



Modelowanie Systemów Informacyjnych (MSI)

dr inż. Mariusz Trzaska
mtrzaska@pjwstk.edu.pl

Wykład 5

Model obiektowy – cz. 3

<http://www.mtrzaska.com>



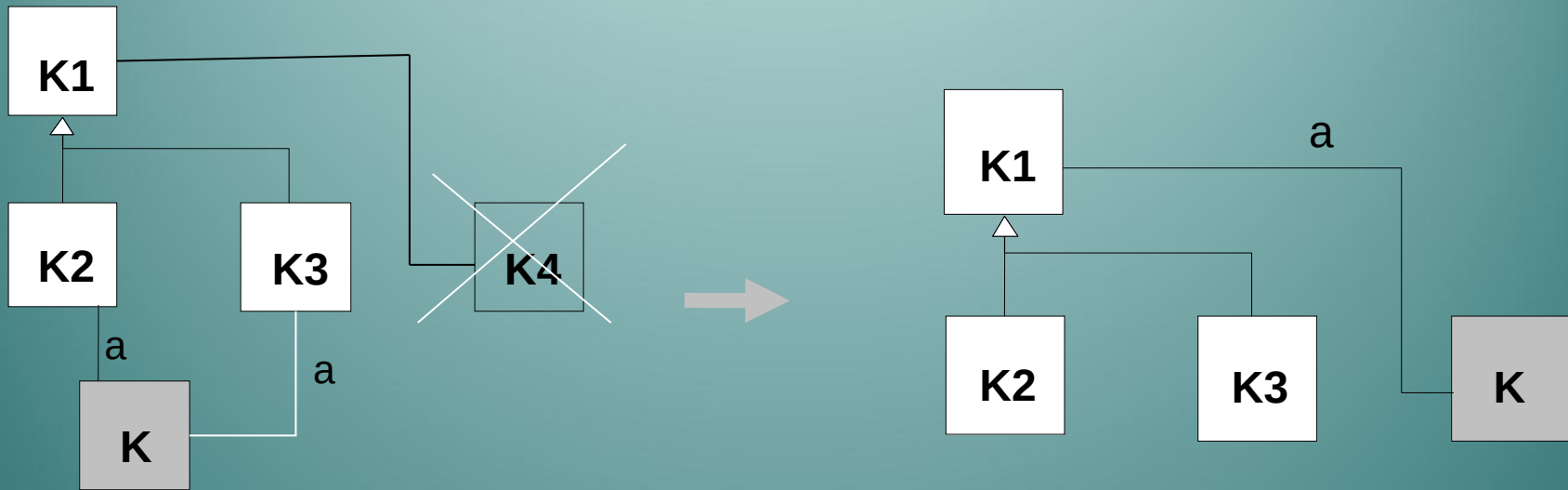
Zagadnienia

Dziedziczenie asocjacji
Asocjacje pochodne
Redukcja liczności
Role wielowartościowe
Trochę więcej o agregacji
Agregacja rekursywna
Trochę więcej o asocjacji kwalifikowanej
Trochę więcej o mechanizmach rozszerzalności



*Wykorzystano materiały z wykładu PRI autorstwa
dr inż. Ewy Stemposz oraz prof. Kazimierza Subiety*

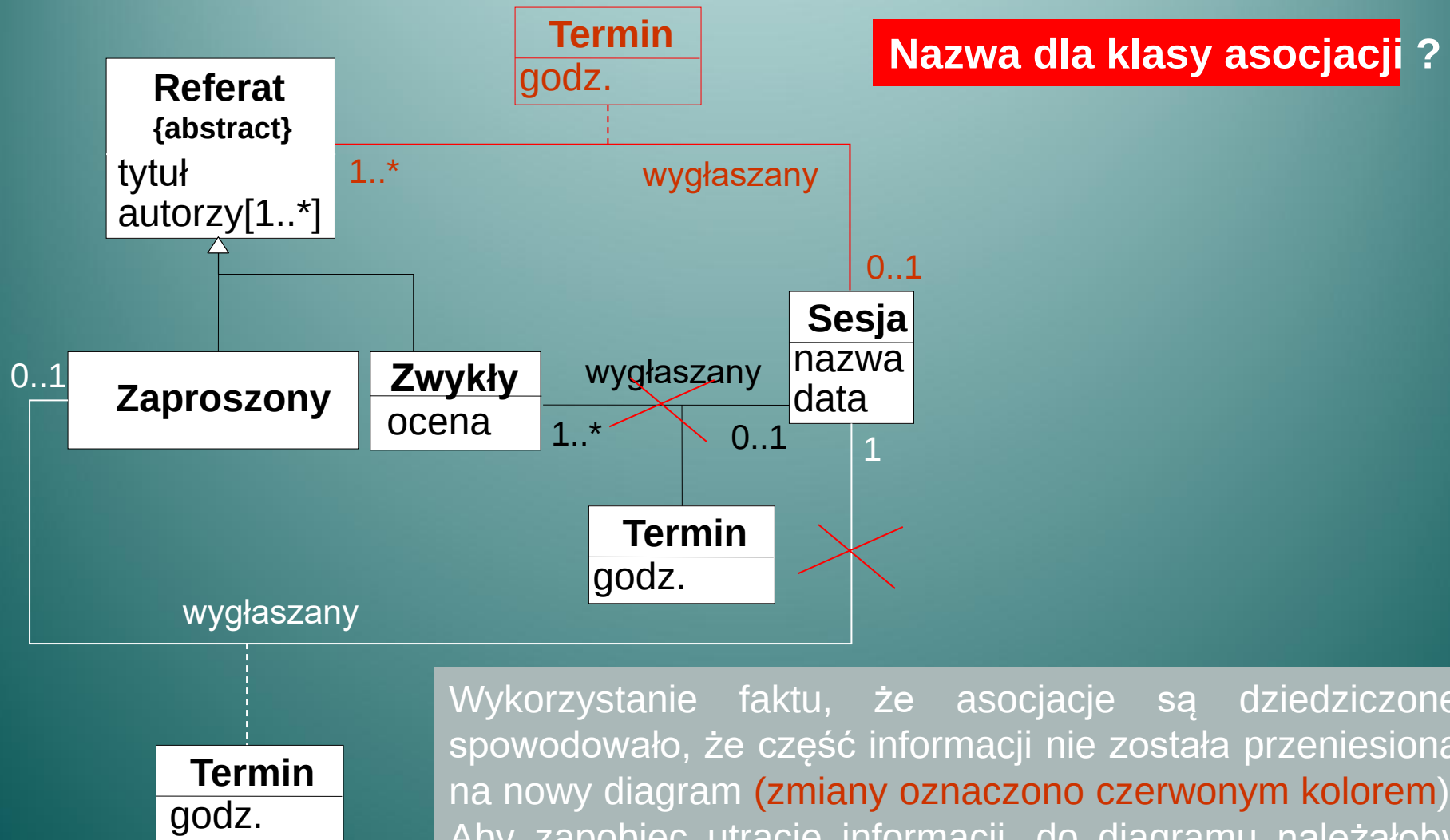
Dziedziczenie asocjacji (1)



