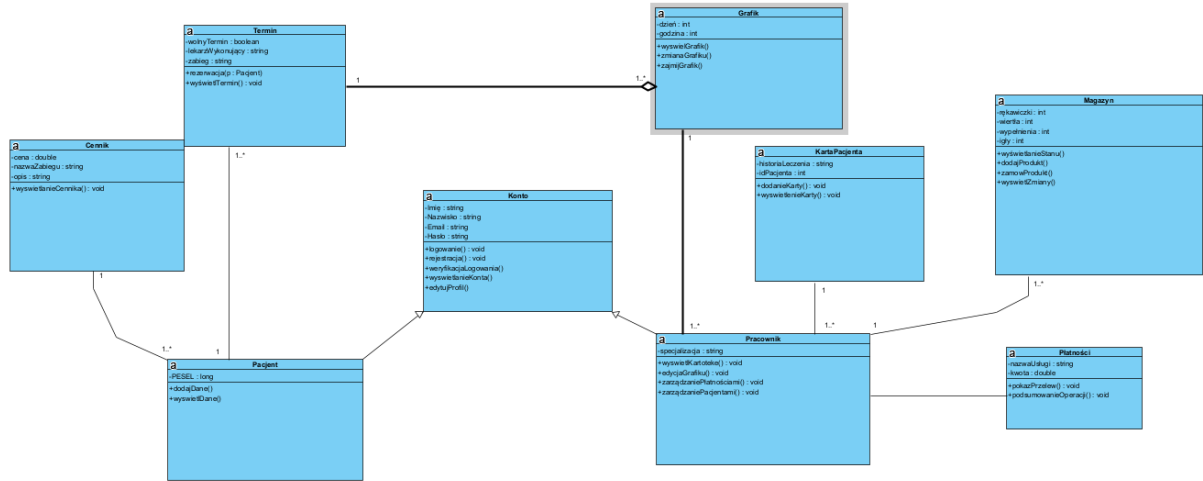


Sprawozdanie wzorce czynnościowe

Wiktoria Sędzik, Łukasz Oprych

Diagram klas przed wprowadzeniem wzorców:



Zastosowane wzorce:

Wzorec stan

Motywacja: Chcieliśmy w klasie termin pozbyć się warunku if-else, który był uzależniony od pola wolnyTermin typu boolean, aby w prosty sposób móc zmienić stan obiektu w trakcie wykonywania programu. Użyliśmy do tego wzorca stan, w którym dokonaliśmy przeniesienia rozgałęzienia warunkowego stanu do klas Wolny i Zajęty. Stan pozwolił nam uniknąć rozbudowanych instrukcji warunkowych, co sprawia, że kod jest elastyczny, łatwiejszy do modernizacji i utrzymania.

Diagram po:

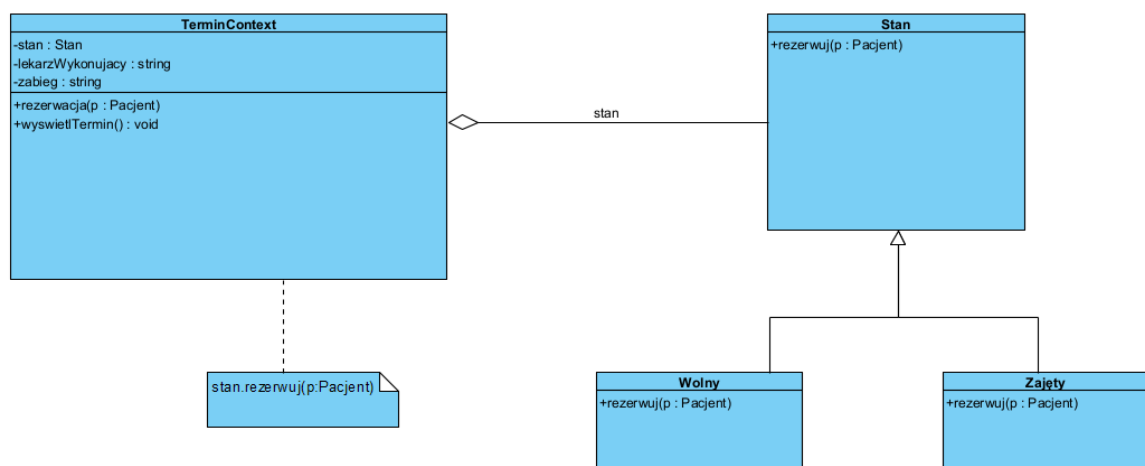
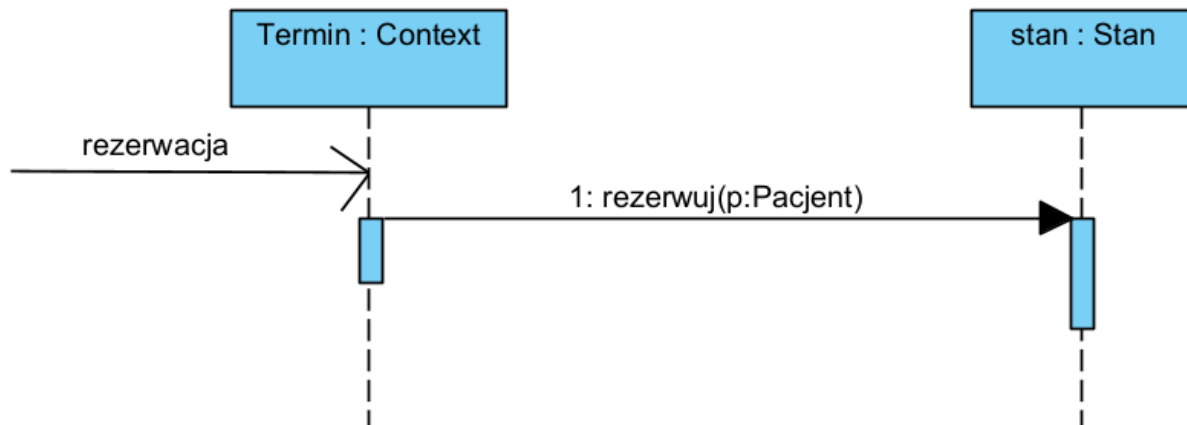


