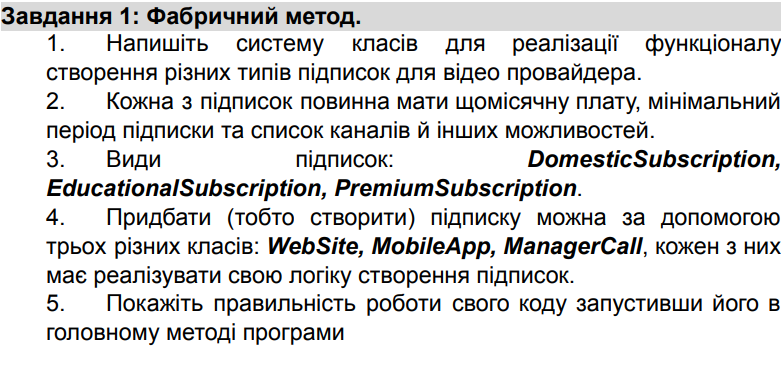
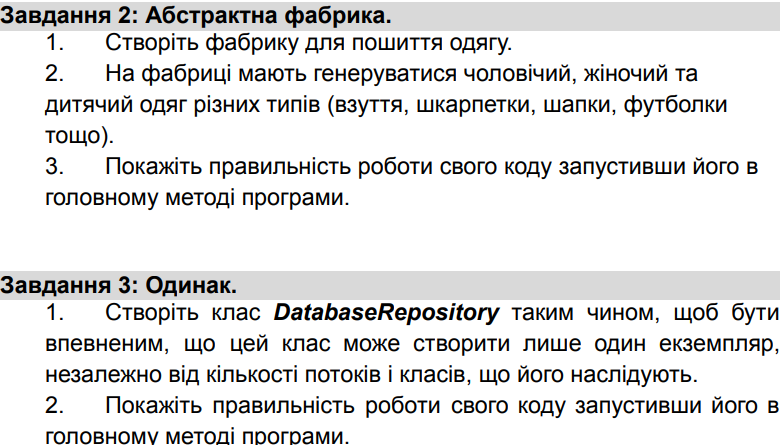
**Лабораторна робота № 2**

