Text

Description automatically generated with medium confidence

wzorce projektowe

Sprawozdanie

Zadanie observer stacja pogodowa

Patryk Figas

Informatyka, programowanie

Grupa 34\_Inf\_P\_NW\_6

Rok akademicki 2024/25

# Cel

Celem dokumentu jest przedstawienie rozwiązania zadania polegającego na zaprojektowaniu i zaimplementowaniu aplikacji stacji pogodowej z wykorzystaniem wzorca projektowego **Observer**. Stworzono system, który automatycznie powiadamia komponenty wyświetlające dane meteorologiczne o zmianach warunków pogodowych.

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

W zadaniu został zastosowany **wzorzec projektowy Observer**, który umożliwia niezależne powiązanie wielu obserwatorów z jednym źródłem danych (podmiotem).

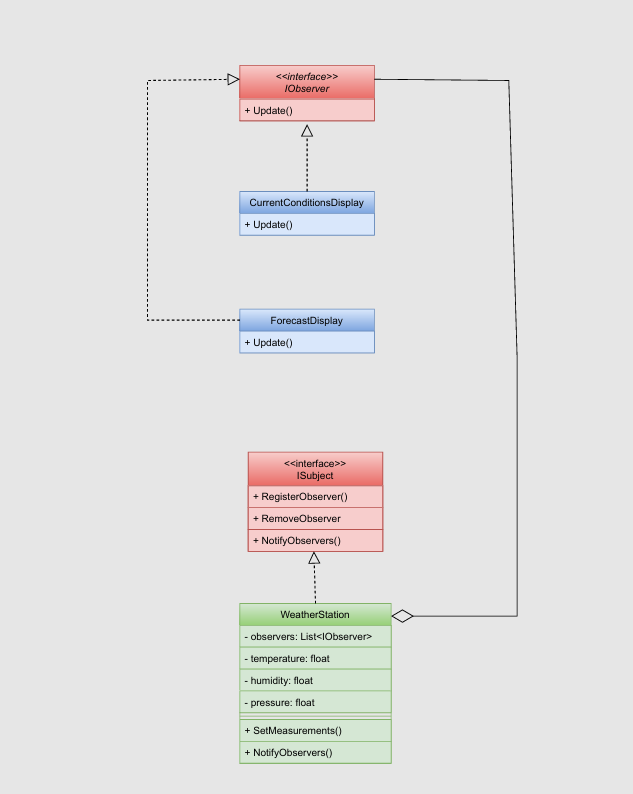
# Opis rozwiązania

Rozwiązanie opiera się na utworzeniu dwóch interfejsów: **ISubject** (podmiot) oraz **IObserver** (obserwator). Podmiotem jest **klasa WeatherStation**, która przechowuje dane pogodowe i informuje o ich zmianie wszystkich zarejestrowanych obserwatorów.

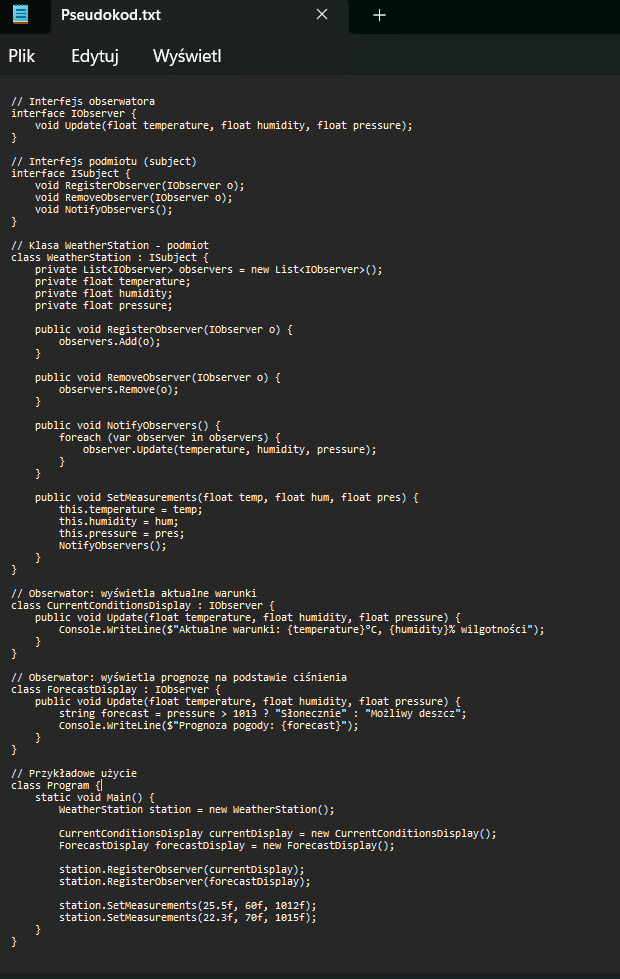
Obserwatorami są **klasy CurrentConditionsDisplay** i **ForecastDisplay**, które implementują **metodę Update()** i reagują na zmiany danych.

