Text

Description automatically generated with medium confidence

wzorce projektowe

Sprawozdanie

Ćwiczenie 1 – Singleton Zadanie

Adam Zajler

Informatyka, programowanie

34\_Inf\_P\_NW\_6

Rok akademicki 2023/24

# Cel

W ramach ćwiczenia powstał program bazujący na wzorcu Singleton. Program posiada możliwość dodania i wyświetlenia wszystkich komunikatów. Wzorzec został wybrany z tego względu iż charakteryzuje się tym że posiada jedną globalną instancję, której nie można nadpisać czy też stworzyć nowej.   
Ma to swoje zastosowanie w przypadku chęci posiadania globalnego dostępu do np. połączenia z bazą danych; chociaż w tym programie jest to system zapisu i odczytu komunikatów; a wzorzec Singleton idealnie wpasowuje się w specyfikę tego zadania.

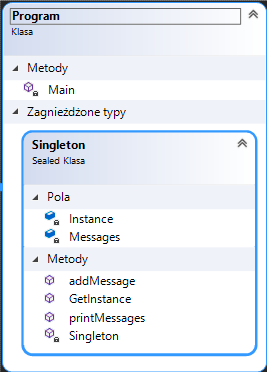
# Opis rozwiązania

W tej części proszę opisać krótko propozycję rozwiązania zadania:

Program posiada dwie metody:

- pierwsza „addMessage” przyjmuje jako parametr string, a później zapisuje go w zmiennej Messages, która jest zabezpieczona i prywatna

- druga „printMessages” wyświetla wszystkie wcześniej zapisane komunikaty. W przypadku braku zapisanych poinformuje o tym użytkownika.



# Implementacja

using System;

using System.Collections.Generic;

namespace Ćwiczenie\_1\_\_\_Singleton

{

class Program

{

public sealed class Singleton

{

private static List<string> Messages = new List<string>();

private Singleton() { }

private static Singleton Instance = null;

public static Singleton GetInstance()

{

if (Instance == null)

{

Instance = new Singleton();

}

return Instance;

}

public void addMessage(string message)

{

Messages.Add(message);

}

public void printMessages()

{

if(Messages.Count == 0)

{

Console.WriteLine("Brak komunikatów");

return;

}

Console.WriteLine("Lista komunikatów: \n");

for (var i = 0; i < Messages.Count; i++)

{

Console.WriteLine("\t{0}: {1}", i+1, Messages[i]);

}

}

}

static void Main(string[] args)

{

Console.WriteLine("Ćwiczenie 1 - Singleton\n\n");

Singleton s1 = Singleton.GetInstance();

s1.addMessage("Komunikat A");

s1.addMessage("Komunikat B");

s1.printMessages();

}

}

}

# Podsumowanie

Podsumowując tą pracę można jednoznacznie stwierdzić że wzorzec Singleton idealnie pasuje do tego zadania. Klasa me jedną globalną instancję, a system dodawania i wyświetlania komunikatów właśnie powinien mieć taki zasięg.  
Implementacja powiodła się, ponieważ nie da się utworzyć dwóch instancji tej samej klasy.

# Lista załączników

[AdamZajler/wzorce-projektowe (github.com)](https://github.com/AdamZajler/wzorce-projektowe/tree/main)