Diagram tříd (Class diagram)

Třída – popisuje skupinu objektů, které mají:

- stejné stavy
- stejné chování

Stav objektu: je určen hodnotami uloženými v objektu.

Chování objektu: je určeno funkcemi, které může objekt vykonávat.

Třída popisuje strukturu objektů. Objekty odvozené z třídy jsou instancemi třídy

Příklad.

Mějme veřejnou knihovnu a uvažujme objekt kniha.

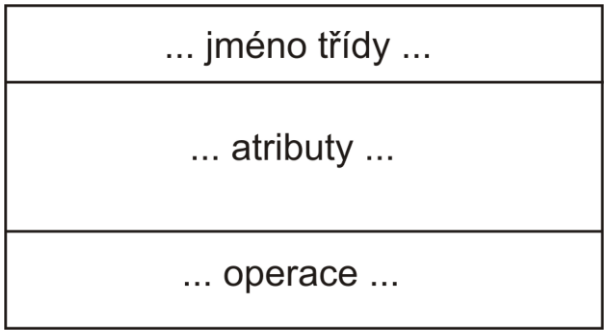
Stav:

název knihy
vydavatel
evidenční číslo
zda je půjčená
zda je poškozená

Chování:

zapůjčení knihy
vrácení knihy
zaznamenání poškození

Chování se vztahuje k objektu, kterého se týká, nikoliv, kdo ho provede (zde knihovník).



Třída má části:

- Jméno třídy
- Atributy – datové členy (členské proměnné) třídy
- Operace – členské funkce třídy

Metoda je termín označující implementaci operace.

Jméno třídy

- začíná velkým písmenem
- slova jsou psána za sebou bez mezer
- každé slovo ve jméně začíná velkým písmenem
- jméno je psáno tučným písmem a je umístěno uprostřed

Příklad.

Kniha
BeznyUcet

Atributy

viditelnost / jméno : typ násobnost = implicitní hodnota { vlastnosti }

Jméno – je jediná povinná část atributu
začíná malým písmenem
další slova začínají velkým písmenem

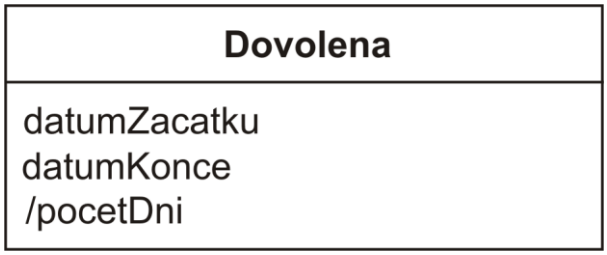
Příklad.

datumPujceni

Viditelnost

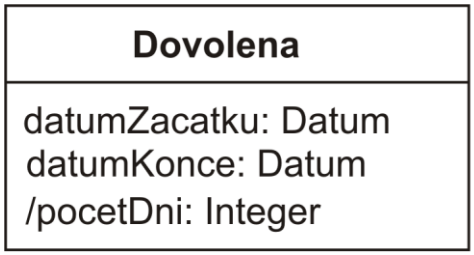
+ veřejný (public) přístupný uvnitř i vně třídy
~ v balíčku (package) přístupný jen v balíčku, kde je třída obsažena
# chráněný (protected) přístupný i pro dědící třídu
- soukromý (private) přístupný jen uvnitř třídy

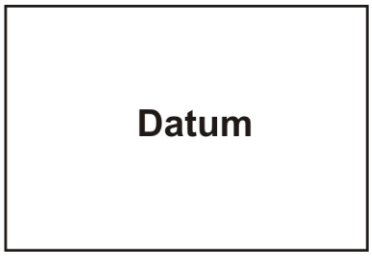
/ - označuje odvozený atribut (atribut, jehož hodnotu lze vypočítat z jiných atributů)



Typ atributu

- Boolean false, true
- Integer -200
- Real 2.718
- String ″UML″
- datové typy jazyka použitého pro implementaci (C++, C#,…) – char, int, …
- vlastní typy – typy vytvořených tříd





Násobnost

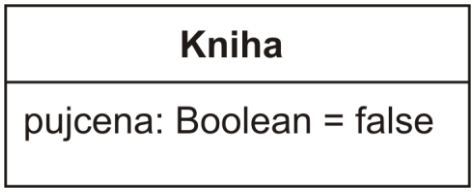
[dolní_hranice … horní_hranice]
[počet]

* - označuje neomezenou horní hranici

Příklad.

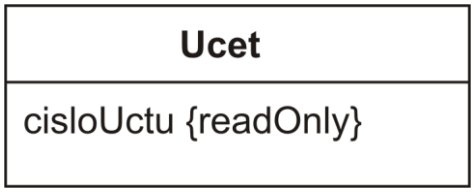
[1..10]
[0..*]
[4]
[*] stejné jako [0..*]
[1] - je implicitní násobnost

Implicitní hodnota



Vlastnosti

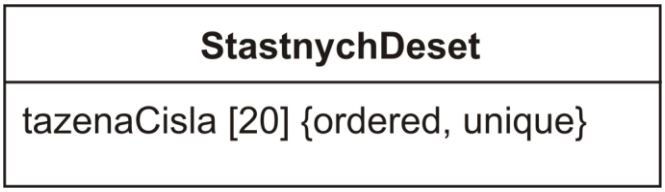
readOnly – jakmile je atributu přiřazena hodnota, nelze ji už změnit



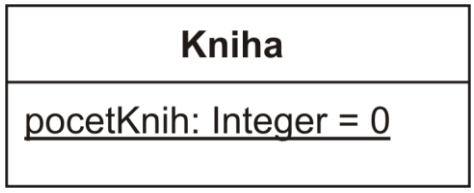
U atributů s násobností

ordered – prvky atributu jsou uspořádané dle velikosti (není-li tato vlastnost uvedena, prvky jsou implicitně neuspořádané - unordered)

unique – prvky atributu jsou vzájemně různé (není-li tato vlastnost uvedena, prvky mohou být stejné - nonunique)



Statický atribut je podtržený.



Operace

viditelnost jméno (…parametry…) : typ_návratové_hodnoty { vlastnosti }

jméno – je jediná povinná část operace, začíná malým písmenem

Parametr

směr jméno : typ násobnost = implicitní_hodnota

jméno – je jediná povinná část parametru, začíná malým písmenem

Směr

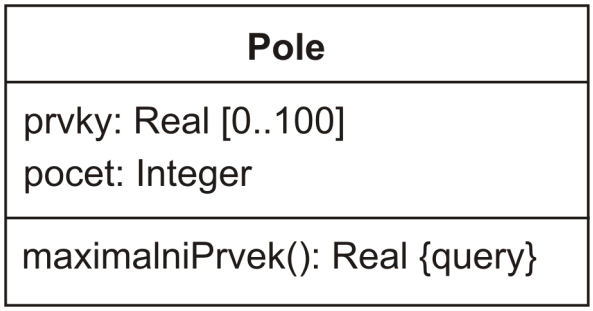
in – vstupní parametr
inout – vstupní i výstupní parametr
out – výstupní parametr

Typ návratové hodnoty

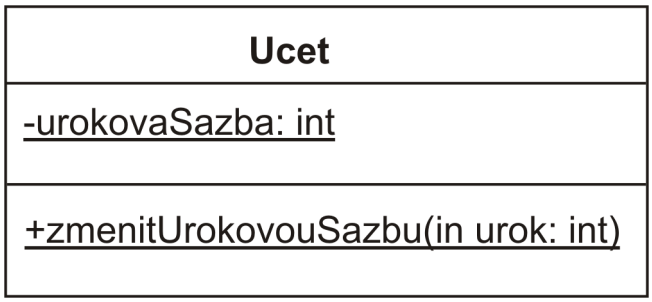
datový_typ násobnost (nepovinné)

Vlastnosti

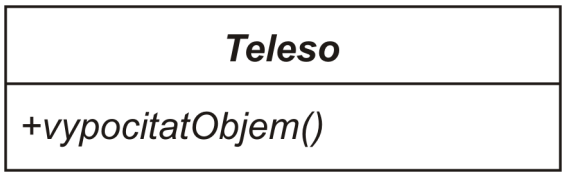
query – dotazovací operace, nemění stav objektu (nemění hodnotu žádného atributu)



Statické operace jsou podtržené.



