Text

Description automatically generated with medium confidence

wzorce projektowe

Sprawozdanie

ZADANIE Adapter biblioteka logowania

Patryk Figas

Informatyka, programowanie

Grupa 34\_Inf\_P\_NW\_6

Rok akademicki 2024/25

# Cel

Dokument powstał w celu przedstawienia sposobu integracji nowej biblioteki logowania z istniejącym systemem, który używa innego, starszego interfejsu logowania.  
 W ramach ćwiczenia zaimplementowano wzorzec projektowy **Adapter**, który umożliwia zachowanie istniejącego kodu aplikacji, a jednocześnie pozwala korzystać z nowej biblioteki bez jej bezpośredniego modyfikowania.

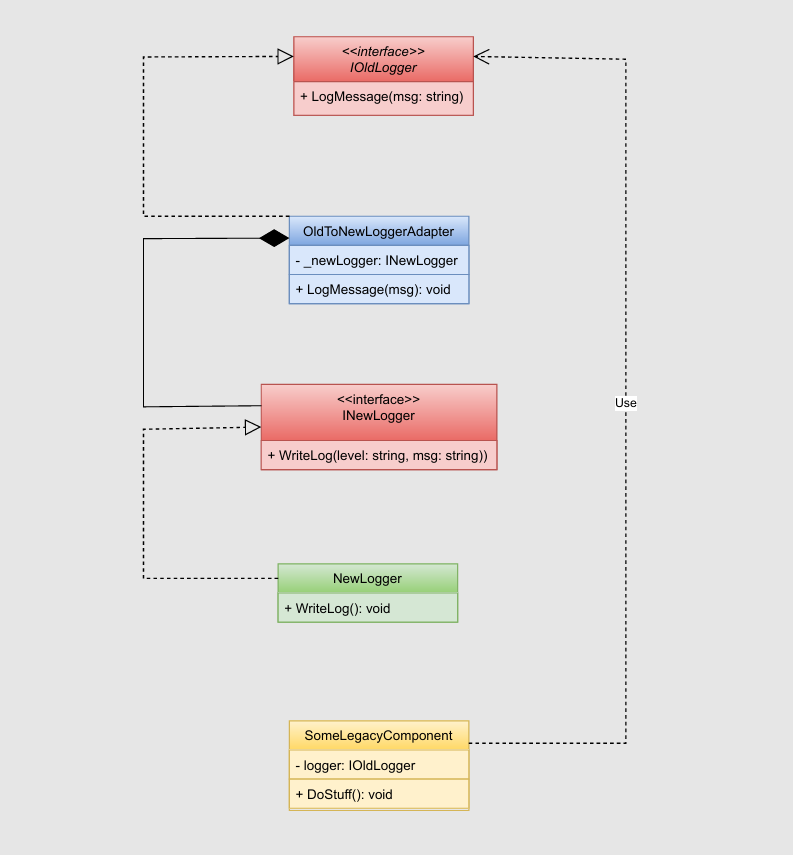
W ramach ćwiczenia zaprojektowano interfejsy i klasy za pomocą „pseudokodu”, diagramu UML i implementacji klas do programu oraz użycie jej w programie Main.

W ramach ćwiczenia zaimplementowano wzorzec projektowy **Adapter**, który umożliwia zachowanie istniejącego kodu aplikacji, a jednocześnie pozwala korzystać z nowej biblioteki bez jej bezpośredniego modyfikowania.

