



Programovanie v jazyku C#

Vzt'ahy medzi triedami

prednáška 9

Ing. Ján Magyar, PhD.

ak. rok. 2022/2023 ZS

Vzt'ahy v OOP

- každý program je definovaný ako interakcia medzi objektmi
- triedy (a objekty) sú definované ako kohézna entita so zväčša sebestačnou funkcionalitou
- v niektorých prípadoch stále potrebujeme, aby objekty spolupracovali a vedieť formálne popísať túto spoluprácu
- vzt'ahy môžu existovať na úrovni tried a inštancií

Vzt'ahy na úrovni tried

- definované medzi triedami
- všetky inštancie tej istej triedy majú automaticky tie isté vzt'ahy
- nevzt'ahujú sa na konkrétne objekty
- vyššie vzt'ahy

Generalizácia

- nazýva sa aj dedenie
- formálny popis vzťahu, ktorý je implementovaný pomocou dedičnosti v OO jazykoch
- jedna trieda je špeciálnym príkladom druhej triedy, ktorá je zovšeobecnením prvej triedy
- vzťah typu “is-a”
- nemôžeme definovať na úrovni inštancií

Realizácia

- nazýva sa aj implementácia
- jeden prvok realizuje správanie definované druhým prvkom
- môže existovať medzi triedami, rozhraniami, komponentmi a balíkmi
- zvyčajne medzi triedou a rozhraním
- vzťah typu “is-a-kind-of”

Vzt'ahy na úrovni inštancií

- vytvorené počas behu, existujú iba za istú časť trvania programu
- objekty tej istej triedy môžu mať rôzne vzt'ahy, ale zvyčajne majú vzt'ahy rovnakého typu
- konkrétne inštancie sú vo vzt'ahu s rôznymi konkrétnymi inštanciami
- nižší vzt'ah

Závislosť

- sémantické prepojenie
- jednosmerný vzťah medzi dvomi prvkami
- závislý prvok (**client**, **source**) je prepojený s nezávislým prvkom (**supplier**, **target**)
- ak zmeníme suppliera, možno potrebujeme zmeniť aj klienta
- najvšeobecnejší vzťah, často definovaný presnejšie

Asociácia

- jedna inštancia môže vyžadovať, aby druhá vykonala akciu pre ňu
- štrukturálny vzťah
- príklady
 - poslanie správy
 - volanie metódy
 - volanie členskej metódy
- môže byť jednosmerná alebo obojsmerná

Definovanie asociácie

- smerovanie
- môže mať názov
- môže definovať úlohy
- môže definovať indikátory vlastníctva
- podpora kardinality

Typy asociácie

- obojsmerná
- jednosmerná
- agregácia (zoskupenie)
 - kompozícia
 - agregácia
- reflexívna

Agregácia

- typ asociácie
- vzťah typu “má”
- predstavuje vzťah typu celok-súčasť alebo súčasť-niečoho
- binárna asociácia
- implementovaná ako asociácia
- inštancie ktoré sú obsiahnuté nemajú silnú životnú závislosť od životného cyklu kontajnera

Kompozícia

- typ asociácie
- vzťah typu “obsahuje” alebo “skladá sa z”
- binárna asociácia
- implementovaná ako asociácia
- inštancie ktoré sú obsiahnuté majú silnú životnú závislosť od životného cyklu kontajnera

Agregácia versus kompozícia

- reprezentuje stav softvéru alebo databázy
- ak vymažeme kontajner, časti môžu ďalej existovať (študijná skupina a študenti)
- reprezentuje súčasti
- pri vymazaní kontajnera sa vymažú aj súčasti (univerzita sa skladá z fakúlt)

UML diagram tried

- pre grafickú reprezentáciu algoritmov používame vývojové diagramy, pre reprezentáciu vzťahov medzi triedami sa používa UML diagram tried
- každá trieda má vlastný blok
- bloky sú prepájané rôznymi šípkami ktoré definujú vzťah
- nezávislý od jazyka a implementácie

Používanie diagramov tried

- základný stavebný blok pre objektovo-orientované modelovanie
- modeluje štruktúru aplikácie
- základ pre implementáciu
- môže sa používať pre modelovanie údajov
- definuje
 - aké triedy potrebujeme implementovať
 - aká je ich štruktúra a správanie
 - aké sú vzťahy medzi triedami

Reprezentácia triedy

- každá trieda sa zobrazí v boxe
- box sa skladá z troch častí
 - názov triedy (alebo rozhrania)
 - atribúty triedy
 - zoznam operácií (metód)

