Text

Description automatically generated with medium confidence

wzorce projektowe

Sprawozdanie

ZADANIE chain of responsibility system zarządzania biletami

Patryk Figas

Informatyka, programowanie

Grupa 34\_Inf\_P\_NW\_6

Rok akademicki 2024/25

# Cel

Celem niniejszego dokumentu jest przedstawienie rozwiązania problemu obsługi zgłoszeń (tzw. „ticketów”) w systemie wsparcia technicznego poprzez zastosowanie wzorca projektowego **Chain of Responsibility**.

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

Zastosowano wzorzec **Łańcuch odpowiedzialności (Chain of Responsibility)**.

# Opis rozwiązania

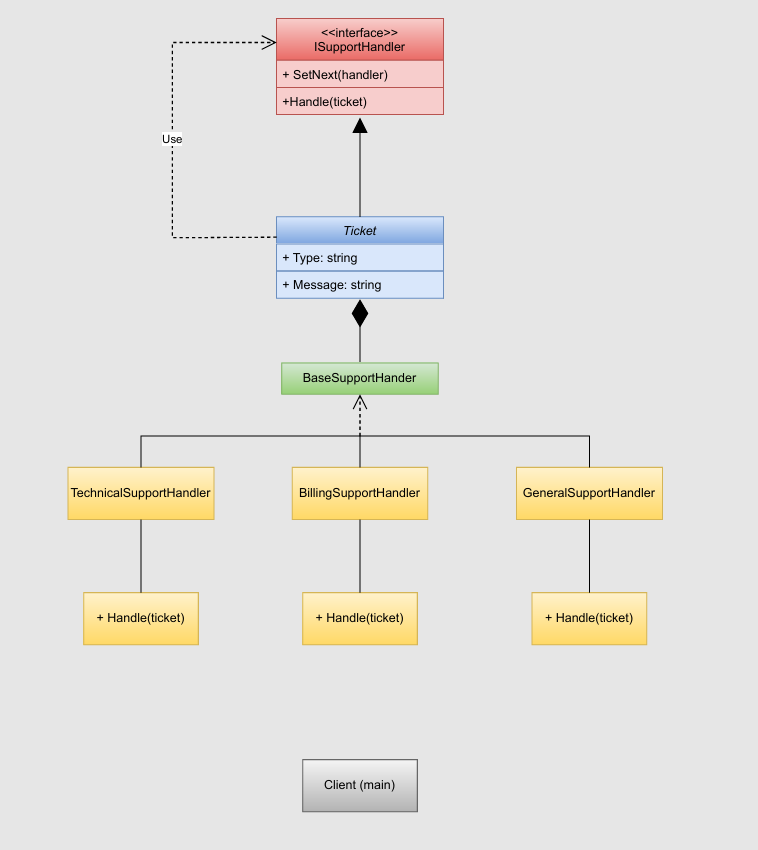
Rozwiązanie opiera się na przekazywaniu zgłoszeń między kolejnymi handlerami (klasami obsługi) do momentu, aż jeden z nich rozpozna typ zgłoszenia i je obsłuży. Każdy handler implementuje wspólny **interfejs** **ISupportHandler**, który zapewnia metodę **Handle()** oraz **SetNext()**, pozwalającą ustawić kolejnego handlera w łańcuchu.

System obsługuje trzy główne typy zgłoszeń:

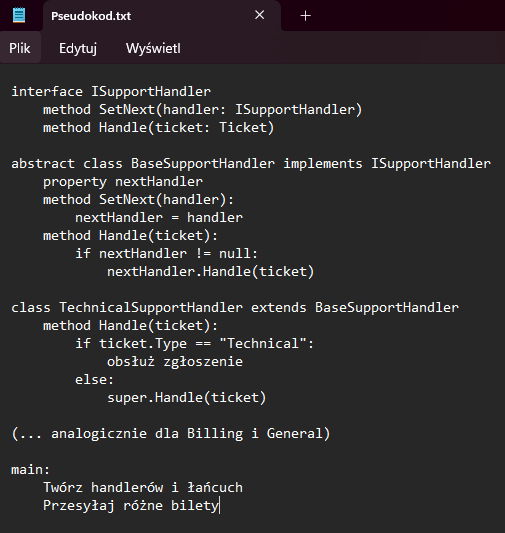
* **Technical** – obsługiwane przez **TechnicalSupportHandler**
* **Billing** – obsługiwane przez **BillingSupportHandler**
* **General** – obsługiwane przez **GeneralSupportHandler**

Zgłoszenia, które nie pasują do żadnego typu, są odrzucane z komunikatem o braku obsługi.