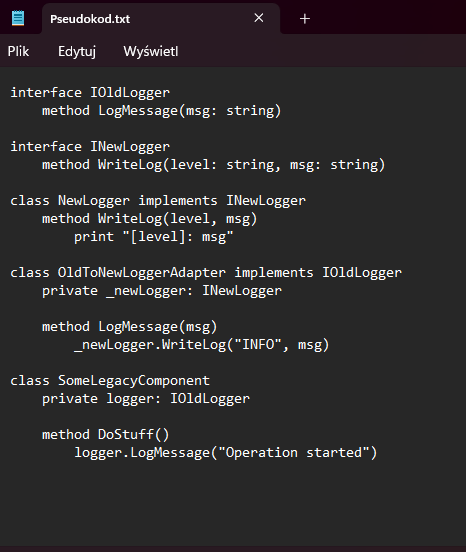
# Opis rozwiązania

W istniejącym systemie funkcjonował interfejs **IOldLogger** z metodą **LogMessage(string)**. Nowa biblioteka logowania operuje na interfejsie **INewLogger**, który wymaga metody **WriteLog(string level, string message)**.  
 Zamiast zmieniać całą aplikację, zastosowano klasę adaptera **OldToNewLoggerAdapter**, która implementuje stary interfejs i wewnętrznie deleguje wywołania do nowej biblioteki. Dzięki temu możliwe jest stopniowe przejście na nowy system bez ryzyka błędów i naruszania istniejącego kodu.