Aby obie asocjacje *a* (diagram po lewej stronie) mogły zostać zastąpione jedną asocjacją *a* poprowadzoną od nadklasy *K1* do klasy *K* (diagram po prawej stronie), asocjacje *a* z diagramu po lewej stronie powinny spełniać następujące warunki:

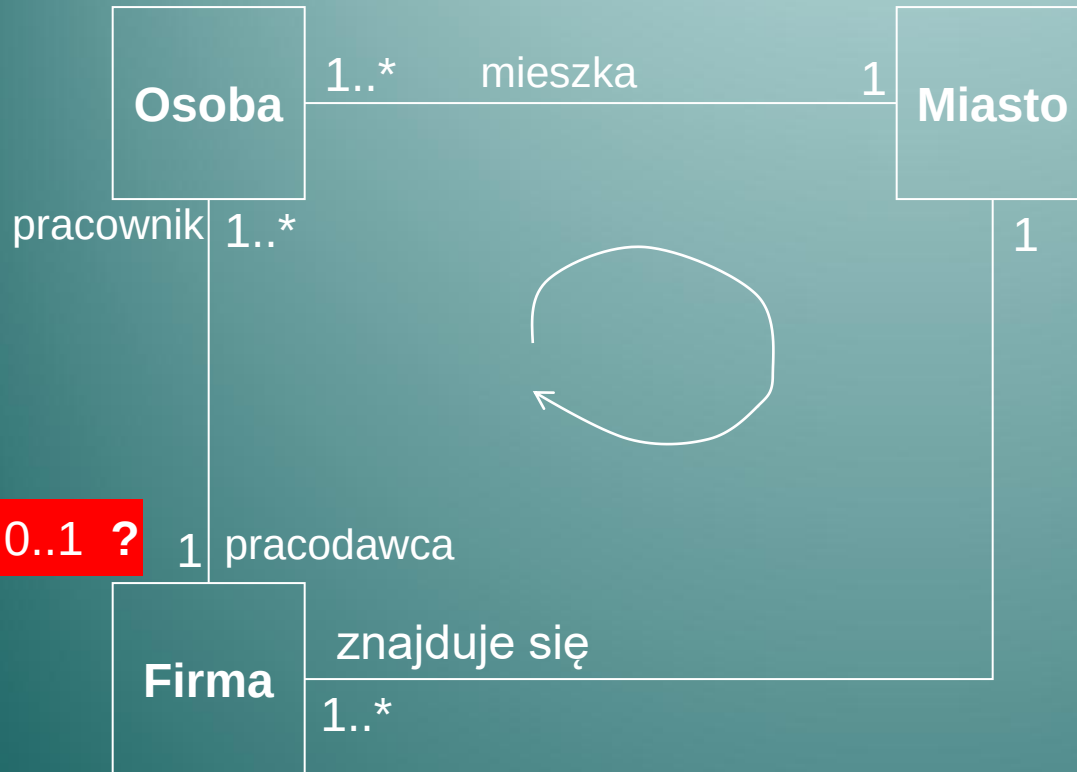
- powinny mieć tę samą semantykę,
- powinny mieć tę samą strukturę,
- byłoby dobrze, gdyby asocjacja *a* łączyła klasę *K* z wszystkimi podklasami klasy *K1* (?)

Dziedziczenie asocjacji (2)



Wykorzystanie faktu, że asocjacje są dziedziczone spowodowało, że część informacji nie została przeniesiona na nowy diagram (**zmiany oznaczone czerwonym kolorem**). Aby zapobiec utracie informacji, do diagramu należałoby wprowadzić odpowiednie ograniczenia.

Asocjacje pochodne



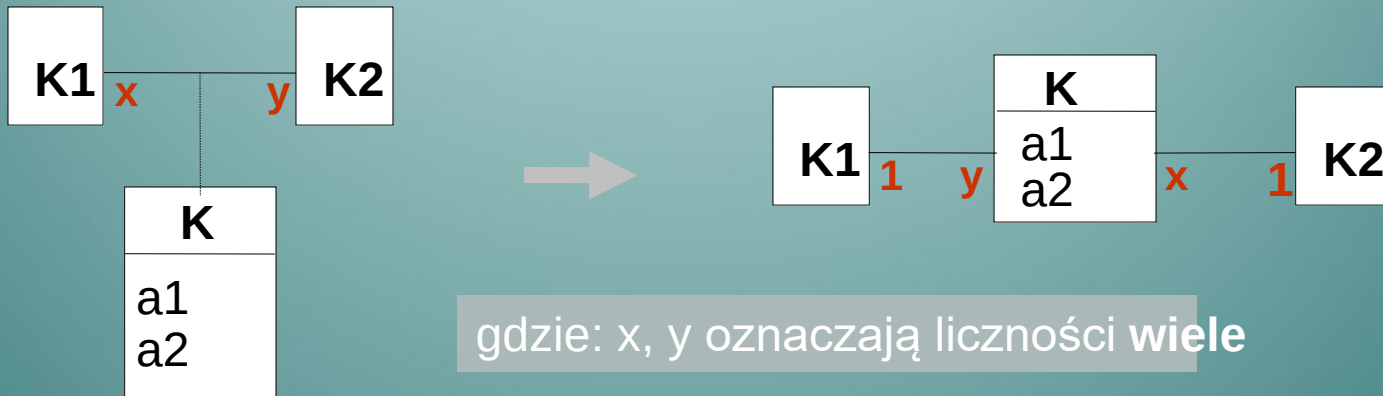
1) Jeśli Osoba mieszka w mieście, w którym pracuje, to jedna z asocjacji: *mieszka* lub *znajduje się* albo powinna zostać oznaczona jako pochodna albo powinna być usunięta z diagramu.

