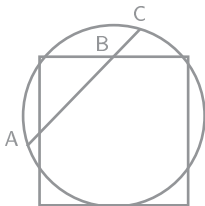


Inżynieria oprogramowania

Radosław Klimek

2015-23



<http://home.agh.edu.pl/rklimek>

1 Relacje modelu obiektowego

1 Relacje modelu obiektowego

Relacje modelu obiektowego



Edouard MANET: *Śniadanie na trawie*

Kategorie relacji w podejściu obiektowym

W klasycznym podejściu obiektowym wyróżnia się **pięć podstawowych** typów relacji występujących pomiędzy klasami/obiektami:


- 1 asocjacja (ang. *association*) lub skojarzenie,
- 2 agregacja (ang. *aggregation*),
- 3 kompozycja (ang. *composition*),
- 4 dziedziczenie (ang. *inheritance*) lub generalizacja,
- 5 uściślenie (ang. *refinement*).

Kategorie relacji w podejściu obiektowym – uwagi

- W przypadku niektórych relacji można zauważyć bliskie podobieństwo, np. w przypadku relacji dziedziczenia i uściślenia. Te dwie relacje opisują zależności pewnych linii generowania klas od specyfikacji bardziej ogólnych do bardziej szczegółowych.
- Jednak istnieje pewna istotna różnica pomiędzy tymi relacjami, a jest nią fakt, że dziedziczenie polega na dostosowaniu zachowania obiektów do określonych wymagań, natomiast uściślenie polega na precyzowaniu typu danych przy zachowanym tym samym (ogólnym) zachowaniu obiektu.

Relacja asocjacji

Definicja

Relacja **asocjacji** (lub **skojarzenia**) (ang. *association*) oznacza możliwość przesyłania informacji pomiędzy obiektami. Fakt komunikacji ma miejsce wtedy, gdy jeden z obiektów korzysta z usług dostarczanych przez obiekt inny. 

Relacja asocjacji

Definicja

Relacja **asocjacji** (lub **skojarzenia**) (ang. *association*) oznacza możliwość przesyłania informacji pomiędzy obiektami. Fakt komunikacji ma miejsce wtedy, gdy jeden z obiektów korzysta z usług dostarczanych przez obiekt inny. └



Relacja asocjacji – uwagi

- asocjacja może łączyć dwie lub więcej klas;
- obiekt korzystający z usług nie jest zawarty (agregacja) wewnątrz dostarczającego usługi;
- czas życia klasy używanej nie jest zakresem działania klasy użytkowników;
- relacja asocjacji wiąże słabiej niż relacja agregacji;
- przykładem relacji może być np. związek między pracownikiem i jego zakładem pracy;

Relacja asocjacji – uwagi (cd.)


- asocjacja umożliwia przechodzenie (nawigację) pomiędzy powiązаныmi obiektami w dowolnym kierunku;
- relacja asocjacji może być określona jako relacja klient-serwer.

Asocjacja określa istnienie przesyłania informacji pomiędzy obiektami, przy czym fakt komunikacji pomiędzy obiektami ma miejsce gdy jeden obiekt korzysta z usług drugiego, stąd też relacja skojarzenia jest często nazywana relacją użycia (np. HOOD).

Skojarzenie to połączenie lub powiązanie idei.

Relacja agregacji

Definicja

Relacja **agregacji** (ang. *aggregation*) ma miejsce wtedy, gdy jeden obiekt zawiera inne. Obiekt zewnętrzny nazywany jest **właścicielem**, a obiekty zawarte (wewnętrzne) **komponentami**. 

Relacja agregacji

Definicja

Relacja **agregacji** (ang. *aggregation*) ma miejsce wtedy, gdy jeden obiekt zawiera inne. Obiekt zewnętrzny nazywany jest **właścicielem**, a obiekty zawarte (wewnętrzne) **komponentami**. └



Relacja agregacji – uwagi

- Dopuszcza się także możliwość istnienia wspólnych komponentów przez kilku właścicieli;
- właściciel jest zazwyczaj uprawniony do kreowania i usuwania obiektów komponentów;
- agregacja może być rozpatrywana jako szczególny przypadek relacji asocjacji;
- agregacja modeluje stosunek całości do jej części (np. samolot i jego śmigła);
- obiekty są powiązane związkiem agregacji, jeżeli jeden z nich można uważać za część drugiego, zaś cykl i czas życia tych obiektów są jednakowe.

