



Technologie obiektowe

UML - diagramy obiektów oraz sekwencji
Wstęp do wzorców projektowych



Czym jest UML

Unified Modelling Language

Przemysłowy standard modelowania

Zbiór notacji graficznych dedykowany modeli wizualnych w projektowaniu zorientowanym obiektowo

Obecnie używana wersja 2.x (2.5.1)



UML 2.5

14 typów diagramów

3 kategorie:

- struktura

- zachowanie

- interakcje



Struktura

Ukazuje statyczne elementy programu i relacje między nimi

Diagramy:

klas, obiektów, paczek, struktur złożonych, komponentów, wdrożeń, profili



Zachowanie

Ukazuje zachowanie elementów programu

Diagramy:

przypadków użycia, aktywności, maszyny stanowej, interakcji



Interakcji

Ukazuje jak elementy systemu oddziałują na siebie

Diagramy:

sekwencji, komunikacji, czasu, interakcji

Diagramy klas UML



Diagram klas

Ukazuje zbiór statycznych elementów modelu: klas, typów danych ich zawartości oraz relacji

Nie zawiera szczegółów implementacji

Nie dotyczy interakcji między elementami



UML: klasa i interfejs

Nazwa klasy - pierwsza część prostokąta

Kursywa - klasa abstrakcyjna

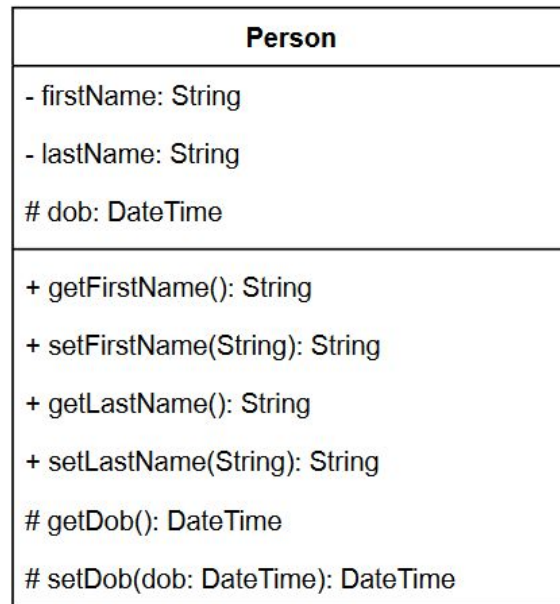
<<interface>> - odnosi się do interfejsu

atrybuty - wszystkie pola klasy

metody - gettery i settery MOGĄ być ominięte

metody interfejsu muszą być zawarte

metody dziedziczone nie powinny być ujęte





UML: klasa i interfejs - atrybuty

Atrybuty: zmienne

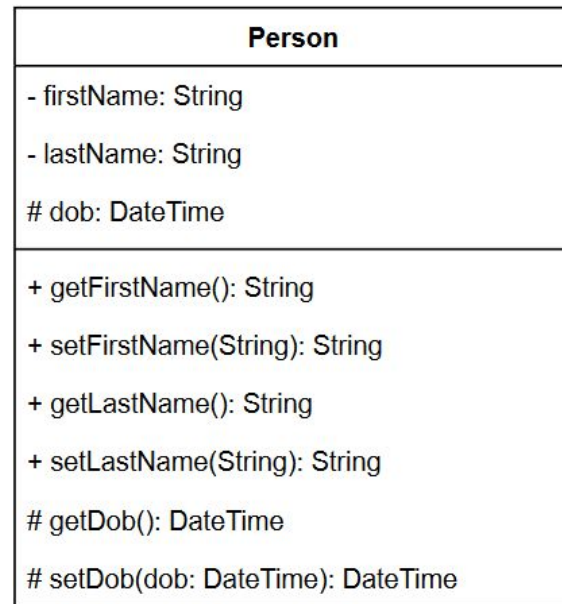
[widzialność] nazwa : [typ] [liczba] = [wartość domyślna]

widzialność:

- + publiczna
- prywatna
- # chroniona
- ~ paczka
- / pochodna

podkreślenie - pole statyczne

/ pochodna - wartość nie jest przechowywana, jest wyliczana na podstawie stanu innych pól