* Propozycja diagramu klas

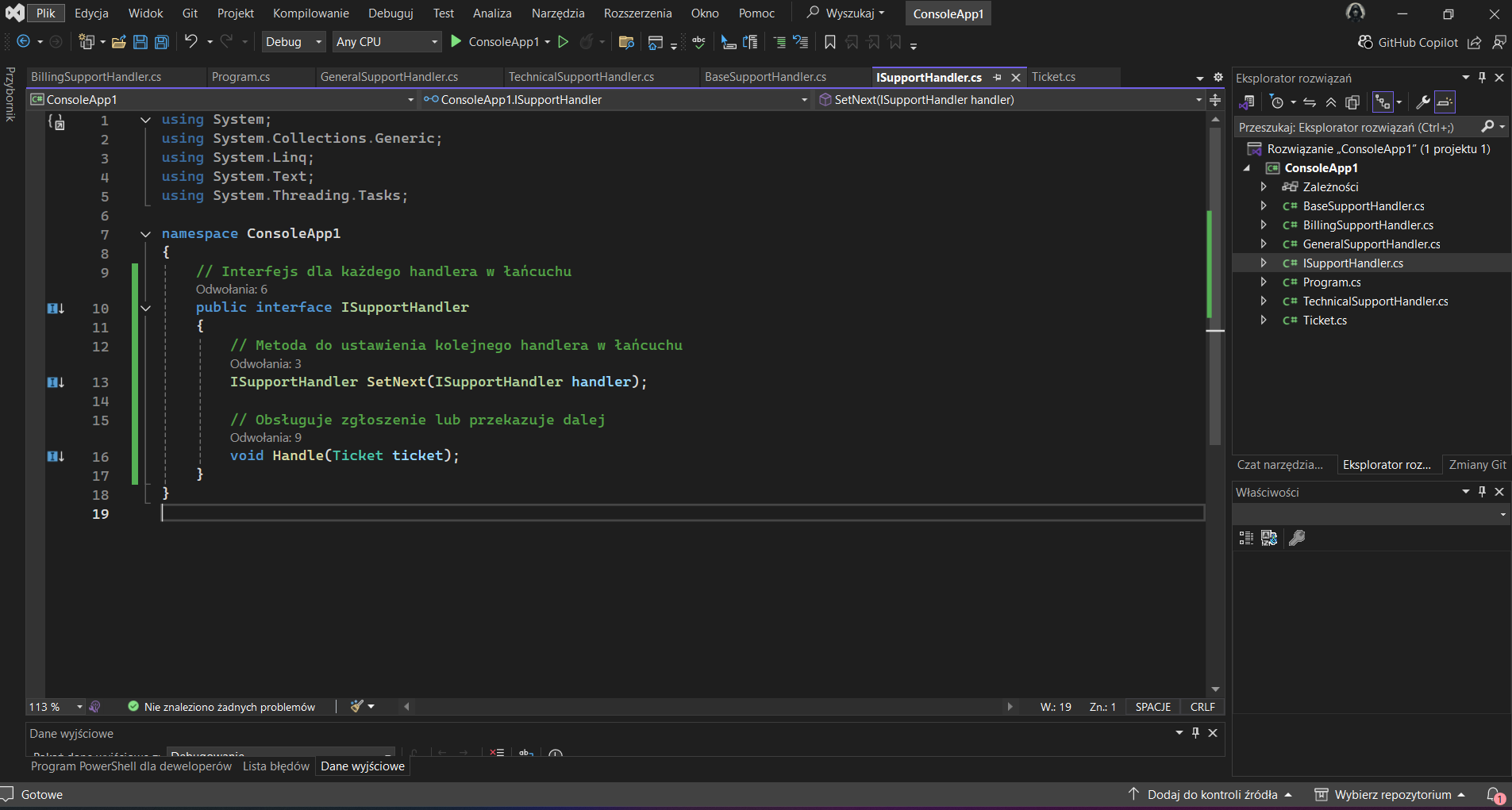


* Pseudokod