* Propozycja diagramu klas

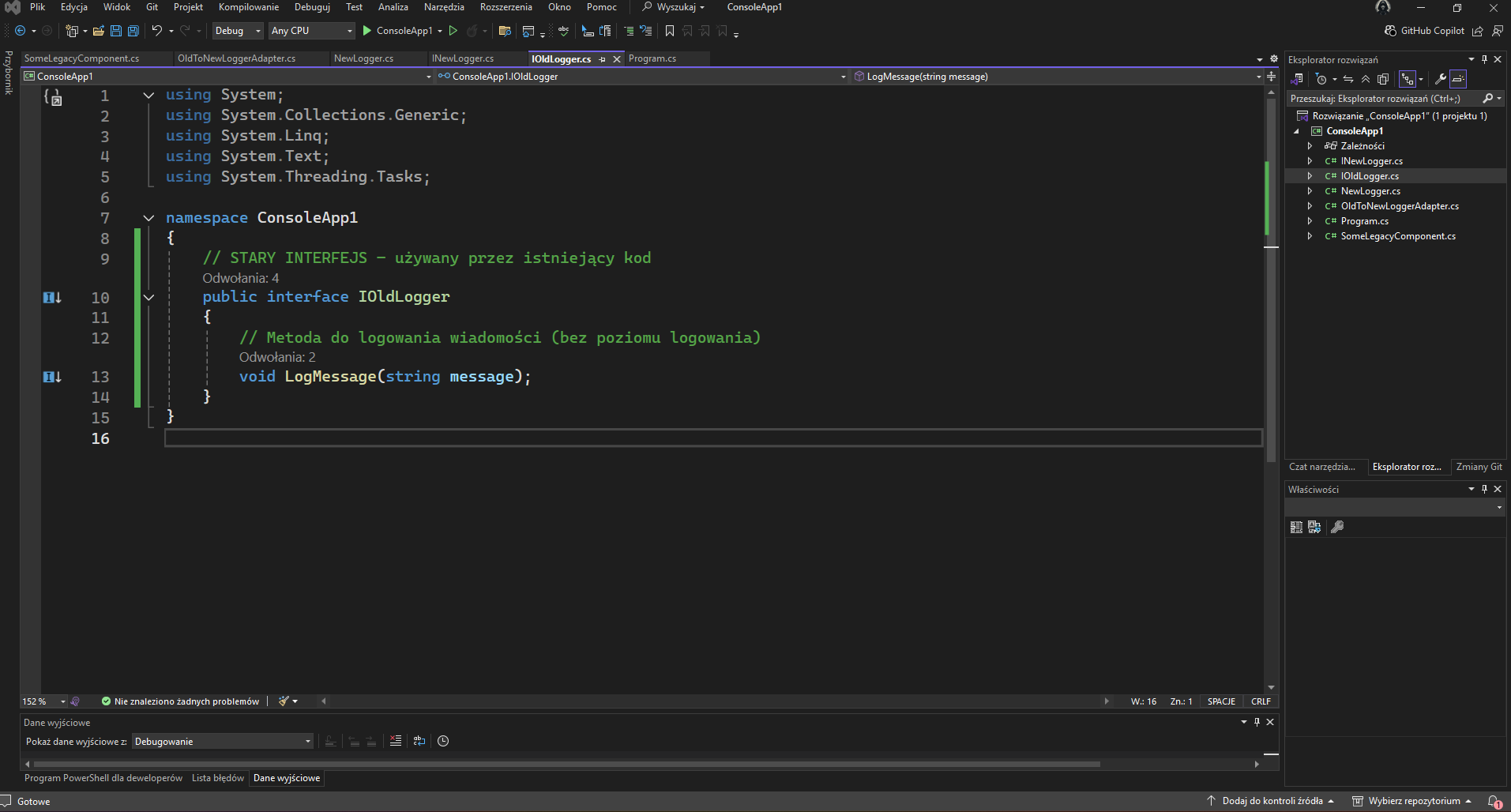


* Pseudokod