Po każdej zmianie danych pogodowych, WeatherStation wywołuje Update() dla wszystkich obserwatorów, które następnie wyświetlają aktualne dane lub prognozę.

* Diagram klas programu

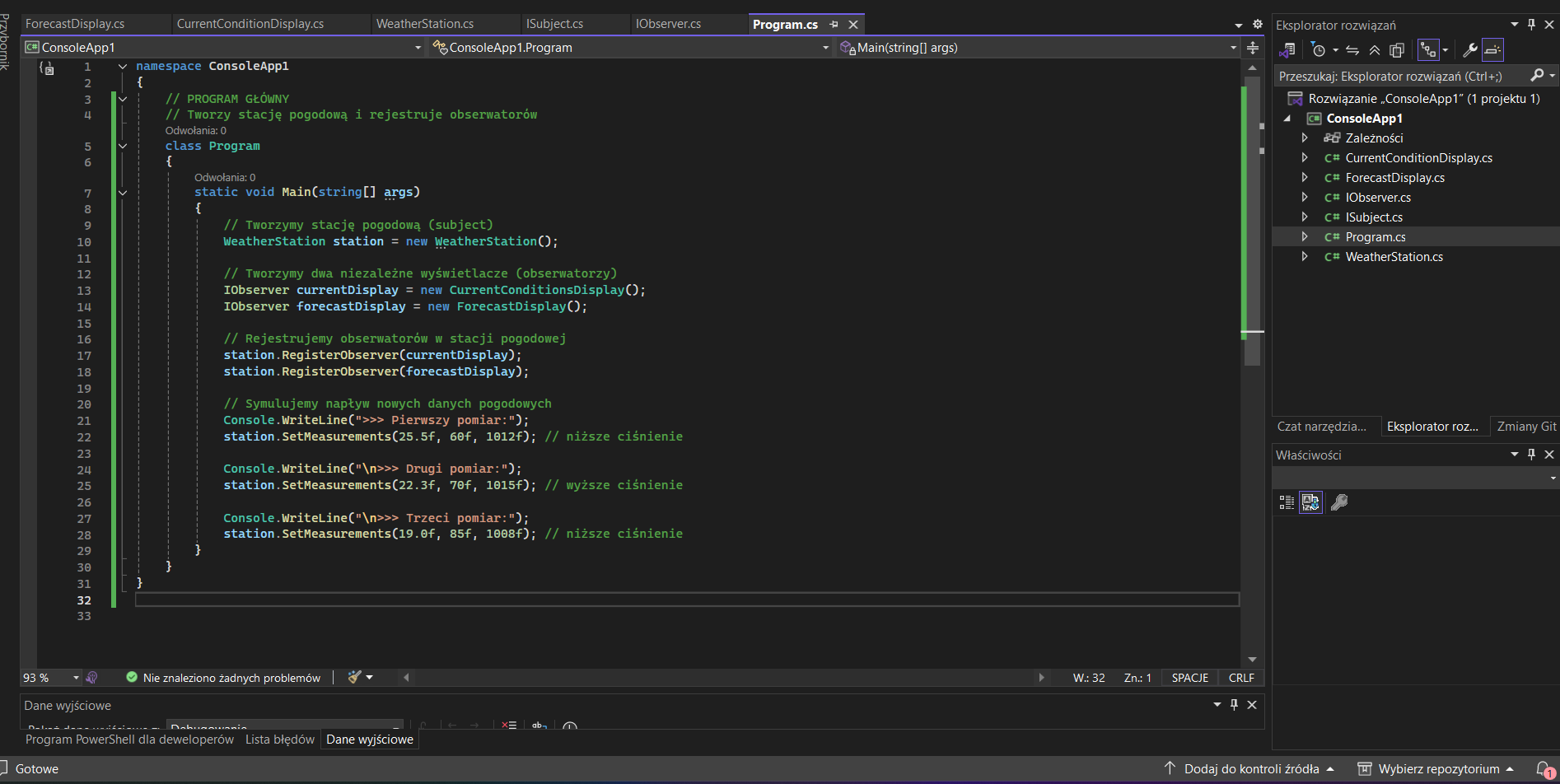


