SOFTVÉROVÉ MODELOVANIE

3.

Ján Ružbarský
Marek Tavač

KST Katedra softvérových technológií
Fakulta riadenia a informatiky, Žilinská univerzita v Žiline

Obsah 3. prednášky

- *Opakovanie*
- *Ďalšie triedy*
- *Ďalšie typy vzťahov*
- *Balíčky*
- *Vaše otázky*

Opakovanie – Objektová teória

- ***Trieda***
- ***Objekt*** *– inštancia triedy*
- ***Atribúty***
- ***Operácie, metódy***
- ***Správa***
- ***Stimul*** *– inštancia správy*

Opakovanie – Objektová teória

Abstrakcia

- všeobecná – trieda, asociácia
- konkrétna – objekt, väzba

Zapúzdrenie

- atribúty a operácie v triede
- ukrytie metód za operácie – ukrývanie informácií

Zovšeobecnenie – generalizácia

Polymorfizmus

- rôzne triedy – rovnaké operácie, ale rôzne metódy
- schopnosť mať viac metód pre jednu operáciu

Opakovanie - Diagramy tried a objektov

- *štruktúra systému*

- ***Trieda***
  - **štrukturálne črty** – vlastnosti, data – atribúty, asociácie
  - **črty správania**- čo môžu robiť - operácie
  - **UML : štvoruholník**
    - môže mať 1 až 3 časti
    - redukcia triedy
    - použitie stereotypov na označenie oddelených častí

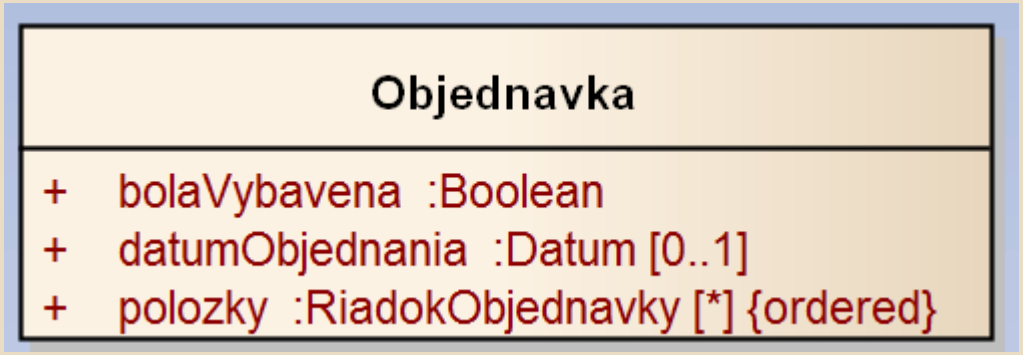
    - **Objekty (inštancie) – podčiarknuté, max. 2 časti**