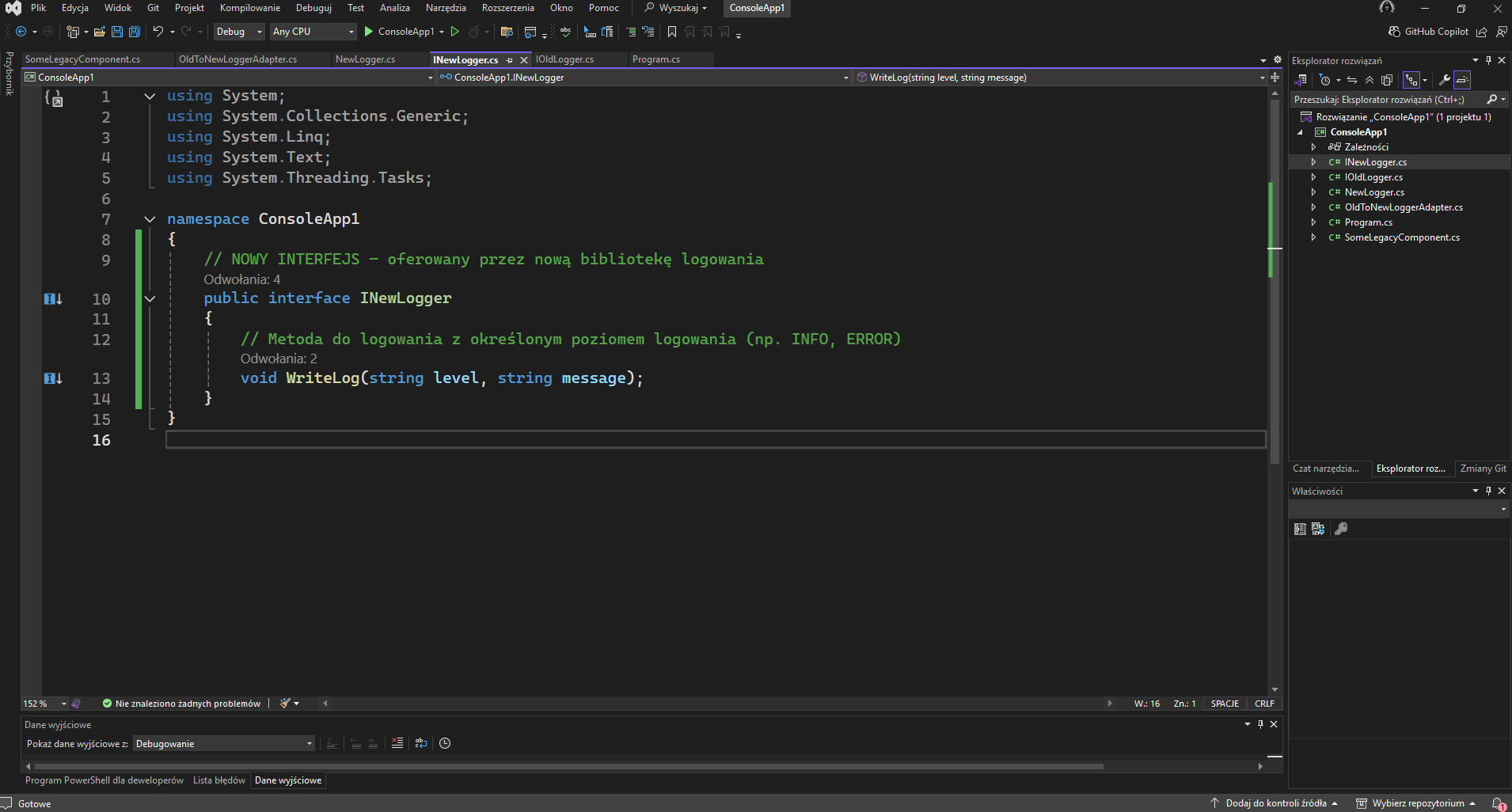


# Implementacja

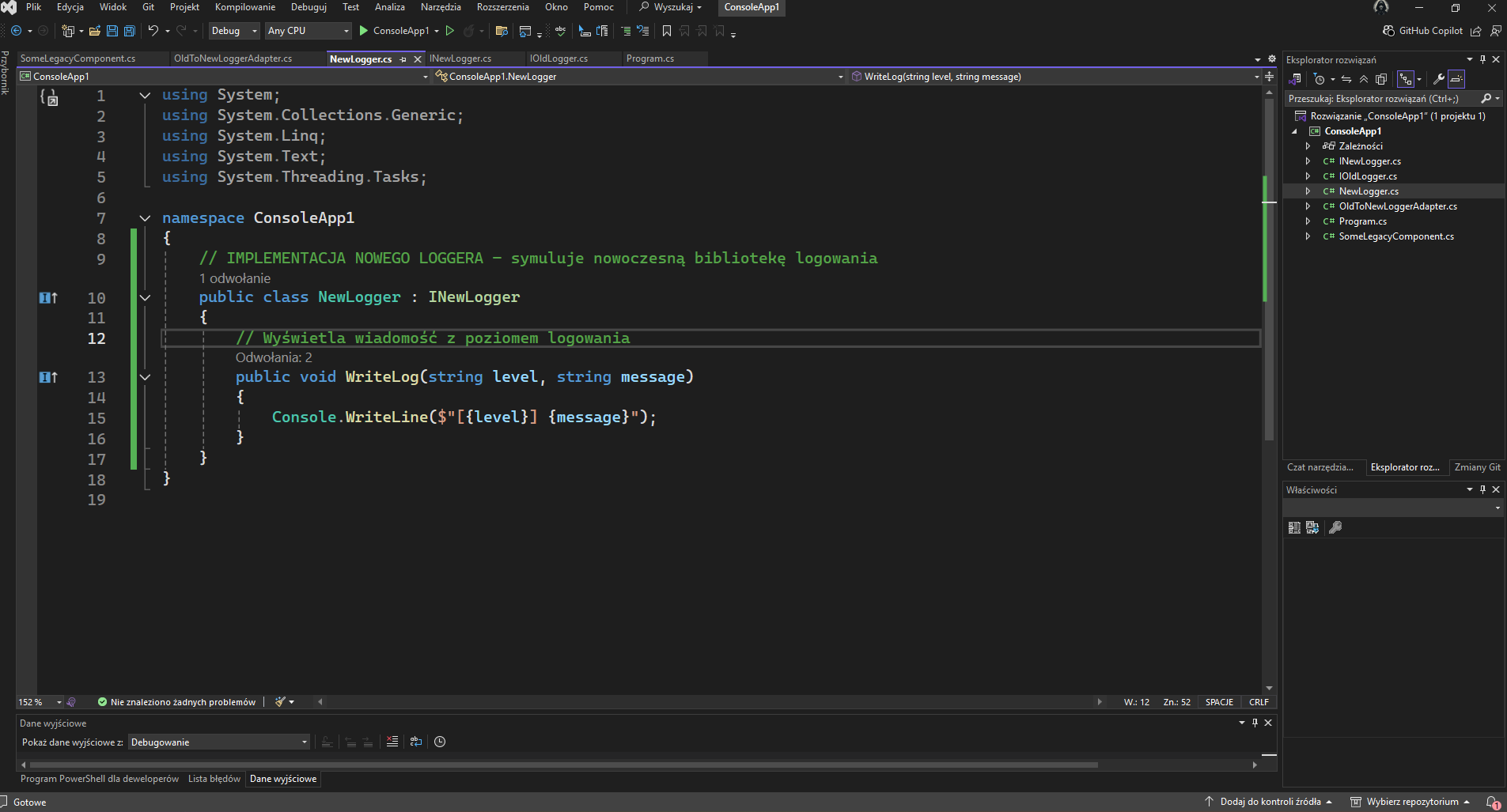