Diagram sekwencyjny:



Po zmianie, na Terminie wywołujemy metodę rezerwacja, to do tego Terminu przypięty jest Stan i metoda rezerwacja jest przekierowywana do tego Stanu i powoduje to, że na Stanie wywołujemy metodę rezerwuj(p:Pacjent). Pod Stan mogą być podpięte obiekty Wolny, Zajęty.

Wzorzec strategia

Motywacja: Zastosowaliśmy wzorzec strategia w klasie Płatności w celu umożliwienia elastycznego zarządzania różnymi opcjami płatności bez ingerencji w domyślną klasę Płatności. Każdej z płatności odpowiada inny wariant algorytmu, między którymi będziemy przełączać w trakcie działania programu w zależności od wyboru metody płatności przez pacjenta.

Diagram po:

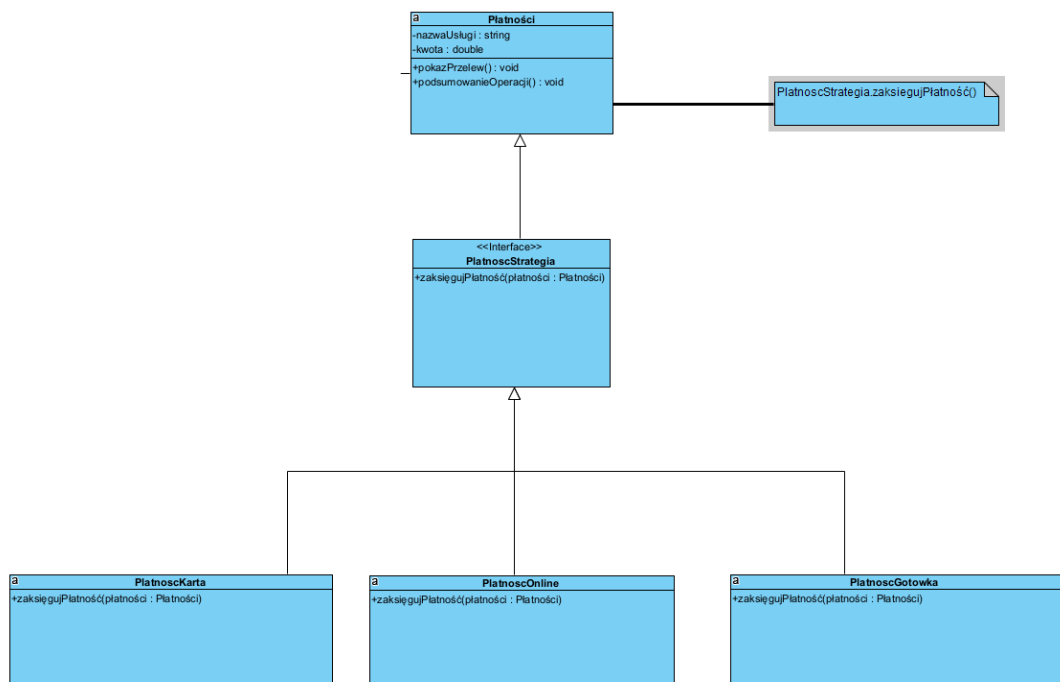
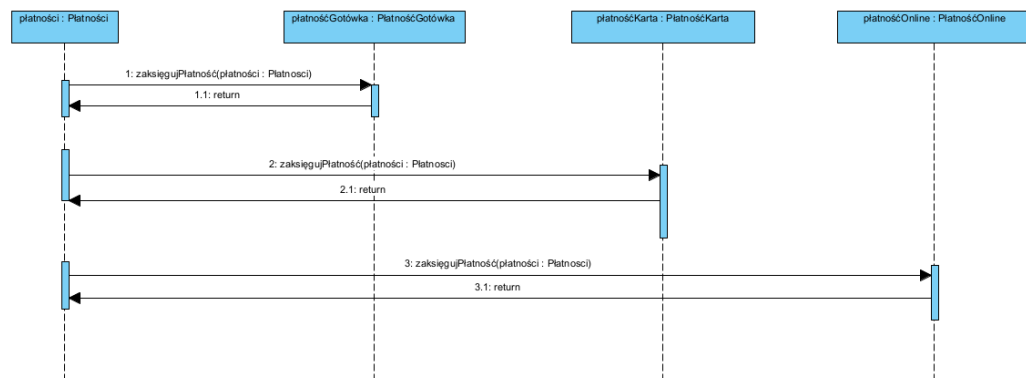


Diagram sekwencji:



Płatność wykonuje algorytm `zaksięgujPłatność` w odniesieniu do interfejsu `PłatnośćStrategia` przyjmując obiekt klasy `Płatności`. Odpowiedzią `PłatnośćGotówka` jest wynik zwracany po wykonaniu operacji, analogiczny schemat postępowania jest dla `PłatnośćKarta` i `PłatnośćOnline`.