* Pseudokod klas i interfejsów programu

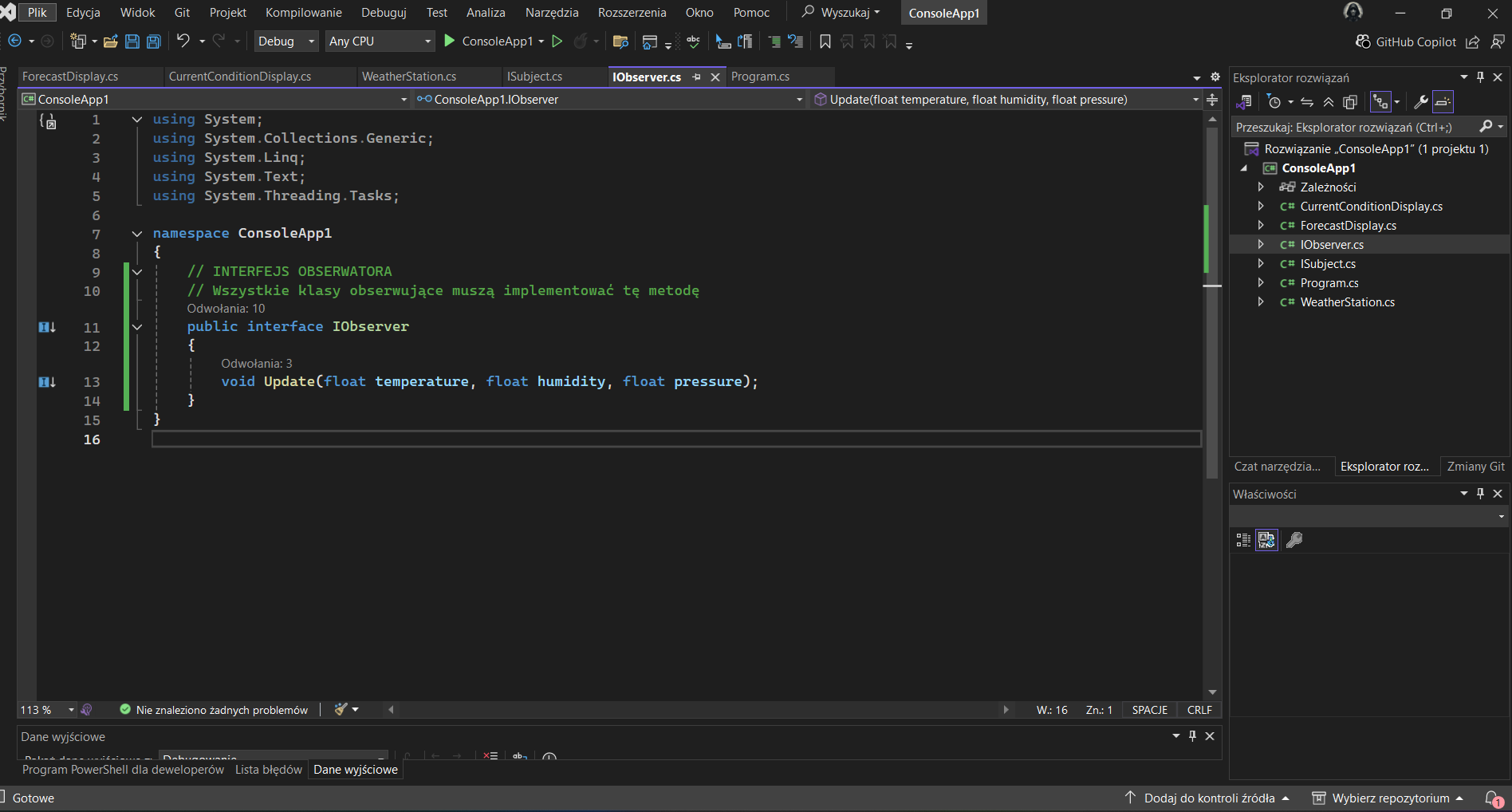


# Implementacja