* Kod **interfejsu IOldLogger.cs**



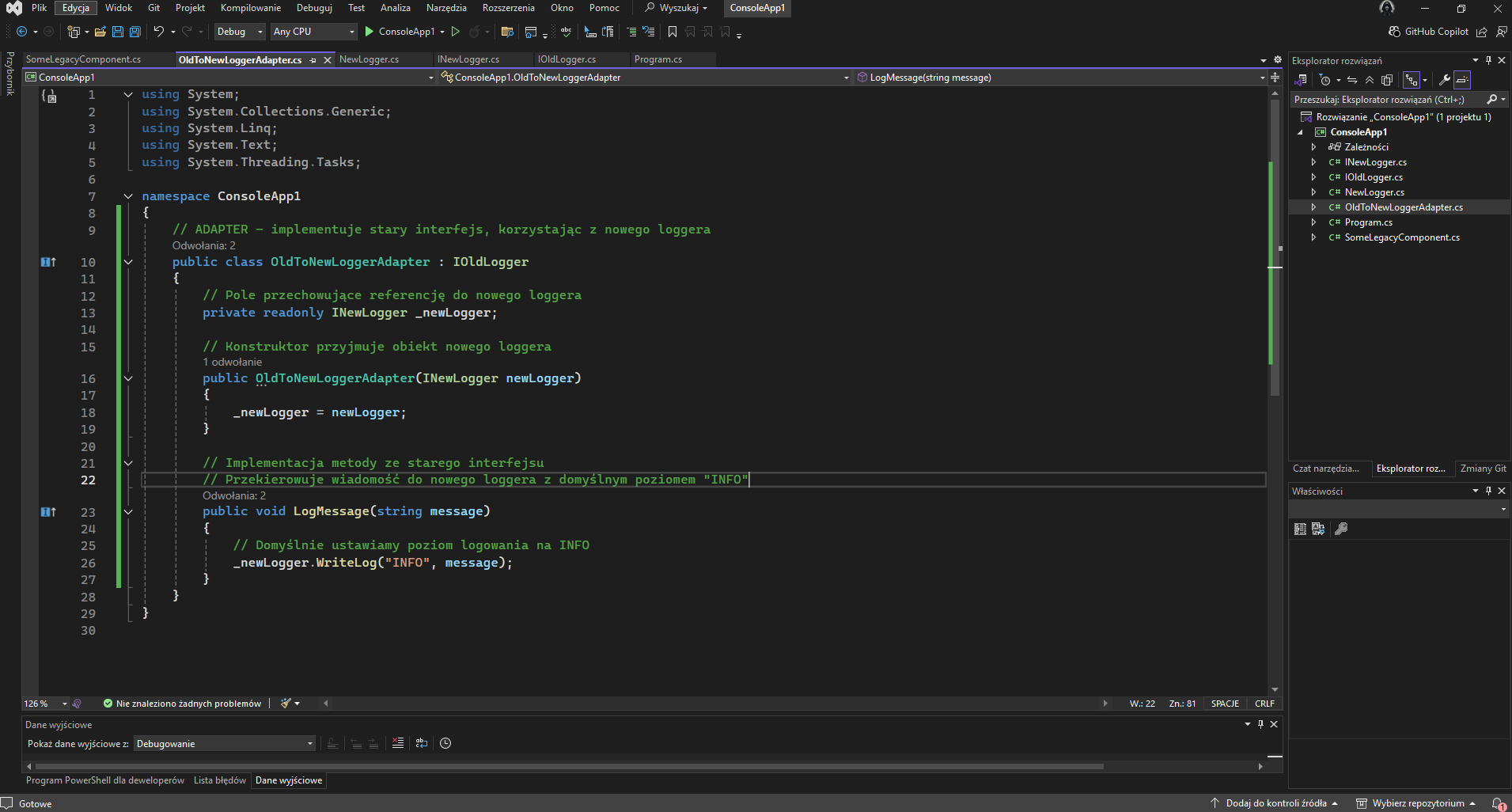
