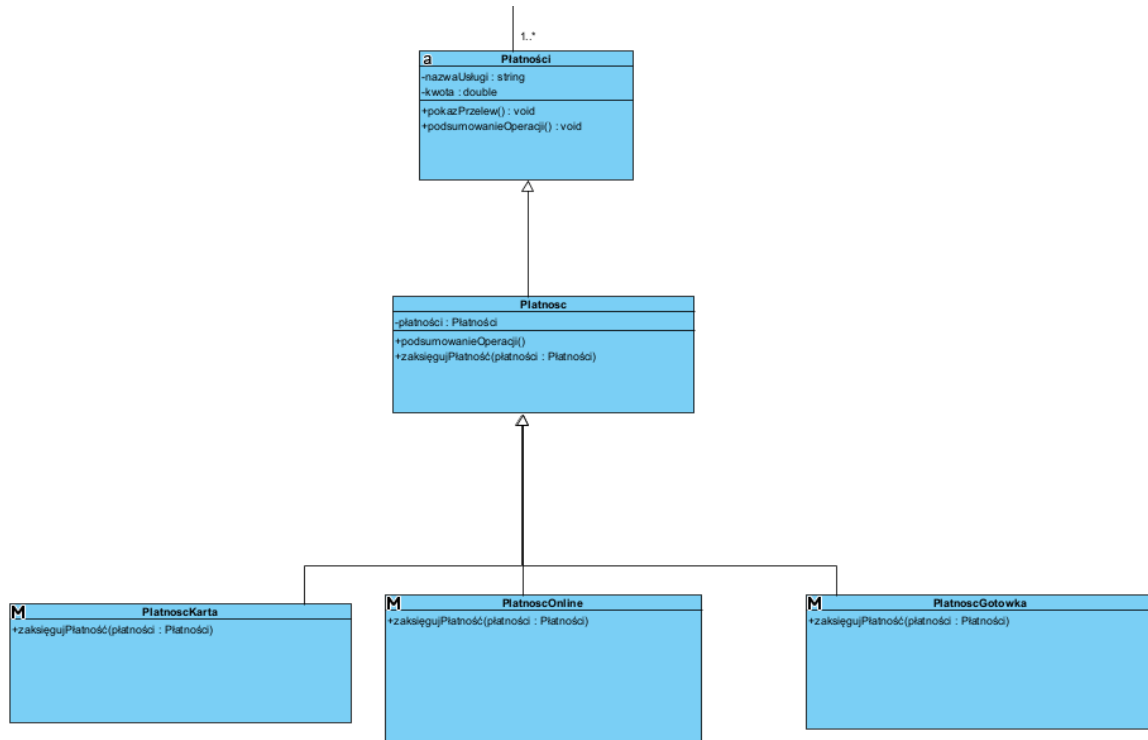


## Open/Closed Principle

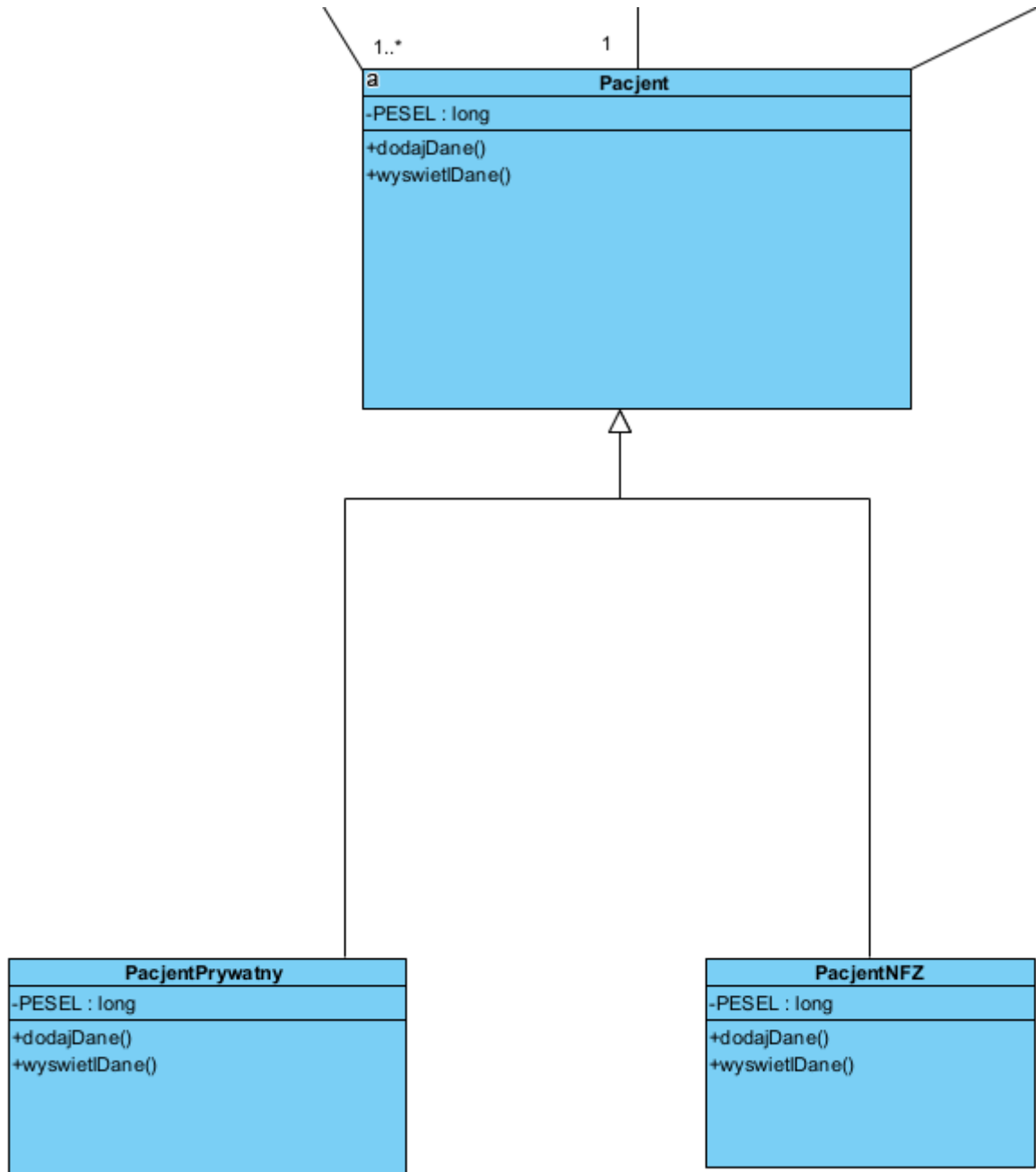
W naszym projekcie elementy systemu odpowiedzialne za płatności, w przypadku dodania kolejnych metod płatności, zabezpieczają, że istniejące płatności nie będą musiały być modyfikowane, a system będzie mógł być rozszerzany o kolejne.