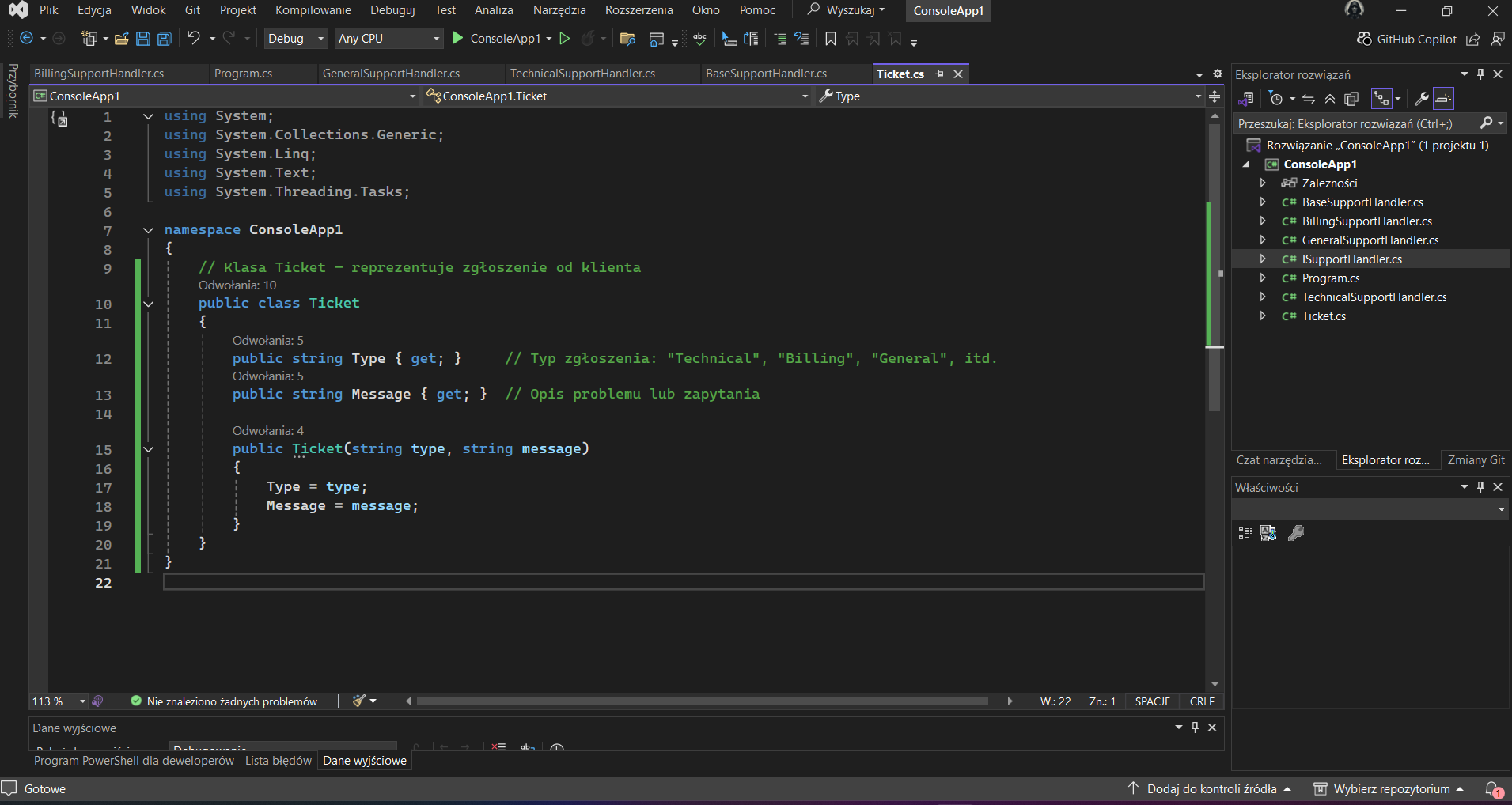


# Implementacja

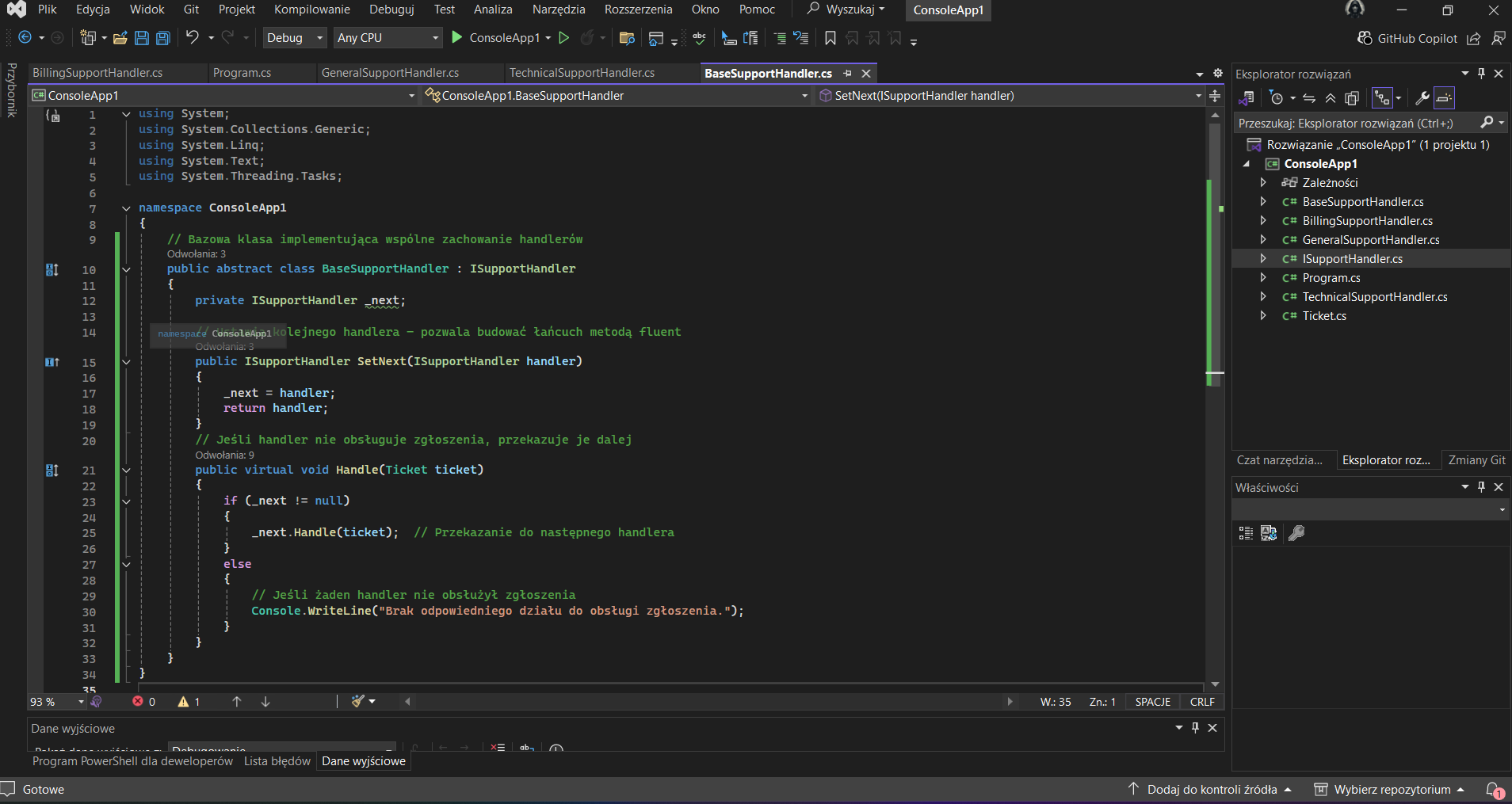