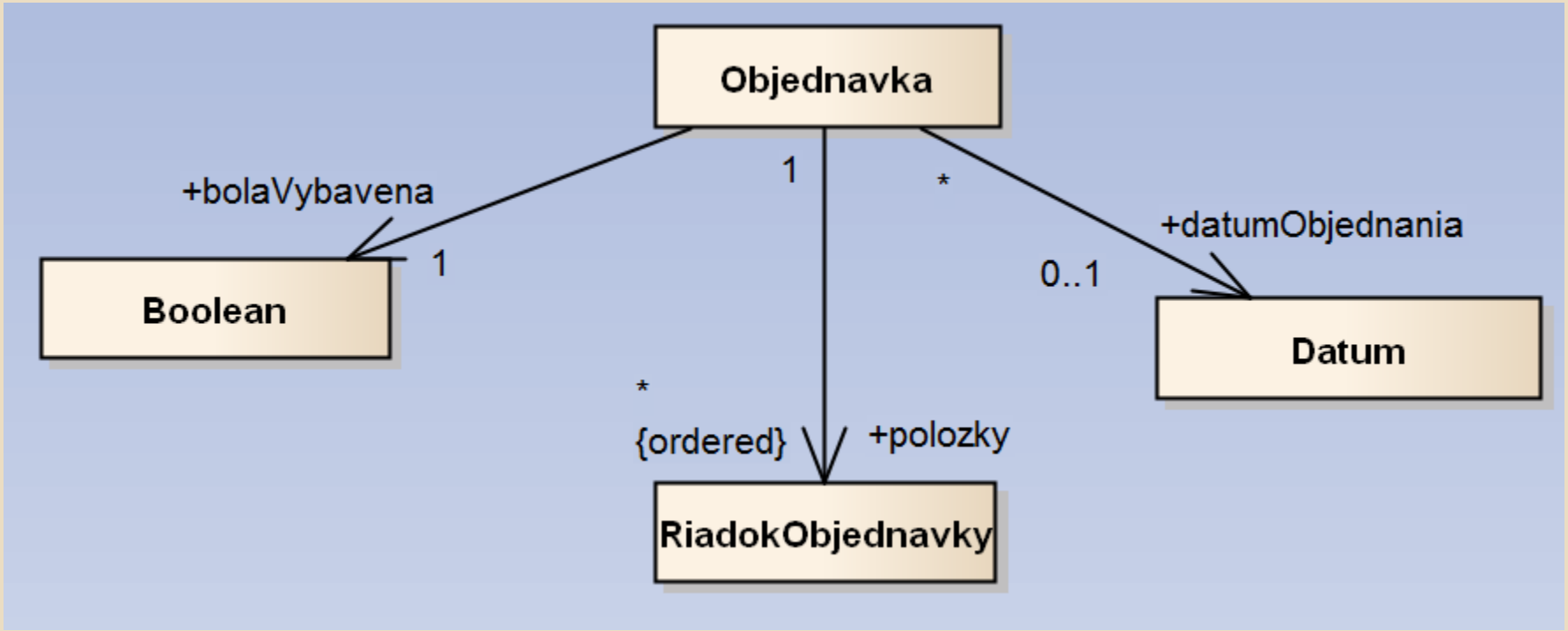
Opakovanie – Asociácie a väzby

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

- ***Binárne asociácie***
- ***N-árne asociácie***
- ***Obmedzenia asociácií***
- ***Asociačná trieda***
  - asociácia má vlastné **atribúty** a (alebo) **operácie**
  - môže mať asociácie aj s ostatnými triedami

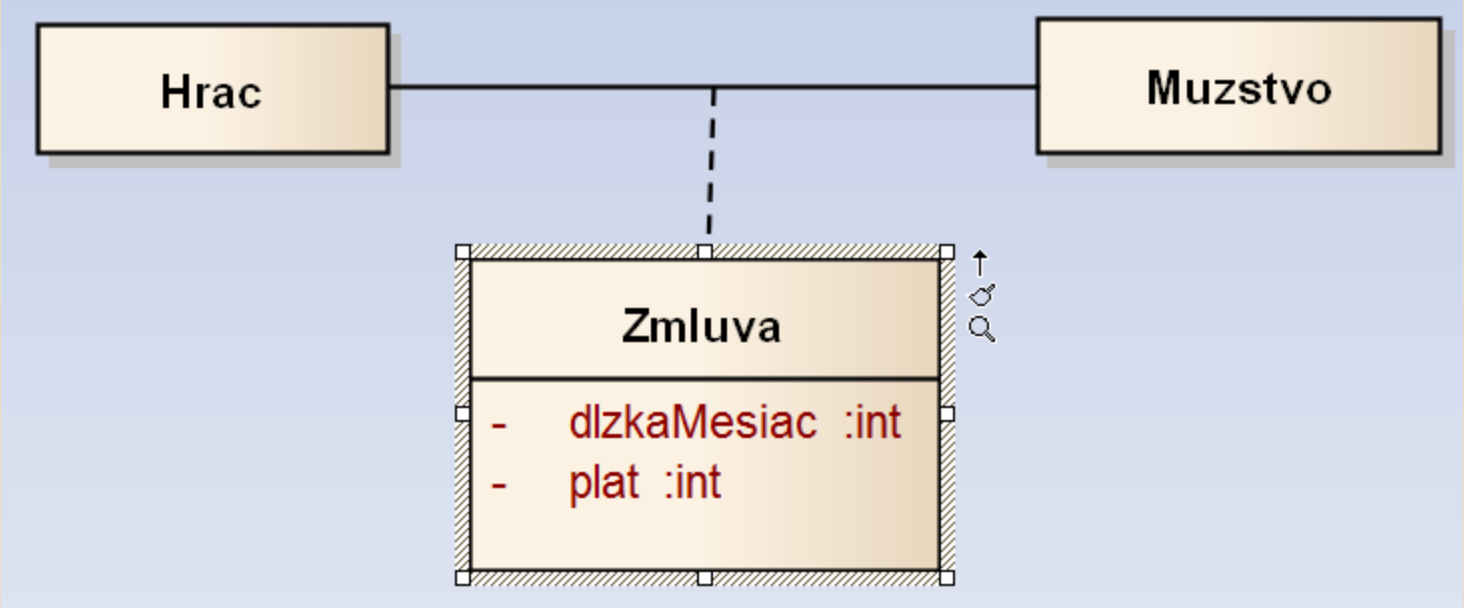
Opakovanie – atribúty alebo asociácie ?