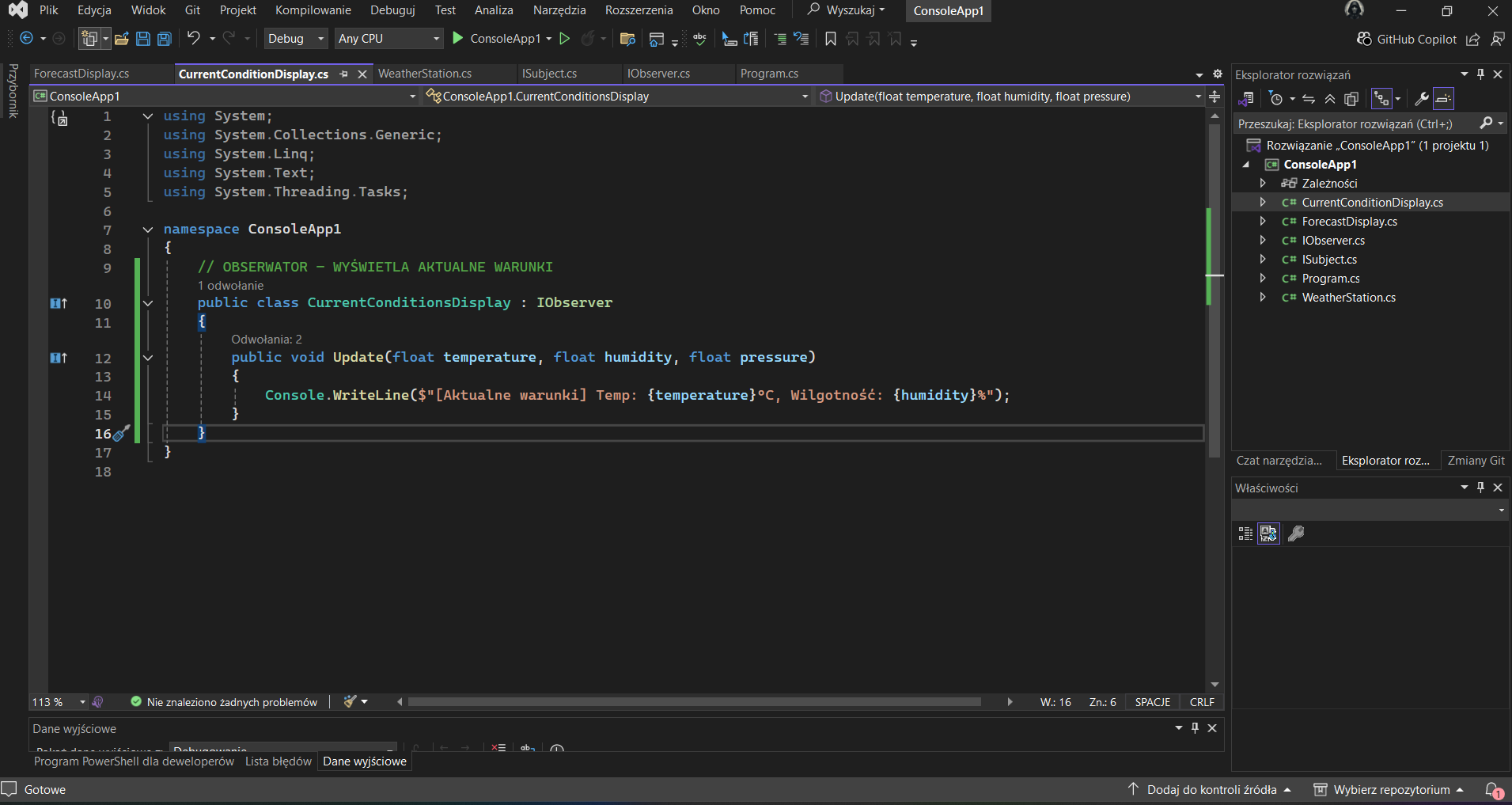
* Kod klasy **Program.cs**



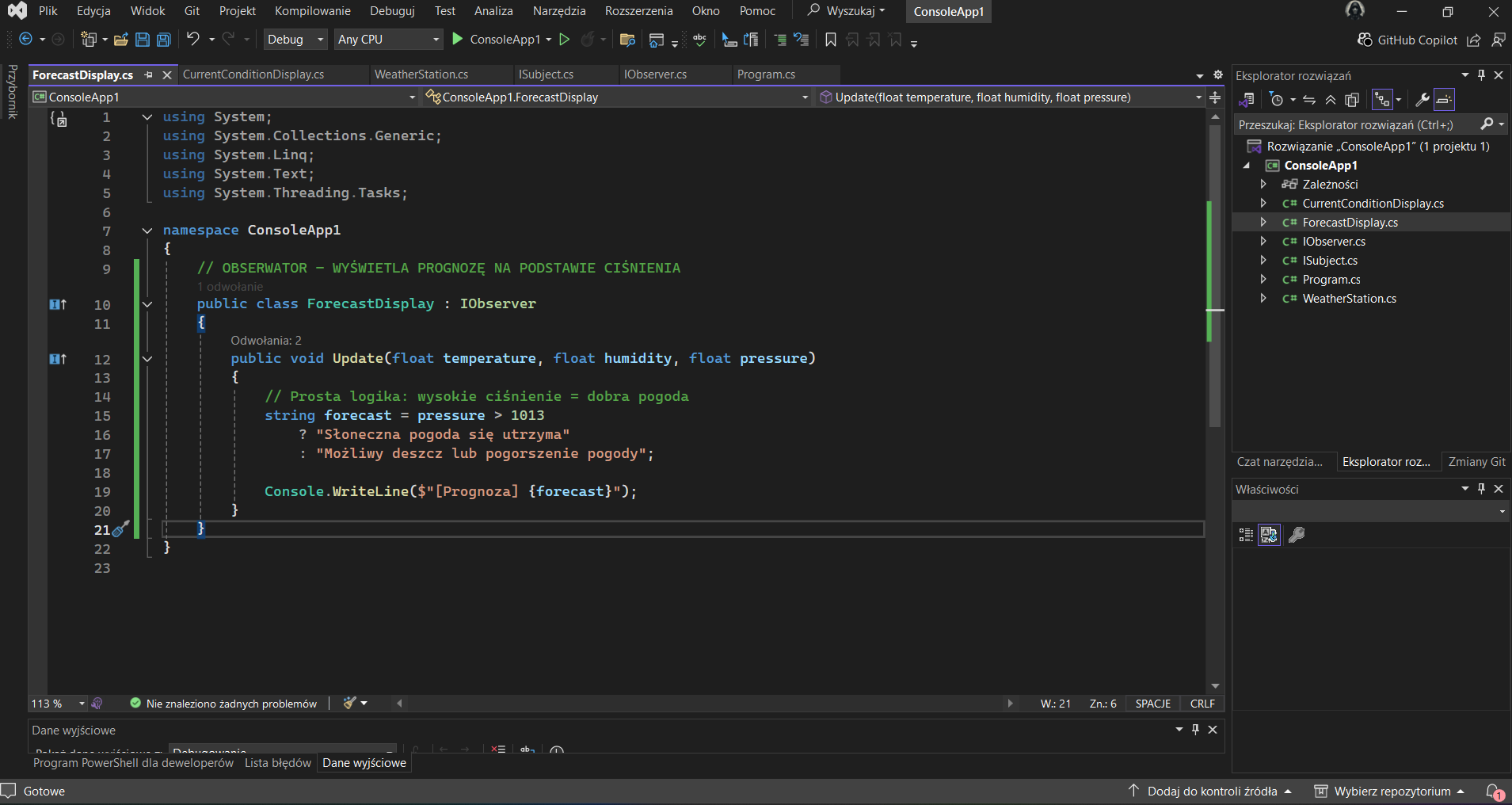