* Kod interfejsu **ISupportHandler.cs**



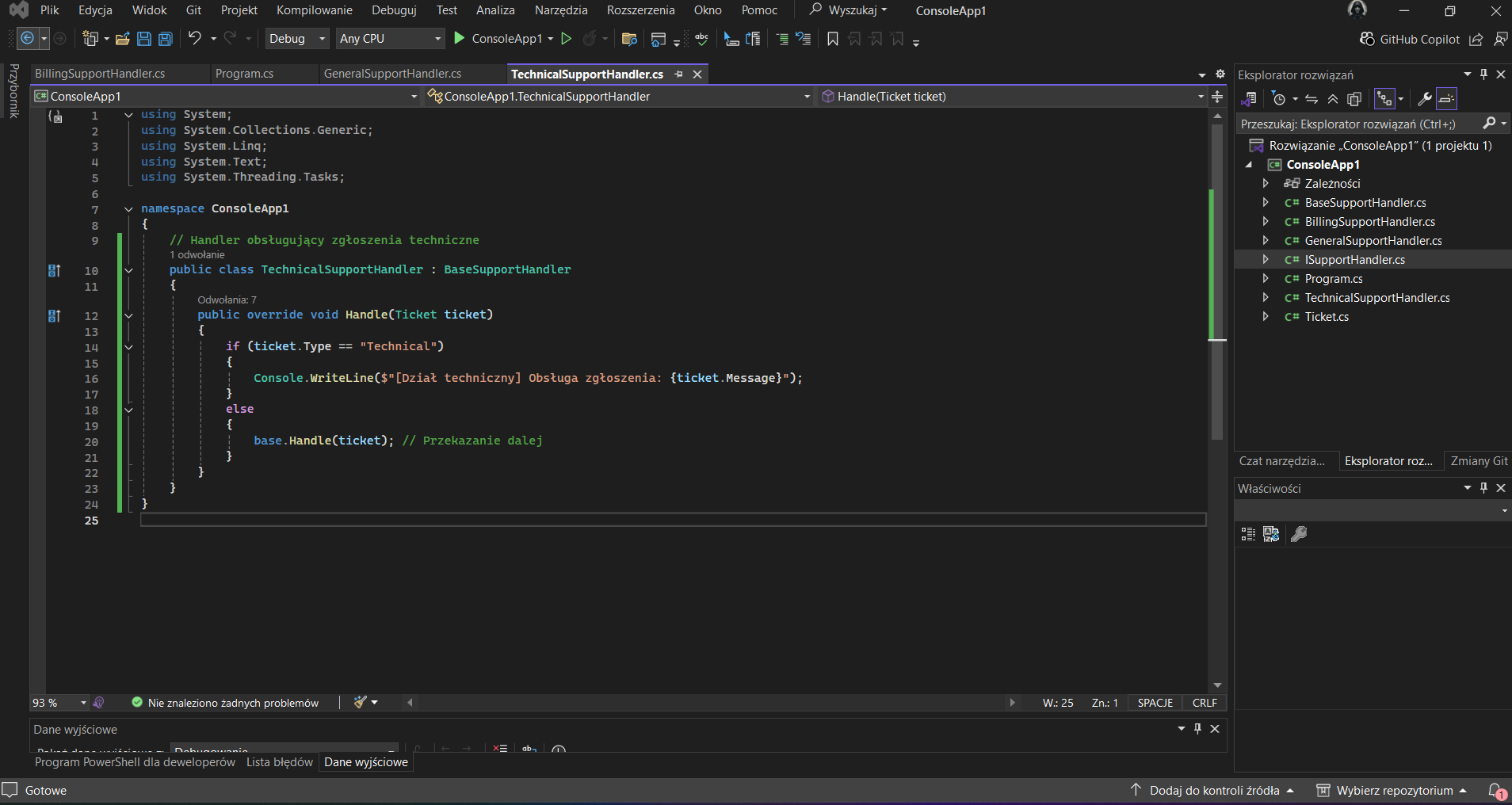