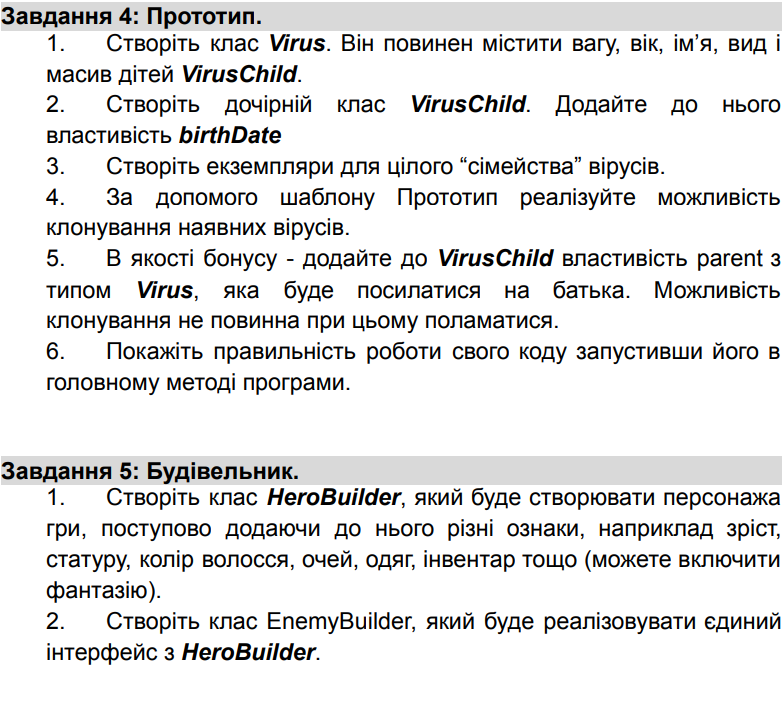
**ПОРОДЖУВАЛЬНІ ШАБЛОНИ**

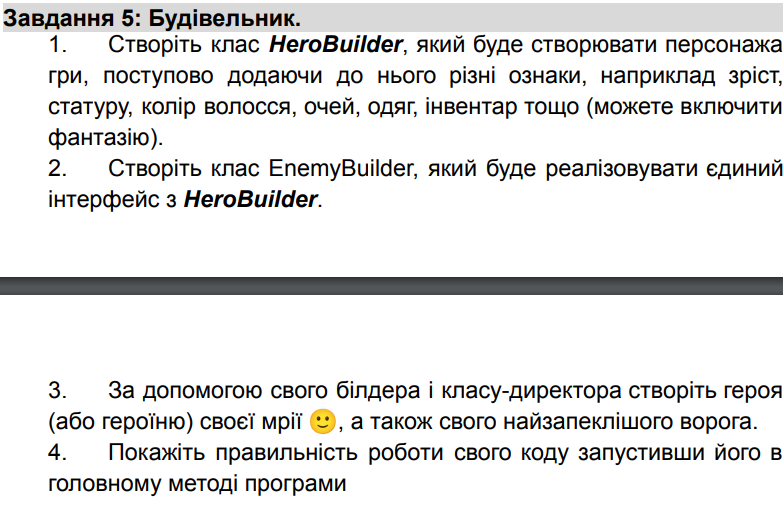
**Мета:** навчитися реалізовувати породжувальні шоблони проєктування

**Хід роботи:**

****

****





**Завдання 1:** Виконати завдання з дотриманням принципів SOLID

Абстрактний клас для передплати

# Лістинг коду:

using PaternsImpl.FactoryMethod.Entities;

using PaternsImpl.FactoryMethod.Interfaces;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternsImpl.FactoryMethod.Classes

{

internal class UsersSubscribes

{

public int Id { get; set; }

public User? User { get; set; }

public ISubscribe? Subscribe { get; set; }

}

}

# Лістинг коду:

//Клас для домашньої передплати

using PaternsImpl.FactoryMethod.Entities;

using PaternsImpl.FactoryMethod.Interfaces;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternsImpl.FactoryMethod.Classes

{

internal class DomesticSubscription : ISubscribe

{

private decimal price=300;

public int Id { get; set; }

public decimal MonthPrice { get=>price; set {

if (value > 0)

{

price = value;

}

}

}

public List<Chanel>? Chanels { get; set ; }

public DateTime Expiration { get; set; }

public string Description { get ; set; }

public int MinPeriodDays => 30;

public User Owner { get; set ; }

}

}

//Клас для освітньої передплати

namespace PaternsImpl.FactoryMethod.Classes

{

internal class StudentSubscription : ISubscribe

{

public int Id { get; set; }

public decimal MonthPrice { get; set; }

public List<Chanel>? Chanels { get; set; }

public DateTime Expiration { get; set; }

public string? Description { get; set; }

public int MinPeriodDays => 15;

public User? Owner { get ; set; }

}

}

//Клас для преміум передплати

namespace PaternsImpl.FactoryMethod.Classes

{

internal class PremiumSubscription : ISubscribe

{

public int Id { get ; set ; }

public decimal MonthPrice { get; set; }

public List<Chanel> Chanels { get; set; }

public DateTime Expiration { get; set; }

public int MinPeriodDays => 10;