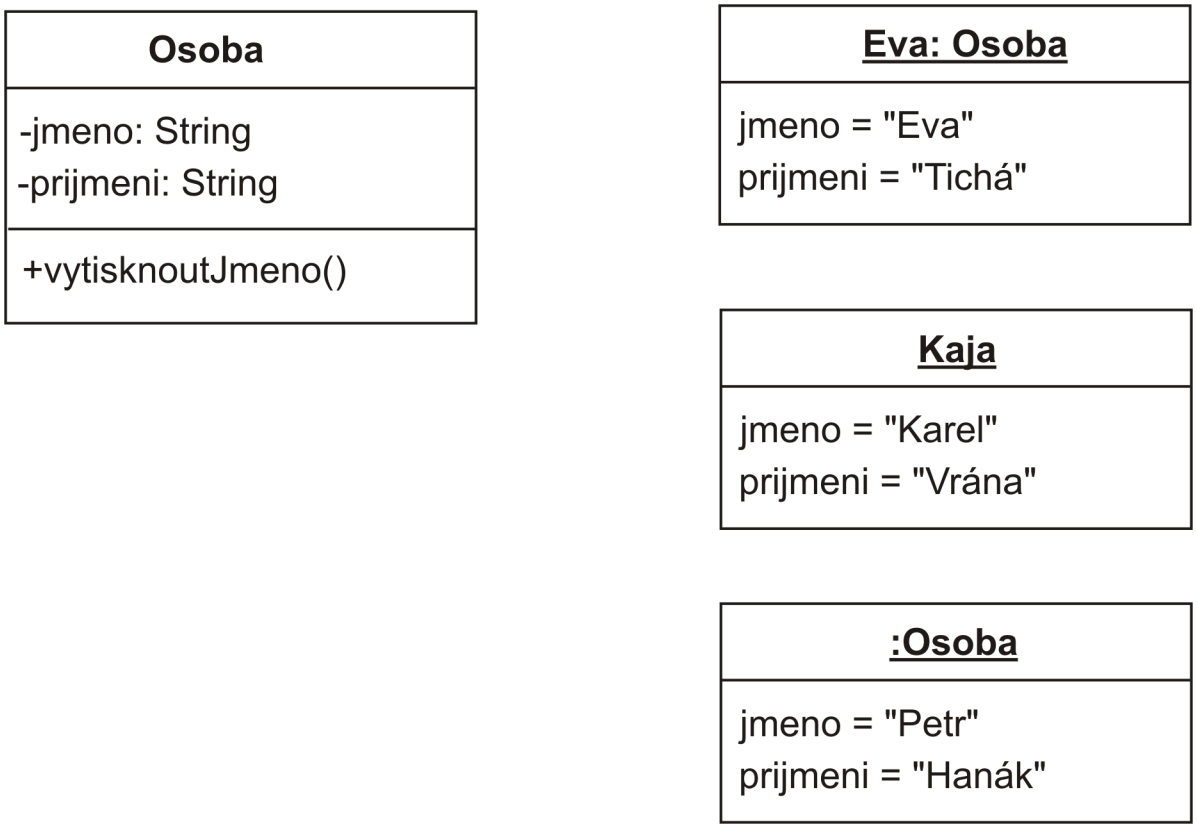
Abstraktní operace se píše kurzívou. Jméno abstraktní třídy se píše kurzívou.



Objekty

- název objektu je podtržený
- Tři možné zápisy názvu objektu:
  jméno_objektu : jméno_třídy
  jméno_objektu
  : jméno_třídy

Atributy se zpravidla uvádí s hodnotami.
Operace se většinou neuvádí.



Stereotypy

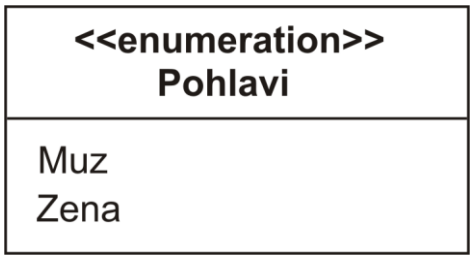
Označují určité variace existujících prvků modelovacího jazyka. Ukazují, že určitý prvek má specifické vlastnosti nebo specifické užití nebo specifický účel. Pro zápis se používají závorky << >> (nazývané zkosené závorky – guillemets).

Jeden prvek může mít více stereotypů.

Stereotypy mohou být

- standardní (součást UML)
- vlastní

Výčtový typ



Vztahy mezi třídami

Objekty si navzájem posílají zprávy (jeden objekt volá operaci jiného objektu). Zasílání zpráv mezí objekty je modelováno vztahy mezi třídami.

Asociace

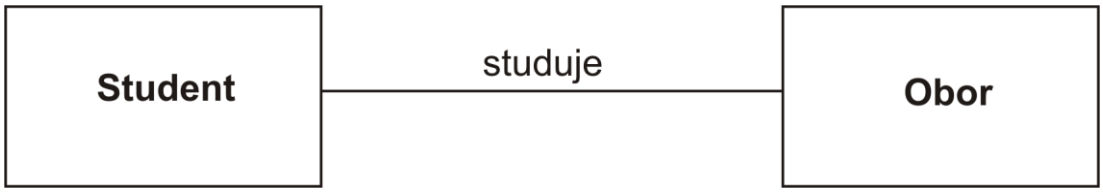
Je základní, jednoduchý vztah mezí třídami.
Asociace jsou

binární – vztah mezi dvěma třídami

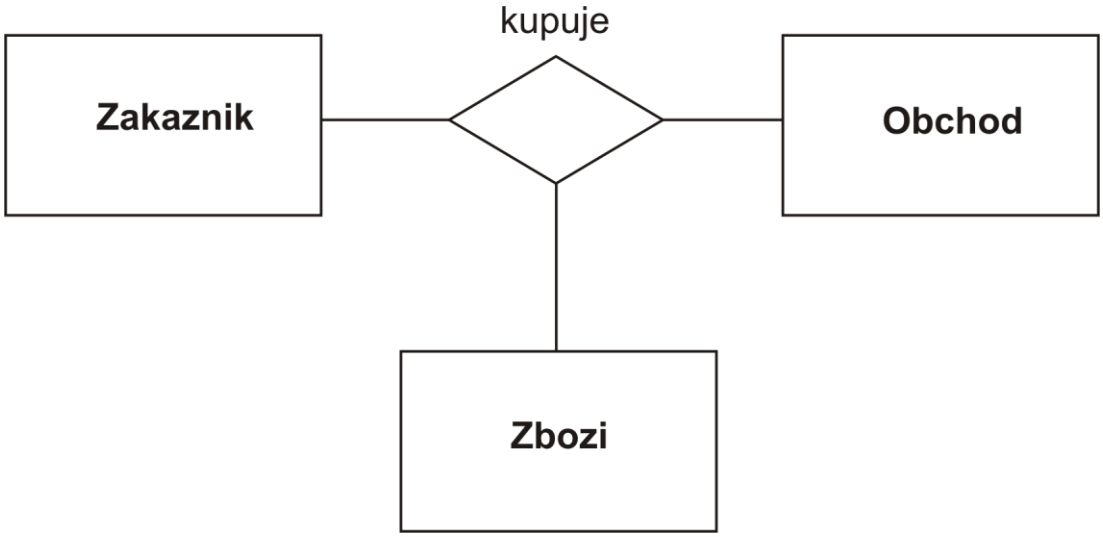
n-ární – vztah mezi n třídami

Binární asociace jsou nejčastějším druhem asociací.