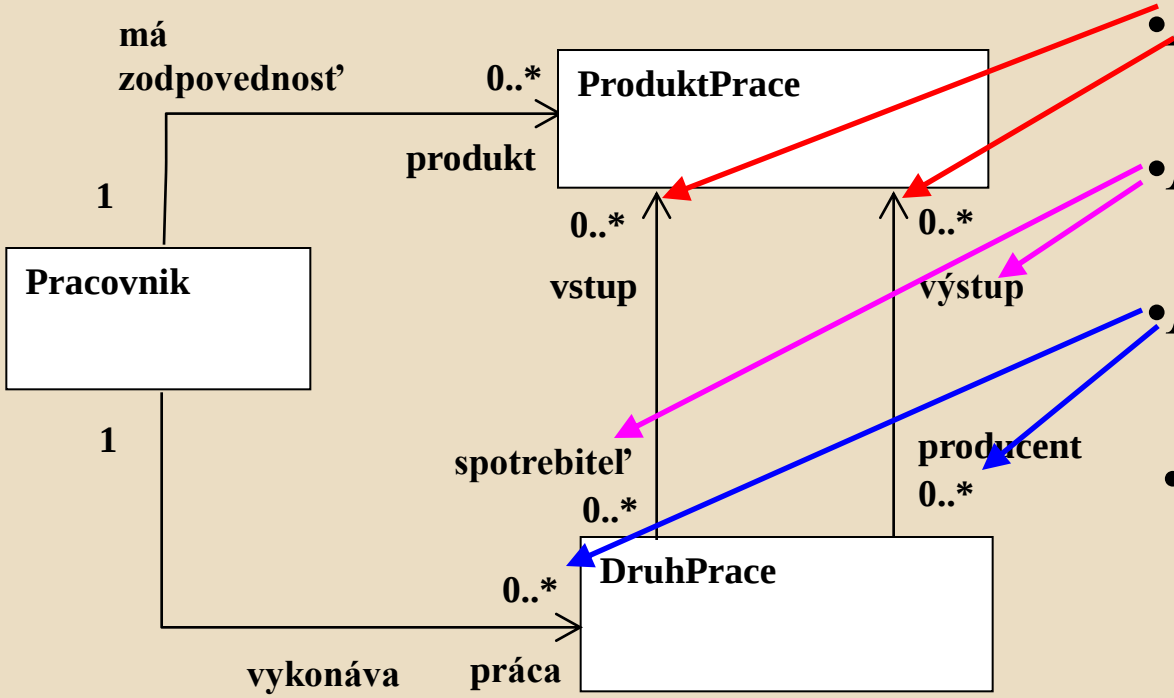


Opakovanie – Asociačná trieda

- *Väzobný objekt – inštancia asociačnej triedy*



Opakovanie – Asociačné konce



- *Navigačné šípky*
- *Mená rolí*
- *Násobnosť*
- *Kvalifikátory*

Opakovanie – Agragácia, kompozícia

- *agregácia*

- *kompozícia – kompozitná agregácia*

  - výhradné vlastníctvo
  - **keď sa ruší Organizacia, ruší sa aj Tim**

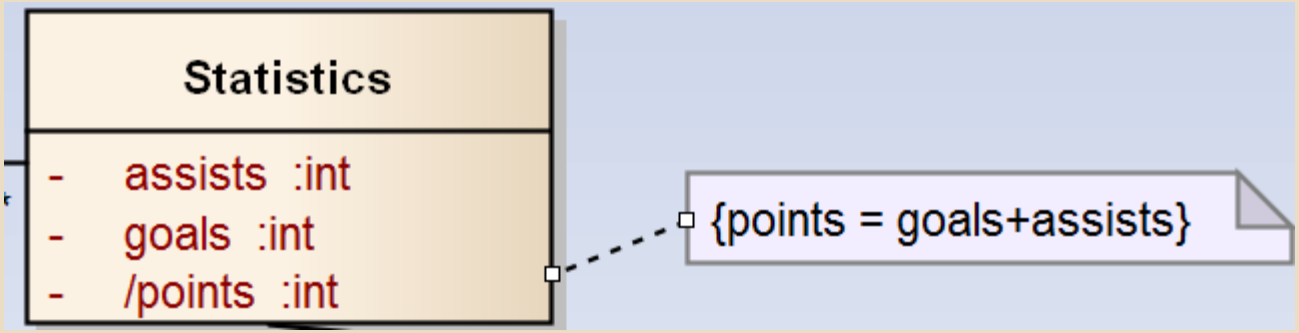
Zodpovednosť tried - Responsibilities

- *V ikone triedy pod zoznamom operácií*
- *Textová forma*
- *CRC karty, analýza UC*

Pracovnik
Responsibilities - pamätať si množstvo vyrobených výrobkov

Odvodené vlastnosti – derived properties

- *Môžu byť **vypočítané** na základe iných hodnôt*



- ***Indikujú obmedzenie** - z troch hodnôt sú dve uložené a jedna vypočítaná – je jedno ktorá*

Abstraktné triedy

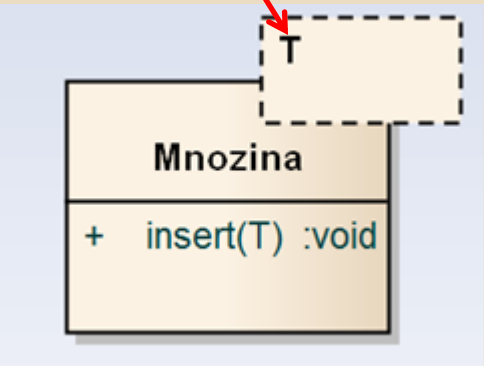
- *Nemajú inštancie*
- *Slúžia na tvorbu subtried*