* Kod interfejsu **IObserver.cs**



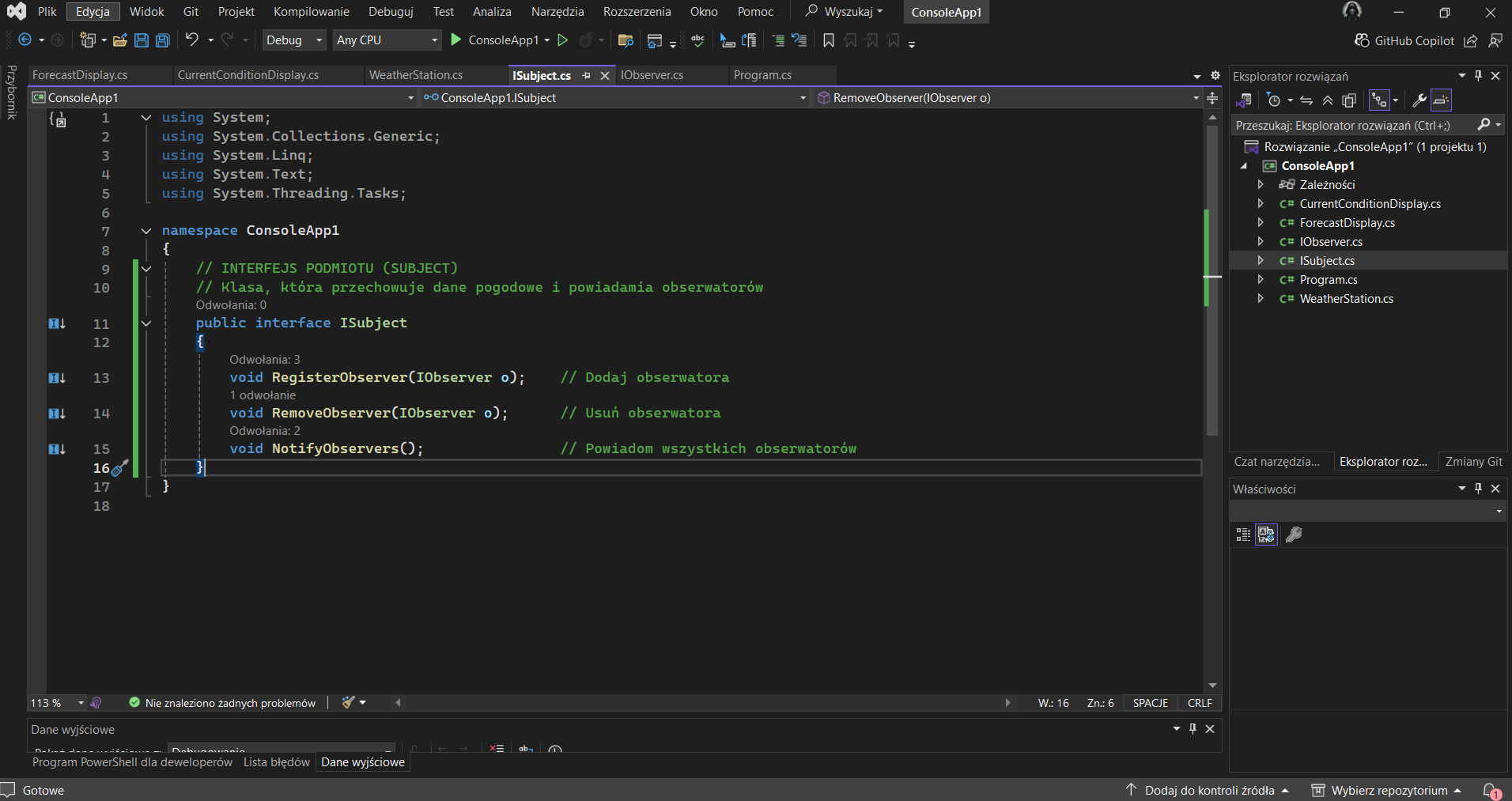
* Kod klasu **CurrentConditionDisplay.cs**