Asociaci kreslíme jako čáru spojující dané třídy.
Asociace může mít pojmenování (zpravidla sloveso nebo slovesná fráze), které vysvětluje, co daný vztah reprezentuje.

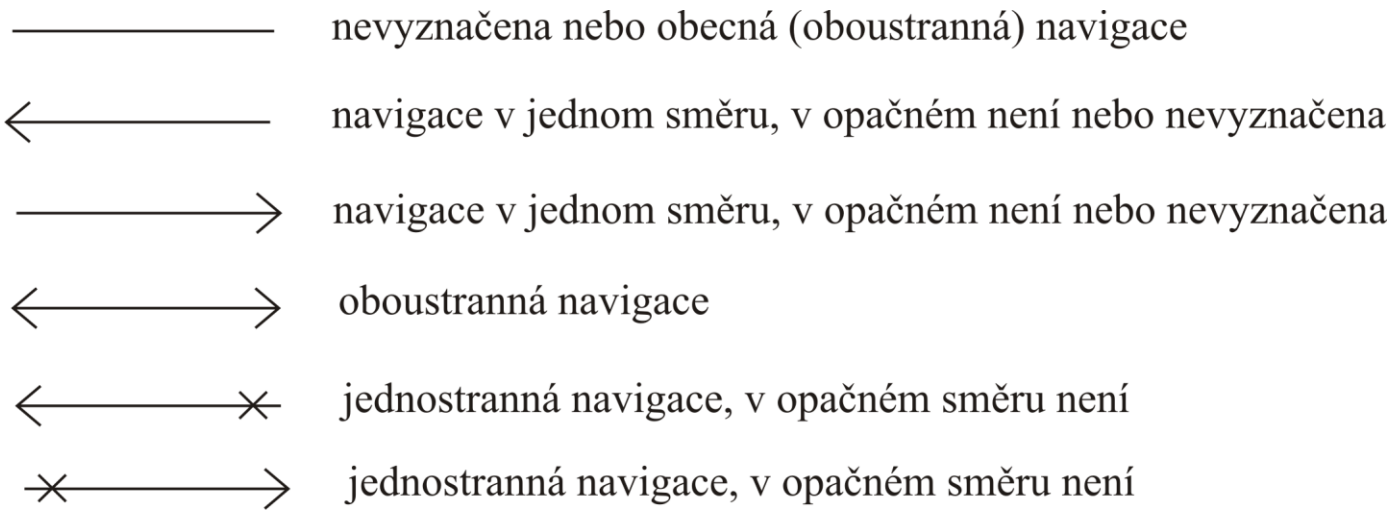


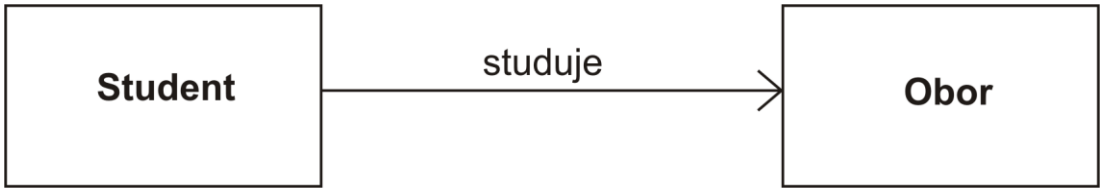
Asociace, které jsou vztahem mezi více třídami, se kreslí se symbolem diamantu uprostřed.

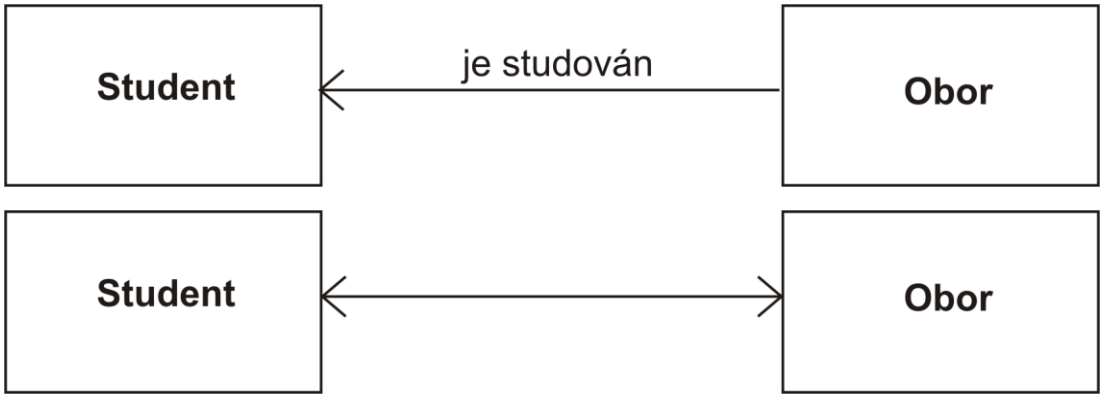


Navigace

Navigace ve vztazích mezi třídami ukazuje, jak jsou objekty dostupné jeden z druhého.