UML: klasa i interfejs - metody

[widzialność] nazwa (parametry) : [typ zwracany]

parametry - nazwa: typ

widzialność:

+ publiczna

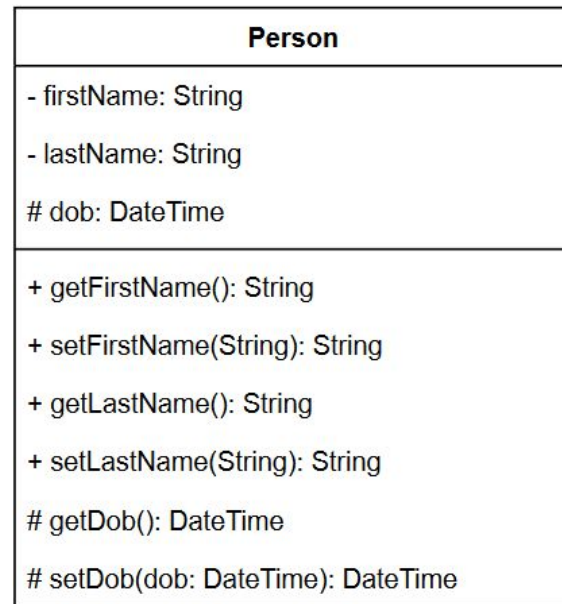
- prywatna

chroniona

~ paczka

podkreślenie - metoda statyczna

Typ jest omijany gdy void lub metoda jest konstruktorem

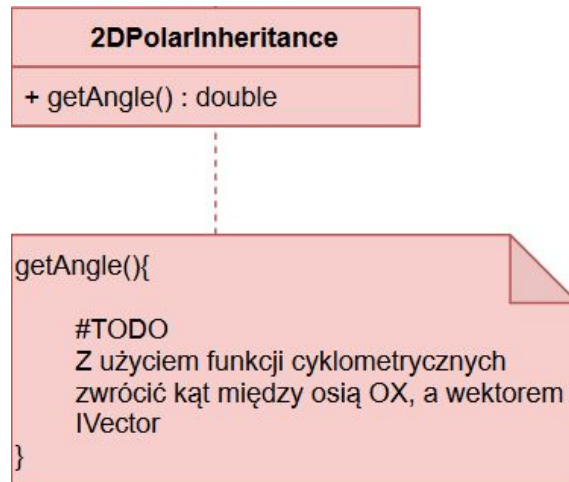




UML: komentarze

Reprezentuje informację w formie zgiętej w jednym rogu ramki

Dołączony do klasy przez przerywaną linię



UML: dziedziczenie

Przekazywanie cech między klasami

Oдноśnie interfejsów - implementacja

Hierarchia rysowana od góry do dołu

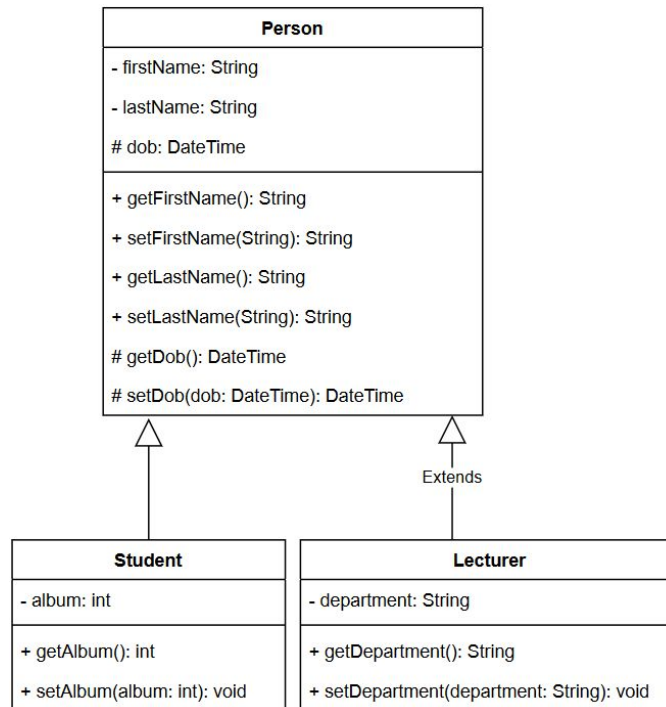
Grot skierowany do rodzica

Styl wskazuje na:

klasę: linia ciągła, grot wypełniony

klasę abstrakcyjną: linia ciągła, grot pusty

interfejs: linia przerywana, grot pusty (dopuszczalna notacja obła)



