намі
The program '[17288] Task5.exe' has exited with code 0 (0x0).

```

Рис. 5. Результат виконання програми

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки: в ході виконання лабораторної роботи я розглянув 4 нових патенти програмування та на практиці застосував їх.

		Пасічник Б.Р			ДУ «Житомирська політехніка».22.121.19.000 – Лр2	Арк.
		Фант М.О				19
Змн.	Арк.	№ докум.	Підпис	Дата		