Text

Description automatically generated with medium confidence

wzorce projektowe

Sprawozdanie

Zadanie singleton/1

Patryk Figas

Informatyka, programowanie

Grupa 34\_Inf\_P\_NW\_6

Rok akademicki 2024/25

# Cel

Dokument powstał w celu przedstawienia rozwiązania zadania polegającego na implementacji klasy do logowania komunikatów z zastosowaniem wzorca projektowego Singleton.

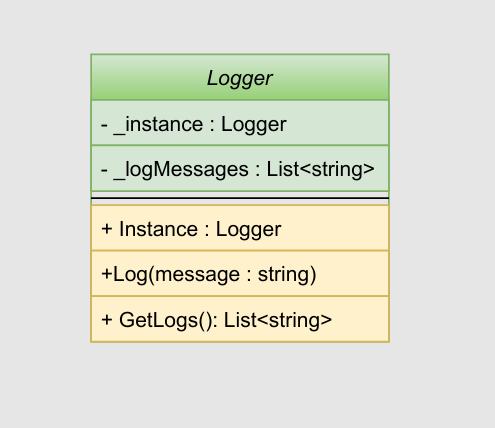
W ramach ćwiczenia zaprojektowano klasę za pomocą „pseudokodu”, diagramu UML i implementacji klasy do programu oraz użycie jej w programie Main.

Zastosowano wzorzec Singleton, który zapewnia, że klasa ma tylko jedną instancję i pozwala z niej korzystać z poziomu całego programu (static).

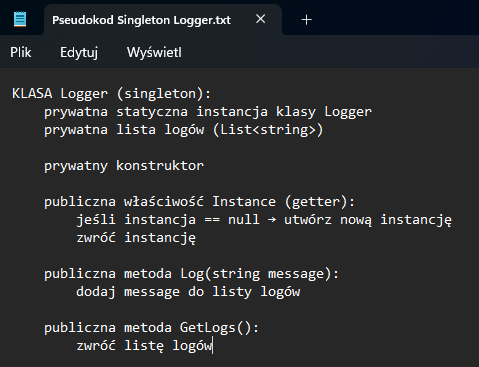
# Opis rozwiązania

W ramach zadania zaprojektowano klasę **Logger**, której celem jest rejestrowanie komunikatów w aplikacji. Klasa została zaimplementowana zgodnie ze wzorcem projektowym ****Singleton****, co oznacza, że przez cały czas działania programu istnieje tylko jedna instancja tej klasy. Dzięki temu wszystkie logi są centralnie gromadzone i dostępne z każdego miejsca w aplikacji. Komunikaty są przechowywane w liście typu **List<string>**, a ich dodawanie odbywa się za pomocą metody **Log()**.

