Національний технічний університет України «КПІ ім. І. Сікорського»

Факультет інформатики і обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

**Лабораторна робота №3**

**з дисципліни**

**«ОРПЗ .NET»**

на тему

# «Шаблони проектування. Породжуючі шаблони»

Виконав

студент групи ІС-03

Овсепян Роман

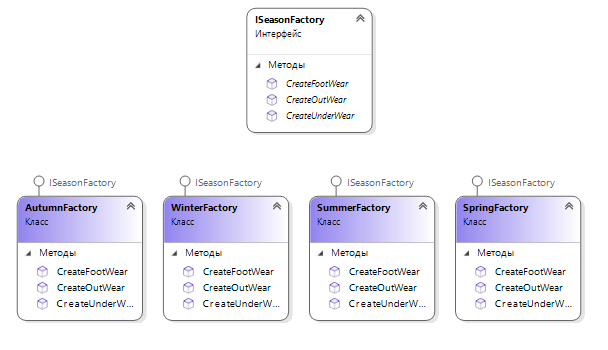
Київ – 2022

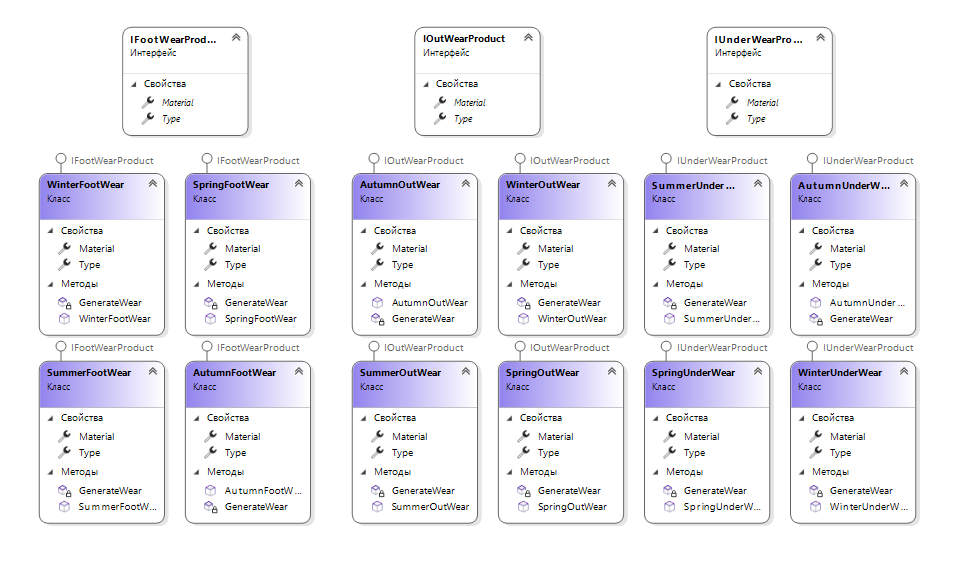


**Використання:**

Я використав породжуючий паттерн Abstract Factory, оскільки у мене є сімейство зв’язаних продуктів(верхній одяг, нижній одяг, взуття) і мені необхідні різні варіації цього сімейства(літній одяг, зимовий, тощо)

Діаграма класів:





Код

Abstract Factory:

using Lab\_3.Products;

namespace Lab\_3.Factories

{

public interface ISeasonFactory

{

IOutWearProduct CreateOutWear();

IUnderWearProduct CreateUnderWear();

IFootWearProduct CreateFootWear();

}

}

Concrete Factories:

using Lab\_3.Products;

using Lab\_3.Products.AutumnProducts;

namespace Lab\_3.Factories

{

public class AutumnFactory : ISeasonFactory

{

public IFootWearProduct CreateFootWear()

{

return new AutumnFootWear();

}

public IOutWearProduct CreateOutWear()

{

return new AutumnOutWear();

}

public IUnderWearProduct CreateUnderWear()

{

return new AutumnUnderWear();

}

}

}

using Lab\_3.Products;

using Lab\_3.Products.SpringProducts;

namespace Lab\_3.Factories

{

public class SpringFactory : ISeasonFactory

{

public IFootWearProduct CreateFootWear()

{

return new SpringFootWear();

}

public IOutWearProduct CreateOutWear()

{

return new SpringOutWear();

}

public IUnderWearProduct CreateUnderWear()