2) Jeśli licznosc roli *pracodawca* zmienimy na 0..1, to asocjacja *mieszka* nie będzie pochodna, ponieważ nie dla wszystkich obiektów powiązania *mieszka* będą mogły być wydedukowane. Podobnie, jeśli licznosc roli *pracownik* zmienimy na *, to asocjacja *znajduje się* nie będzie pochodna.

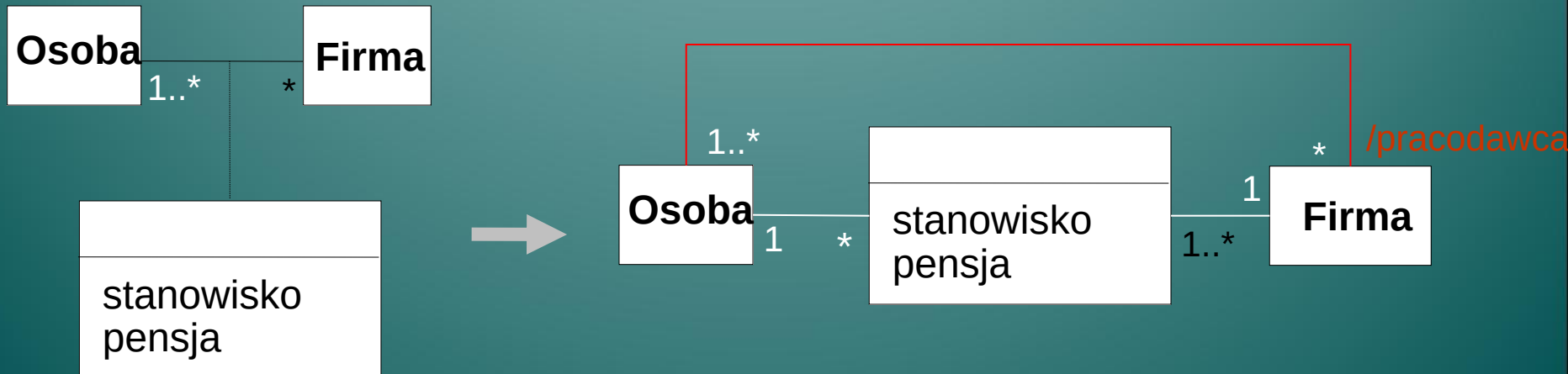
Możliwe asocjacje pochodne:
/mieszka lub */znajduje się*

Redukcja liczności

Wykorzystanie klasy pośredniczącej dla redukcji liczności związków wiele-do-wiele

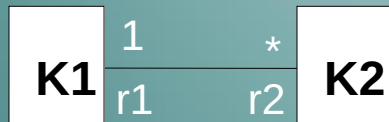


Przykład:



Role wielowartościowe (1)

Rola wielowartościowa to taka rola, dla której górna granica liczności jest większa od 1.



Rola **r2** jest tu rolą wielowartościową.

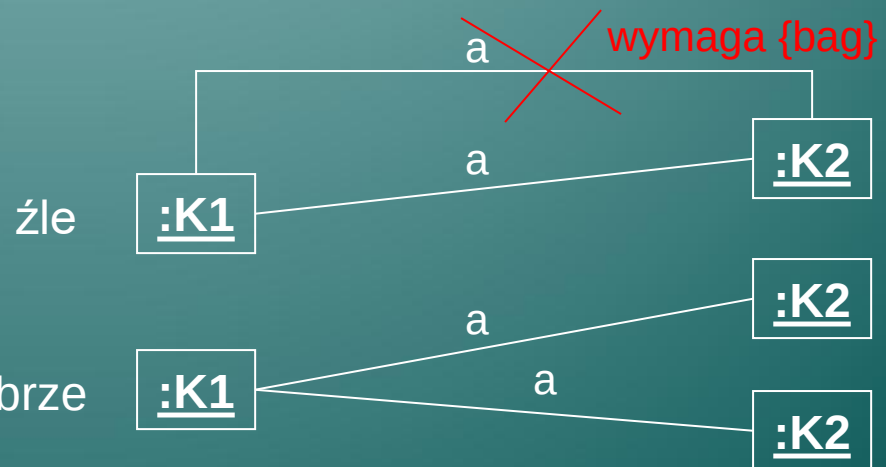
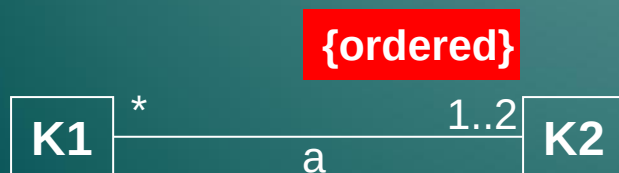
Uwaga:

W sensie dosłownym, liczności obu końców asocjacji oznaczają liczności obu ról.