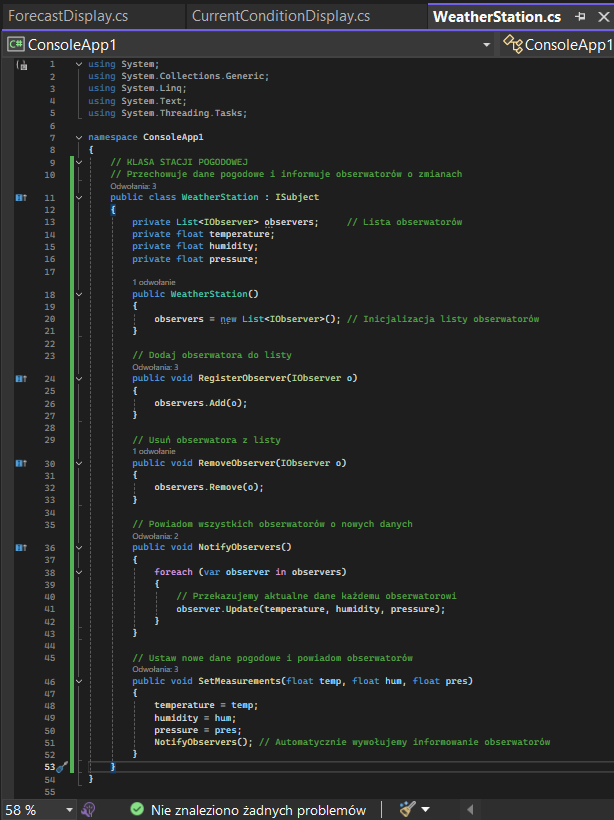
* Kod klasy **ForecastDisplay.cs**



* Kod interfejsu **ISubject.cs**



* Kod klasy **WeatherStation.cs**

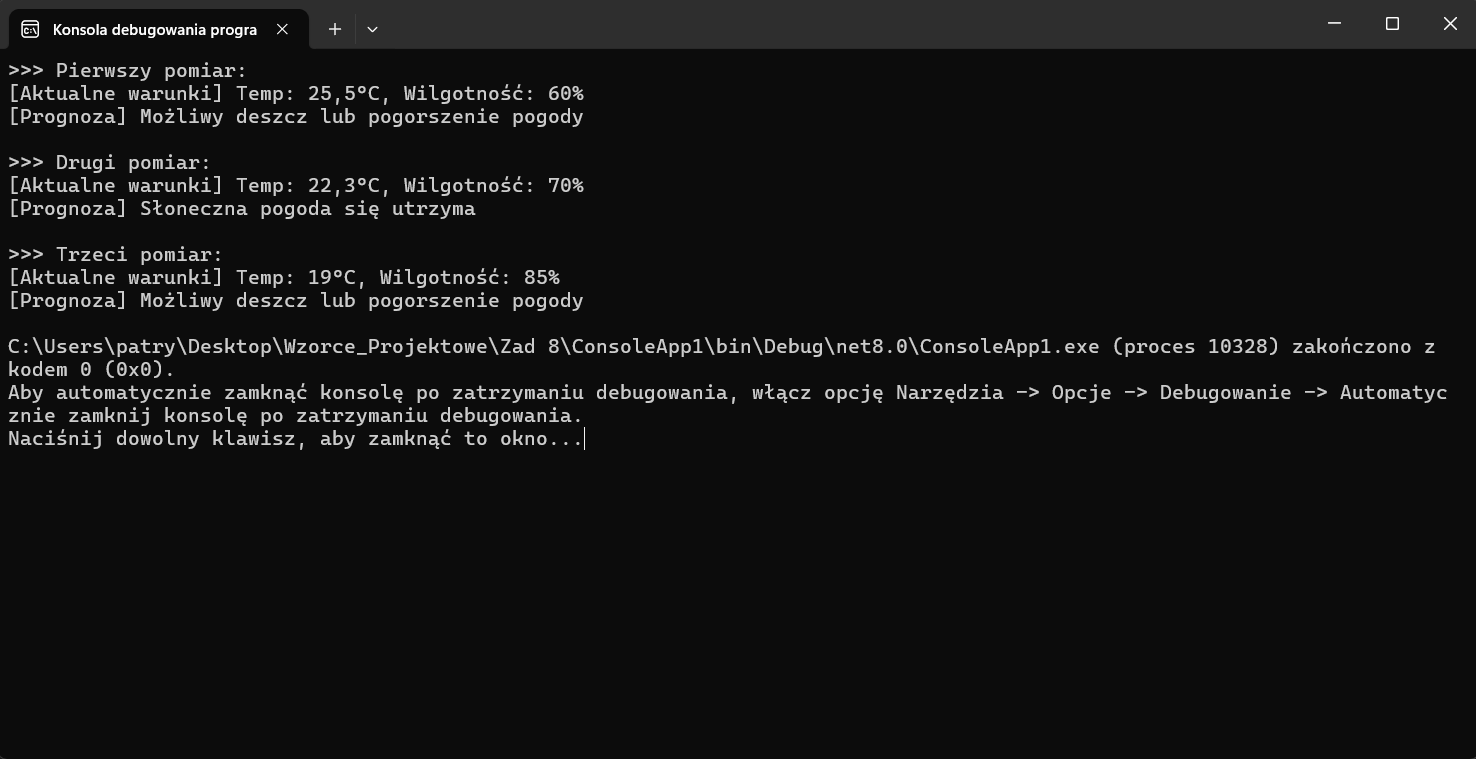


# Podsumowanie

Do realizacji zadania zastosowano wzorzec **Observer**, ponieważ:

* umożliwia niezależną reakcję wielu komponentów na zmianę stanu jednego obiektu,
* pozwala dynamicznie dodawać i usuwać obserwatorów bez modyfikacji klasy WeatherStation,
* ułatwia rozdzielenie odpowiedzialności (zasada SOLID – Single Responsibility).

Implementacja wzorca zakończyła się sukcesem – aplikacja działa zgodnie z założeniami i umożliwia łatwą rozbudowę (np. dodanie nowych typów wyświetlaczy).



Alternatywne wzorce (np. Mediator lub Publisher-Subscriber) mogłyby działać, ale **Observer jest najprostszy i najbardziej adekwatny** w przypadku bezpośrednich relacji między obiektami.

# Lista załączników

Repozytorium GITHUB z projektem:

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