- *Abstraktná operácia – nemá implementáciu*
- *Kurzíva*

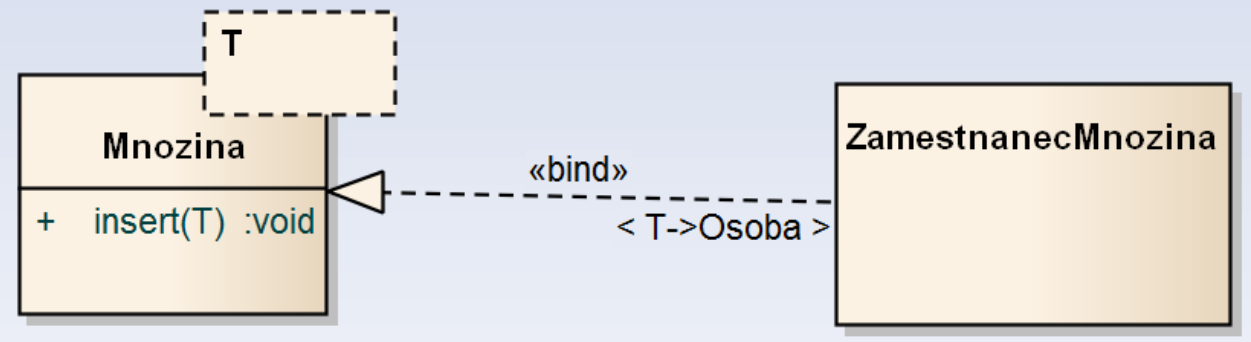
- *Rozhranie-interface*

Šablóny – parametrizované triedy

- *Definujú správanie pre nejakú všeobecnú množinu použitím parametra*
- *definícia*