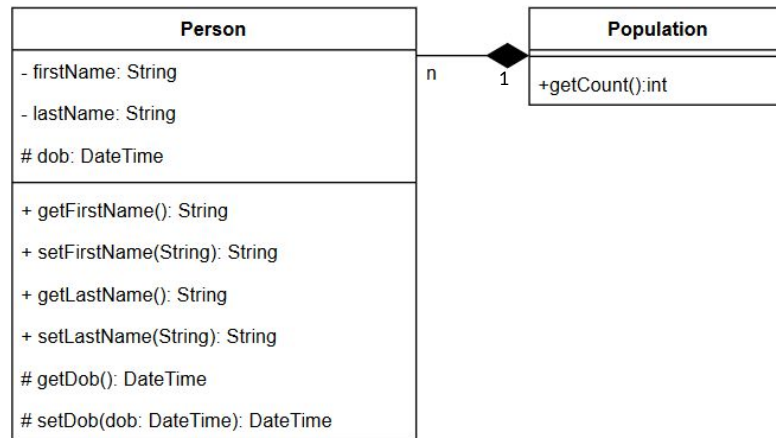
UML: kompozycja

Relacja: w całości składa się z

Silniejsza wersja agregacji

Części mają taki sam czas życia jak ich zbiór

Czarny wypełniony romb



UML: kompozycja

Dodatkowe adnotacje:

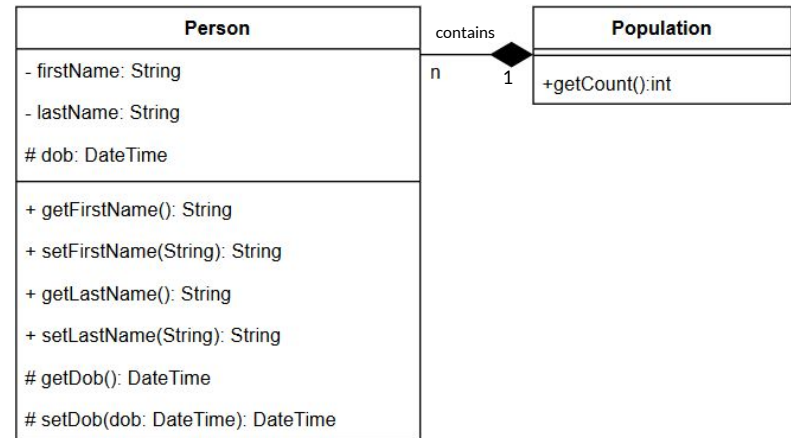
krotność:

* - 0, 1 lub więcej

1 dokładnie 1

2..4 - między 2 a 4

nazwa

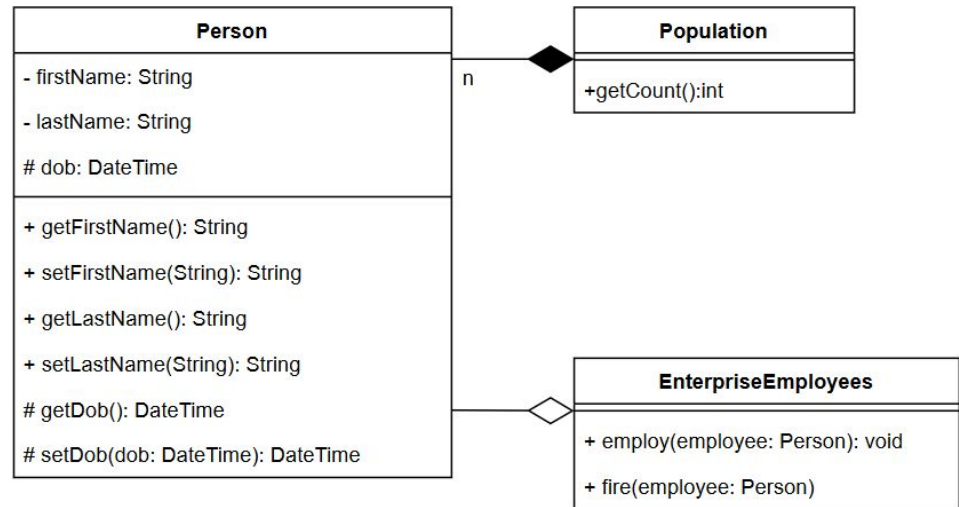


UML: agregacja

Relacja: jest częścią

Zbiór nie zarządza elementami

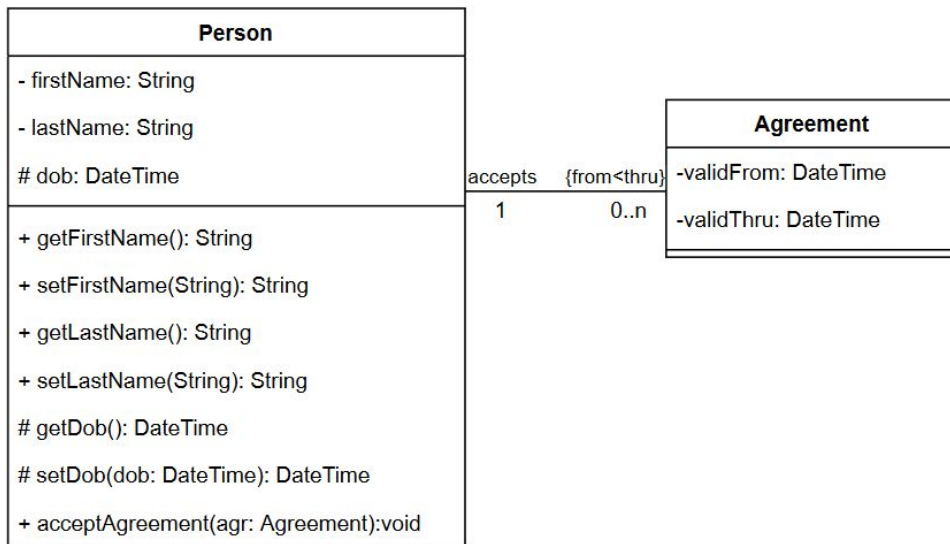
Romb bez wypełnienia



UML: asocjacja

Czasowe powiązanie między obiektami

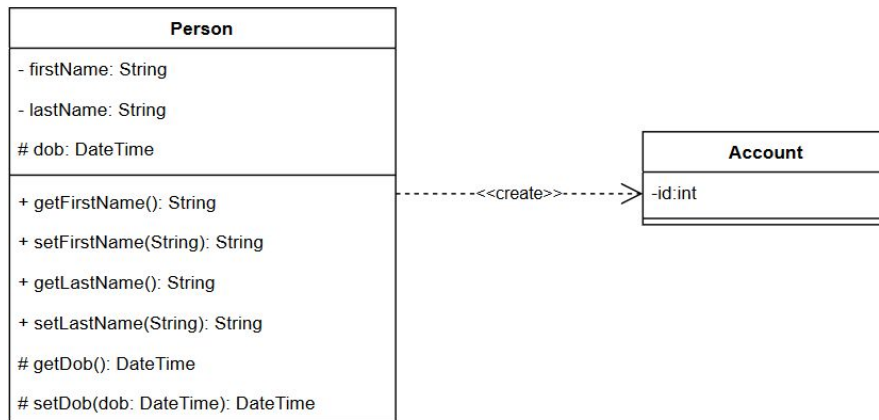
Adnotacja pozwalająca zrozumieć ich znaczenie



UML: zależność

Ukazuje to jak zależą od siebie pewne elementy systemu

<<derive>>
<<friend>>
<<use>>
<<include>>
...