## Liskov Substitution Protocole

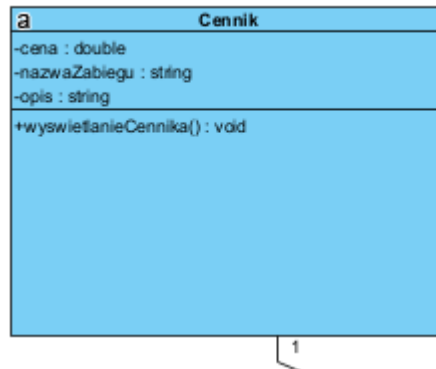
Klasa bazowa Pacjent, może korzystać z obiektów klas dziedziczących Pacjent prywatny, Pacjent NFZ.

Funkcja wyświetlDane() będzie działać również po przekazaniu instancji z klasy PacjentNFZ do PacjentPrywatny.



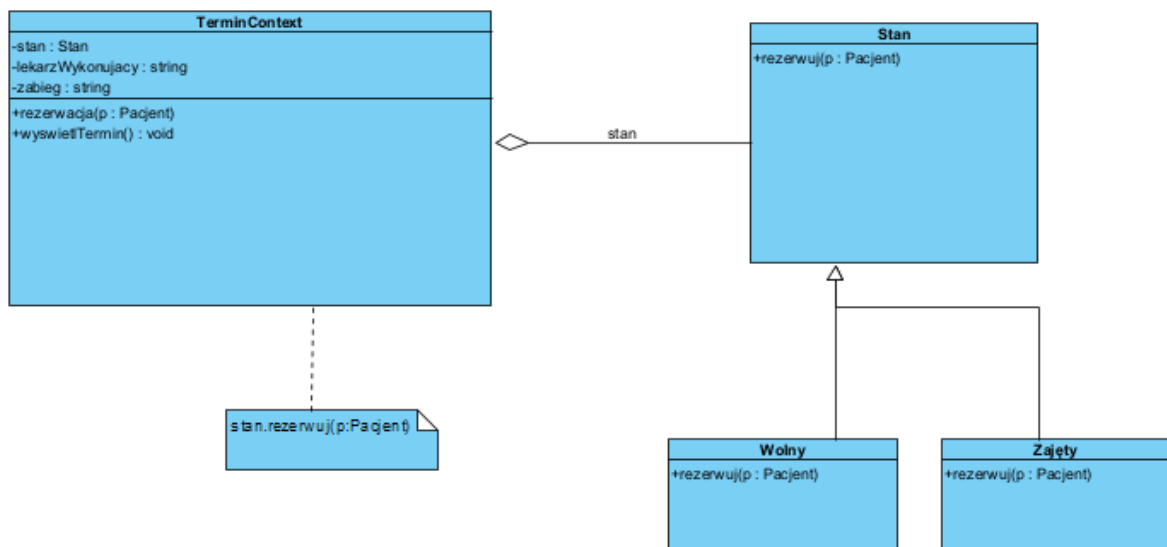
## Single Responsibility Principle

W przypadku naszego projektu, klasa Cennik jest odpowiedzialna jedynie za wyświetlanie cennika, co pozwala nam na elastyczne dalsze rozwijanie systemu.



## Controller

Klasa **TerminContext** działa jako kierownik obsługując operacje związane z terminem, takie jak rezerwacja. Kontroluje również zmiany stanu terminu.



## Information Expert

Klasa Grafik jest ekspertem w zarządzaniu stanem grafiku w systemie, posiada on informacje o dniu i godzinie do ustalenia pracy gabinetu, którymi zarządza. Grafik otrzymał odpowiedzialność do sprawdzenia grafiku oraz zmieniania go, ponieważ ma wszystkie niezbędne informacje.