- ***Odvodenie(Derivation)** - použitie parametrizovanej triedy*



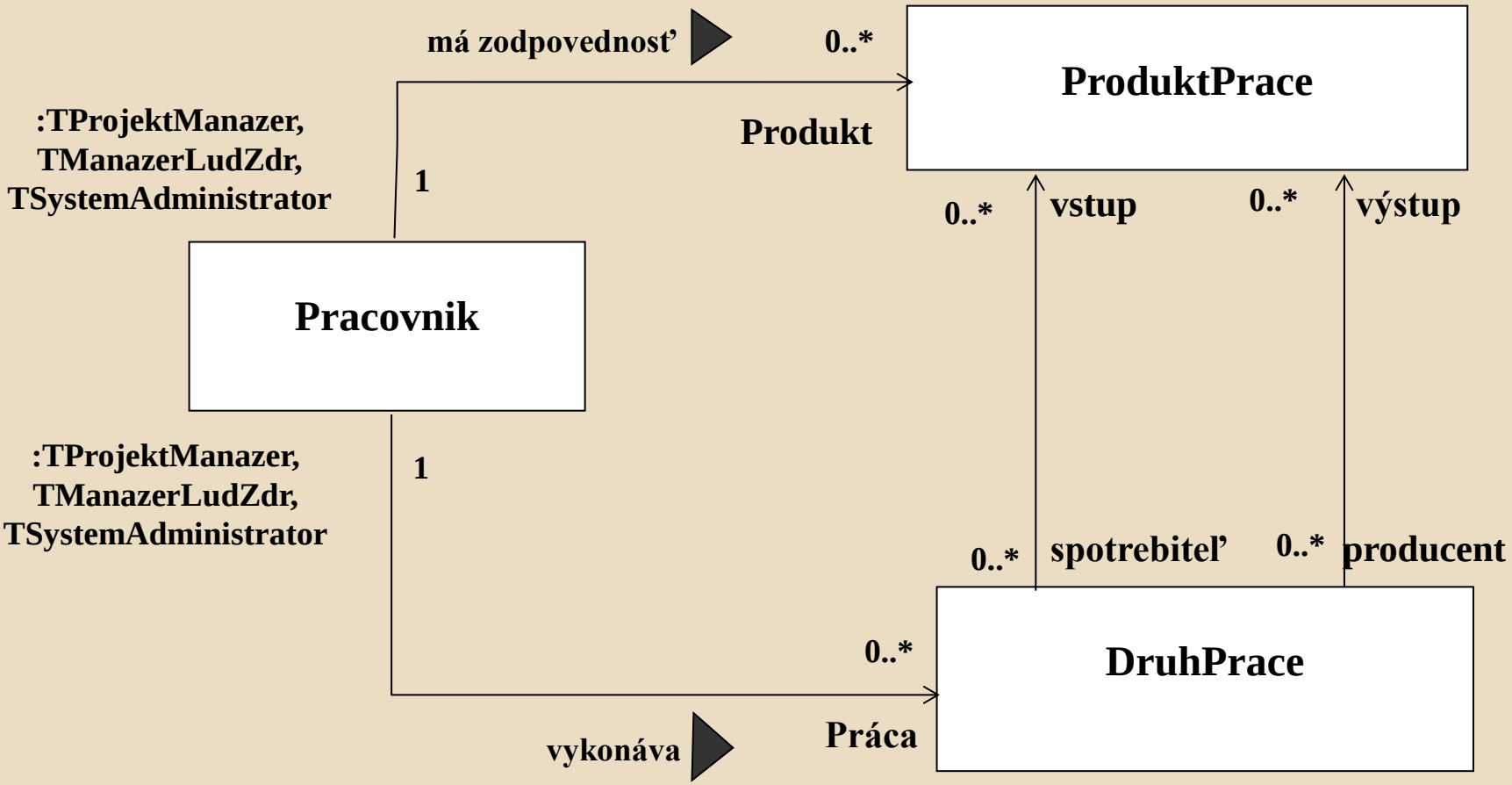
Vymenované typy - enumerations

- *Fixná množina hodnôt*

Triedy viac

- ***nediferencované triedy***
  - doteraz – bežne používané
- ***diferencované triedy***
  - typové triedy
  - implementačné triedy
  - interface

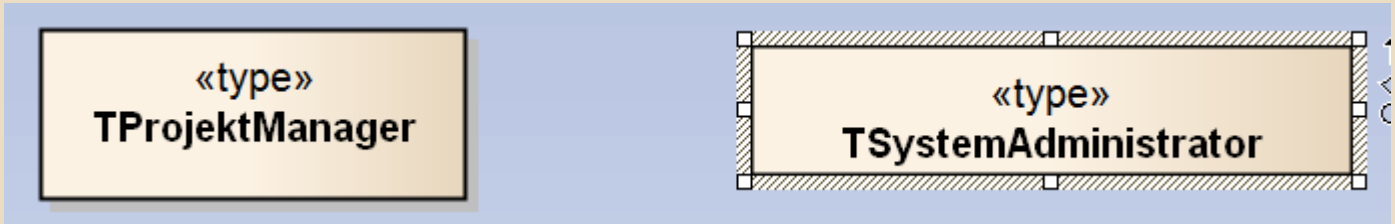
Príklad - Typy



Typy – typové triedy

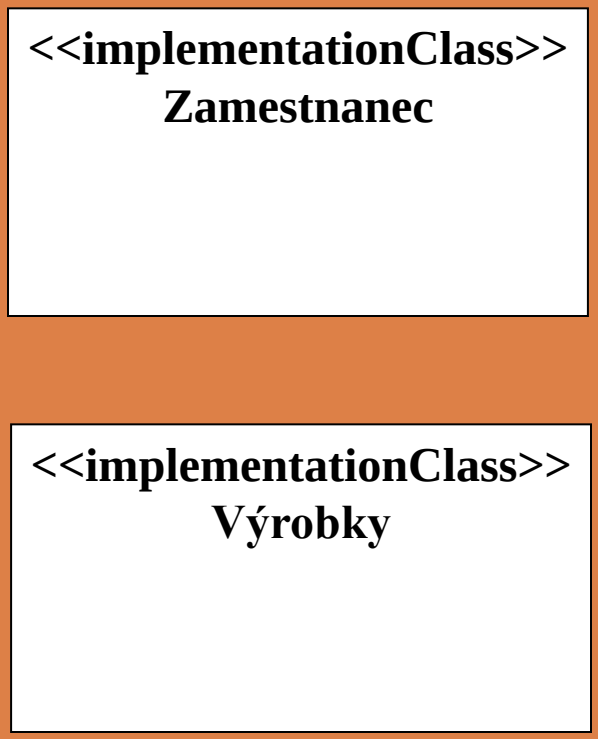
- *Trieda, ktorá môže mať*
  - atribúty – v analýze
  - operácie
  - Asociácie
- ***nemá metódy***
- ***definuje rolu*** *objektu vo vzťahu k iným objektom*

  (podobne ako rola v asociácii)
- *počas analýzy*



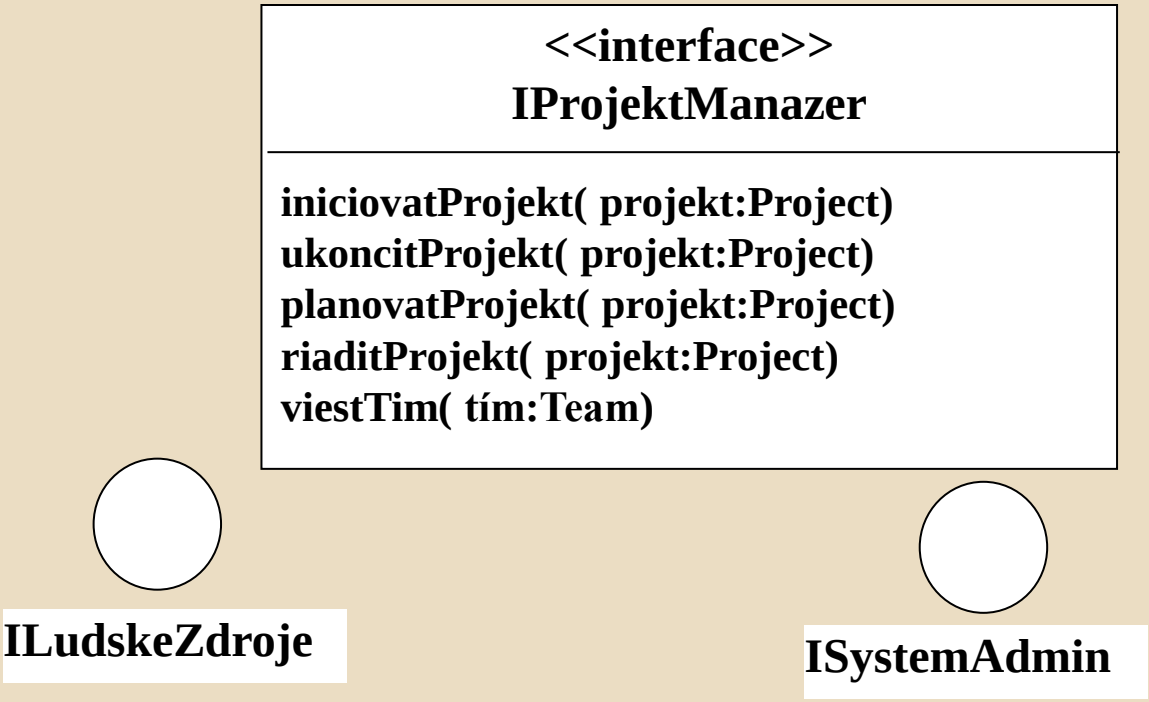
Implementačné triedy

- *Môžu mať*
  - atribúty
  - asociácie
  - operácie
  - metódy
- ***Definujú fyzickú reprezentáciu** objektov triedy*
- *V neskorej fáze návrhu a počas implementácie*



Interface

- *Môžu mať*
  - **operácie**
- *Nesmú mať*
  - **atribúty**
  - **asociácie**
  - **metódy**



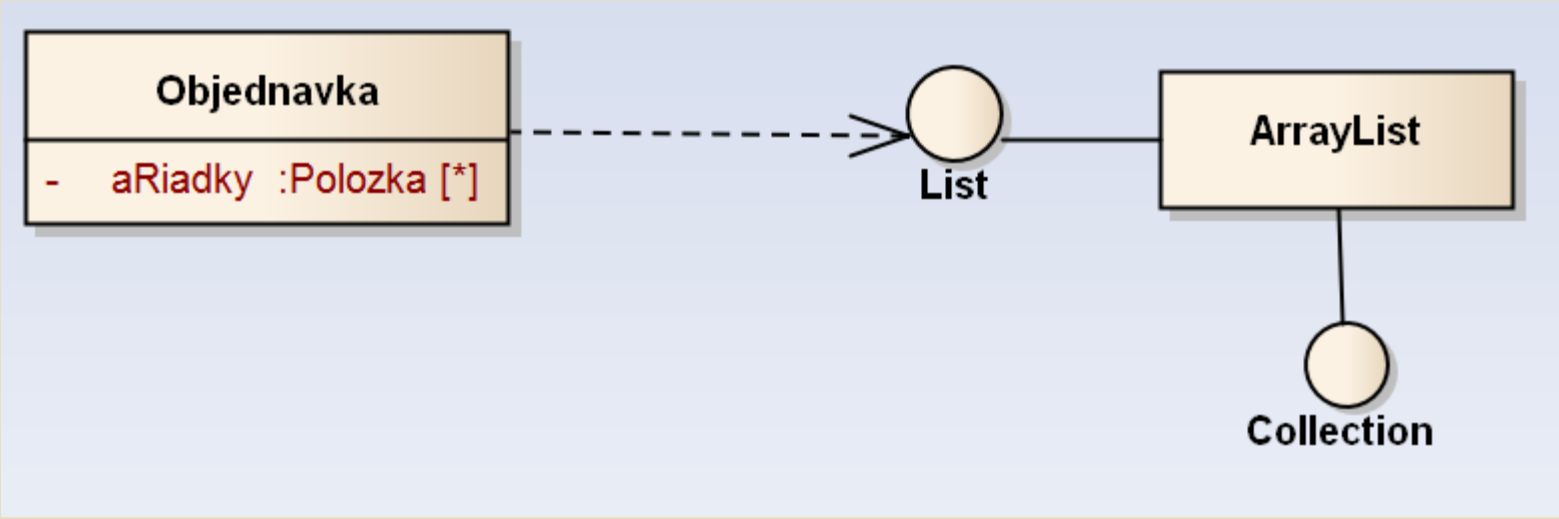
- ***Definuje funkcie**, spoluprácu pomocou množiny verejných (public) operácií*