{

return new SpringUnderWear();

}

}

}

using Lab\_3.Products;

using Lab\_3.Products.SummerProducts;

namespace Lab\_3.Factories

{

public class SummerFactory : ISeasonFactory

{

public IFootWearProduct CreateFootWear()

{

return new SummerFootWear();

}

public IOutWearProduct CreateOutWear()

{

return new SummerOutWear();

}

public IUnderWearProduct CreateUnderWear()

{

return new SummerUnderWear();

}

}

}

using Lab\_3.Products;

using Lab\_3.Products.WinterProducts;

namespace Lab\_3.Factories

{

public class WinterFactory : ISeasonFactory

{

public IFootWearProduct CreateFootWear()

{

return new WinterFootWear();

}

public IOutWearProduct CreateOutWear()

{

return new WinterOutWear();

}

public IUnderWearProduct CreateUnderWear()

{

return new WinterUnderWear();

}

}

}

Wear interfaces:

namespace Lab\_3.Products

{

public interface IFootWearProduct

{

string Type { get; set; }

string Material { get; set; }

}

public interface IOutWearProduct

{

string Type { get; set; }

string Material { get; set; }

}

public interface IUnderWearProduct

{

string Type { get; set; }

string Material { get; set; }

}

}

Concrete Wears

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_3.Products.WinterProducts

{

public class WinterOutWear : IOutWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public WinterOutWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Hoodie";

Material = "Cotton";

}

}

public class WinterUnderWear : IUnderWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public WinterUnderWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Pants";

Material = "Cotton";

}

}

public class WinterFootWear : IFootWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public WinterFootWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Boots";

Material = "Fur";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_3.Products.SpringProducts

{

public class SpringOutWear : IOutWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public SpringOutWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Sweatshirt";

Material = "Jersey";

}

}

public class SpringUnderWear : IUnderWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public SpringUnderWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Pants";

Material = "Jeans";

}

}

public class SpringFootWear : IFootWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public SpringFootWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Sneakers";

Material = "Jersey";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_3.Products.AutumnProducts

{

public class AutumnOutWear : IOutWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public AutumnOutWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Sweatshirt";

Material = "Jersey";

}

}

public class AutumnUnderWear : IUnderWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public AutumnUnderWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Pants";

Material = "Jeans";

}

}

public class AutumnFootWear : IFootWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public AutumnFootWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Sneakers";

Material = "Leatherette";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_3.Products.SummerProducts

{

public class SummerFootWear : IFootWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public SummerFootWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Sandals";

Material = "Leather";

}

}

public class SummerOutWear : IOutWearProduct

{

public string Type { get ; set ; }

public string Material { get; set; }

public SummerOutWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "T-shirt";

Material = "Jersey";

}

}

public class SummerUnderWear : IUnderWearProduct

{

public string Type { get; set; }

public string Material { get; set; }

public SummerUnderWear()

{

GenerateWear();

}

private void GenerateWear()

{

Type = "Shorts";

Material = "Jersey";

}

}

}

Client.cs

using Lab\_3.Factories;

using Lab\_3.Products;

using System;

namespace Lab\_3

{

public class Client

{

private IOutWearProduct OutWear;

private IUnderWearProduct UnderWear;

private IFootWearProduct FootWear;

public Client(ISeasonFactory factory)

{

OutWear = factory.CreateOutWear();

UnderWear = factory.CreateUnderWear();

FootWear = factory.CreateFootWear();

}

public void Run()

{

Console.WriteLine($"OutWear: {OutWear.Type} {OutWear.Material}");

Console.WriteLine($"UnderWear: {UnderWear.Type} {UnderWear.Material}");

Console.WriteLine($"FootWear: {FootWear.Type} {FootWear.Material}\n");

}

}

}

Program.cs

using Lab\_3.Factories;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_3

{

internal class Program

{

static void Main(string[] args)

{

Client summerClient = new Client(new SummerFactory());

Console.WriteLine("Wear for Summer");

summerClient.Run();

Client autumnClient = new Client(new AutumnFactory());

Console.WriteLine("Wear for Autumn");

autumnClient.Run();

Client winterClient = new Client(new WinterFactory());

Console.WriteLine("Wear for Winter");

winterClient.Run();

Client springClient = new Client(new SpringFactory());

Console.WriteLine("Wear for Spring");

springClient.Run();

}

}

}