**10 Паттерны проектирования**

Задание 1: Реализация паттерна Singleton

Условие: В каждой задаче необходимо создать один единственный

экземпляр класса, ограничив его создание через приватный конструктор и

предоставляя доступ через статический метод.

Листинг программы:

using System;

using System.Collections.Generic;

public class ConfigManager{

private static readonly ConfigManager \_instance = new ConfigManager();

private readonly Dictionary<string, string> \_configurations;

private ConfigManager() {

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

}

public static ConfigManager Instance {

get { return \_instance; }

}

public void SetConfig(string key, string value) {

\_configurations[key] = value;

}

public string GetConfig(string key) {

\_configurations.TryGetValue(key, out var value);

return value;

}

}

class Program{

static void Main(string[] args) {

ConfigManager.Instance.SetConfig("AppName", "My Application");

ConfigManager.Instance.SetConfig("Version", "1.0.0");

string appName = ConfigManager.Instance.GetConfig("AppName");

string version = ConfigManager.Instance.GetConfig("Version");

Console.WriteLine($"App Name: {appName}");

Console.WriteLine($"Version: {version}");

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| "AppName", "My Application" "Version", "1.0.0" | App Name: My Application  Version: 1.0.0 |

Анализ результатов:



Рисунок 1 – Результат работы программы