- *Počas fázy analýzy a návrhu*

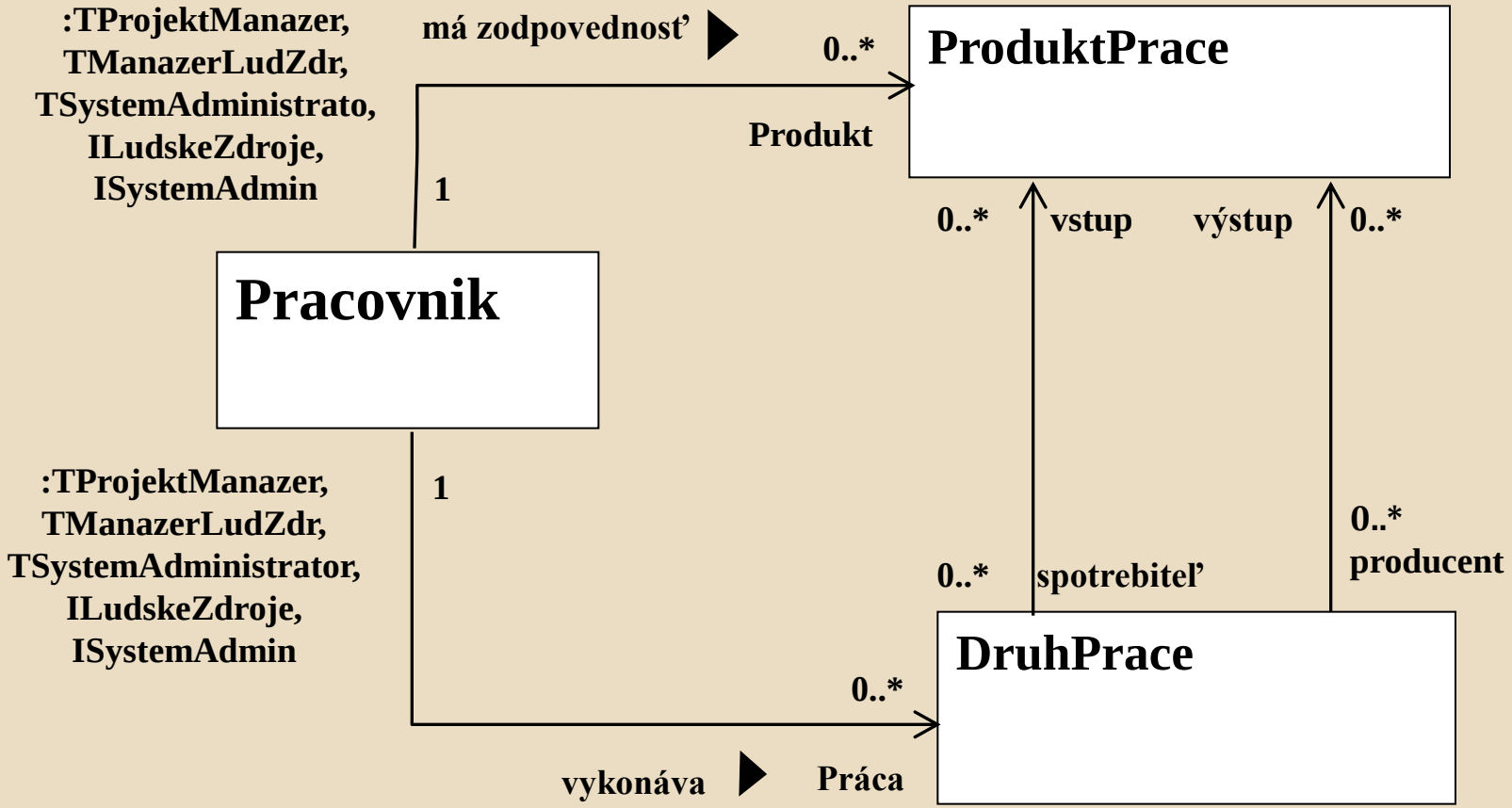
Interface

- ***Poskytovanie rozhrania*** *- realizácia*
  - Trieda implementuje rozhranie

- ***Požadovanie rozhrania*** *- závislosť na rozhraní*
  - Trieda potrebuje inštanciu triedy implemetujúcej rozhranie



Príklad


:TProjektManazer,
TManazerLudZdr,
TSystemAdministrato,
ILudskeZdroje,
ISystemAdmin
má zodpovednosť
0..*
ProduktPrace
Produkt
1
0..*
vstup
výstup
0..*
Pracovnik
:TProjektManazer,
TManazerLudZdr,
TSystemAdministrator,
ILudskeZdroje,
ISystemAdmin
1
0..*
producent
0..*
spotrebiteľ
DruhPrace
0..*
vykonáva
Práca

Aké triedy ?