* Kod klasy **Ticket.cs**



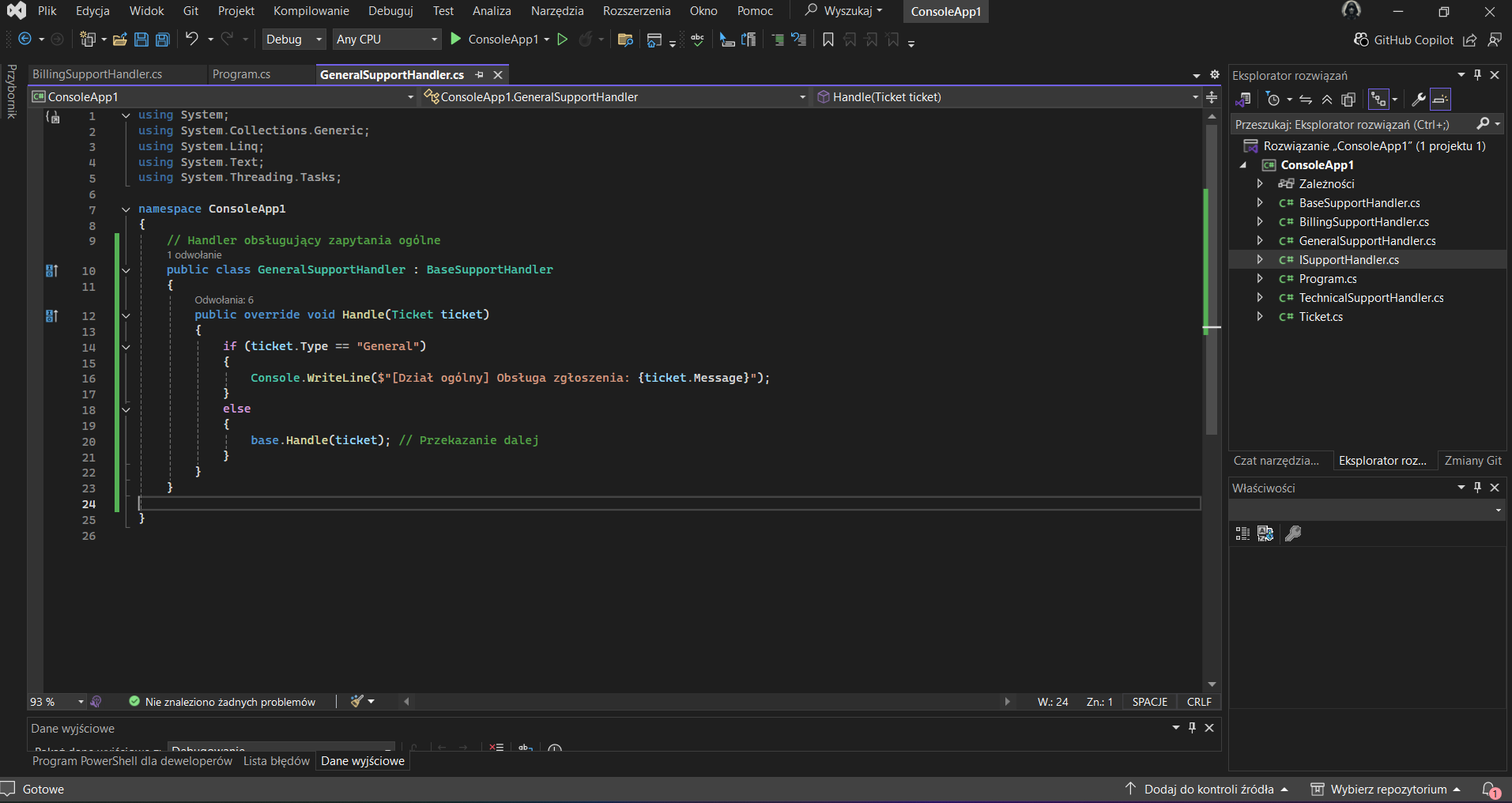
* Kod klasy **BaseSupportHandler.cs**