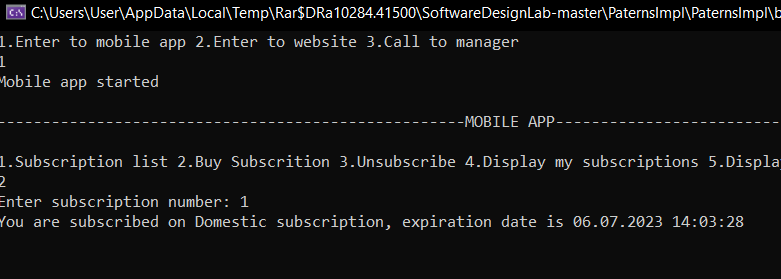
public string Description { get; set; }

public User Owner { get ; set ; }

}

}

# Результат виконання програми:



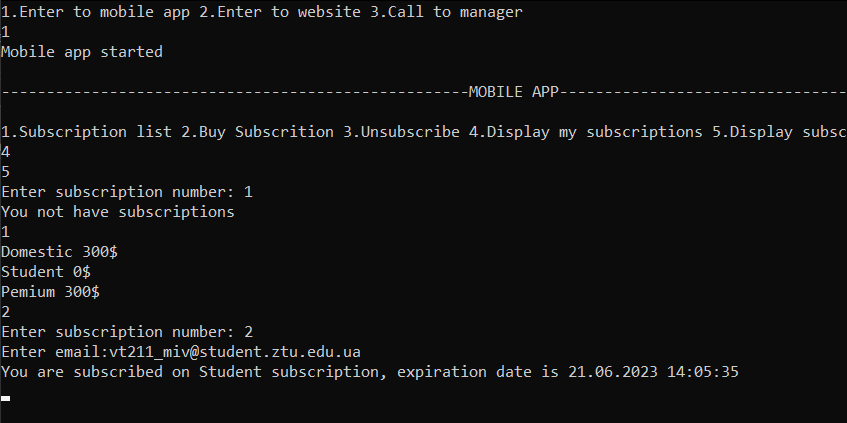


Рис.3.1-2. Реалізація завдання.

**Завдання 2:** Абстрактна фабрика.

Клас для створення фабрик

# Лістинг коду:

using PaternsImpl.AbsctractFactoryMethod.Interfaces;

using PaternsImpl.AbsctractFactoryMethod.Products;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternsImpl.AbsctractFactoryMethod.Factories

{

internal class ChildClothesFactory : IClothesFactory

{

public ISocks CreateSocks()

{

return new ChildSocks();

}

public ITrousers CreateTrousers()

{

return new ChildPants();

}

public ITShirt CreateTShirt()

{

return new ChildTshirt();

}

}

}

namespace PaternsImpl.AbsctractFactoryMethod.Factories

{

internal class MensClothesFactory : IClothesFactory

{

public ISocks CreateSocks()

{

return new MenSocks();

}

public ITrousers CreateTrousers()

{

return new MenPants() { BrandName = "Gucci",Color="Red", PocketsNumber=4,Size=15,TissueType="A1\_RL"};

}

public ITShirt CreateTShirt()

{

return new MenTShirt();

}

}

}

namespace PaternsImpl.AbsctractFactoryMethod.Factories

{

internal class WomansClotherFactory : IClothesFactory

{

public ISocks CreateSocks()

{

return new WomanSocks();

}

public ITrousers CreateTrousers()

{

return new WomanPants() { TissueType="LG2R", PocketsNumber=12};

}

public ITShirt CreateTShirt()

{

return new WomanTshit();

}

}

}

Клас для створення одягу

# Лістинг коду:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternsImpl.AbsctractFactoryMethod.Interfaces

{

internal interface IClothesFactory

{

public ITShirt CreateTShirt();

public ITrousers CreateTrousers();

public ISocks CreateSocks();

}

}

# Перевірка роботи

# Лістинг коду:

using PaternsImpl.AbsctractFactoryMethod.Interfaces;

using PaternsImpl.AbsctractFactoryMethod.Products;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternsImpl.AbsctractFactoryMethod

{

internal class App

{

private IClothesFactory \_factory;

private ITShirt \_tshirt;

private ITrousers \_trousers;

private ISocks \_socks;

public App(IClothesFactory factory)

{

\_factory= factory;

}

public void CreateTshirt()

{

\_tshirt = \_factory.CreateTShirt();

}

public void PrintTshirtInfo()

{

\_tshirt.AboutProduct();

}

public void CreateTrousers()

{

\_trousers= \_factory.CreateTrousers();

}

public void PrintTrousersInfo()

{

Console.WriteLine($"Tissue type:{\_trousers.TissueType}\nPockets number:{\_trousers.PocketsNumber}");

}

public void CreateSocks()

{

\_socks = \_factory.CreateSocks();

}

public void PrintSocksInfo()

{

Console.Write("Color: ");

\_socks.GetColor();

Console.WriteLine();

Console.Write("Size: ");

\_socks.GetSize();

Console.WriteLine();

Console.Write("Tissue type: ");

\_socks.GetTisuueType();

Console.WriteLine();

}

}

}

**Завдання 3:** Одинак.

Лістинг коду:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternsImpl.Singletone

{

sealed class DatabaseRepository

{

protected string ConnetionString = "mybasd.casd";

private static DatabaseRepository \_instance;

private static object lockRef = new object();

private DatabaseRepository()

{

Console.WriteLine("Db init...");

}

public static DatabaseRepository GetInstance()

{

lock (lockRef)

{

if (\_instance == null)

{

\_instance = new DatabaseRepository();

}

}

return \_instance;

}

}

}

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рис.3.3. Робота завдання.