W UML przyjmuje się domyślnie, że:

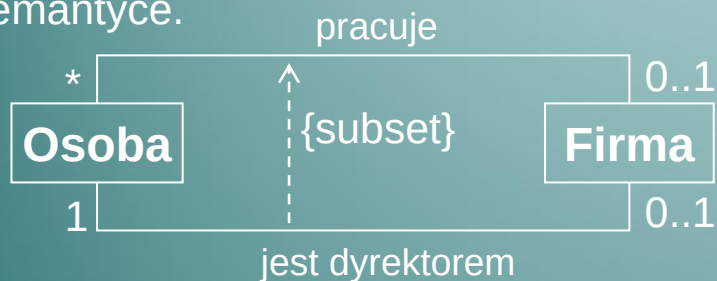
- zbiór obiektów, opisywany daną rolą, jest nieuporządkowany,
- dany obiekt pojawia się tylko jeden raz w w zbiorze obiektów opisanym rolą,
- powyższe reguły mogą zostać zmienione dzięki ograniczeniom **{ordered}**, **{bag}** i stereotypowi «**history**».

Ograniczenie **{ordered}** pozwala na uporządkowanie zbioru obiektów opisanego daną rolą.

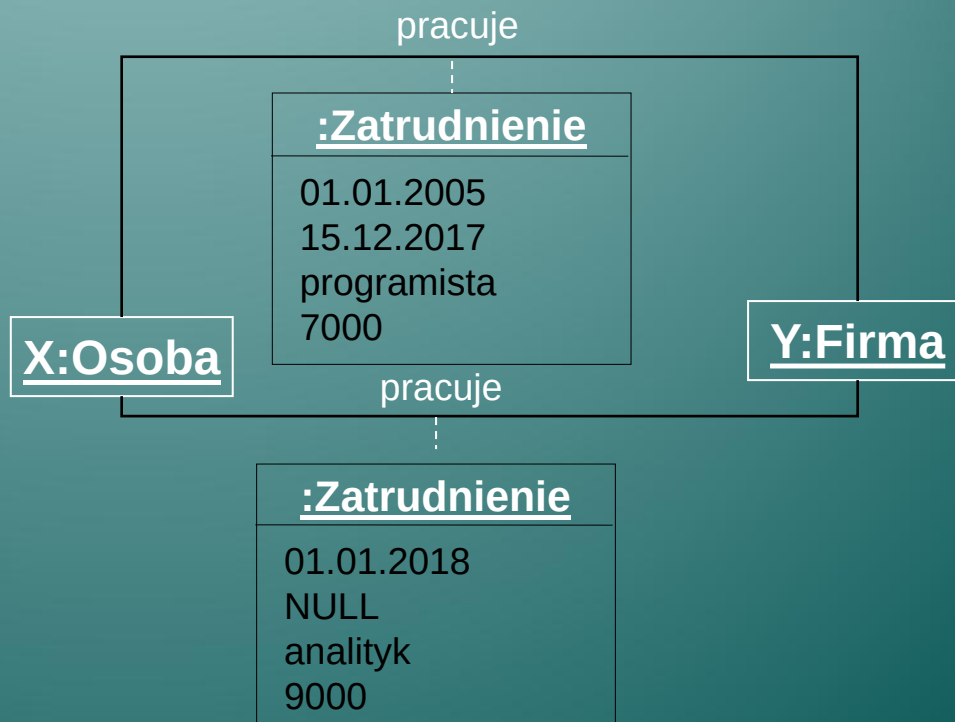
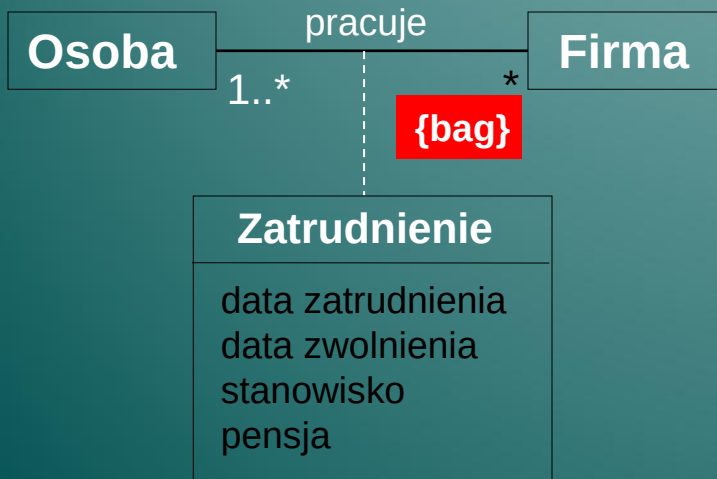


Role wielowartościowe (2)

Między dwoma tymi samymi obiektami może wystąpić więcej niż jedno powiązanie (np. jak na diagramie poniżej), ale nie mogą to być – jak poprzednio – powiązania o tej samej semantyce.