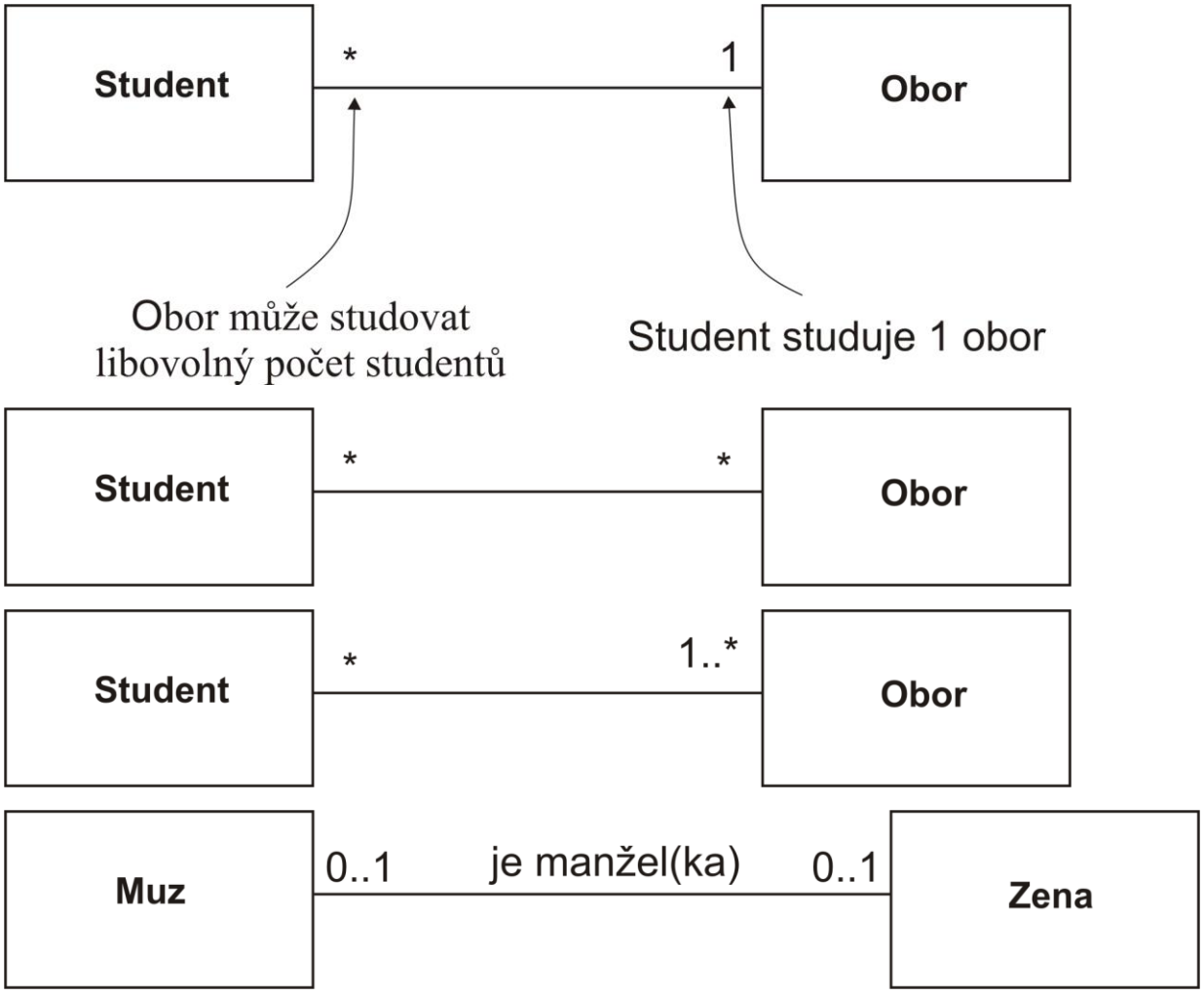




Násobnost

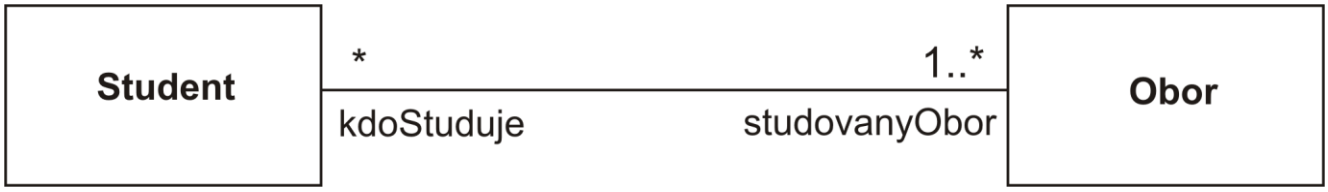
Násobnost ve vztazích mezi třídami vyznačuje, s kolika objekty druhé třídy může být daný objekt propojen. Způsob označení je stejný jako u atributů, někdy se dále používá výčet hodnot – 2,3,5 apod.

Násobnost 1 je implicitní (nemusí být uvedena).

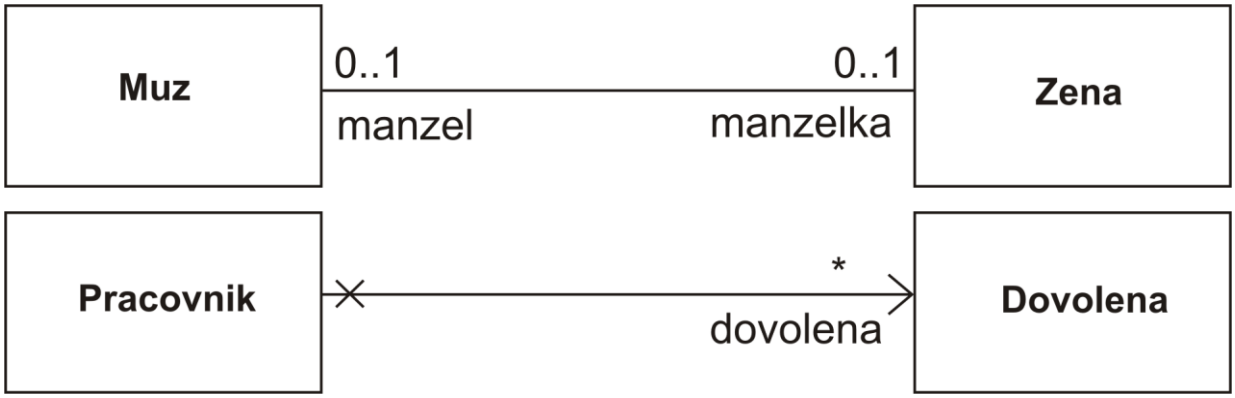


Role

Jsou umístěny na koncích asociací a vysvětlují, jakým způsobem je daný objekt účasten v daném vztahu (jakou v něm má roli). Role často jsou následně atributy příslušných tříd a pro jejich jména se pak používá stejný způsob zápisu jména jako pro atributy.

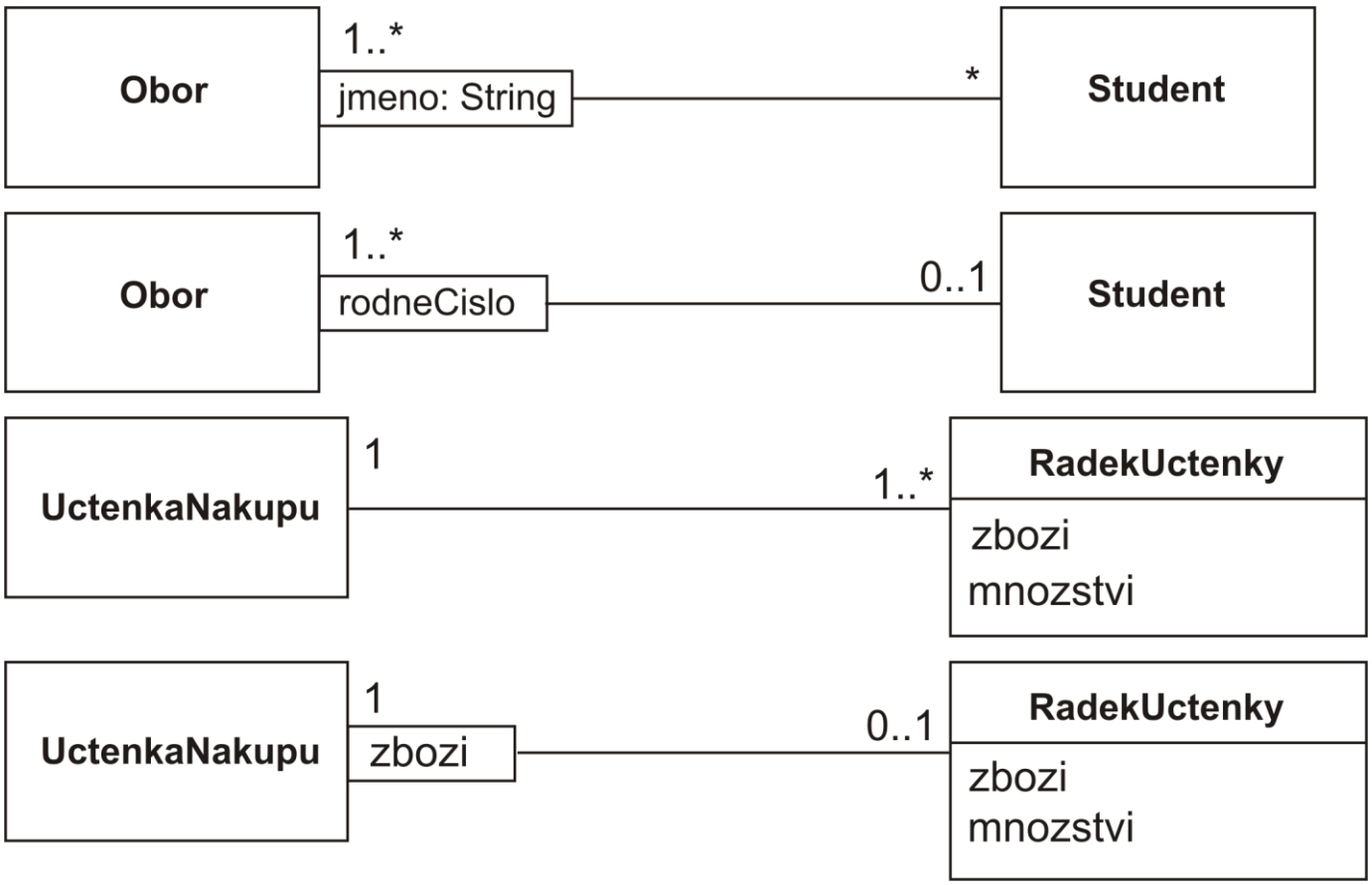


U vyšších násobností se používá i množné číslo – *studovaneObory*.



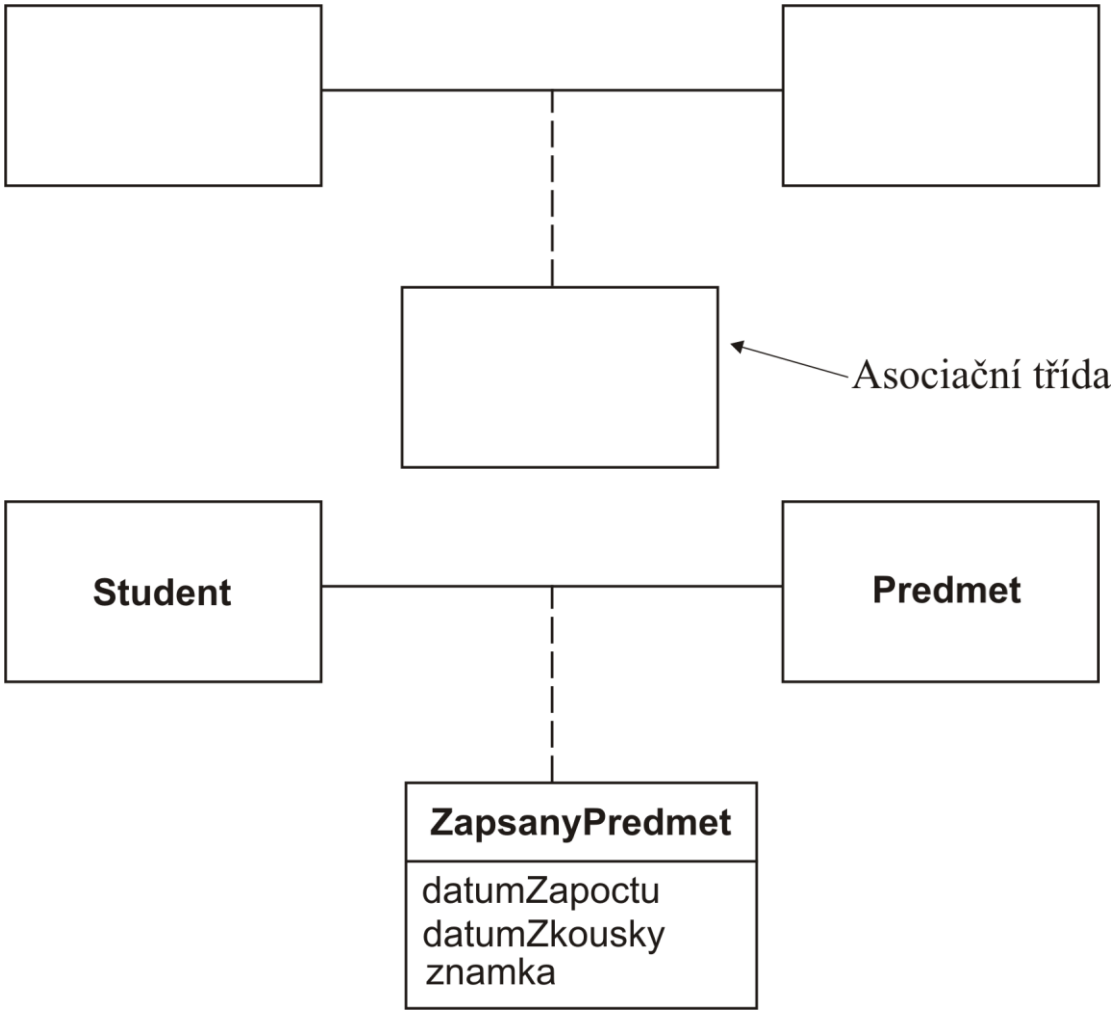
Kvalifikovaná asociace

Modeluje koncept známý jako asociativní pole, mapování. Dá se přirovnat ke klíči, který určuje hledaný objekt nebo objekty. Je to zpravidla atribut (nebo více atributů) cílové třídy, jehož hodnota nebo hodnoty) určují příslušný objekt nebo objekty.

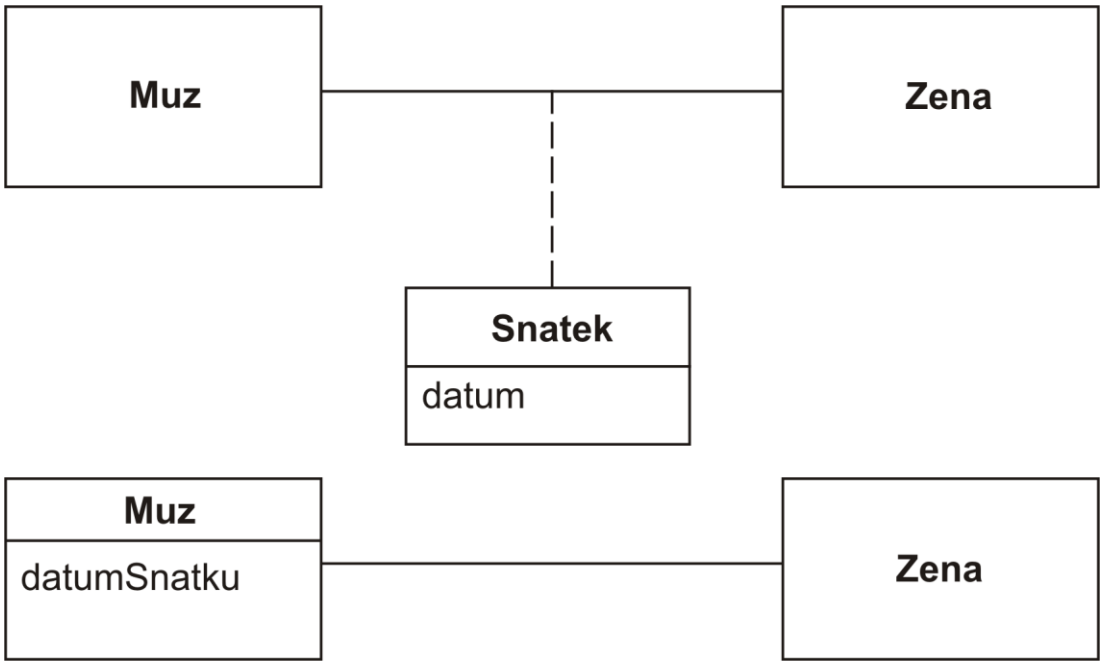


Asociační třída

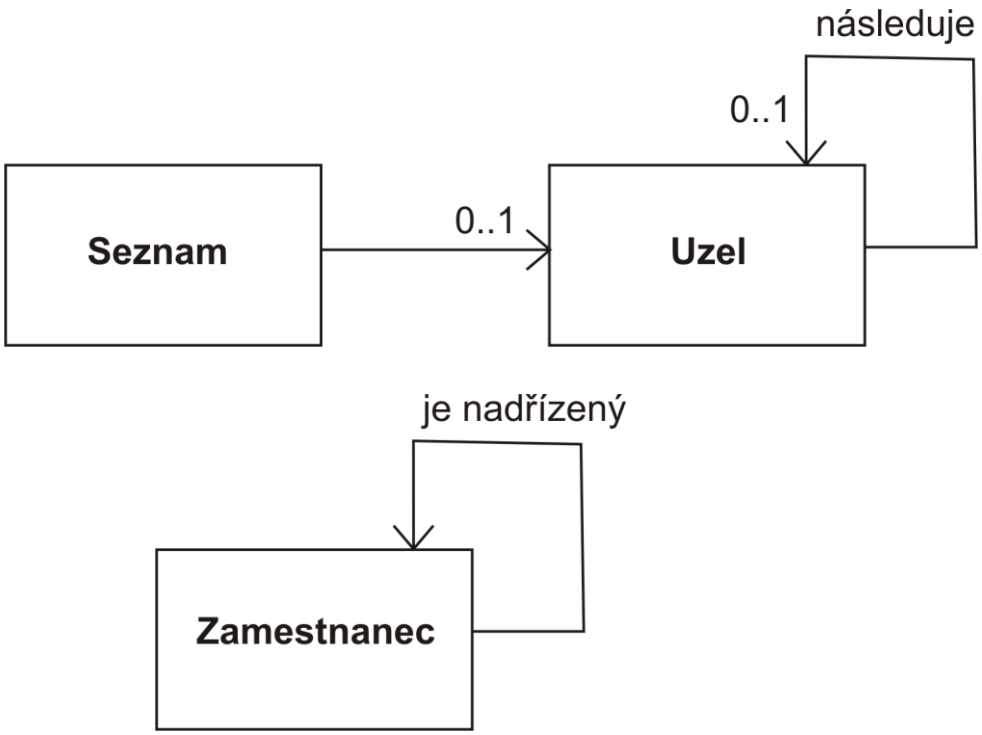
Asociační třída umožňuje podrobněji modelovat asociace. Lze v ní uvést atributy nebo operace, které se asociace týkají. Je vhodná pro složitější případy asociace a používá se zejména u asociací s vyšší násobností (1:*, *:*).



Pokud násobnost asociace není větší než 1, lze atributy a operace spojené s asociací řešit začleněním do třídy, která je na konci asociace.

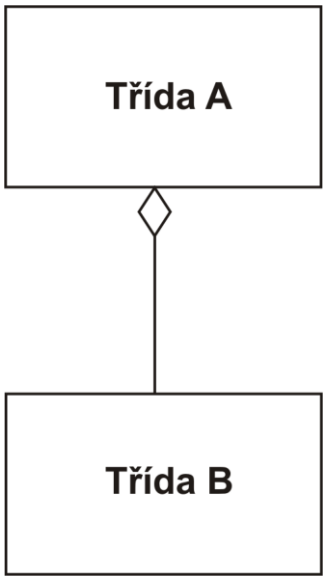


Asociace může být i se stejnou třídou.

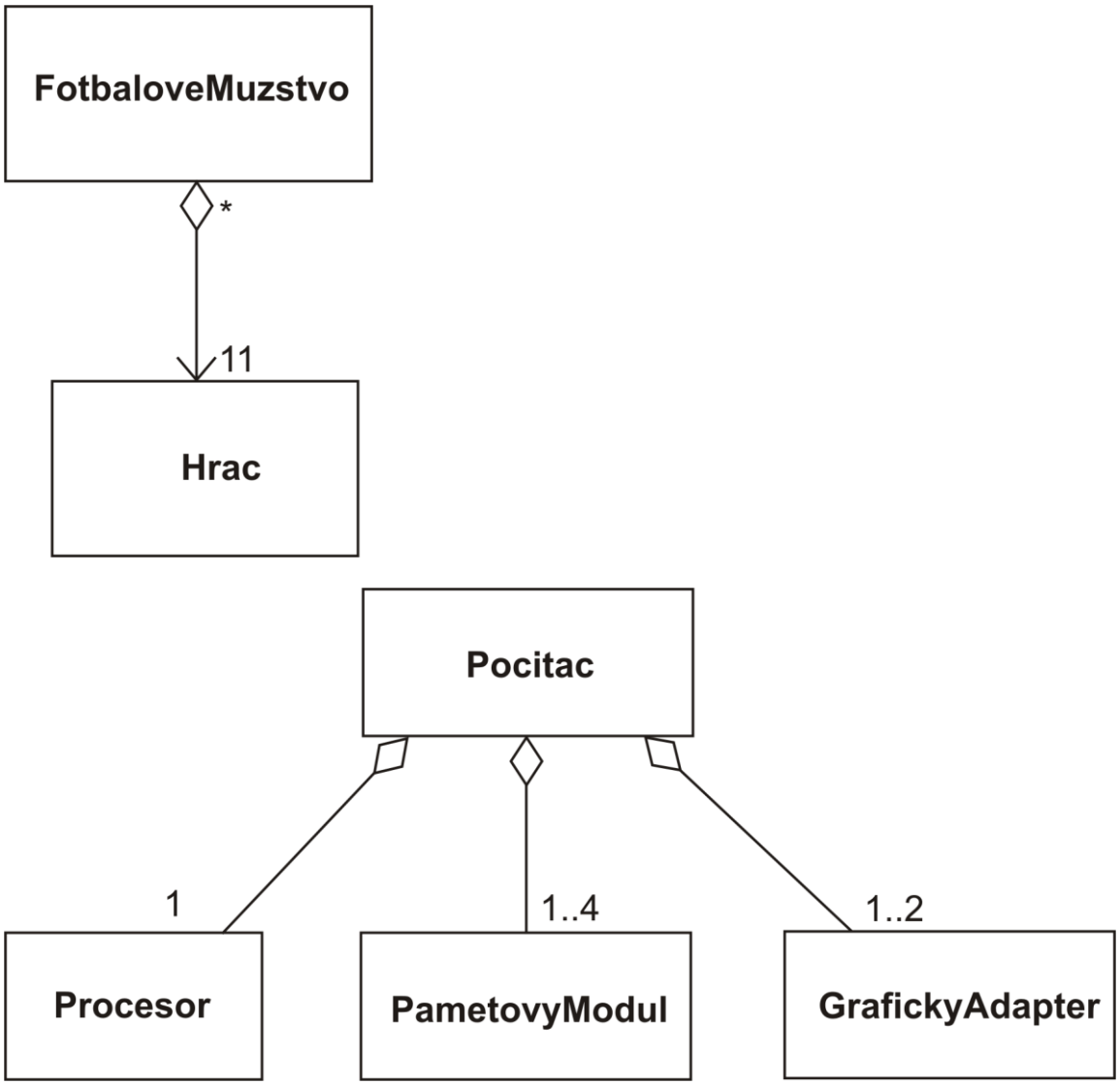


Agregace

Agregace je silnější typ asociace. Je vztahem mezi celkem a jeho částmi. Jedna nebo více tříd v agregaci je částí celku.

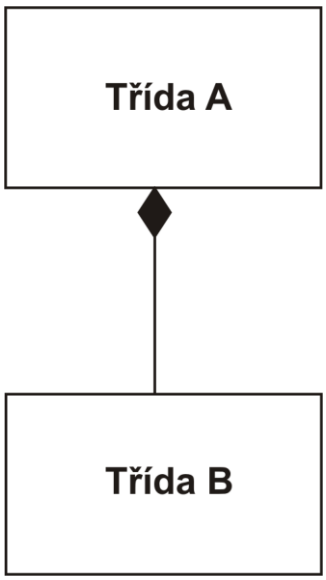


Třída *B* je částí celku, který je reprezentován třídou *A*.