* diagram klasy **Logger**

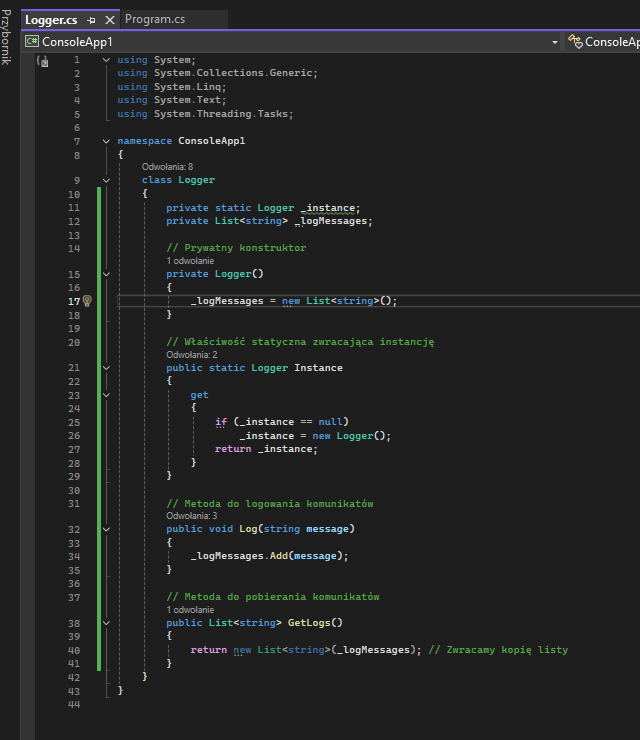


* pseudokod klasy **Logger**

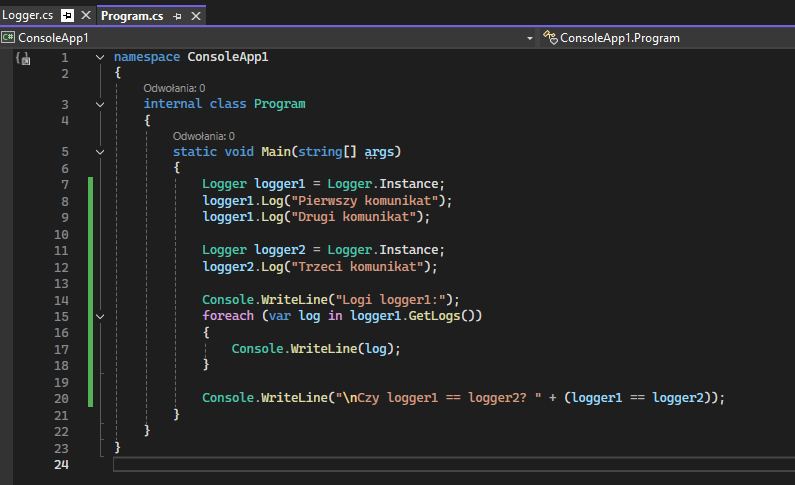


# Implementacja

* kod klasy **Logger**:



* kod klasy **Program**

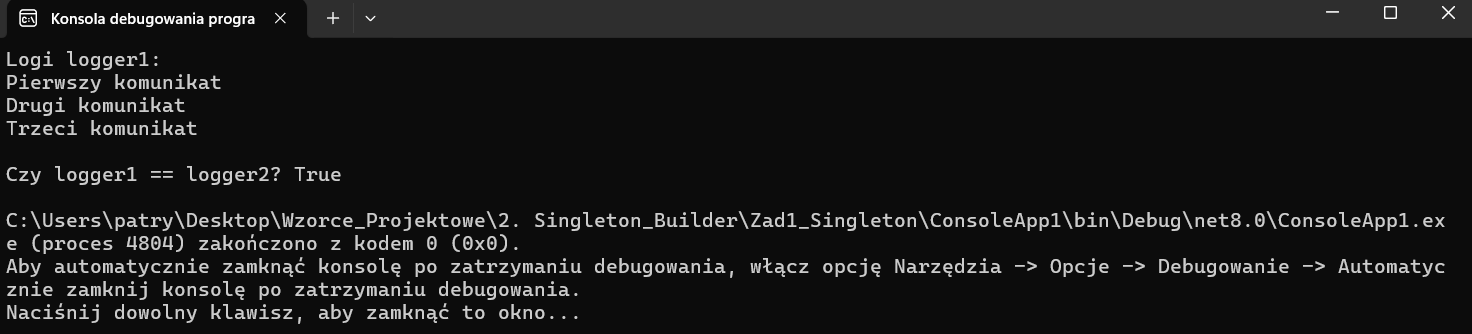


# Podsumowanie

W zadaniu zastosowano wzorzec projektowy ****Singleton****, ponieważ jego głównym celem jest zapewnienie, że dana klasa posiada tylko **jedną instancję** w całym programie. Jest to przydatne w przypadku klasy **Logger**, której instancja powinna być **wspólna dla całej aplikacji**, aby wszystkie logi były przechowywane w jednym miejscu.

Implementacja wzorca Singleton **powiodła się**, ponieważ:

* wszystkie odwołania do **Logger.Instance** zwracają **tę samą instancję**,
* rejestrowane komunikaty są przechowywane w jednej liście,
* test wykazał, że **logger1 == logger2**, co potwierdza jednolitość instancji.



Zaproponowany wzorzec jest w tym przypadku **najbardziej odpowiedni**, ponieważ: inne wzorce, takie jak ****Factory Method**** czy ****Prototype****, nie pasują do sytuacji, w której potrzebujemy jednej instancji. Z tego powodu wzorzec Singleton jest **najlepszym wyborem** do realizacji tego typu zadania.

# Lista załączników

W tej części proszę podać listę załączników z kodem lub link do zdalnego repozytorium.

Repozytorium GITHUB: