

SOFTVÉROVÉ MODELOVANIE

2.

Ján Ružbarský
Marek Tavač

Obsah 2. prednášky

- *Opakovanie*
- *UML a teória objektov*
- *Diagram tried a diagram objektov*
- *Vaše otázky*

Opakovanie - Pojmy

- **System** – kombinácia softvéru a hardvéru, umožňujúca riešenie konkrétneho problému
- **Vývoj systému** – vytváranie systému pre klienta, osobu, ktorá má nejaký problém
- **Reuse** – znovupoužiteľnosť
- **Modelovanie** - návrh softvérovej aplikácie pred písaním kódu

Opakovanie - Softvérový model

- ***model** = zjednodušenie reality = abstrakcia reálneho systému*
 - ukrytie alebo zobrazenie detailov
 - možnosť zamerať sa na rôzne aspekty projektu
- *súhrn rôznych pohľadov na systém na určitej úrovni abstrakcie*
- *zvyšuje pravdepodobnosť úspešného ukončenia*
- *znižuje náklady na vývoj*
- *redukuje čas riešenia problému*

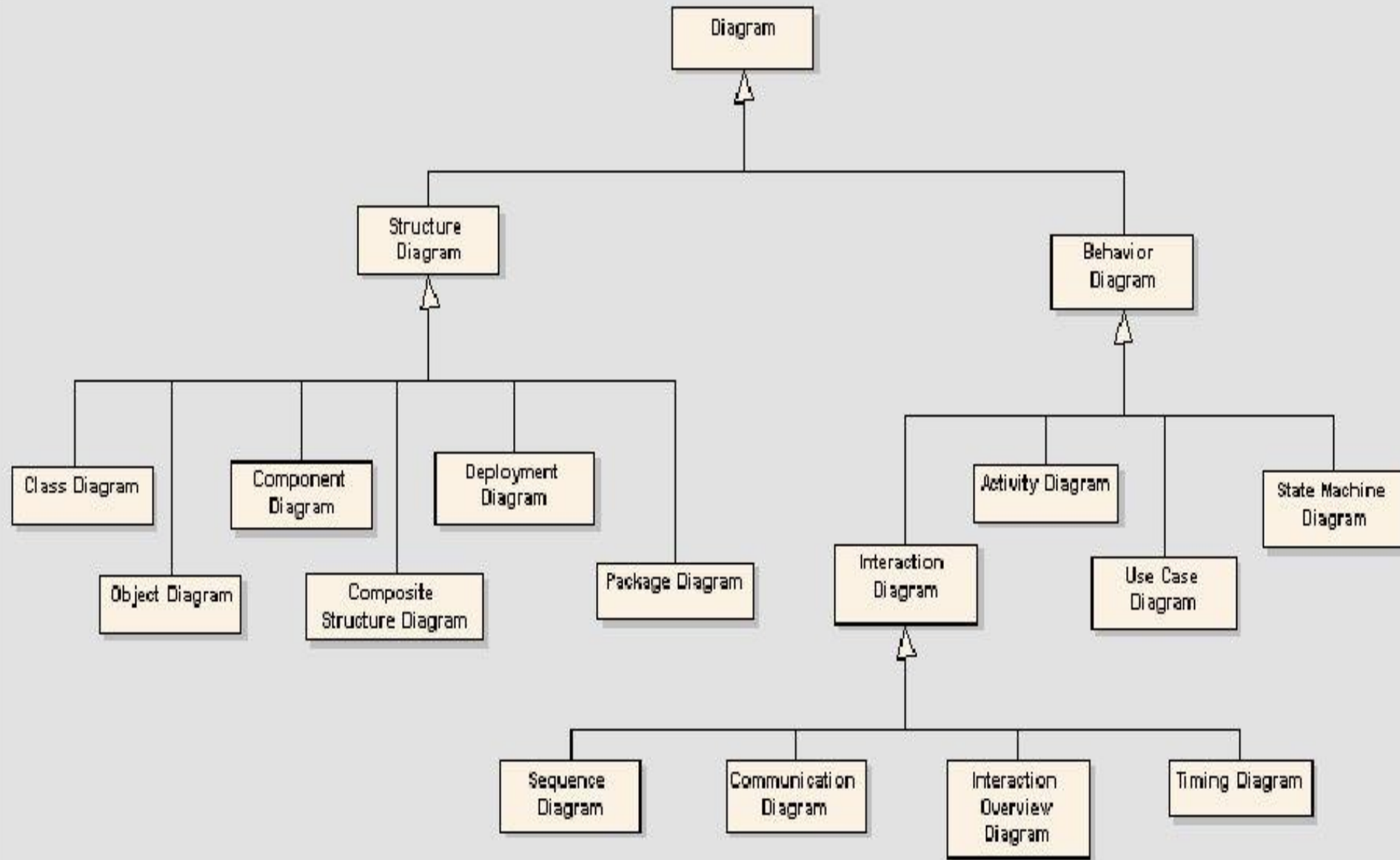
Opakovanie - U M L

- *Módny trend*
 - *Štandard – de facto, de jure (OMG 2.4.1)*
1. *jazyk určený pre **špecifikovanie, vizualizáciu, konštrukciu a dokumentovanie** súčastí **softvérových** systémov ako aj pre biznis modelovanie alebo modelovanie iných **nesoftvérových** systémov*
 2. *je vizuálny jazyk na modelovanie a komunikáciu o systéme pomocou diagramov a podporného textu*

Opakovanie - UML

- *Čo má systém urobiť, ale nič nehovorí o tom, ako sa má systém implementovať*
- *Diagramy (modely)*
 - Každý projekt iba tie, ktoré potrebuje
 - Závisí to od úrovne abstrakcie popisovaného systému
- *Všetky fázy vývoja systému*
- *Nestanovuje žiadny osobitný prístup, ale natívne podporuje OOP*

Opakovanie - Štruktúra UML diagramov

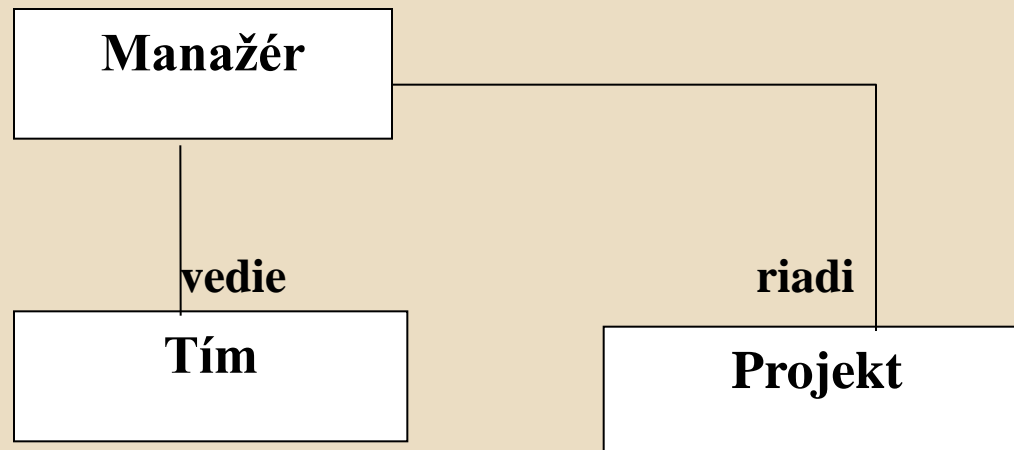


Stručne o objektoch

- **Používané pojmy:**
 - triedy, objekty,
 - asociácie, väzby
 - skladanie, dedičnosť, polymorfizmus, zapúzdrenie
 - správy, abstrakcia
 - typovo – inštančná dichotómia

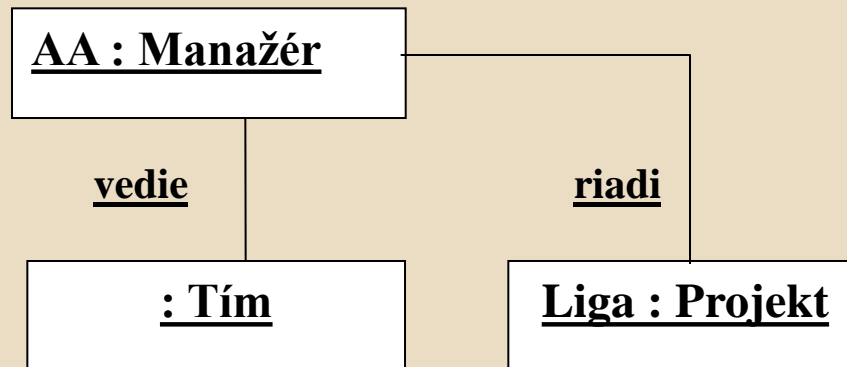
Trieda

- *Trieda*
 - základný pojem
 - definuje typ objektu, jeho charakteristiky
 - všeobecný pojem