Relacja agregacji jest czasem zwana relacją zawierania (np. HOOD).

Relacja kompozycji

Definicja

Relacja **kompozycji** (ang. *composition*) jest silniejszą odmianą relacji agregacji poprzez wprowadzenie następujących ograniczeń:

- komponenty nie mogą być dzielone pomiędzy wielu właścicieli,
- komponenty są powoływane i usuwane przez właściciela.



Relacja kompozycji

Definicja

Relacja **kompozycji** (ang. *composition*) jest silniejszą odmianą relacji agregacji poprzez wprowadzenie następujących ograniczeń:

- komponenty nie mogą być dzielone pomiędzy wielu właścicieli,
- komponenty są powoływane i usuwane przez właściciela.




Relacja kompozycji – uwagi

Relacja ta jest stosowana w sytuacjach, gdy pewne wątki są powoływane przez obiekt-właściciel, który grupuje korzeń wątków wykonywania, natomiast komponenty realizują poszczególne wątki.

Relacja dziedziczenia

Definicja

Relacja **dziedziczenia** (ang. *inheritance*) lub **generalizacja** ma miejsce wtedy, gdy pomiędzy klasami obiektów zachodzi przekazywanie cech (definicji atrybutów, metod, itd.) z **nadklasy** do jej **podklas**.

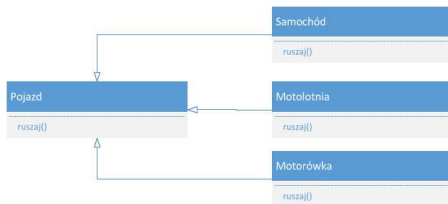
Albo też równoważnie, dziedziczenie ma miejsce wtedy, gdy dana klasa może być zdefiniowana z wykorzystaniem charakterystyk innej klasy, będącej **klasą-rodzicem**, z ewentualnym dodaniem dodatkowych atrybutów lub rozszerzeniem zachowań. 

Relacja dziedziczenia

Definicja

Relacja **dziedziczenia** (ang. *inheritance*) lub **generalizacja** ma miejsce wtedy, gdy pomiędzy klasami obiektów zachodzi przekazywanie cech (definicji atrybutów, metod, itd.) z **nadklasy** do jej **podklasy**.

Albo też równoważnie, dziedziczenie ma miejsce wtedy, gdy dana klasa może być zdefiniowana z wykorzystaniem charakterystyk innej klasy, będącej **klasą-rodzicem**, z ewentualnym dodaniem dodatkowych atrybutów lub rozszerzeniem zachowań.



Relacja dziedziczenia – uwagi

- Bardzo efektywny mechanizm definiowania nowych klas z wykorzystaniem rodziców, tzn. potomek dziedziczy cechy rodzica, za wyjątkiem tych cech które są określone na nowo;
- mechanizm wyrażania podobieństwa pomiędzy klasami, pozwalający na upraszczanie definicji klas podobnych do już zdefiniowanych;
- przykład: obiekt klasy „Student” dziedziczy wszystkie własności (definicje atrybutów, metody) określone w ramach klasy „Osoba”;
- dziedziczenie to podstawowy mechanizmem sprzyjającym ponownemu użyciu.

Relacja dziedziczenia – uwagi końcowe

- Dziedziczenie jest realizowane w poszczególnych językach i systemach na wiele różnych sposobów, a mianowicie możemy mieć do czynienia z dziedziczeniem pojedynczym lub prostym (od jednego rodzica), np. Ada, Smalltalk, albo też dziedziczeniem wielokrotnym, np. C++.
- Dziedziczenie przebiega zazwyczaj wzdłuż pojedynczej charakterystyki, względnie niewielkiego zbioru charakterystyk.

Relacja dziedziczenia – rodzaje

Istnieje wiele form dziedziczenia, np. dziedziczenie oparte na prototypach lub delegacja, dziedziczenie dynamiczne, wielokrotne dziedziczenie, itd.