Article
- name: string - price: int - taxRate: float
+ Article(name: string, price: int, taxRate: float) + getPrice() : string + setTaxRate(newTaxRate: float): void

Reprezentácia prístupu

+ public

- private

protected

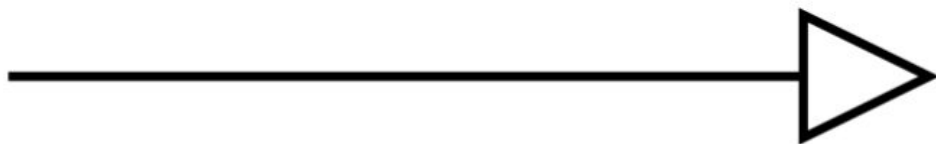
~ package/internal

Členské a statické premenné

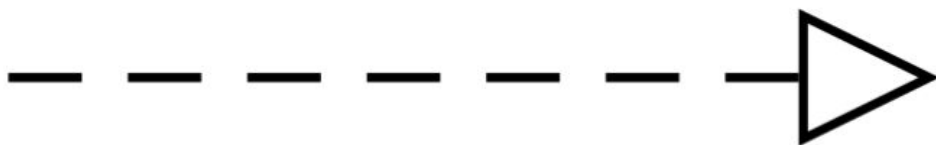
- člen inštalácie
 - každý objekt triedy má vlastnú kópiu
 - zavolanie metódy spôsobuje zmenu stavu objektu
- člen triedy (statický člen)
 - hodnota je rovnaká pre každú inštanciu
 - pri volaní metódy nedochádza k zmene stavu
 - reprezentované podčiarknutými názvami

Reprezentácia vzťahov na úrovni tried

- generalizácia
 - šípka s prázdny trojuholníkom z podtriedy do nadtriedy
- realizácia
 - šípka s čiarkovanou čiarou a prázdny trojuholníkom



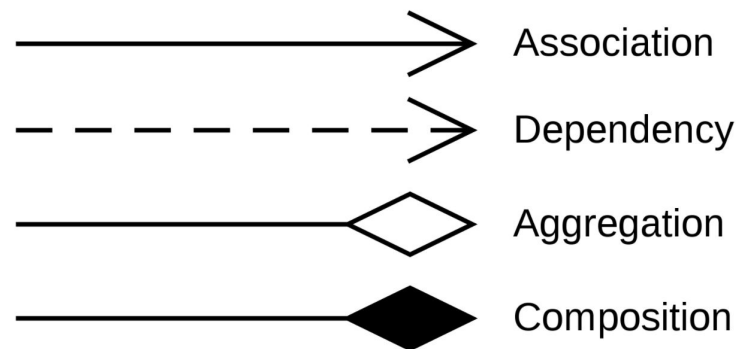
Inheritance



Realization /
Implementation

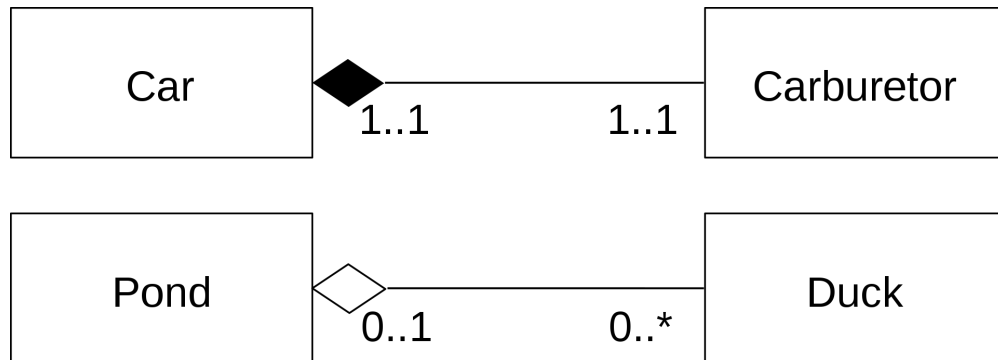
Reprezentácia vzťahov na úrovni inštancií

- závislosť
 - čiarkovaná šípka z klienta do suppliera
- asociácia
 - plná šípka
 - bez šípok v prípade obojsmernej asociácie
- agregácia
 - plná čiara s prázdny kosoštvorcom
- kompozícia
 - plná čiara s plným kosoštvorcom



Kardinalita

- 0 - žiadna inštancia
- 0..1 - žiadna inštancia alebo presne jedna
- 1/1..1 - presne jedna inštancia
- 0..*/* - žiadna inštancia alebo viac inštancií
- 1..* - jedna alebo viac inštancií



otázky?