Кам’янець-Подільський національний університет  
імені Івана Огієнка

КАФЕДРА КОМП’ЮТЕРНИХ НАУК

ЛАБОРАТОРНА РОБОТА #0113

з дисципліни «Об’єктно-орієнтоване програмування»

Патерни проектування в C#.

**Виконала**:  
студентка 2-го курса  
гр. KN1-B23  
Серебрянський Артем

**Прийняв**:  
старший викладач,  
Слободянюк О.В.

Дата здачі 25.05.2025р.

Кам’янець-Подільський – 2025

# Завдання для виконання

Варіант 9   
Реалізуйте Singleton для кешування даних у програмі.  
**Покликання на Git Hub:**[**https://github.com/Arhangell228/OOP\_LAB\_13\_-\_KN1-B23**](https://github.com/Arhangell228/OOP_LAB_13_-_KN1-B23)

**Код програми:**

public class ConfigurationManager

{

private static ConfigurationManager \_instance;

private static readonly object \_lock = new object();

private readonly Dictionary<string, string> \_cache;

// Приватний конструктор

private ConfigurationManager()

{

\_cache = new Dictionary<string, string>();

// Ініціалізація кешу з деякими початковими даними

InitializeCache();

}

// Реалізація Singleton

public static ConfigurationManager Instance

{

get

{

lock (\_lock)

{

if (\_instance == null)

{

\_instance = new ConfigurationManager();

}

return \_instance;

}

}

}

// Ініціалізація кешу

private void InitializeCache()

{

// Додаємо початкові конфігураційні дані

\_cache.Add("DatabaseConnection", "Server=localhost;Database=MyApp;Trusted\_Connection=True;");

\_cache.Add("ApiEndpoint", "https://api.github.com"); // Замінено з https://api.example.com на https://api.github.com

\_cache.Add("MaxRetries", "3");

}

// Отримання значення з кешу за ключем

public string GetConfiguration(string key)

{

if (\_cache.ContainsKey(key))

{

return \_cache[key];

}

return null; // або можна кидати виняток

}

// Додавання або оновлення значення в кеші

public void SetConfiguration(string key, string value)

{

lock (\_lock)

{

if (\_cache.ContainsKey(key))

{

\_cache[key] = value;

}

else

{

\_cache.Add(key, value);

}

}

}

// Очищення кешу

public void ClearCache()

{

lock (\_lock)

{

\_cache.Clear();

InitializeCache(); // Поновлюємо початкові значення

}

}

// Отримання всіх конфігураційних даних

public IReadOnlyDictionary<string, string> GetAllConfigurations()

{

return \_cache;

}

}

class Program

{

static void Main()

{

// Отримання єдиного екземпляру ConfigurationManager

var config = ConfigurationManager.Instance;

// Приклади використання

Console.WriteLine("Database Connection: " + config.GetConfiguration("DatabaseConnection"));

Console.WriteLine("API Endpoint: " + config.GetConfiguration("ApiEndpoint"));

// Додавання нової конфігурації

config.SetConfiguration("NewSetting", "NewValue");

Console.WriteLine("New Setting: " + config.GetConfiguration("NewSetting"));

// Виведення всіх конфігурацій

Console.WriteLine("\nAll configurations:");

foreach (var item in config.GetAllConfigurations())

{

Console.WriteLine($"{item.Key}: {item.Value}");

}

// Очищення кешу

config.ClearCache();

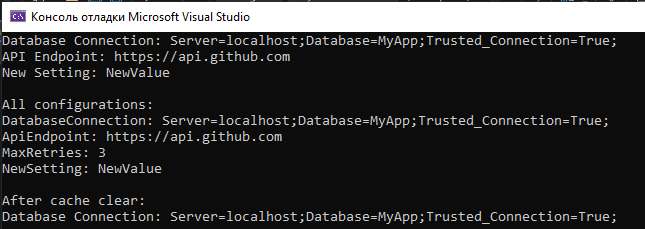
Console.WriteLine("\nAfter cache clear:");

Console.WriteLine("Database Connection: " + config.GetConfiguration("DatabaseConnection"));

}

}

**Результат:**

****

Варіант 9

Реалізуйте Observer для повідомлення про нові публікації в соціальній мережі.  
**Покликання на Git Hub:**[**https://github.com/Arhangell228/OOP\_LAB\_13\_-\_KN1-B23**](https://github.com/Arhangell228/OOP_LAB_13_-_KN1-B23)

**Код програми:**

public interface IObserver

{

void Update(string message);

}

public interface ISubject

{

void RegisterObserver(IObserver observer);

void RemoveObserver(IObserver observer);

void NotifyObservers();

}

public class SocialNetwork : ISubject

{

private readonly List<IObserver> \_observers = new List<IObserver>();

private string \_latestPost;

public void RegisterObserver(IObserver observer)

{

if (!\_observers.Contains(observer))

{

\_observers.Add(observer);

}

}

public void RemoveObserver(IObserver observer)

{

\_observers.Remove(observer);

}

public void NotifyObservers()

{

foreach (var observer in \_observers)

{

observer.Update(\_latestPost);

}

}

public void CreatePost(string postContent)

{

\_latestPost = postContent;

Console.WriteLine($"New post created: {postContent}");

NotifyObservers();

}

}

public class User : IObserver

{

private readonly string \_username;

public User(string username)

{

\_username = username;

}

public void Update(string message)

{

Console.WriteLine($"{\_username} received notification: {message}");

}

}

class Program

{

static void Main()

{

// Створюємо соціальну мережу

var socialNetwork = new SocialNetwork();

var user1 = new User("Alice");

var user2 = new User("Bob");

var user3 = new User("Charlie");

// Підписуємо користувачів на оновлення

socialNetwork.RegisterObserver(user1);

socialNetwork.RegisterObserver(user2);

socialNetwork.CreatePost("Hello, this is the first post!");

// Додаємо ще одного підписника

socialNetwork.RegisterObserver(user3);

socialNetwork.CreatePost("Check out my new photo!");

// Відписуємо одного користувача

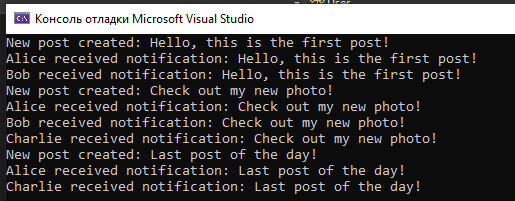
socialNetwork.RemoveObserver(user2);

socialNetwork.CreatePost("Last post of the day!");

}

}

**Результат:**

****