Objekt

- *Objekt*
 - inštancia triedy
 - špecifický, konkrétny pojem

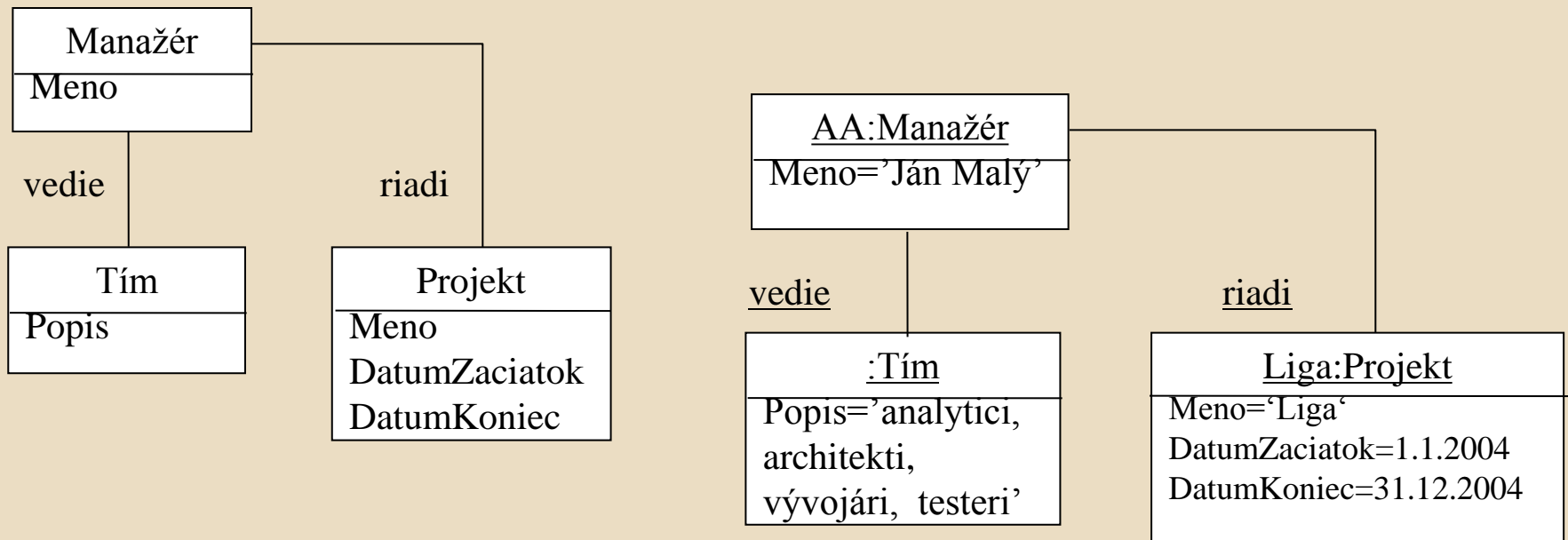


Typovo-inštančná dichotómia

- *Dichotómia – pohľad na nejakú vec z dvoch uhlov*
- *Konvencie používané v UML na popísanie vzťahov medzi*
 - *triedou jej inštanciami*
 - *asociáciou a jej inštanciami*
- *Konkrétna veta v UML je vždy založená na všeobecnej vete*

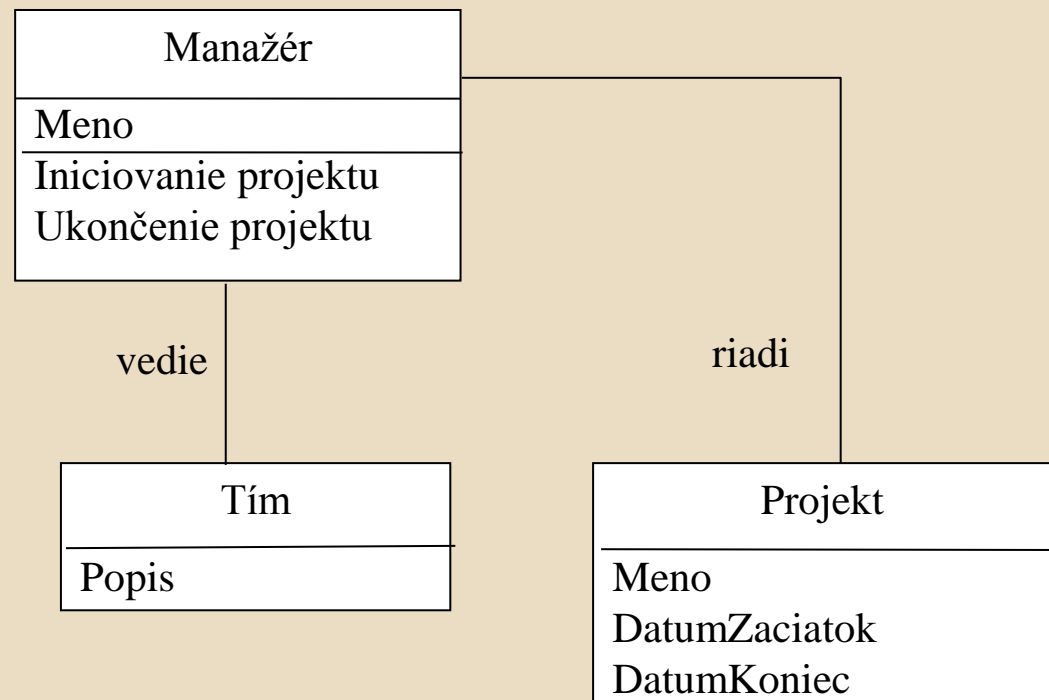
Atribúty

- *reprezentujú data*
- *to, čo objekt pozná, vie*
- *štruktúrálna črta (+asociácie)*



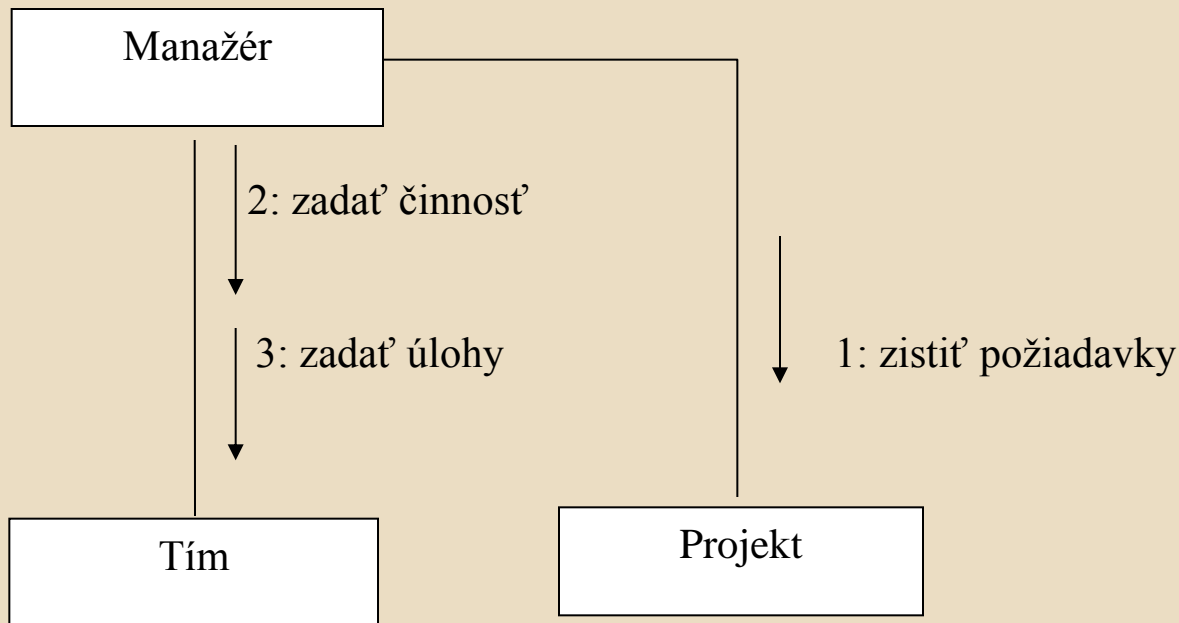
Operácie, metódy

- *operácia* – čo objekt môže robiť
- *metóda* – ako to urobí
- *trieda definuje operácia a metódy, ktoré aplikujú jej objekty*
- *črty chovania*



Správy, stimuly

- *správa* – komunikácia medzi triedami
- *stimul* – komunikácia medzi objektami
- Inštancia správy



Základné princípy OO

Abstrakcia

- pojmy a vzťahy
- všeobecná abstrakcia – trieda, asociácia
- konkrétna abstrakcia – objekt, väzba
- dobrá abstrakcia = filtrovanie

Základné princípy OO

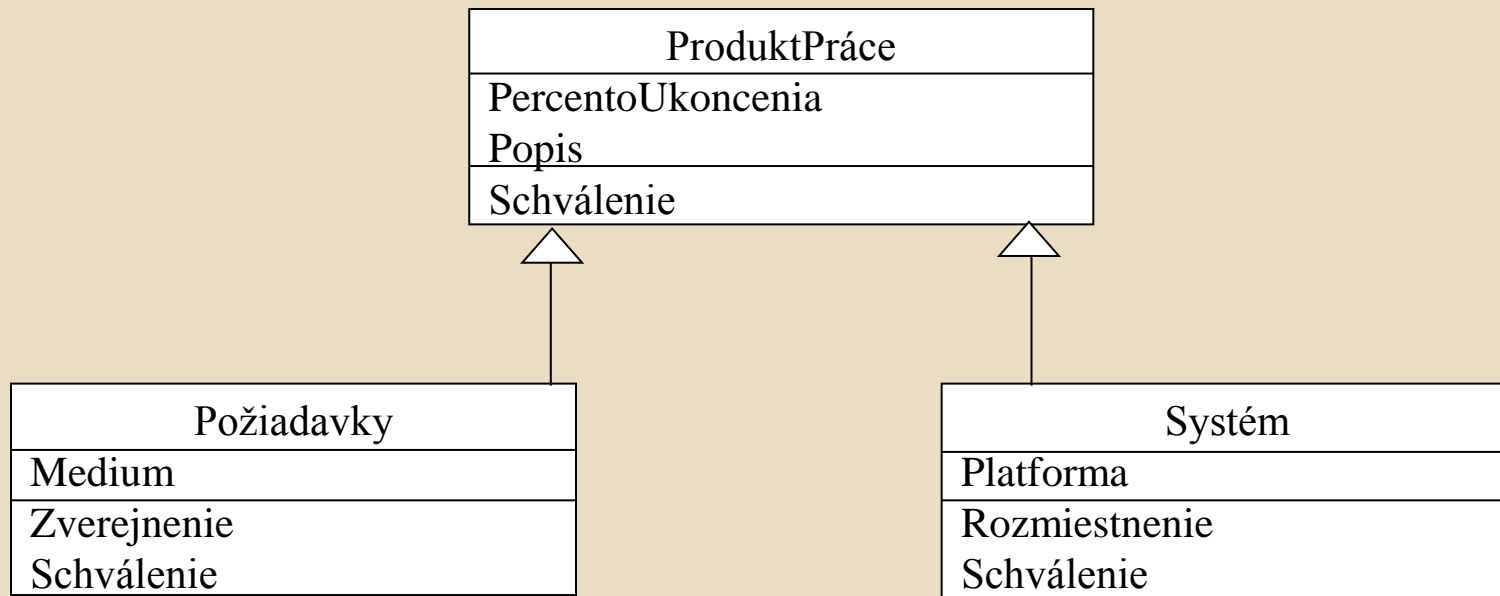
Zapúzdzrenie

- Zapúzdzrenie atribútov a operácií do tried
- Ukrytie metód za operácie – ukrývanie informácií
- Jednoduché celky
- Zmeny na jednom mieste
- Rozhrania

Základné princípy OO

Zovšeobecnenie – generalizácia

- vzťah dedičnosť



Základné princípy OO

Polymorfizmus

- rôzne triedy – rovnaké operácie, ale rôzne metódy
- trieda vie, ktorá metóda sa má vykonať
- schopnosť mať viac metód pre jednu operáciu
- dôležité aj pre analytika – terminológia
Např. otvoriť – dvere, konzervu, súbor, ...

Diagramy tried, diagramy objektov

- *popisujú štruktúru systému*
- *počas celého procesu vývoja*
- ***najprv:***
 - fáza analýzy a návrhu
- ***neskôr:***
 - paralelne s modelovaním spolupráce
 - zjemňovanie detailov elementov a ich vzťahov

Triedy v UML diagramoch

- *štrukturálne črty – vlastnosti, data*
 - Asociácie
 - Atribúty
- *črty správania - čo môžu robiť - operácie*
- *symbol v UML : štvoruholník*
môže mať 1 až 3 časti
 - Meno triedy
 - Atribúty
 - Operácie
- *redukcia triedy*
- *použitie stereotypov na označenie oddelených častí*

Atribúty

- *vlastnosti triedy reprezentujúce dáta*
- *trieda nemusí mať atribúty*
- *UML reprezentácia*
 - *druhá časť v symbole triedy*

Atribúty - syntax

- EmailAdresa [1..5unordered] : String = „ Žiadna adresa“

Pracovník
<ul style="list-style-type: none">-<u>DalsiID:integer=1</u>-ID:integer-Meno:String-EmailAdresa[1..5 unordered]:String=„Žiadna adresa“-Telefon[1..* ordered]:String#Heslo:String

Atribúty - syntax

viditeľnosť meno [násobnosť usporiadanie]:typ = inicializačná hodnota

viditeľnosť: + - #

násobnosť: 0..1, 1..5, *, 1

usporiadanie: ordered, unordered

typ: Integer, Real, Boolean, String, iná trieda

inicializačná hodnota - default nie je nastavená

Operácie

- *čo môže trieda vykonávať*
- *čo s ňou môže robiť iná trieda*
- *príklad: funkcie, procedúry*
 - deklarácia, definícia

+ pridatEmailAdresu (in nováEmailAdresa:String=“”) : Boolean

Operácie - syntax

➤ viditeľnosť meno_operácie (zoznam parametrov) : návratový typ

- viditeľnosť: + - #

- zoznam parametrov

druh meno : typ = defaultná_hodnota

druh: in, out, inout

- návratový typ: Integer, Real, Boolean, String

Príklad

Pracovník

+create():Pracovník

+getID():Integer

+getMeno():String

+setMeno(in Meno:String)

+getTelefon(in Priorita:Integer):String

+setTelefon(in Priorita:Integer, in Cislo:String)

+pridajEmailAdresu(in Adresa:String=""):Boolean

+odstranEmailAdresu(in Adresa:String)

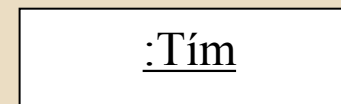
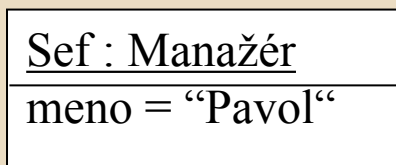
getEmailAdresa(out PrvaAdresa:String, out DruhaAdresa:String, out TretiaAdresa:String, out
StvrtaAdresa:String, out PiataAdresa : String)

#setHeslo(in Heslo:String)

+destroy()

Objekt

- *identita*
- *symbol v UML – max. 2 časti*
 - meno
 - atribúty
 - viditeľnosť meno[index]: typ = hodnota*
- operácie – nezobrazujú sa

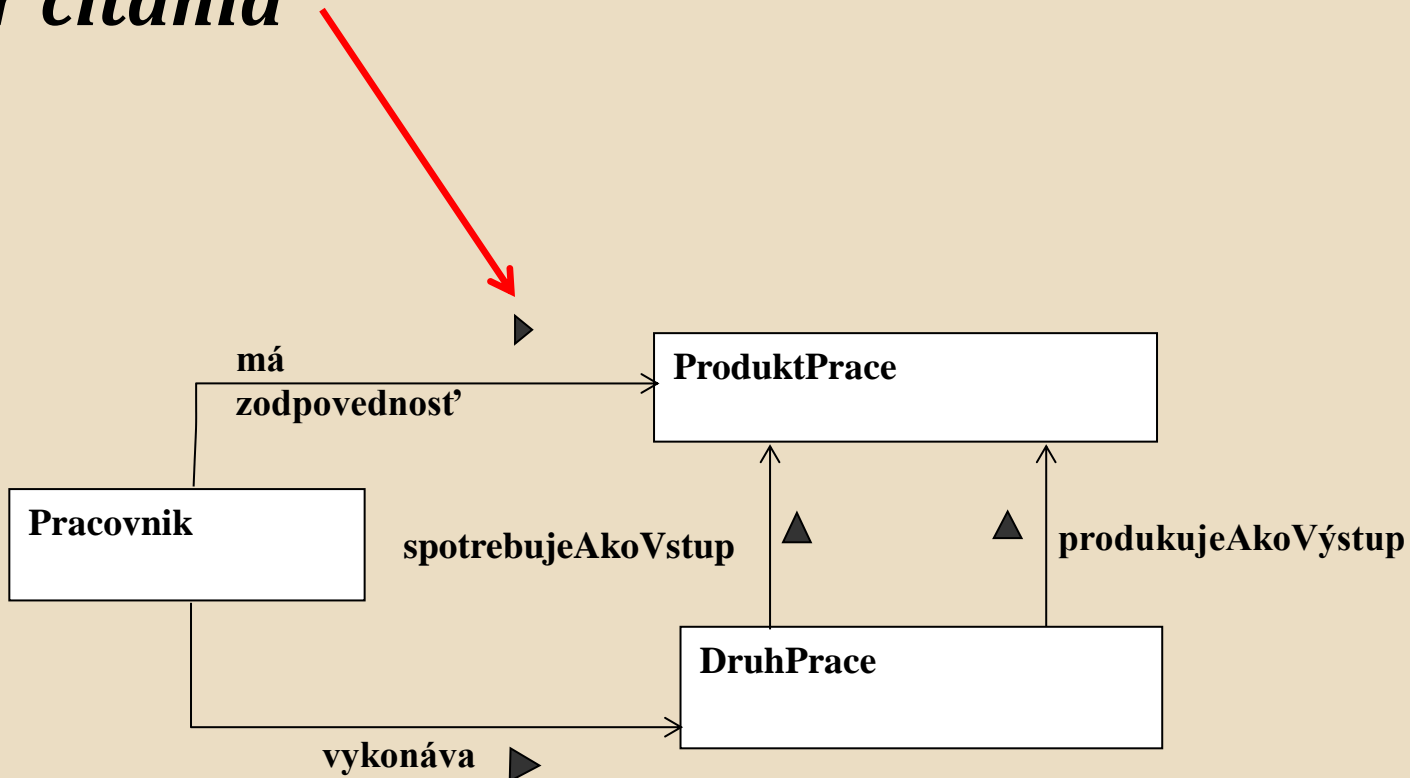


Asociácie a väzby

- *Základné vzťahy medzi triedami a objektami*
- **Asociácia**
 - **všeobecný** vzťah medzi triedami (v diagrame tried)
- **Väzba**
 - **konkrétny** vzťah medzi objektami (v diagrame objektov)

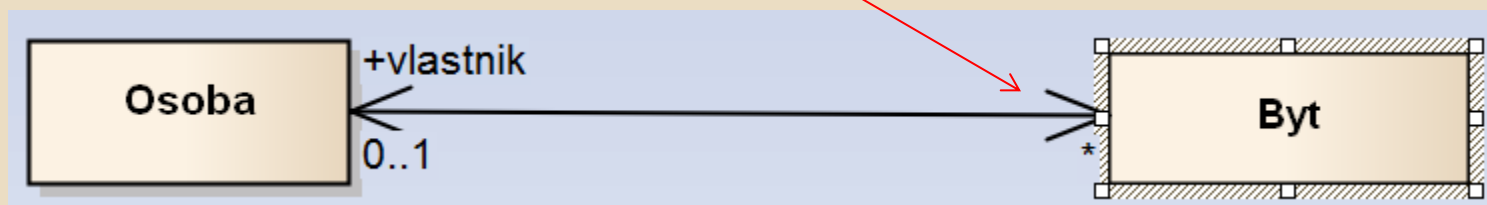
Binárne asociácie

- asociácie medzi dvoma triedami, **môžu** byť popísané menom
- zvyčajne sa popisujú **slovesami**
- **smer čítania**