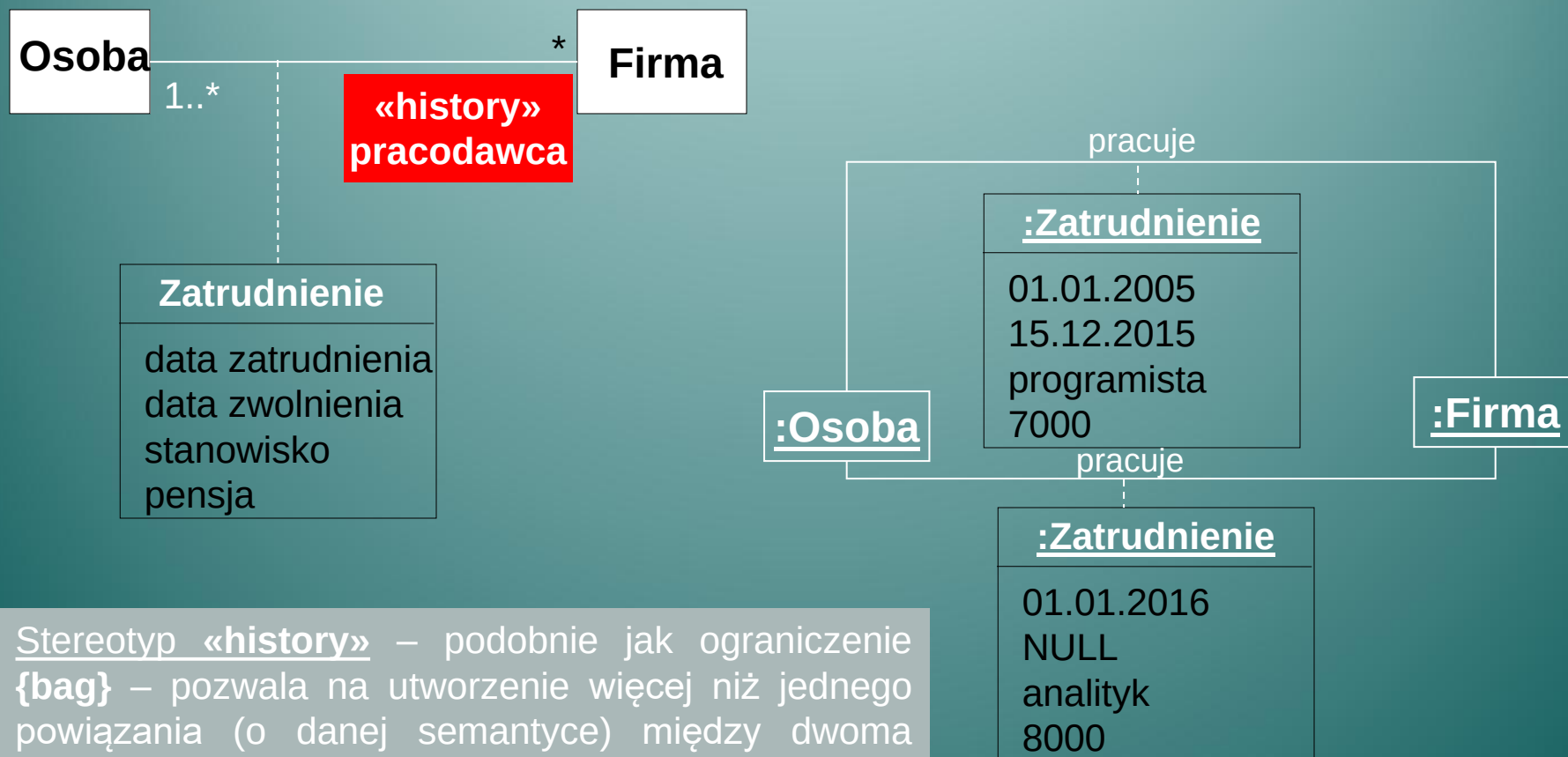


Ograniczenie: {bag}



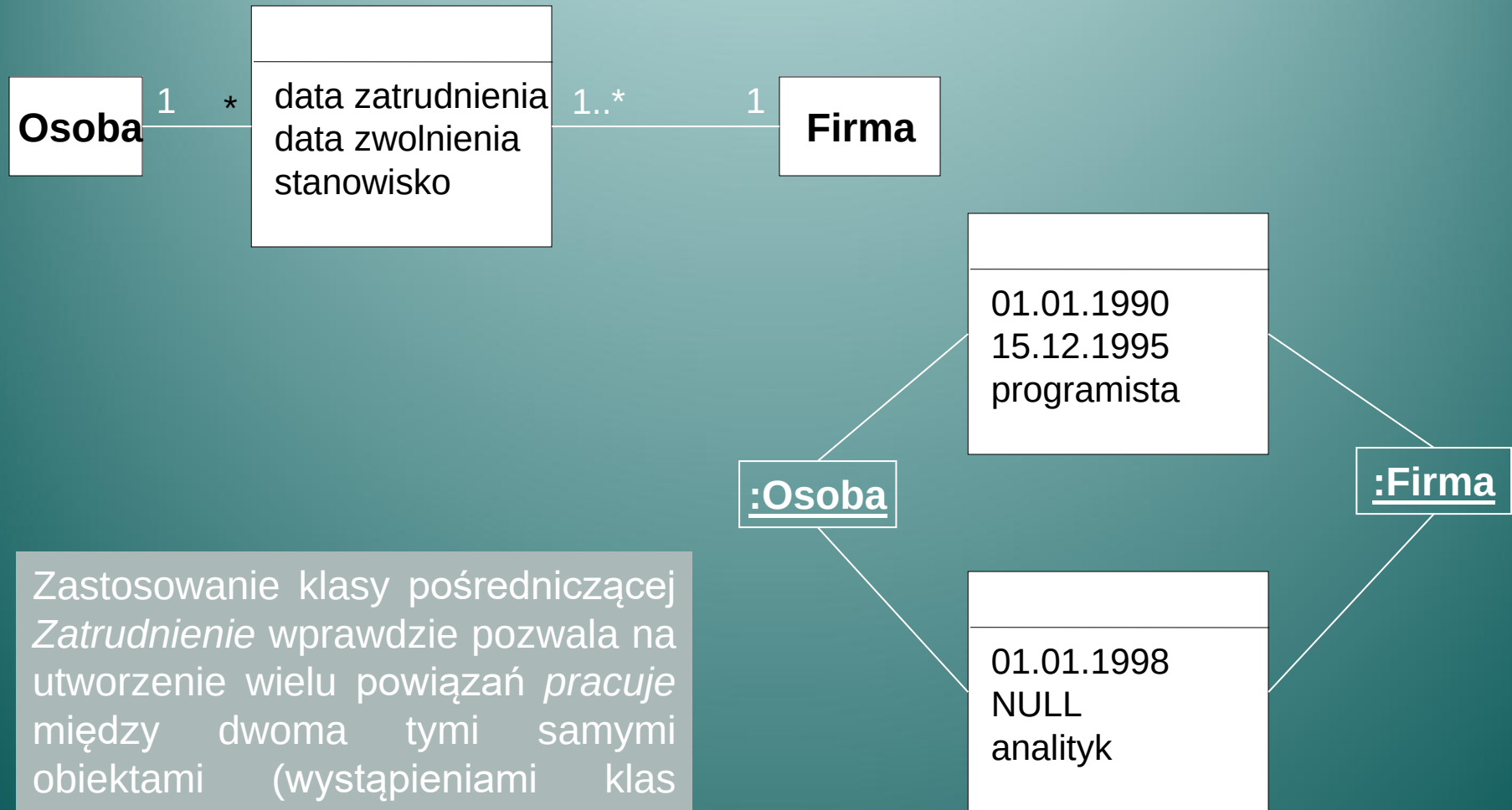
Role wielowartościowe (3)

Stereotyp: «history» dla oznaczenia roli pracodawca



Stereotyp «**history**» – podobnie jak ograniczenie {**bag**} – pozwala na utworzenie więcej niż jednego powiązania (o danej semantyce) między dwoma obiektami; wykorzystywanie tego stereotypu jest ukierunkowane na rejestrowanie zmian w czasie.