**Завдання 4:** Прототип.

Лістинг коду:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace PaternsImpl.Prototype

{

internal class Virus : ICloneable

{

public List<Virus> viruses = new List<Virus>();

public Virus(int weight, int age, string? name, string? view)

{

Weight = weight;

Age = age;

Name = name;

View = view;

}

public Virus(Virus prototype)

{

Weight =prototype.Weight;

Age = prototype.Age;

Name = prototype.Name;

View = prototype.View;

}

public void AddVirus(Virus virus)

{

viruses.Add(virus);

}

public object Clone()

{

return new Virus(this);

}

public int Weight { get; set; }

public int Age { get; set; }

public string? Name { get; set; }

public string? View { get; set; }

}

}

namespace PaternsImpl.Prototype

{

internal class VirusChild : Virus, ICloneable

{

public VirusChild(int weight, int age, string? name, string? view, DateTime? birthDate, Virus parent) : base(weight, age, name, view)

{

Weight = weight;

Age = age;

Name = name;

View = view;

BirthDate = birthDate;

Parent = parent;

}

public VirusChild(VirusChild prototype):base(prototype)

{

Weight= prototype.Weight;

Age= prototype.Age;

Name= prototype.Name;

View = prototype.View;

BirthDate= prototype.BirthDate;

Parent = prototype.Parent;

}

public Virus Parent { get; set; }

public DateTime? BirthDate { get; set; }

public object Clone()

{

return new VirusChild(this);

}

}

}

**Завдання 5:** Будівельник.

Лістинг коду:

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternsImpl.Builder

{

internal interface IBuilder

{

public IBuilder AddWeapon(string weapon);

public IBuilder AddArmor(int points);

public IBuilder AddHairColor(string color);

public IBuilder AddClothes(string clothesName);

public IBuilder AddDamage(int damage);

public IBuilder AddLifeTime(int lifetime);

}

}

namespace PaternsImpl.Builder

{

internal class HeroBuilder : IBuilder

{

private HeroExample \_pers = new HeroExample();

private void \_reset()

{

\_pers = new HeroExample();

}

public HeroExample GetPersonality()

{

HeroExample pers = \_pers;

\_reset();

return pers;

}

public IBuilder AddWeapon(string weapon)

{

\_pers.SetWeapon(weapon);

return this;

}

public IBuilder AddArmor(int points)

{

\_pers.SetArmorPoints(points);

return this;

}

public IBuilder AddHairColor(string color)

{

\_pers.HairColor = color;

return this;

}

public IBuilder AddClothes(string clothesName)

{

\_pers.SetInventory(clothesName);

return this;

}

public IBuilder AddDamage(int damage)

{

\_pers.SetDamage(damage);

return this;

}

public IBuilder AddLifeTime(int lifetime)

{

\_pers.SetLifeTime(lifetime);

return this;

}

public HeroBuilder()

{

\_reset();

}

}

}

namespace PaternsImpl.Builder

{

internal class EnemyBuilder : IBuilder

{

private EnemyExample \_life = new EnemyExample();

private void \_reset()

{

\_life = new EnemyExample();

}

public EnemyBuilder()

{

\_reset();

}

public EnemyExample GetLife()

{

EnemyExample life = \_life;

\_reset();

return life;

}

public IBuilder AddWeapon(string weapon)

{

\_life.SetWeapon(weapon);

return this;

}

public IBuilder AddArmor(int points)

{

\_life.SetArmorPoints(points);

return this;

}

public IBuilder AddHairColor(string color)

{

\_life.HairColor = color;

return this;

}

public IBuilder AddClothes(string clothesName)

{

\_life.SetInventory(clothesName);

return this;

}

public IBuilder AddDamage(int damage)

{

\_life.SetDamage(damage);

return this;

}

public IBuilder AddLifeTime(int lifetime)

{

\_life.SetLifeTime(lifetime);

return this;

}

}

}

**Висновок:** на лабораторній роботі було отримано навички з реалізування породжувальних шаблонів проєктування.