* Kod **interfejsu INewLogger.cs**



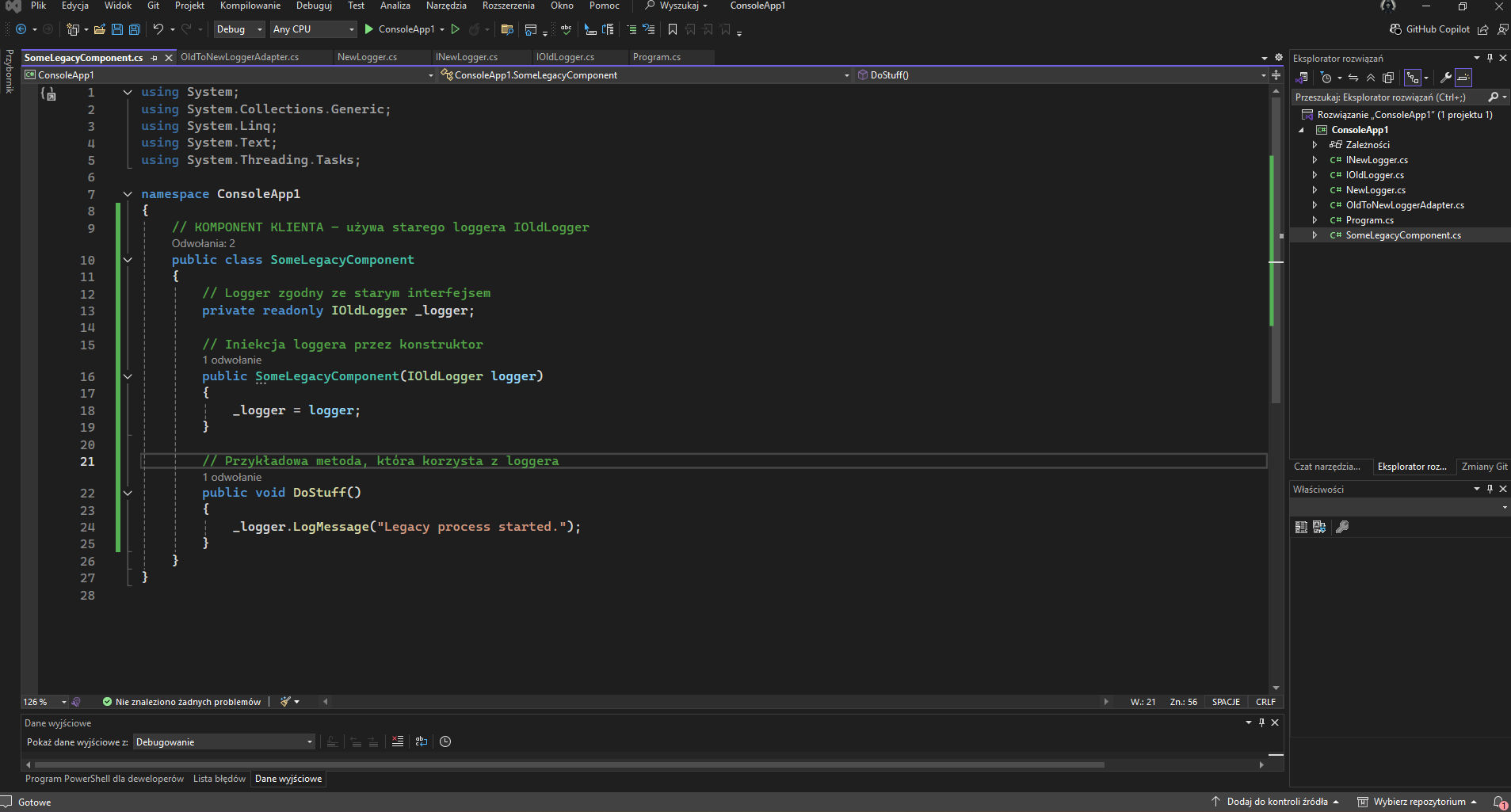
* Kod **klasy NewLogger.cs**