Role wielowartościowe (4)

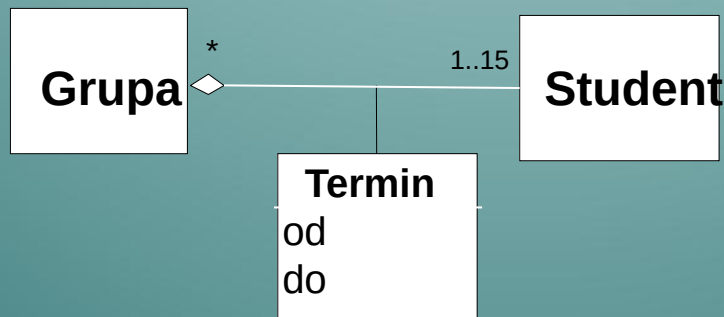


Zastosowanie klasy pośredniczącej *Zatrudnienie* wprowadzie pozwala na utworzenie wielu powiązań *pracuje* między dwoma tymi samymi obiektami (wystąpieniami klas *Osoba* i *Firma*), ale nie pozwala na uwidocznienie tego faktu.

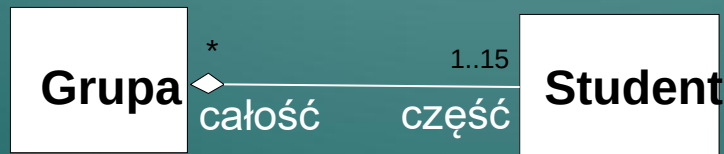
Agregacja (1)

Agregacja jest rodzajem asocjacji; zadaniem agregacji jest **modelowanie związku całość-część**.

- agregacja jest asocjacja: dla obu jej końców są określane licznosci, ponadto (jak każda asocjacja) może mieć atrybuty, np.



- agregacja jest wykorzystywana do modelowania związku całość-część



Agregacja (2)

Inne nazwy dla ról agregacji:

całość

- ✓ składa się z
- ✓ zawiera
- ✓ obejmuje, itp.

część

- ✓ wchodzi w skład
- ✓ należy
- ✓ jest zawarta w, itp.

Nazwa agregacji i nazwy jej ról, jako oczywiste, są z reguły (?) pomijane.

Własności agregacji:



- jest **relacją niesymetryczną**, tzn. jeśli B jest częścią A, to A nie jest częścią B



- jest **relacją przechodnią** (tranzytywną), tzn. jeśli C jest częścią B i B jest częścią A, to C jest częścią A

Kompozycja

- o Kompozycja jest mocniejszą formą agregacji.
- o Czyli jest również asocjacją, a zatem posiada wszystkie jej cechy.
- o Inaczej niż w przypadku agregacji, wykorzystanie kompozycji niesie pewne konsekwencje:
 - Część nie może być współdzielona (liczności),
 - Część nie może istnieć bez całości (całość może),
 - Usunięcie całości oznacza usunięcie również wszystkich jej części.

Kompozycja (2)

o Agregacja



o Kompozycja



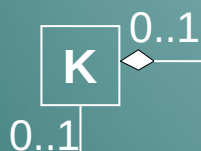
Agregacja rekursywna (1)

Agregacja rekursywna



Obiekt klasy *K* może zarówno wchodzić w skład innych obiektów klasy *K*, jak i może zawierać obiekty klasy *K*.

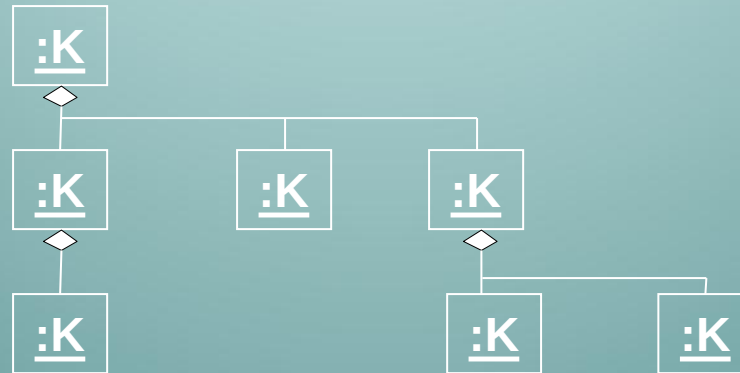
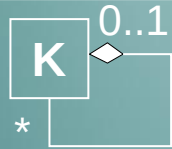
1



- ✓ Co by było, gdyby któryś z końców tej agregacji (lub oba końce) oznaczyć licznnością dokładnie **1** zamiast licznności opcjonalnej **0..1** ?
- ✓ Jakie zmiany wprowadziłoby do powyższego diagramu zastosowanie zwykłej asocjacji zamiast agregacji ?
- ✓ Czy można tu zastosować kompozycję?

Agregacja rekursywna (2)

2



- ✓ Czy można tu zastosować licznosc dokładnie 1 zamiast 0..1 i licznosc 1..* zamiast licznosci * ?
- ✓ Czy można tu zastosować kompozycję?

I



II



- ✓ Dla którego z obu powyższych diagramów możliwość zastosowania kompozycji wydaje się być bezdyskusyjna?