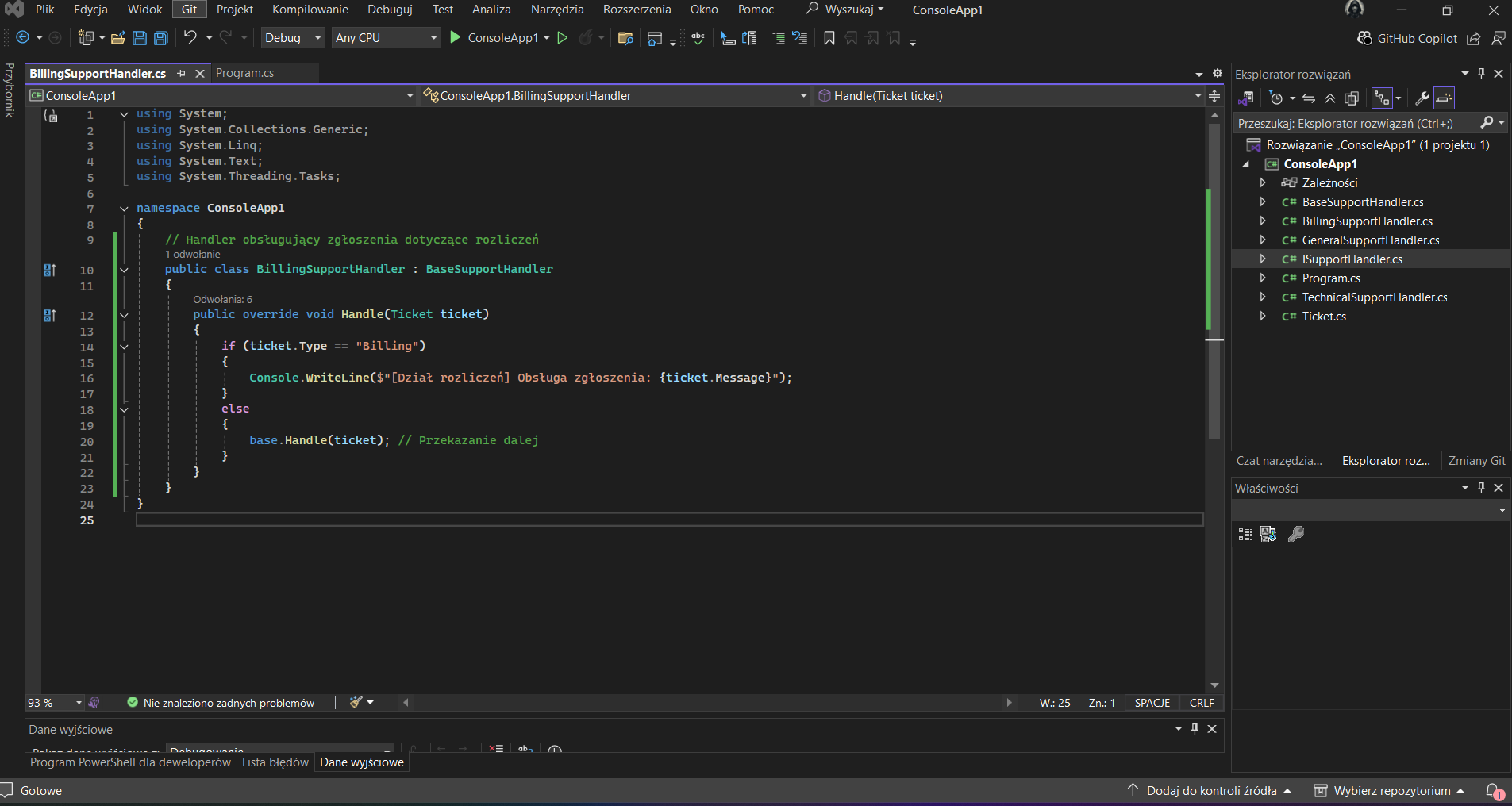
* Kod klasy **TechnicalSupportHandler.cs**