Zalety i wady diagramów klas UML

Wady:

Nie nadają się do wykrywania zależności wynikających ze specyfiki algorytmów/danych

Nie nadają się do wyznaczania przepływów między obiektami

Nie nadają się do analizy kontroli przepływu (EDP, Web, seq)



Zalety i wady diagramów klas UML

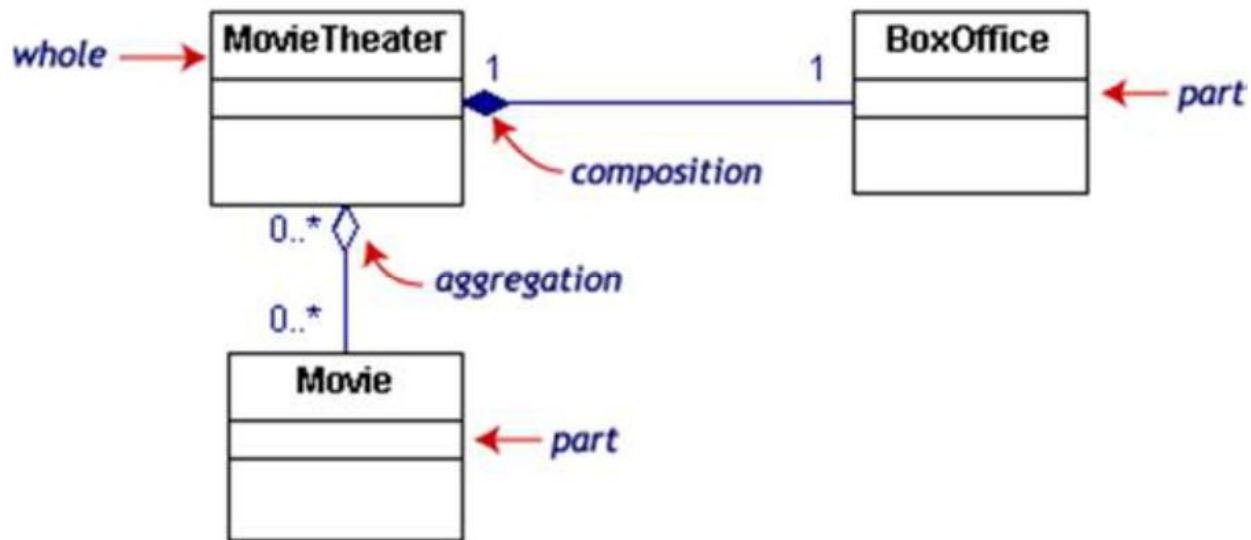
Zalety:

Wykrycie zależności pomiędzy danymi

Szybki obraz ważnych elementów systemu

Wykrycie zależności między klasami

Przykłady



Przykłady

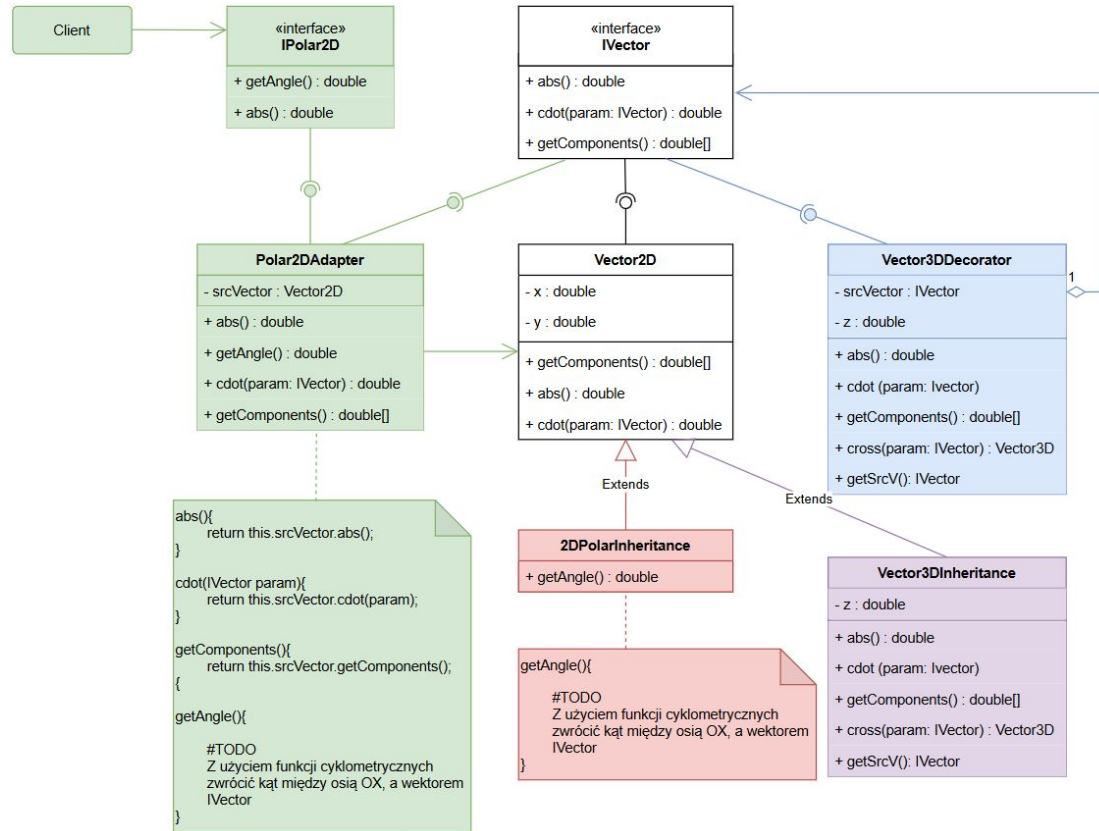


Diagram obiektów

Wizualizuje stany przykładowych instancji obiektów

Egzemplarze z diagramów klas

Zbiór obiektów i relacji między nimi

Ukazuje przypadki rzeczywiste lub prototypowe

```
nazwa_obiektu : nazwa_klasy  
nazwa_atrybutu1 = wartość_atrybutu1  
nazwa_atrybutu2 = wartość_atrybutu2  
...  
nazwa_atrybutuN = wartość_atrybutuN
```