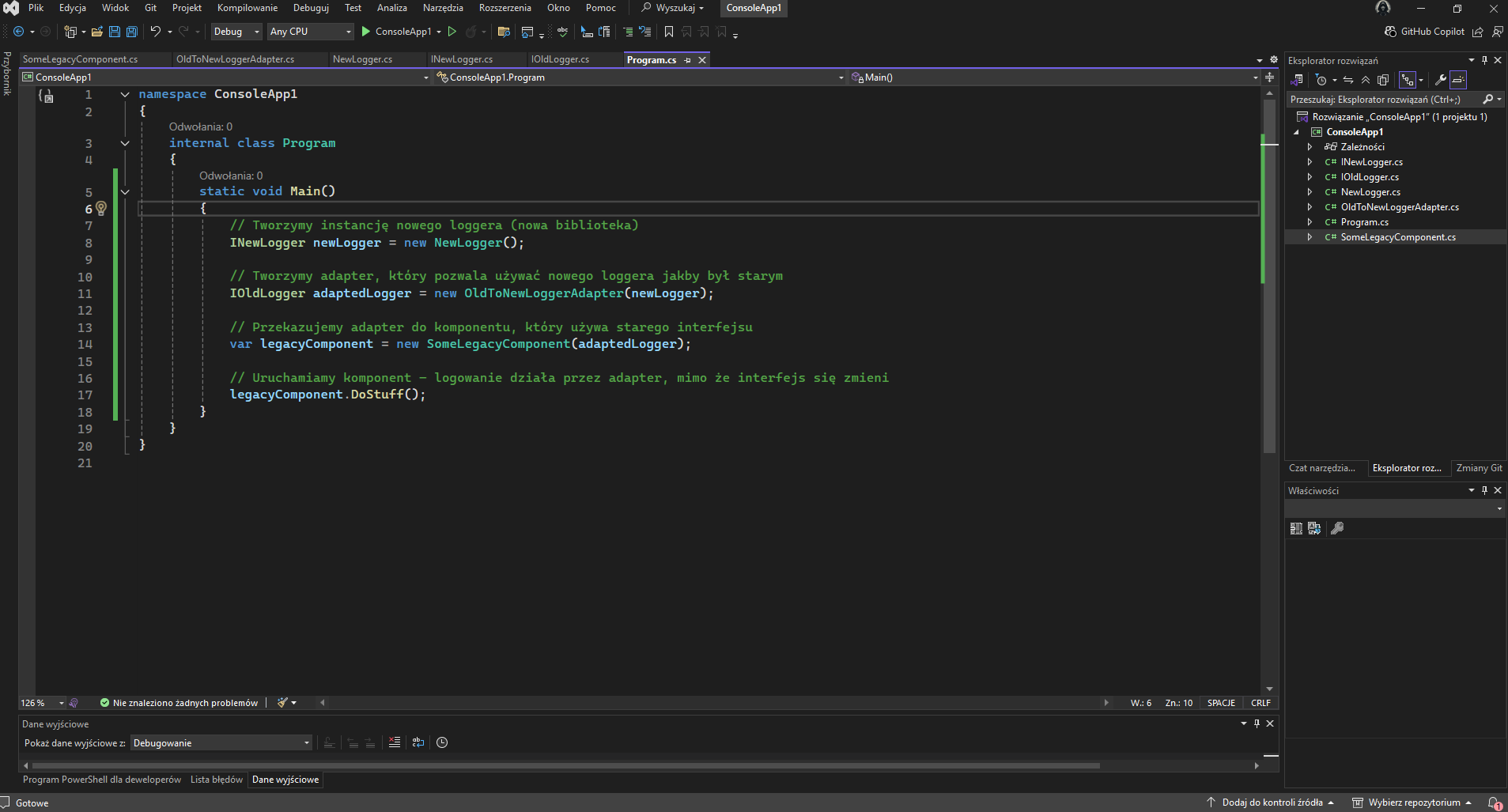
* Kod **klasy OldToNewLoggerAdapter.cs**