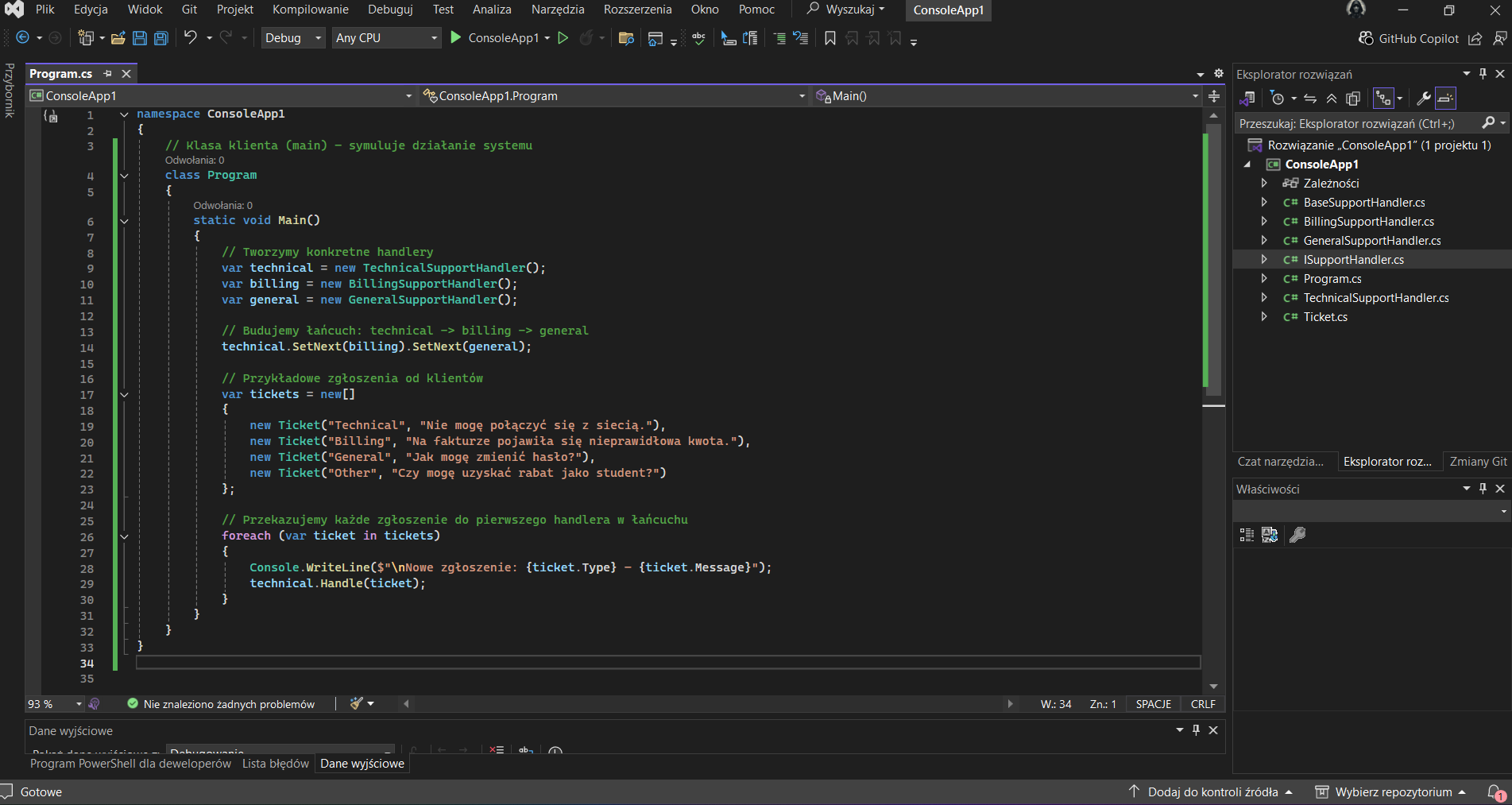
* Kod **klasy GeneralSupportHandler.cs**