- **Dziedziczenie dynamiczne** (ang. *dynamic inheritance*) – dziedziczenie przez obiekt cech klas lub cech innych obiektów (np. ich stanu) podczas czasu wykonania; często powiązane z możliwością dynamicznej zmiany dziedziczonych cech lub dynamicznej zmiany przynależności obiektu do klasy. Dziedziczenie dynamiczne można osiągnąć wyłącznie w językach/systemach, gdzie klasa jest obywatelem pierwszej kategorii lub w językach/systemach opartych o koncepcję prototypu.
- **Dziedziczenie statyczne** (ang. *static inheritance*) – dziedziczenie, które można rozstrzygnąć podczas kompilacji.

Relacja uściślenia

Definicja

Relacja **uściślenia** (ang. *refinement*) to relacja umożliwiająca niekompletne specyfikowanie elementów (tutaj klas), które mogą być konkretyzowane poprzez dodanie elementów wcześniej nieokreślonych. Niekompletna specyfikacja stanowi definicję wyższego poziomu, będąc rodziną klas nieprzydatną do bezpośredniego użycia, a dopiero po uściśleniu definicja staje się kompletna i może być stosowana w odpowiedniej konstrukcji. ┘

Relacja uściślenia – uwagi

W poszczególnych językach takie niekompletne specyfikacje nazywają się odmiennie, np. w C++ używane jest pojęcie **szablonu** (ang. *template*), natomiast w Adzie jest to **jednostka rodzajowa** (ang. *generic*), bardzo dokładnie uściślenie zostało zdefiniowane w języku UML.

Porównanie terminologii obiektowej

Martin/Odell	Booch	Coad/Yourdon	Jacobson	Shlaer/Mellor	Rumbaugh
Typ obiektowy	Klasa	Klasa i obiekt	Typ obiektowy	Obiekt	Klasa
Obiekt	Obiekt	Obiekt	Egzemplarz	Egzemplarz	Obiekt
Typ związków	Używanie relacji	Związek obiektów		Związek	Powiązanie
Odwzorowanie	Rola		Związek znajomości	Odwzorowanie	Rola
Uogólnienie	Dziedziczenie	Gen-spec	Dziedziczy	Podtyp-nadtyp	Uogólnienie
Składanie	Zawieranie	Część-całość			Agregacja

Identyfikowanie klas i obiektów – prosty przykład

Identyfikowanie klas obiektów na przykładzie pewnego opisu problemu:

„System rezerwacji sprzedaży biletów na przedstawienia
w różnych teatrach”

Wstępnie można tu wyróżnić następujące klasy obiektów:

- System,
- Rezerwacja,
- Bilet,
- Przedstawienie,
- Teatr.

W kolejnym kroku szuka się związków pomiędzy nimi poprzez odnajdywanie podobieństw, co prowadzi do eliminacji niewłaściwych klas obiektów wg. odpowiednich reguł.

Reguły eliminowania klas

Przykładowe reguły eliminacji niewłaściwych klas obiektów:

- 1 redukcjonowanie klas zamiarowych – gdy powstają klasy o różnych nazwach, lecz dotyczące tego samego rodzaju obiektu;
- 2 redukcjonowanie klas nieistotnych – usunięcie klas nie mających związku z opisywanym problemem (decyzja raczej arbitralna);
- 3 poprawianie klas źle określonych – każda klasa powinna mieć dobrze określone granice swojego występowania;
- 4 redukcjonowanie klas będących cechami – nazwy klas mogą opisywać własności;

Reguły eliminowania klas (dok.)

- 5 redukowanie klas będących operacjami – podobnie jak powyżej należy dokonać redukcji, gdyż operacje określone będą jako sekwencje w tzw. modelu dynamicznym;
- 6 stosowanie odpowiednich nazw klas – nazwa klasy powinna odzwierciedlać naturę obiektu, a nie jego rolę jaką obiekt odgrywa w systemie;
- 7 eliminowanie klas obiektów implementacyjnych – klasy muszą odzwierciedlać obiekty rzeczywiste, a nie implementacyjne, takie np. jak: procesory, procedury itd.