* Kod **klasy SomeLegacyComponent.cs**



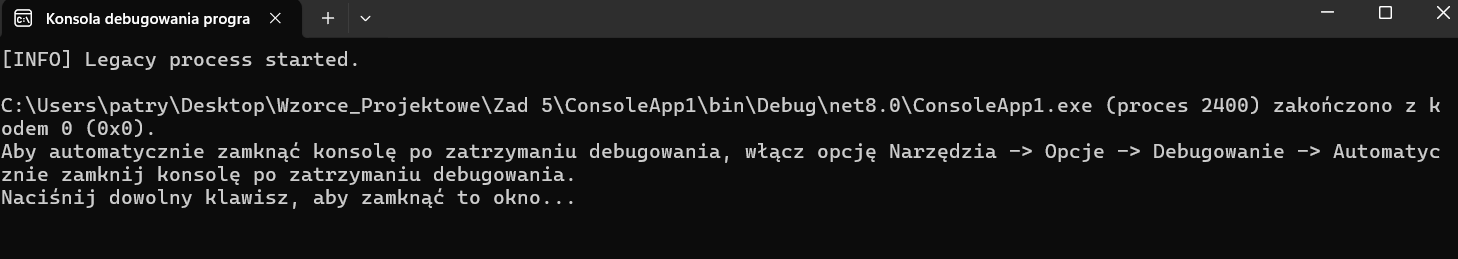
* Kod **klasy Program.cs**



# Podsumowanie

W rozwiązaniu zastosowano wzorzec projektowy **Adapter**, ponieważ idealnie odpowiadał on na potrzebę połączenia dwóch niespójnych interfejsów – nowego i starego loggera. Adapter pozwolił utrzymać działający kod klienta bez jego modyfikacji, co znacznie ułatwia migrację systemu.

Nowa biblioteka logowania została zintegrowana bez naruszania istniejącej architektury. Kod stał się bardziej elastyczny i zgodny z zasadą **Open/Closed** z zasad SOLID – można go rozszerzyć bez ingerencji w już działające komponenty.



Alternatywne wzorce, takie jak **Fasada** czy **Dekorator**, nie byłyby tu odpowiednie, ponieważ nie chodziło o uproszczenie interfejsu ani o dodanie funkcjonalności, lecz o dostosowanie interfejsu. Dlatego wzorzec **Adapter** był najlepszym wyborem.

# Lista załączników

Repozytorium GITHUB z projektem:

https://github.com/PatrykFigas/Wzorce-projektowe.git