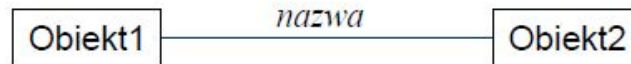


Diagram sekwencji

Interakcje między elementami systemu

Sekwencje komunikatów

Przeływ sterowania

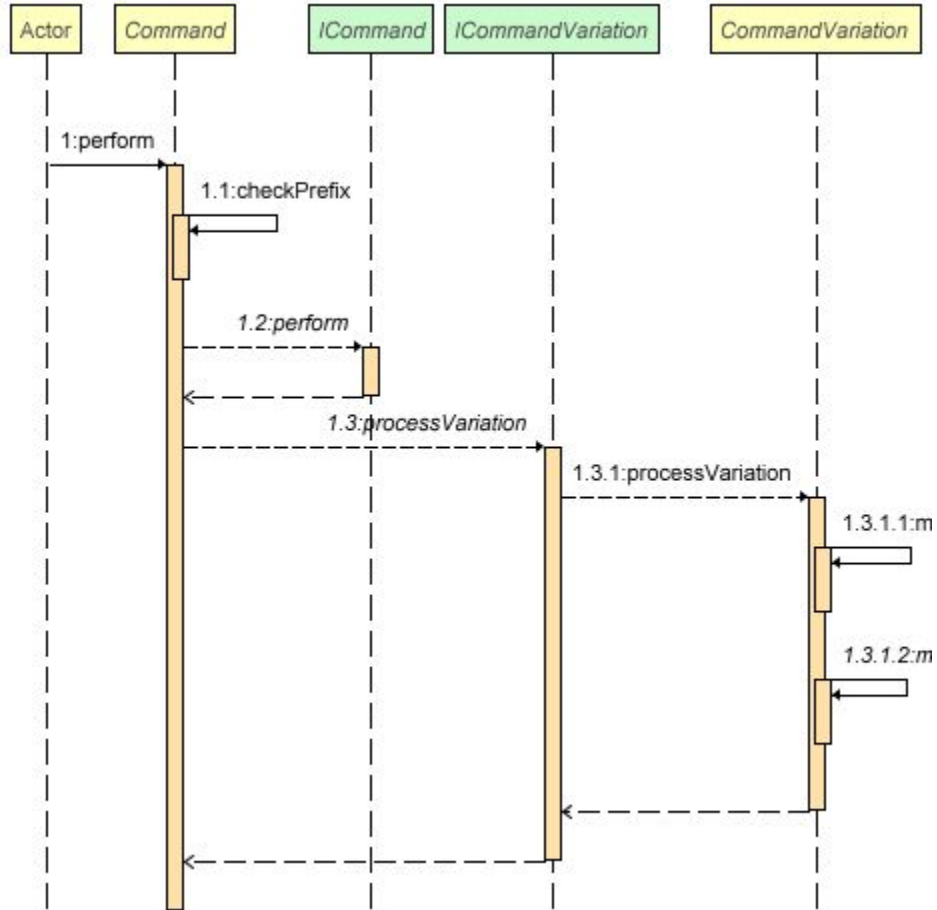
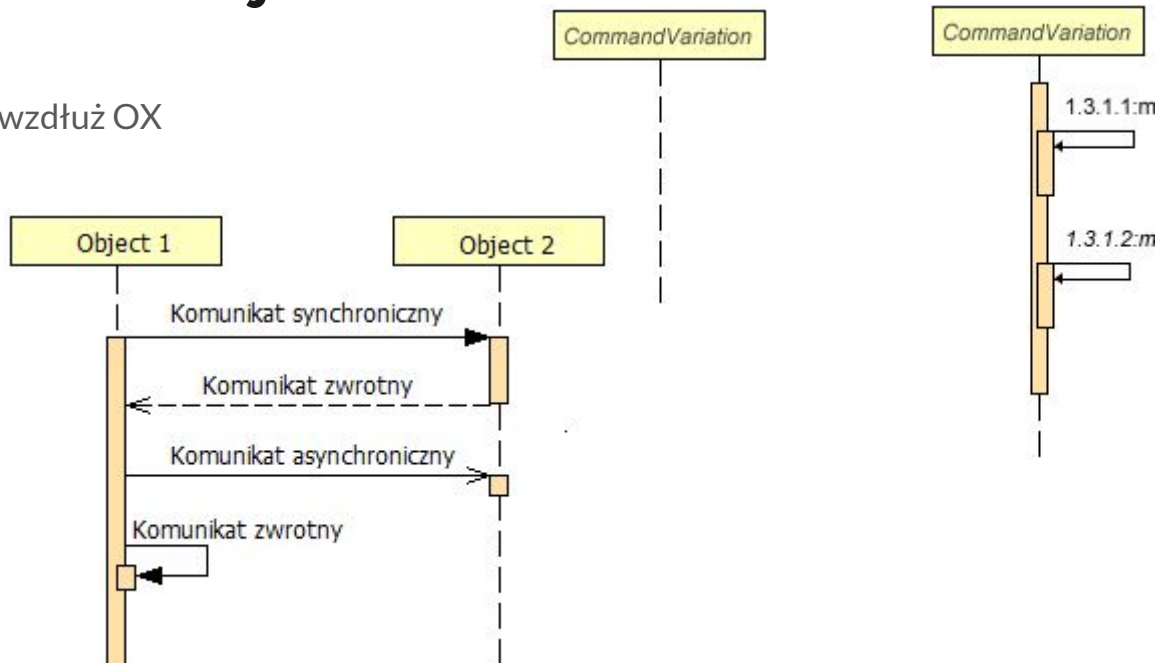


Diagram sekwencji

Obiekty - ułożone wzdłuż OX

Czas - oś OY

Linia życia



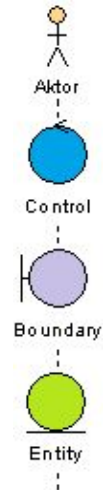
Stereotypy

Aktor - rola zewnętrzna inicjatora względem systemu

Control - element sterujący działaniem

Boundary - element oddziałujący spoza systemu

Entity - element podlegający persystencji



Bloki

Grupa powiązanych komunikatów

alt (alternative) - if else

opt (optional) - if

par (parallel) - przetwarzanie współbieżne

critical - obszar krytyczny

loop - for / while

break - przerwanie

seq - słaba sekwencja, zdarzenia z kilku linii

ignore/consider - ignoruj lub rozważ pewne komunikaty