* Kod **klasy BillingSupportHandler.cs**



* Kod **klasy Program.cs**

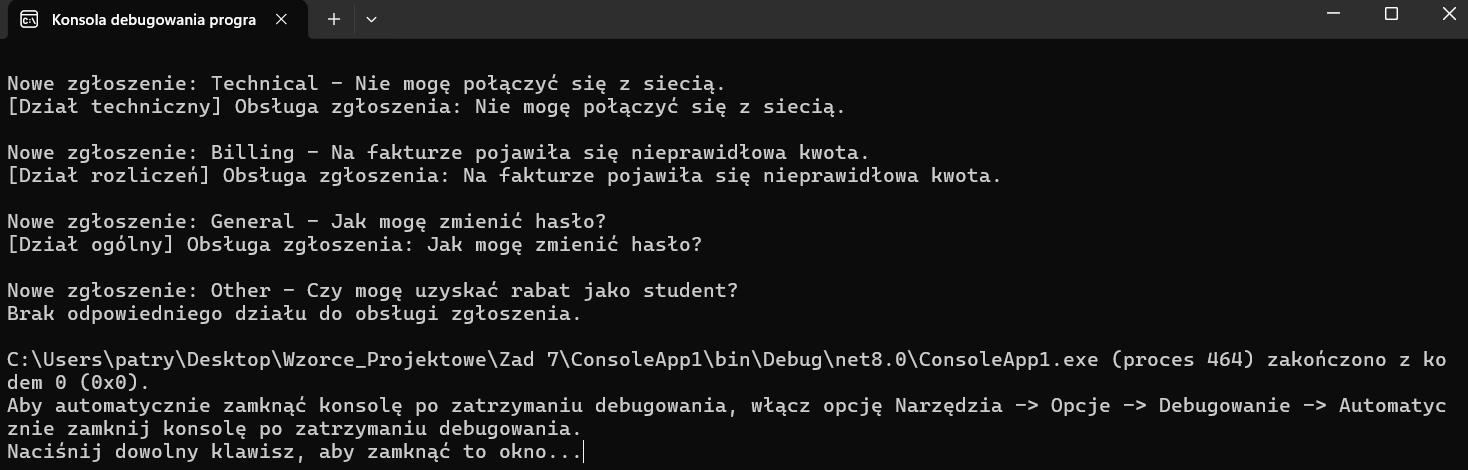


# Podsumowanie

Do rozwiązania zadania zastosowano wzorzec **Chain of Responsibility**, ponieważ idealnie odwzorowuje proces przetwarzania zgłoszeń w systemie wsparcia technicznego. Umożliwia on modularne i rozszerzalne podejście – dodanie nowego typu zgłoszenia (np. Security, Legal) nie wymaga modyfikacji pozostałego kodu.

**Implementacja powiodła się**, ponieważ:

* Zgłoszenia są poprawnie obsługiwane przez odpowiednie handlery.
* Niezgodne zgłoszenia są obsługiwane zgodnie z logiką końcową („brak obsługi”).
* Kod jest elastyczny i zgodny z zasadą **Open/Closed (SOLID)**.



Alternatywne wzorce (np. Strategy, Mediator) mogłyby być użyte do innej architektury przekazywania zgłoszeń, ale **Chain of Responsibility** jest tutaj najlepszym wyborem, ponieważ odwzorowuje naturalny przepływ decyzyjny i przetwarzania.

# Lista załączników

Repozytorium GITHUB z projektem:

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