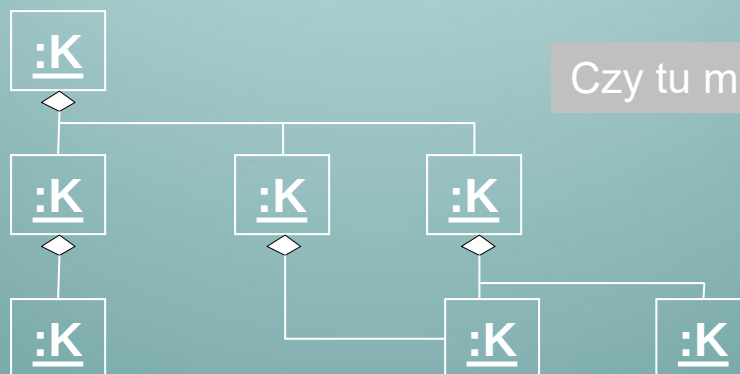
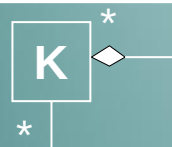
Agregacja rekursywna (3)

- o Modelowanie nie ogranicza się tylko do opisywania kwestii biznesowych, np.
 - produktów,
 - klientów,
 - pracowników.
- o Jak wyglądałby diagram klas służący do przechowywania wyrażeń matematycznych, np.:

$$(x + y/2) * (x/3 - y)$$

Agregacja rekursywna (4)

3



Czy tu można zastosować kompozycję?

Przykłady agregacji rekursywnych

I

Program

II

2-gi operand

1-szy operand

1

Człon

✓ Jak wyglądałby diagram obiektowy dla wyrażenia, np.
 $(x + y/2) * (x/3 - y)$

1..*

Blok

Instrukcja
złożona

Instrukcja
prosta

Wyrażenie

operator binarny

Zmienna
nazwa

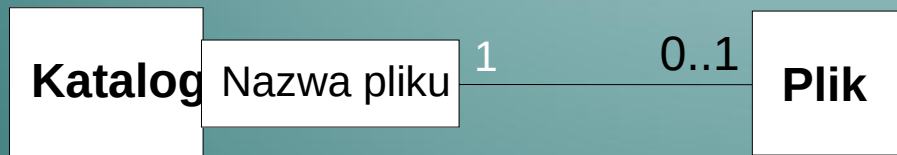
Stała
wartość

Gdzie można by tu zastosować kompozycję – w I czy w II?

Asocjacja kwalifikowana (1)



{ nazwa pliku jest unikatowa w ramach katalogu }



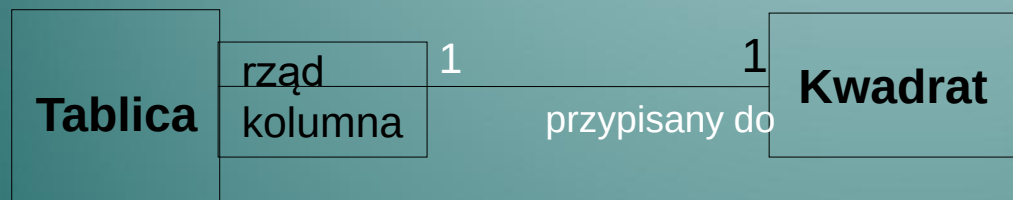
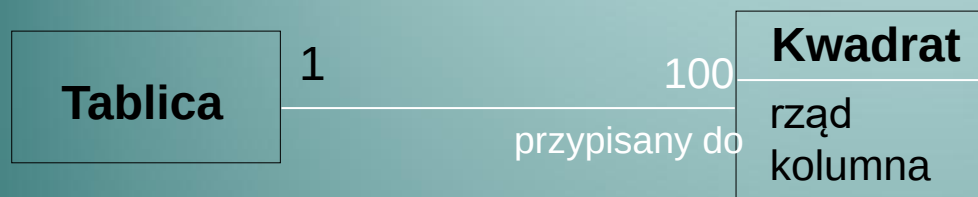
1

Perspektywa pojęciowa – plik jest w ramach katalogu jednoznacznie identyfikowany przez nazwę.

2

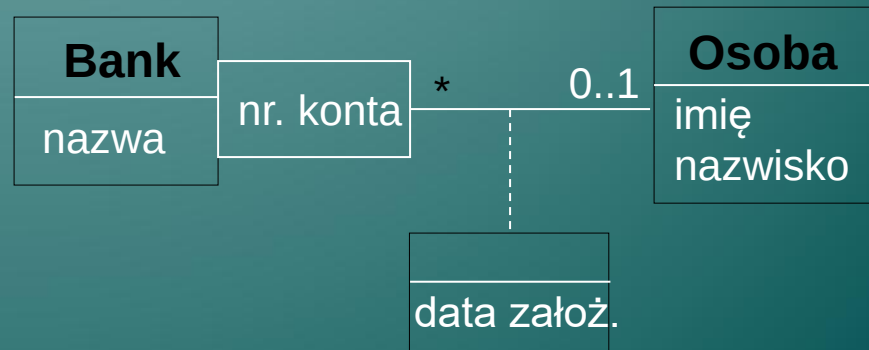
Perspektywa projektowa – wskazanie na to, że katalog plików można zorganizować jako tablicę asocjacyjną (słownik) (przeszukiwanie za pomocą nazwy pliku).

Asocjacja kwalifikowana (2)



Kwalifikator asocjacji może być określony przez więcej niż jeden atrybut. Warunek – wartości tych atrybutów muszą pozwolić na jednoznaczną identyfikację obiektu/ grupy obiektów w ramach pewnego zbioru obiektów (tutaj – w ramach zbioru kwadratów przypisanych do jednej konkretnej tablicy, czyli do jednego obiektu klasy *Tablica*).

