

ZÁKLADNÉ PRVKY CLASS DIAGRAM-u - súčasť UML (Unified Modeling Language)

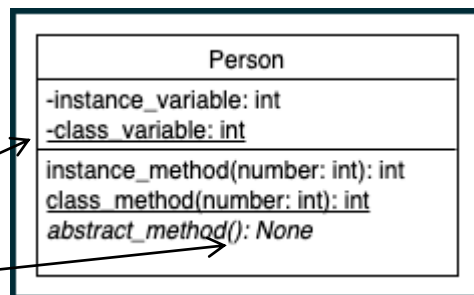
Class specification - špecifikácia triedy

* - označuje viacero

názvy premenných s - t.j. nie sú z vonku prístupné (enkapsulácia) očakávaný dátový typ návrat. hod. je int

označenie class variable

parametre metódy správania



- Názov triedy. Začína veľkým písmenom a ak je abstraktná je to *Italic-om*
- Inštančné premenné a ich očakávaný dátový typ navratovej hodnoty je int
- Premenné triedy a ich očakávaný typ. Ak je - resp. podčiarknutý ide o **private**
- Inštančná metóda s očakávaním jedného parametra správania typu int
- Metóda triedy s jedným parametrom typu int očakávajúca vrátenie hod. int
- Abstraktná metóda vo fonte *Italic* s premennými ktoré nie sú zvonka prístupné a metóda nič nevráti (None) Používa sa tu **enkapsulácia**

- je privátna, + je verejná _____ - označuje class variable

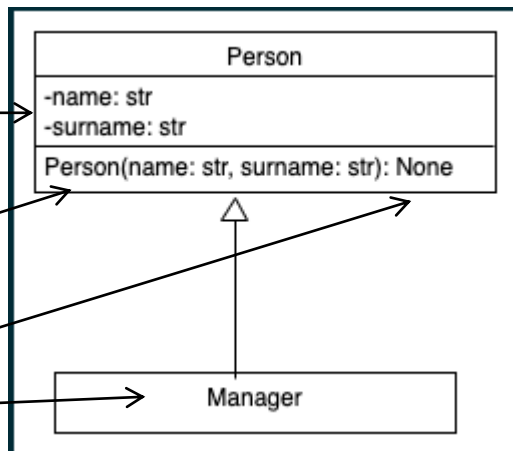
Generalization – dedenie

Inštančné atribúty ak by pri nich nebolo žiadne znamienko tak je to ako + t.j. public čiže zvonka prístupné

Konštruktor ktorého názov je ako názov triedy

Neočakáva sa žiadna hodnota

Nemá nič špecifikované



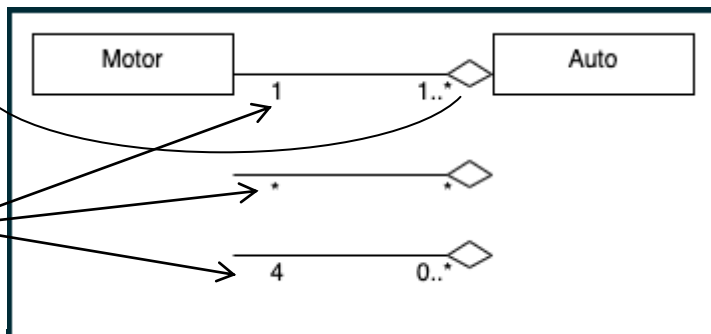
- Trieda Person
- Trieda Osoba má inštančné parametre name a surname ktoré sú privat typu string
- Trieda Person má konštruktor Person ktorý nič nevracia (None) s atribútmi

- Trieda Manager je dcerou triedy Person a z nej dedí čo ukazuje smer šípky

Agregácia – spojenie medzi triedami. Používa sa vtedy ak dané objekty vedia existovať aj bez seba resp. ak jeden objekt obsahuje druhý

Motor vie existovať aj bez auta a auto vie existovať bez motora.

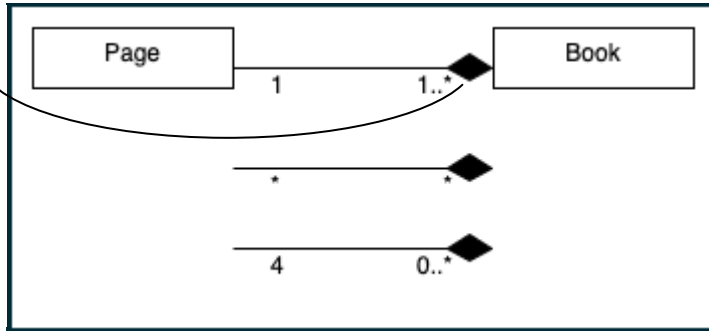
Čísla vyjadrujú početnosti



- V jednom alebo vo viacerých autách sa nachádza jeden Motor **alebo (..)** viacero Aut má práve jeden Motor. Vzťah tried nie je podmienený a tak existuje aj spojenie pri nereálnych vzťahoch (viac aut ma 1 a ten istý motor)
- Viacero Motorov sa môže nachádzať vo viacero Autach
- Práve 4 Motory sa nachádzajú v nula(0) autách alebo vo viacerých autách t.j. aj bez Motora môže **existovať** Auto a chceme tento prípad zohľadniť

Composition – skladanie s existenčnou závislosťou t.j. používa sa vtedy ak objekty nevedia (nemôžu) existovať bez seba

Kniha nevie existovať bez strán resp. každá kniha musí mať stránky. Ináč to není kniha ale napr. obal.



- Kniha nemôže existovať bez jednej strany resp. bez Book-u **neexistuje** žiadna Page t.j. kniha bez strán nemá zmysel alebo (..) Viacero Book-ov má práve jednu Page
- Viacero Book má viacero Page resp. viacero strán má viacero kníh
- Práve 4 stránky sa nachádzajú v nula (0) alebo viacerých knihách

* - označuje viacero

Príklad použitia: je vytvorený v SW www.umletino.com

chýba tu konštruktor s 2 parametrami ktoré sú uvedené nižšie

premenné triedy resp. parametre konštruktora

metóda ktorá môže byť static alebo class (ak iba metódu použijem a nechcem s ňou pracovať tak ju použijeme ako static)

modul autá