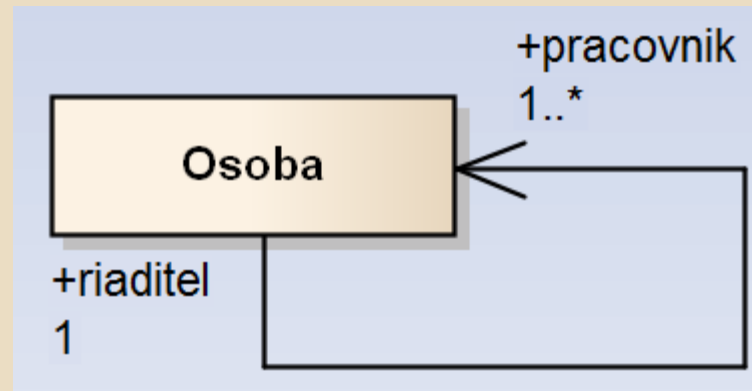
Obojsmerná asociácia

- Vyjadruje „párové property“
- navigačné šípky



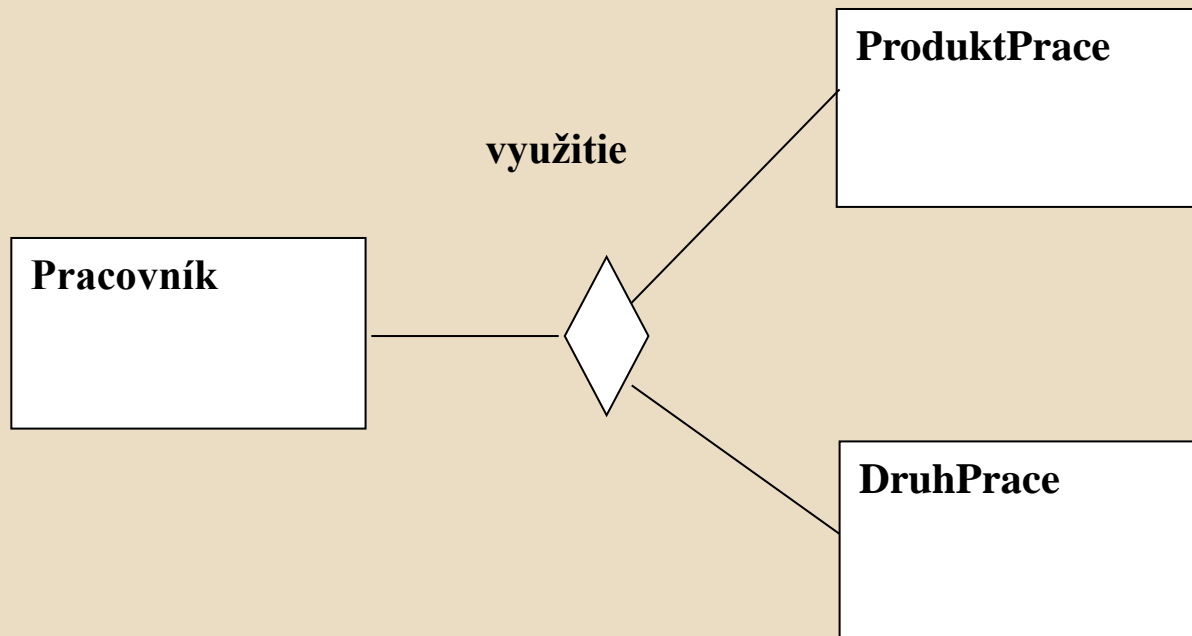
Reflexná asociácia

- *Vzťah medzi prvkami rovnakej triedy*



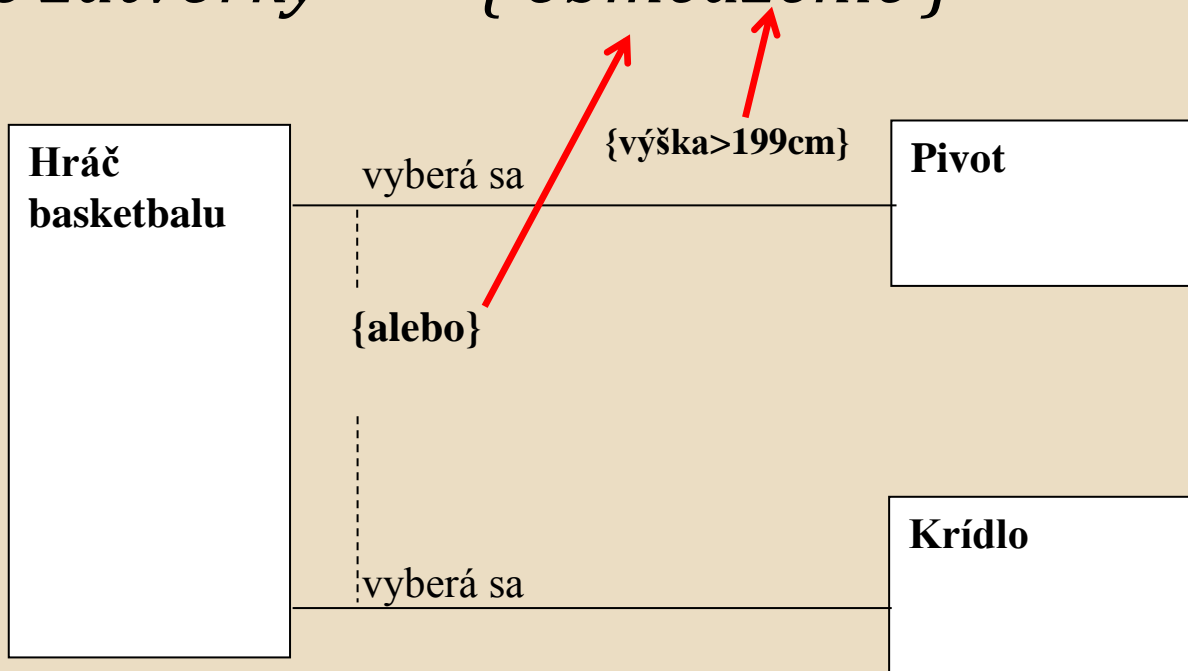
N – árne asociácie

- Vzťah medzi viacerými triedami
- Označenie slovesom alebo podstatným menom

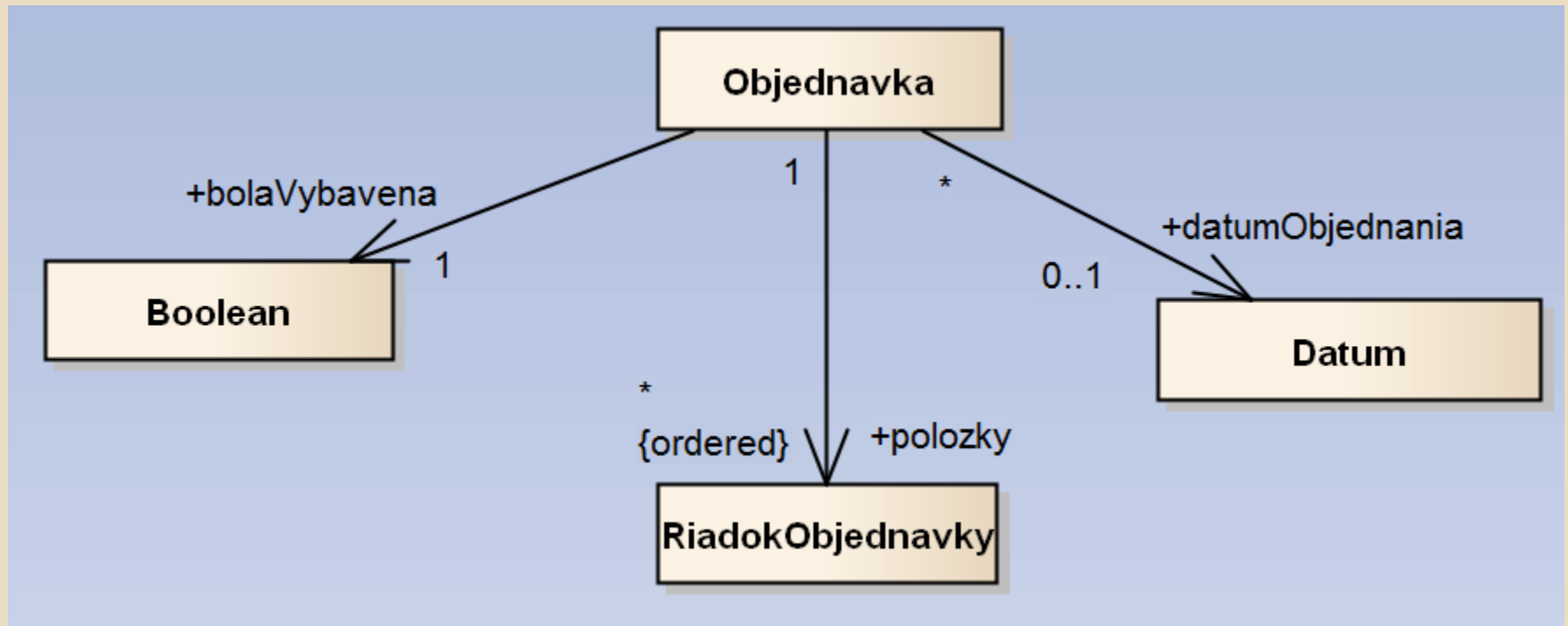
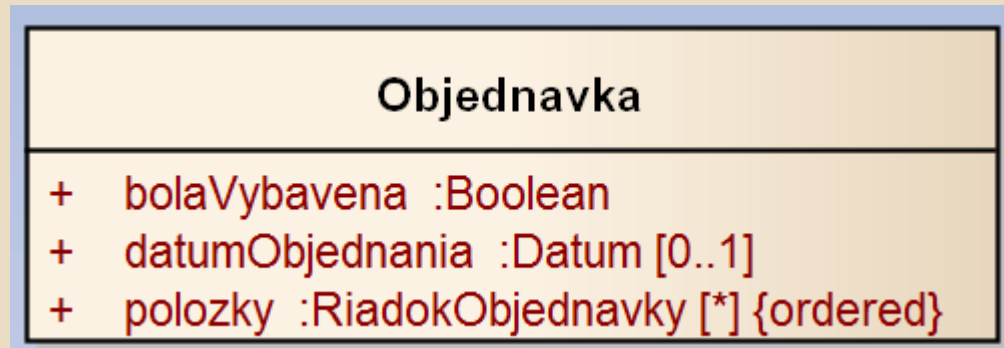


Obmedzenia asociácií

- *Vzťah tried riadený pravidlom*
- *Výber z rôznych možností*
- *Zložené zátvorky* { obmedzenie }



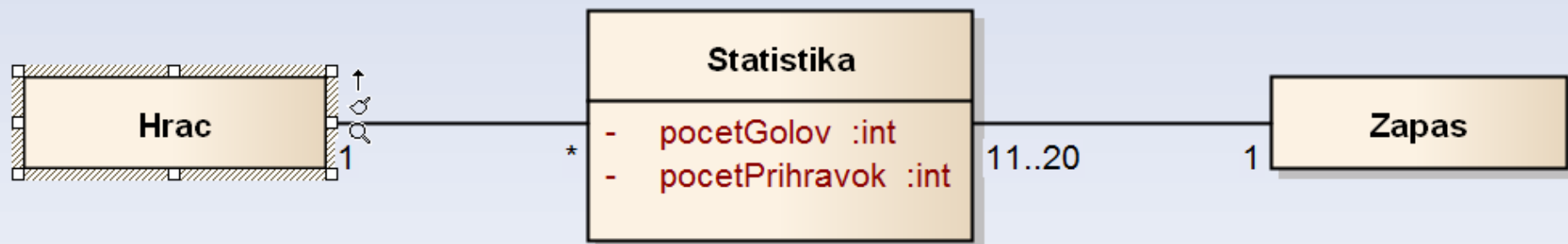
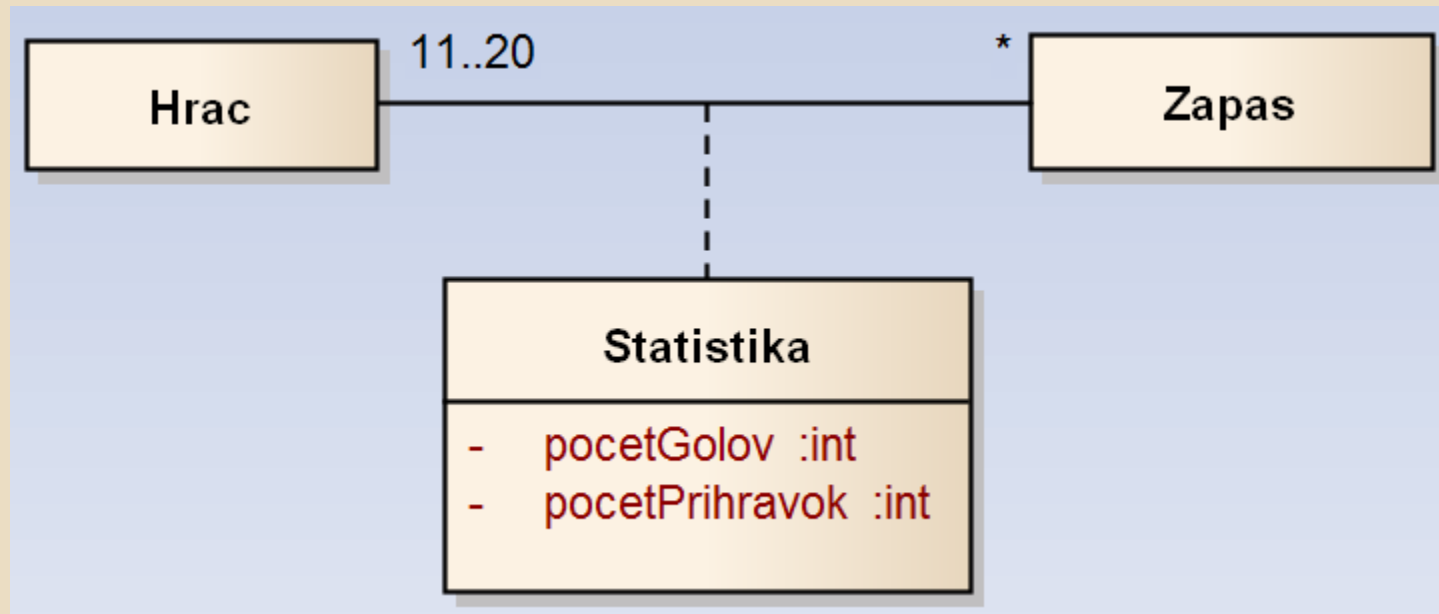
Atribúty a asociácie - porovnanie



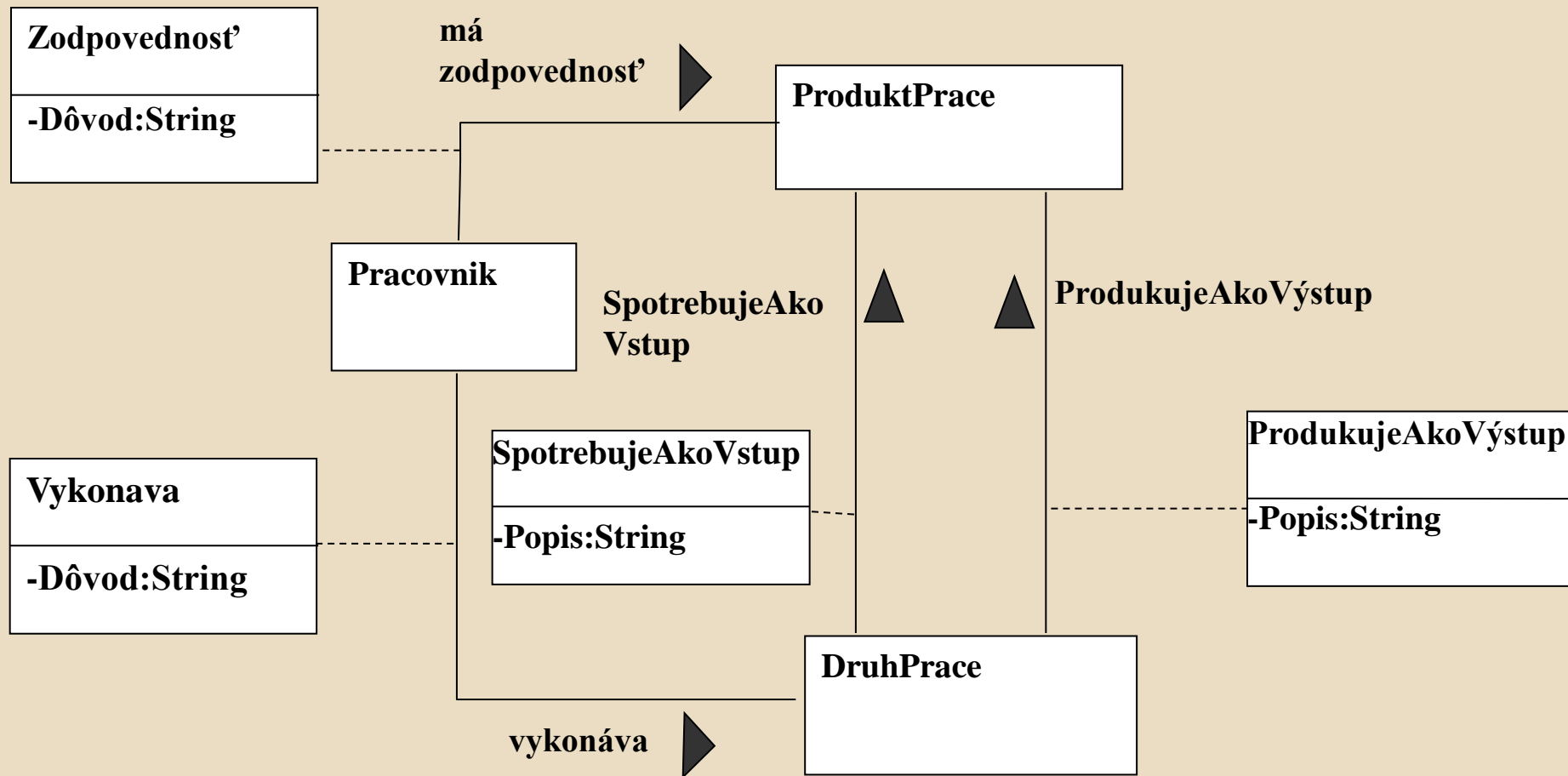
Asociačné triedy

- *asociácia má vlastné **atribúty** a (alebo) **operácie***
- *môže mať asociácie s ostatnými triedami*
- *mala by mať rovnaké meno ako asociácia (nie je nutnou podmienkou)*
- *pre **binárne** aj **n-árne** asociácie*

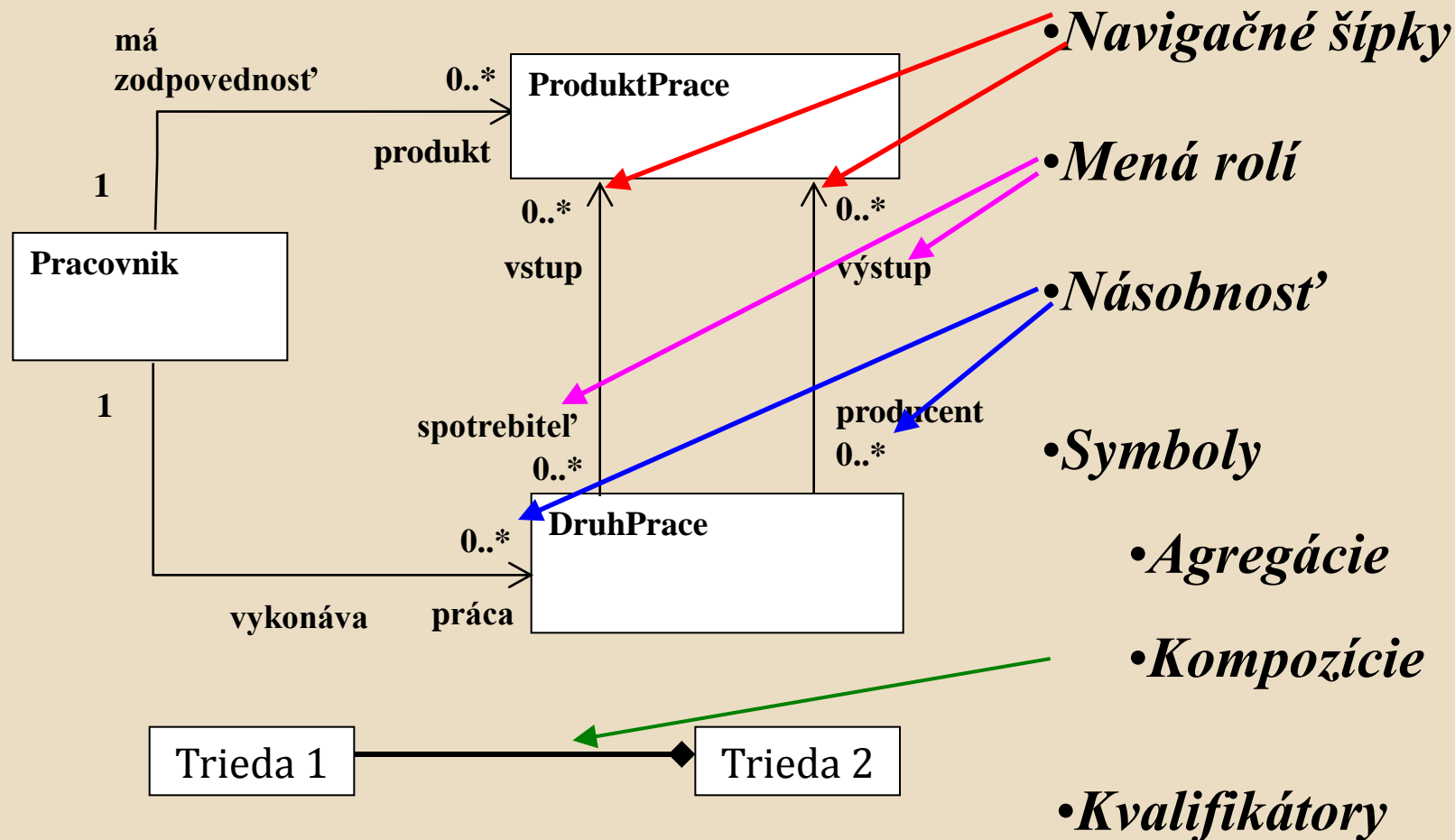
Asociačné triedy - príklad



Príklad asociácií a asociačných tried

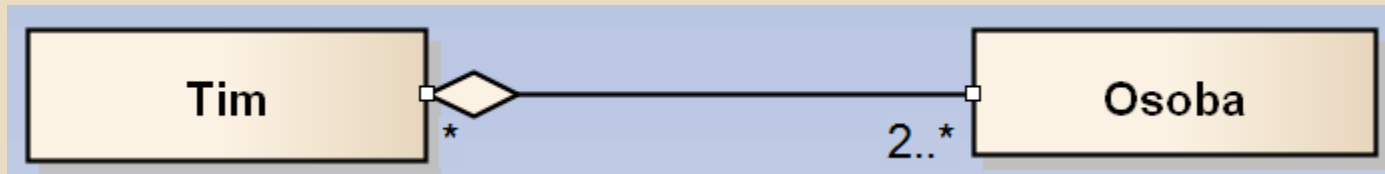


Asociačné konce



Agregácia, kompozícia

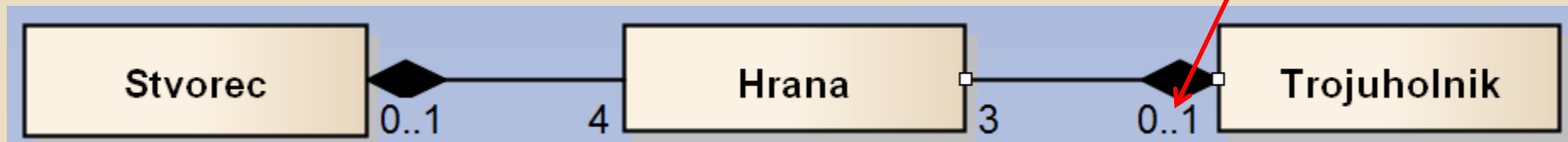
➤ Agregácia



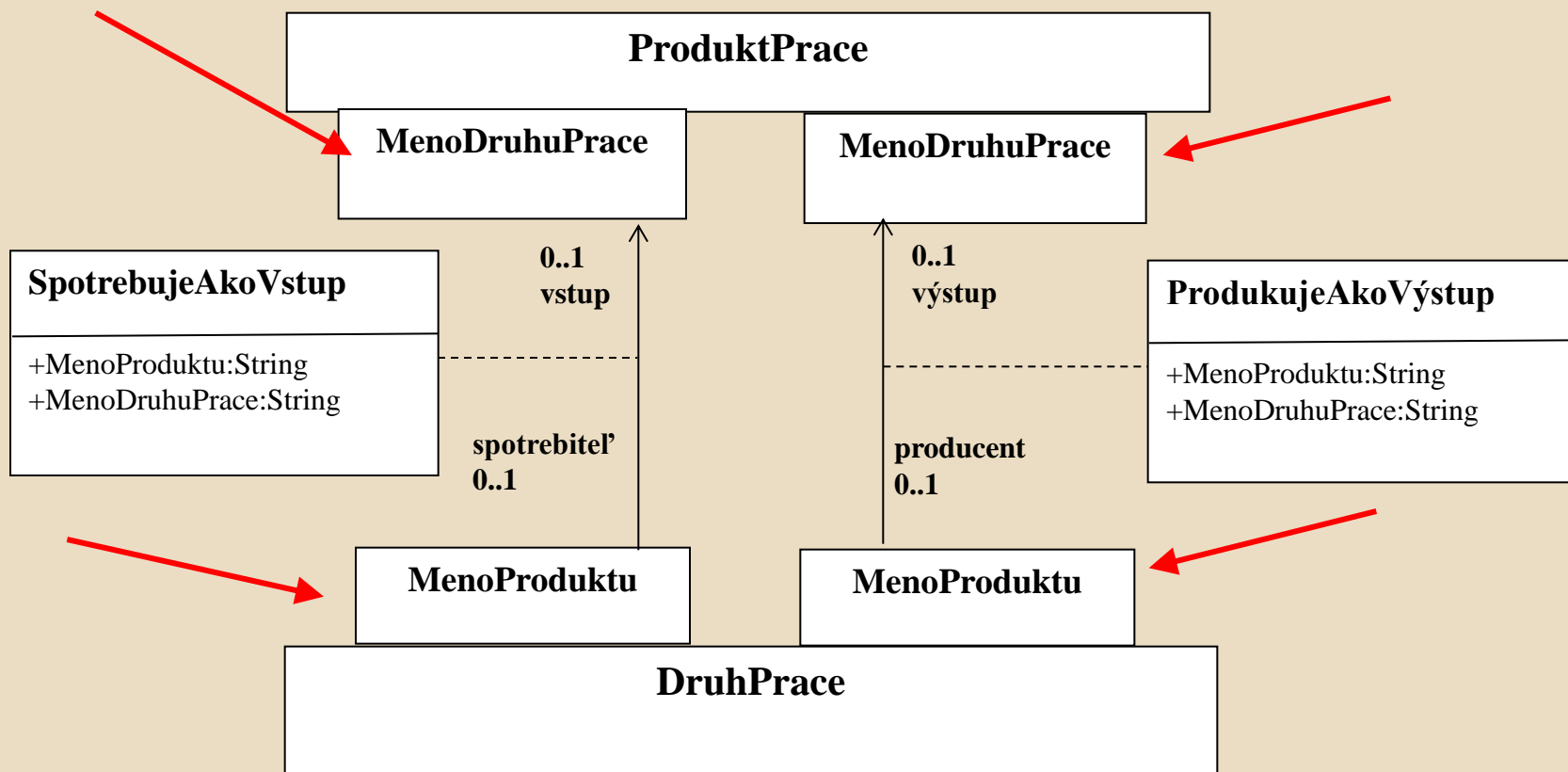
➤ Kompozícia – kompozitná agregácia



- výhradné vlastníctvo
- **Tim nemôže existovať bez Organizacie**
- **Tim sa ruší, keď sa ruší Organizacia**

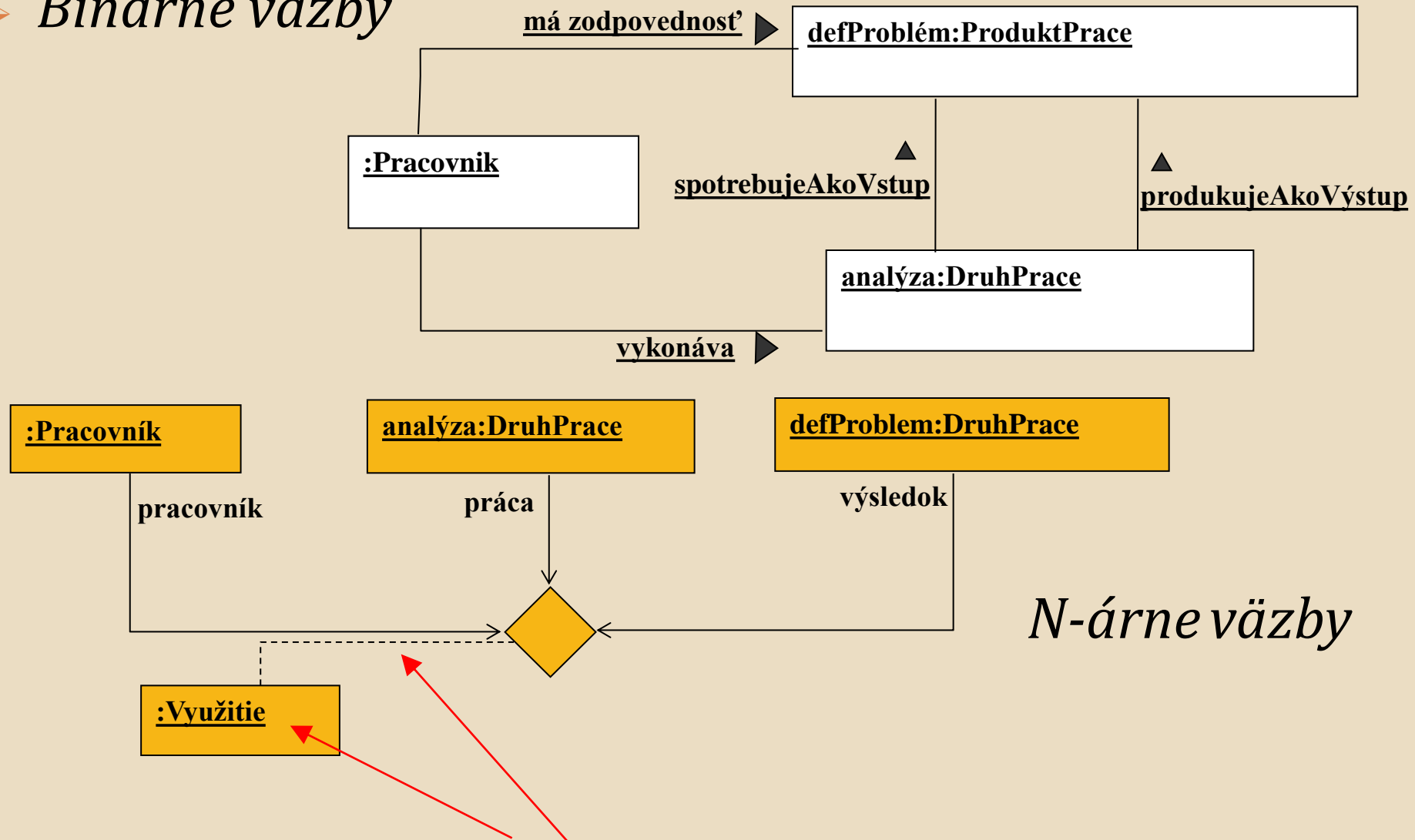


Asociačné konce - Kvalifikátory



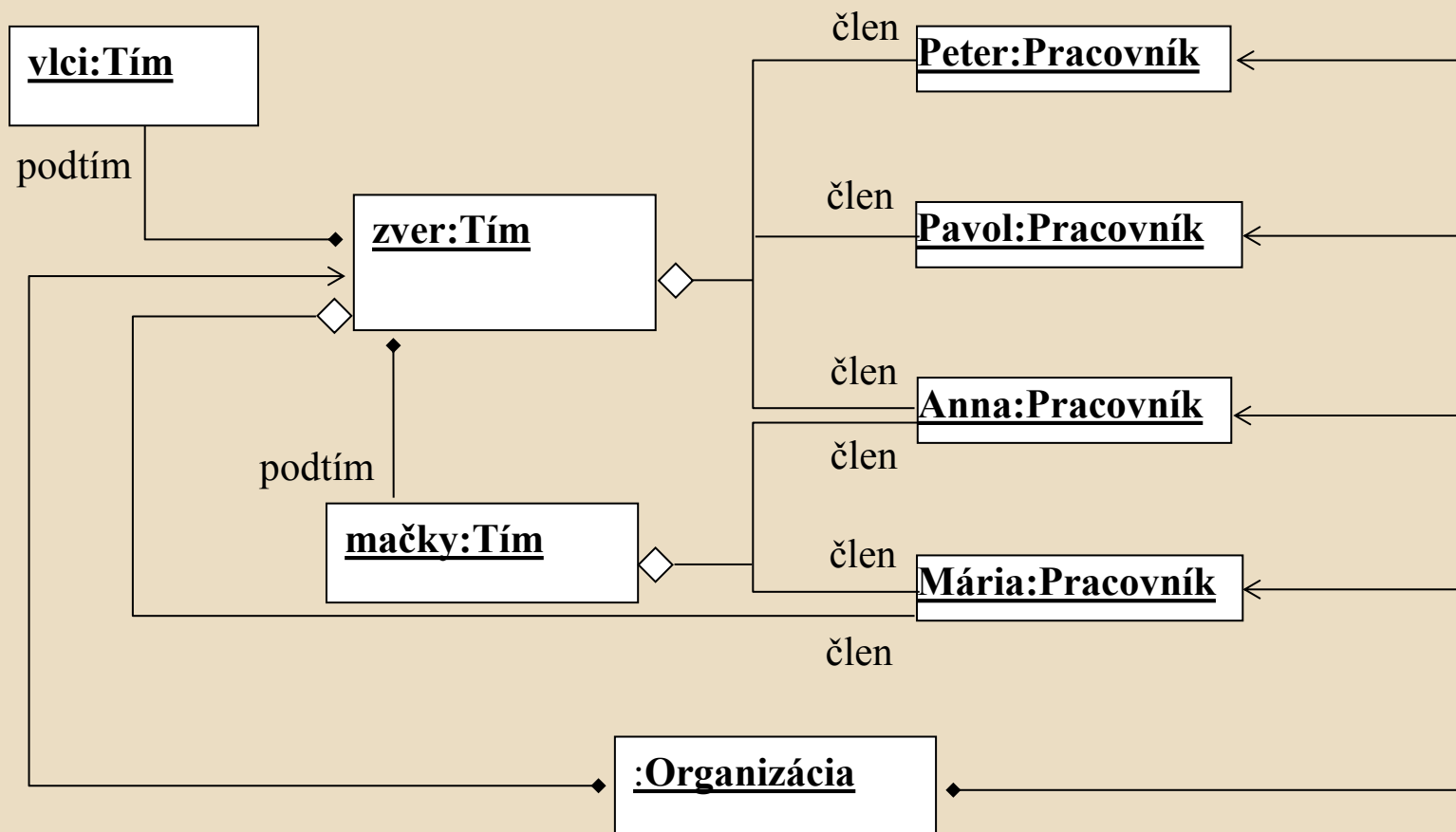
- **Konkrétny vzťah medzi objektmi**
- *Inštancia asociácie*
- **Meno väzby** : *podčiarknuté meno asociácie*
- **Rovnaký koniec väzby ako koniec asociácie**
 - Rola
 - Navigačné šípky
 - Symboly agregácie, kompozície
 - Kvalifikátory
- **Nutnosť rozložiť násobnosť asociácie do jednej alebo viacerých väzieb**

➤ Binárne väzby



Väzobné objekty – inštancie asociačných tried

Príklad ukončení väzieb



Ďakujem vám za pozornosť