Лабораторна робота зараховується тільки при наявності в звіті повних відповідей на всі питання.

1. На мові програмування C# напишіть код паттерну Singleton.
2. Докажіть в коді, що Ваш клас дійсно є Singleton.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp6

{

public class Singleton

{

public static Singleton Instance { get; } = new Singleton();

static Singleton()

{

}

private Singleton()

{

}

public void IsSingleton()

{

Console.WriteLine("Да");

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Ты действительно синглтон?");

Singleton.Instance.IsSingleton();

Console.ReadKey();

}

}

}

Контрольні питання:

1. Дайте визначення паттернів.

**Паттерни** – це відповідь на якусь проблему програмування, яка в процесі своєї еволюції обігнала інші способи рішення цієї проблеми.

1. Хто такі GoF?

Команда авторів книги «**Design Patterns** — Elements of Reusable Object-Oriented Software» відома загальності під назвою «Банда четырёх» (англ. Gang of Four, часто скорочується до GoF) (**Еріха Гамма**, Ричарда Хелма (Richard Helm), Ральфа Джонсона (Ralph Johnson) и Джона Вліссідеса (John Vlissides))

В цій книзі описані 23 шаблони проектування.

1. Скільки паттернів GoF Ви знаєте?

* [Абстрактная фабрика (Abstract Factory)](https://bool.dev/blog/detail/porozhdayushchie-shablony-abstract-factory)
* [Строитель (Builder)](https://bool.dev/blog/detail/porozhdayushchie-patterny-pattern-builder)
* [Фабричный метод (Factory Method)](https://bool.dev/blog/detail/porozhdayushchie-patterny-fabrichnyy-metod)
* [Прототип (Prototype)](https://bool.dev/blog/detail/porozhdayushchie-patterny-prototype-sharp)
* [Одиночка (Singleton)](https://bool.dev/blog/detail/pattern-singleton)
* [Адаптер (Adapter)](https://bool.dev/blog/detail/strukturnye-patterny-adapter-csharp)
* [Мост (Bridge)](https://jopr.org/blog/detail/strukturnye-patterny-most-csharp)
* [Компоновщик (Composite)](https://bool.dev/blog/detail/strukturnye-patterny-komponovshchik-csharp)
* [Декоратор (Decorator)](https://bool.dev/blog/detail/strukturnye-patterny-decorator-csharp)
* [Фасад (Facade)](https://bool.dev/blog/detail/strukturnye-patterny-facade-csharp)
* [Приспособленец (Flyweight)](https://bool.dev/blog/detail/strukturnye-shablony-flyweight-csharp)
* [Заместитель (Proxy)](https://bool.dev/blog/detail/strukturnye-patterny-zamestitel-c)
* [Цепочка обязанностей (Chain of responsibility)](https://bool.dev/blog/detail/pattern-tsepochka-obyazannostey-chain-of-responsibility)
* [Команда (Command)](https://bool.dev/blog/detail/pattern-komanda)
* [Интерпретатор (Interpreter)](https://bool.dev/blog/detail/povedencheskie-patterny-interpretator-csharp)
* [Итератор (Iterator)](https://bool.dev/blog/detail/pattern-iterator-iterator)
* [Посредник (Mediator)](https://bool.dev/blog/detail/pattern-posrednik-mediator)
* [Хранитель (Memento)](https://bool.dev/blog/detail/povedencheskie-patterny-memento-csharp)
* [Наблюдатель (Observer)](https://bool.dev/blog/detail/pattern-nablyudatel-observer)
* [Состояние (State)](https://bool.dev/blog/detail/pattern-sostoyanie)
* [Стратегия (Strategy)](https://bool.dev/blog/detail/pattern-strategiya-strategy)
* [Шаблонный метод (Template method)](https://bool.dev/blog/detail/pattern--shablonnyy-metod-template-method)
* [Посетитель (Visitor)](https://bool.dev/blog/detail/pattern-posetitel)

1. Скільки всього існує паттернів?

Кількість паттернів росте з потребами, тому на це питання точної відповіді дати не можна

5.         На які групи поділяються паттерни GoF?

* Породжуючі (Creational)
* Структурні (Structural)
* Патерни поведінки (Поведенческие)(Behavioral)

6.         Переваги застосування паттернів

* Застосування шаблонів проектування підвищує ефективність праці.
* Шаблони проектування, вивчені належним чином, істотно допомагають спільного розуміння основних принципів об'єктно-орієнтованого проектування.
* Шаблони проектування дозволяють розробнику знаходити проектні рішення для складних проблем, не створюючи громіздкої ієрархії успадкування класів. Навіть якщо шаблони не використовуються в проекті безпосередньо, одне тільки зменшення розміру ієрархій успадкування класів уже сприятиме підвищенню якості проекту

7.         Структура паттерну Singleton. Його призначення.

**Основа паттерну – программне забеспечення одного екземпляру класа.**

**Конструктор повинен бути прихований від клієнтів. Виклик методу повинен стати єдиним способом отримати об'єкт цього класу.**

Призначення паттерна Сінглтон (або Одинак) полягає в забезпеченні доступу до деякого унікального об'єкту з будь-якої точки прикладання. Під унікальністю мається на увазі, що такий об'єкт існує в пам'яті в єдиному екземплярі і інші екземпляри створені бути не можуть.

8.         З яким типом паттернів можна пов‘язати вислів:»пироги должен печь пирожник, а сапоги точить сапожник»?

Породжуючий тип

9.         Эрик Фримен, Элизабет Фримен – «Паттерны проектирования». Знайти в інтернеті і прочитати загальний опис.