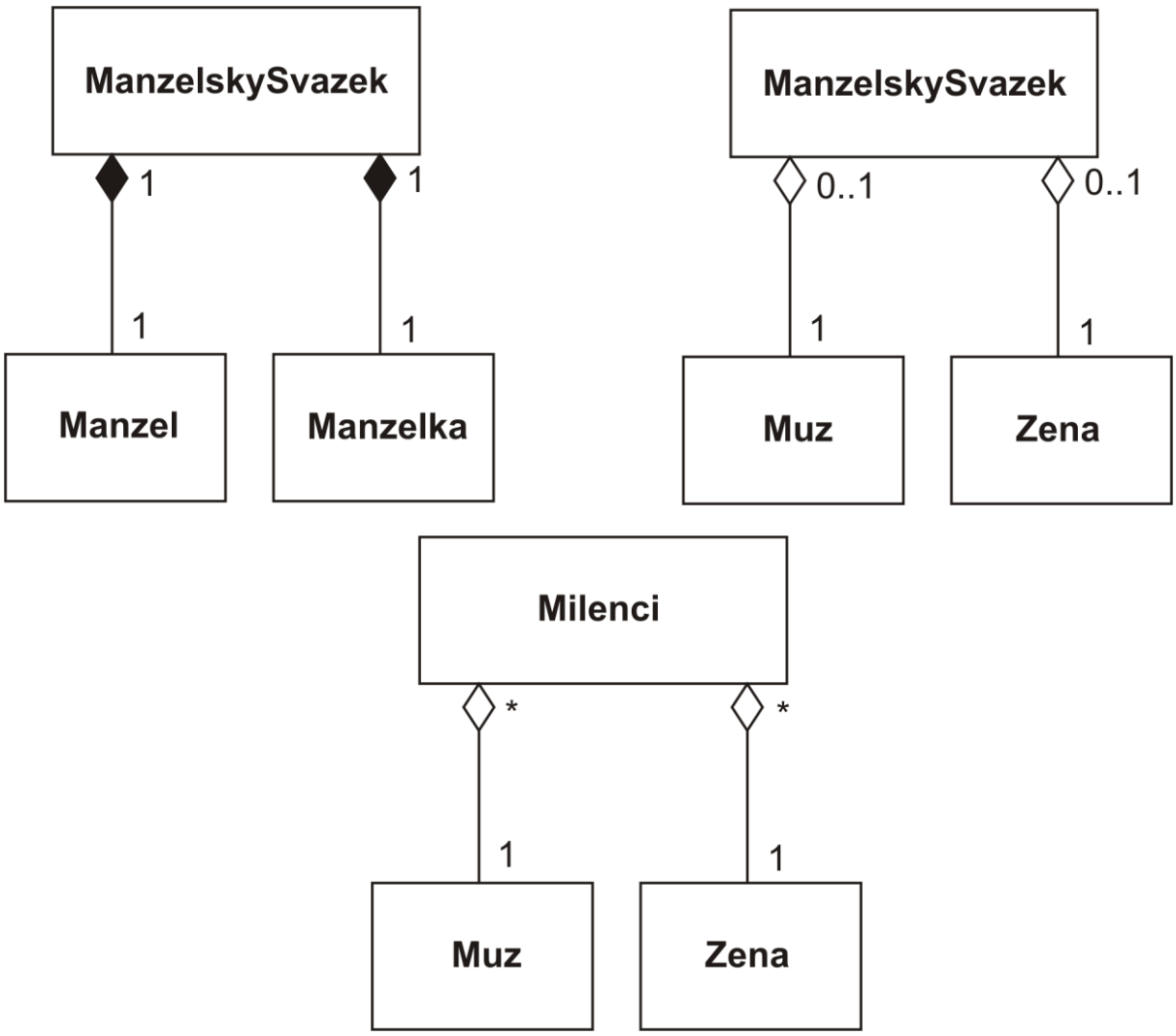


Kompozice

Kompozice je rovněž vztah mezi celkem a jeho částmi. Části v kompozici vznikají se vznikem kompozice a zanikají, když vztah kompozice zanikne.

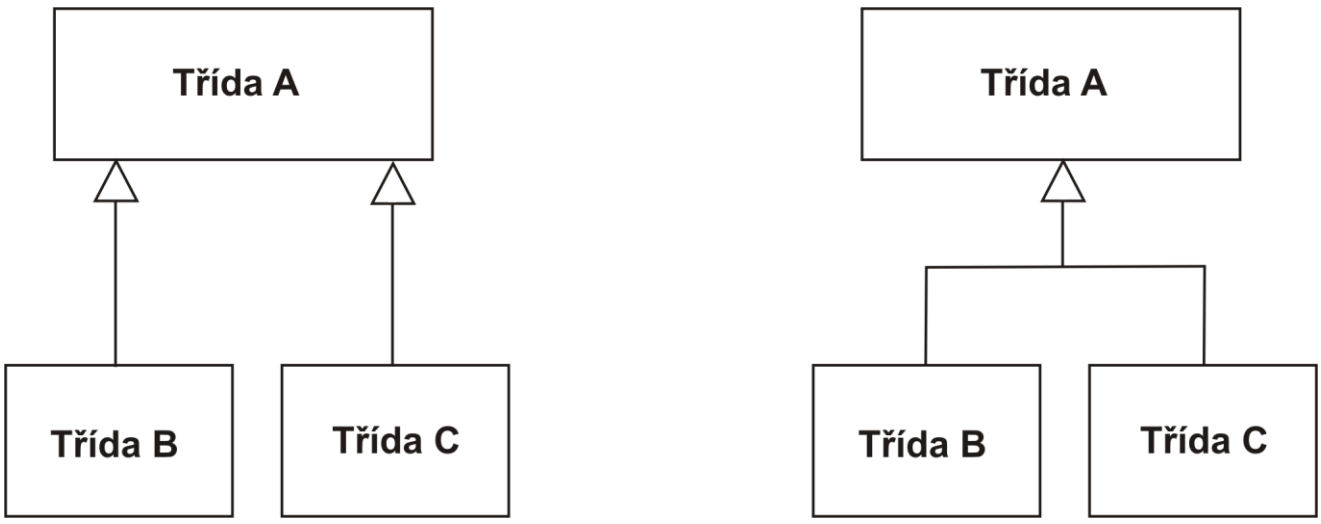


Po vzniku objektu třídy *A* vzniká objekt třídy *B*. Se zánikem objektu třídy *A* zaniká i objekt třídy *B*.



Dědičnost

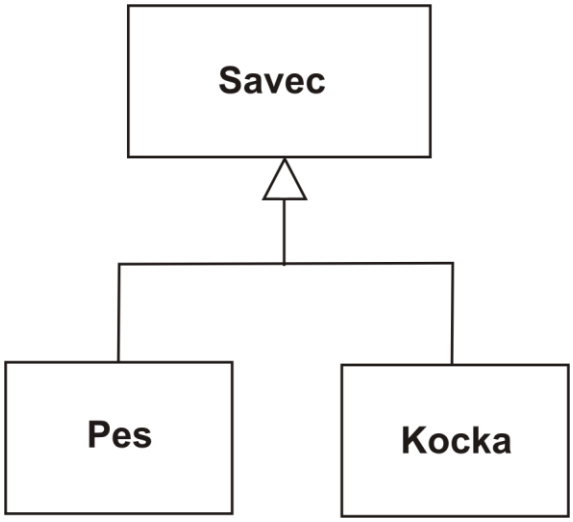
Dědičnost je vztah, kdy děděním existující třídy vzniká nová třída nebo třídy.