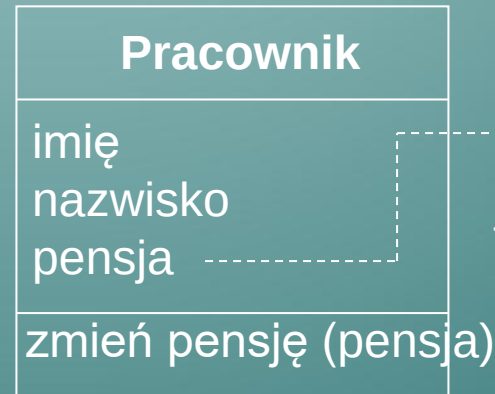
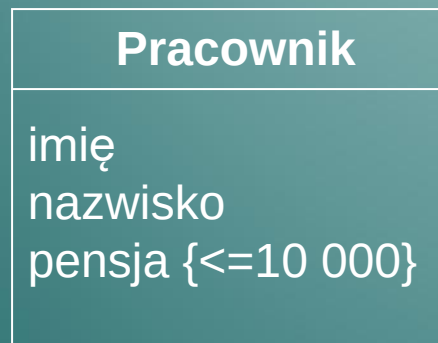
Asocjacja kwalifikowana, jak każda asocjacja, może posiadać atrybuty.



Ograniczenia

Ograniczenia specyfikują restrykcje nakładane na elementy modelu. Mogą stanowić wyrażenia języka naturalnego czy języka formalnego (np. OCL w UML), mogą też przyjmować postać formuły matematycznej lub fragmentu kodu (czy też pseudokodu).

Notacja: **Ograniczenia** są zawarte wewnątrz {} i umieszczane za elementem w klasie, lub poza klasą. Z reguły są umieszczane w komentarzu (przykład na następnej folii).



ograniczenie statyczne

{<=10 000}

{pensja nie wzrasta o więcej niż 300}

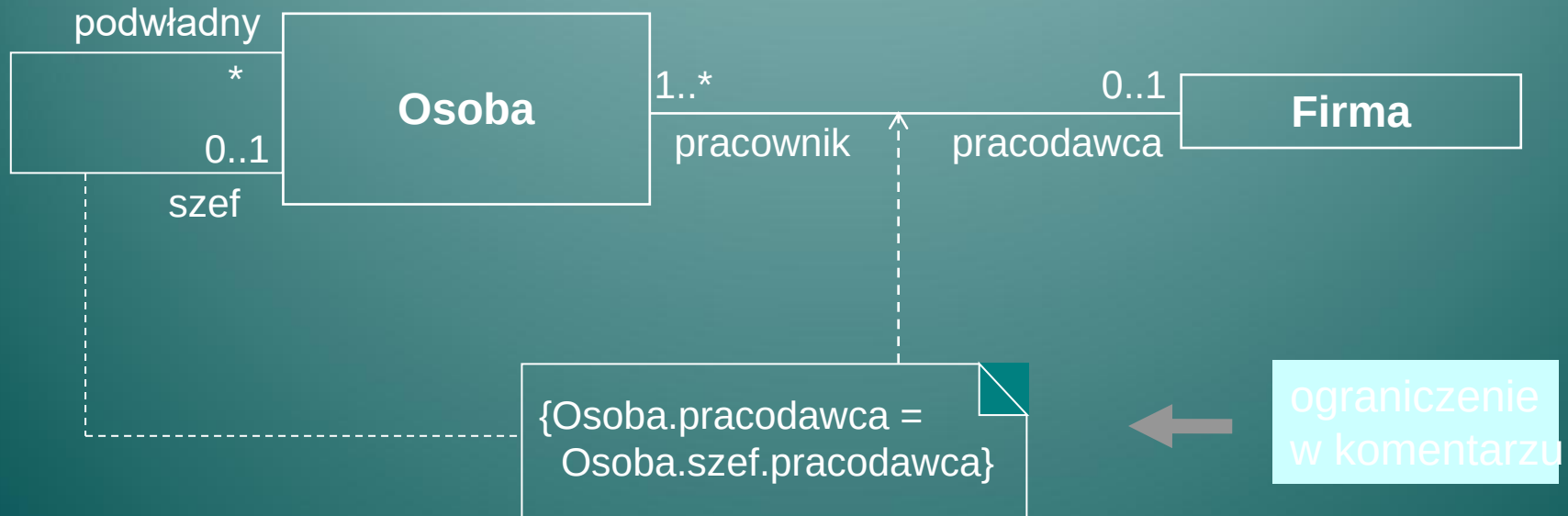
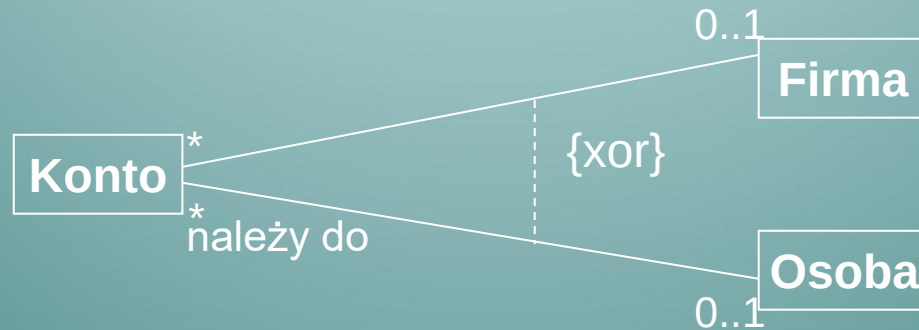
ograniczenie dynamiczne

W przypadku ograniczenia dynamicznego – w przeciwieństwie do ograniczenia statycznego – interesuje nas poprzedni stan elementu, dla którego wyspecyfikowano ograniczenie.

Czy powiedzie się próba zmiany pensji z 2500 na 5500, przy ograniczeniach jak powyżej?

Ograniczenia; przykłady

Symbole, takie jak - - - - oraz - - - - > są używane do wskazywania elementów, na które zostały nałożone ograniczenia.



Ograniczenia predefiniowane; przykłady

j. angielski

{complete}

{incomplete}

{disjoint}

{overlapping}

{or}

{xor}

{ordered}

{subset}

{bag}

{hierarchy}

{dag}

dag - directed acyclic graph

j. polski

{podział całkowity}

{podział nie całkowity}

{podział rozłączny}

{podział nierozłączny}

{lub} (suma logiczna)

{albo} (różnica symetryczna)

{uporządkowane}

{podzbiór}

{wielozbiór}

{hierarchia}

{graf acykliczny skierowany}