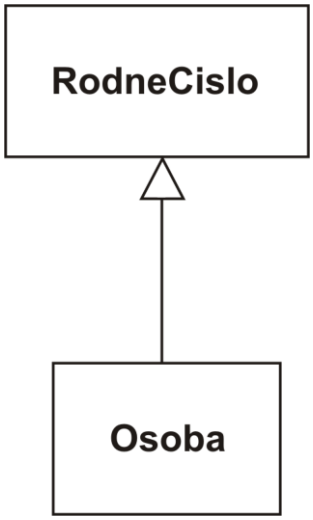
Děděná třída *A* je zobecněním (abstrakcí) tříd *B* a *C*, které ji dědí.

Třídy *B*, *C* jsou odvozené z třídy a. Jsou speciálním případem třídy *A*.

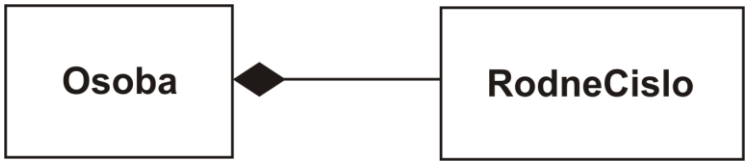
Dědičnost je „je“ (is a) vztah. Tedy mělo by platit, že lze říct „Třída *B* je třída *A*“.



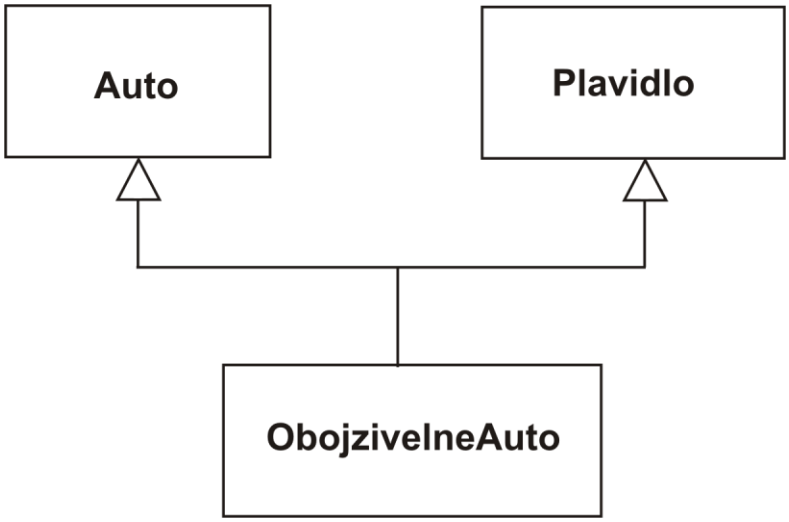
Lze říct (platí): „Pes je savec.“ „Kočka je savec.“



Nelze říct: „Osoba je rodné číslo.“ Zde je vhodnější použít kompozici.



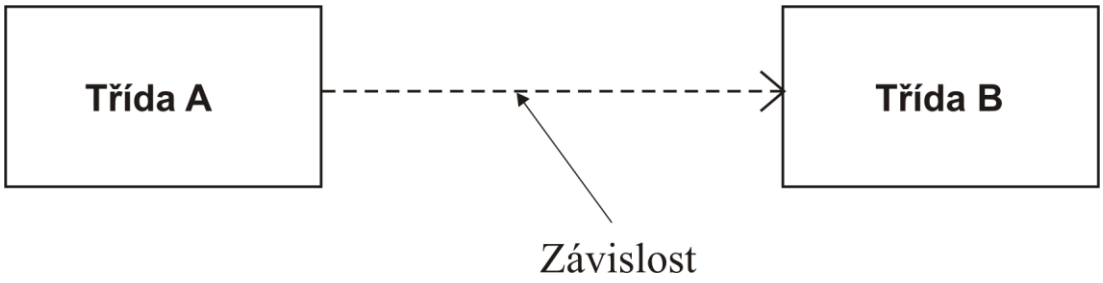
Můžeme dědit i více tříd.



Lze říct (platí): „Obojživelné auto je auto.“ „Obojživelné auto je plavidlo.“

Závislost

Je nejslabší vztah mezi třídami.



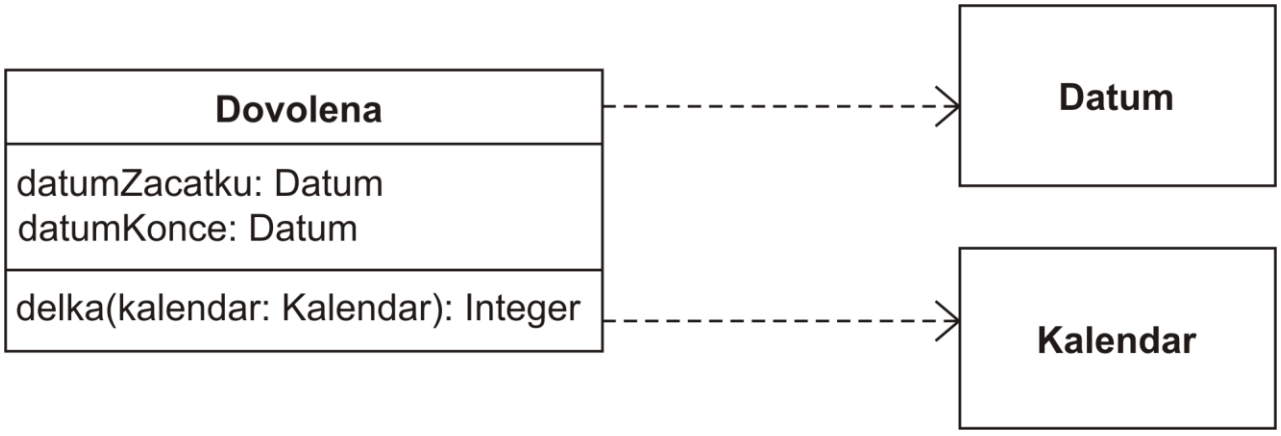
Třída *A* je závislá na třídě *B*.

Závislost je případ vztahu mezi třídami, kdy ve třídě *A* je nějakým způsobem použita nebo využita třída *B*, její atributy, operace.

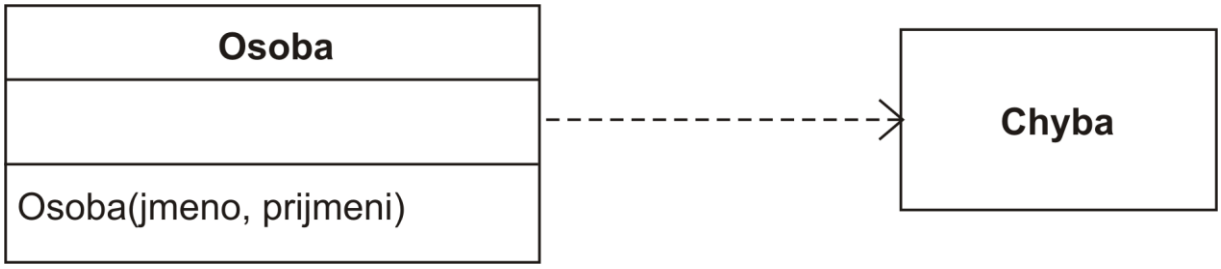
Závislost nastává například v případech:

- Typem atributu ve třídě *A* je třída *B*.

- Třída *B* je typem parametru operace ve třídě *A*.



- Třída *B* se používá pro výjimky. Například nelze vytvořit objekt třídy *A*, protože v konstruktoru objektu jsou uvedeny chybné hodnoty jeho parametrů. Konstruktor vytvářeného objektu generuje objekt třídy *A* pro výjimku.



Šablona (template)