- *typy – fáza analýzy*
- *rozhrania – fáza analýzy a návrhu*
- *nediferencované triedy – fáza návrhu*
- *implementačné triedy – neskoršia fáza návrhu a implementácia*

Vzťahy

- ***Asociácie*** *– štrukturálne vzťahy*
  - Agregácia, kompozícia
- ***Závislosti*** *(dependency) – vzťahy používania*
- ***Generalizácie*** *– spájajú všeobecné triedy so špecializovanými*

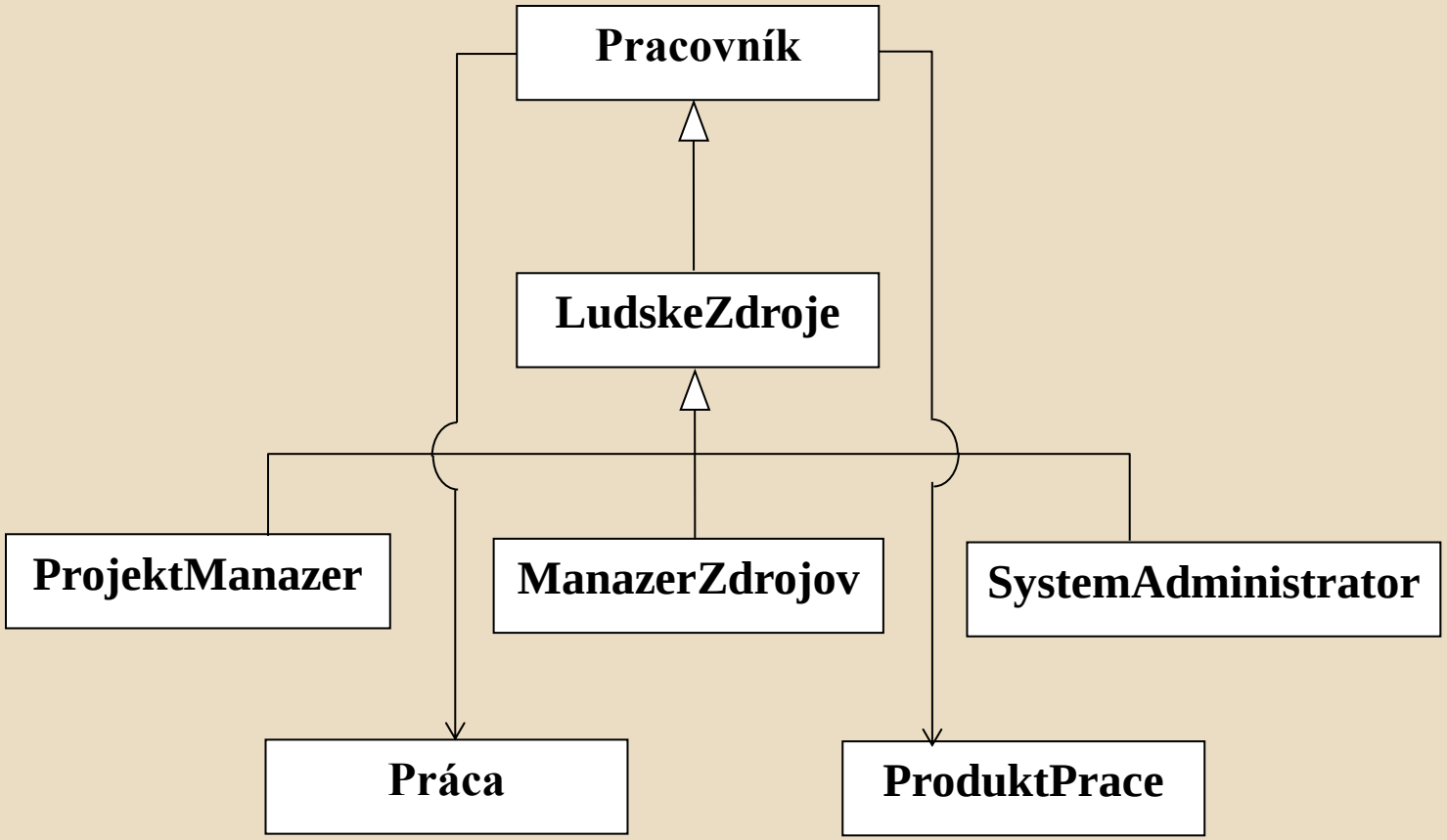
Generalizácie

- *indikujú, že konkrétny prvok získava od všeobecného:*
  - Atribúty, asociácie a iné vzťahy
  - Operácie, metódy
- *prvky musia byť rovnakého druhu*
- ***plná čiara** ukončená **prázdnym trojuholníkom** pri všeobecnom prvku*
- ***Nahraditeľnosť - Subtyping***
  - **Subclassing – použitie dedičnosť**
  - **implementácia rozhraní**
  - **...**

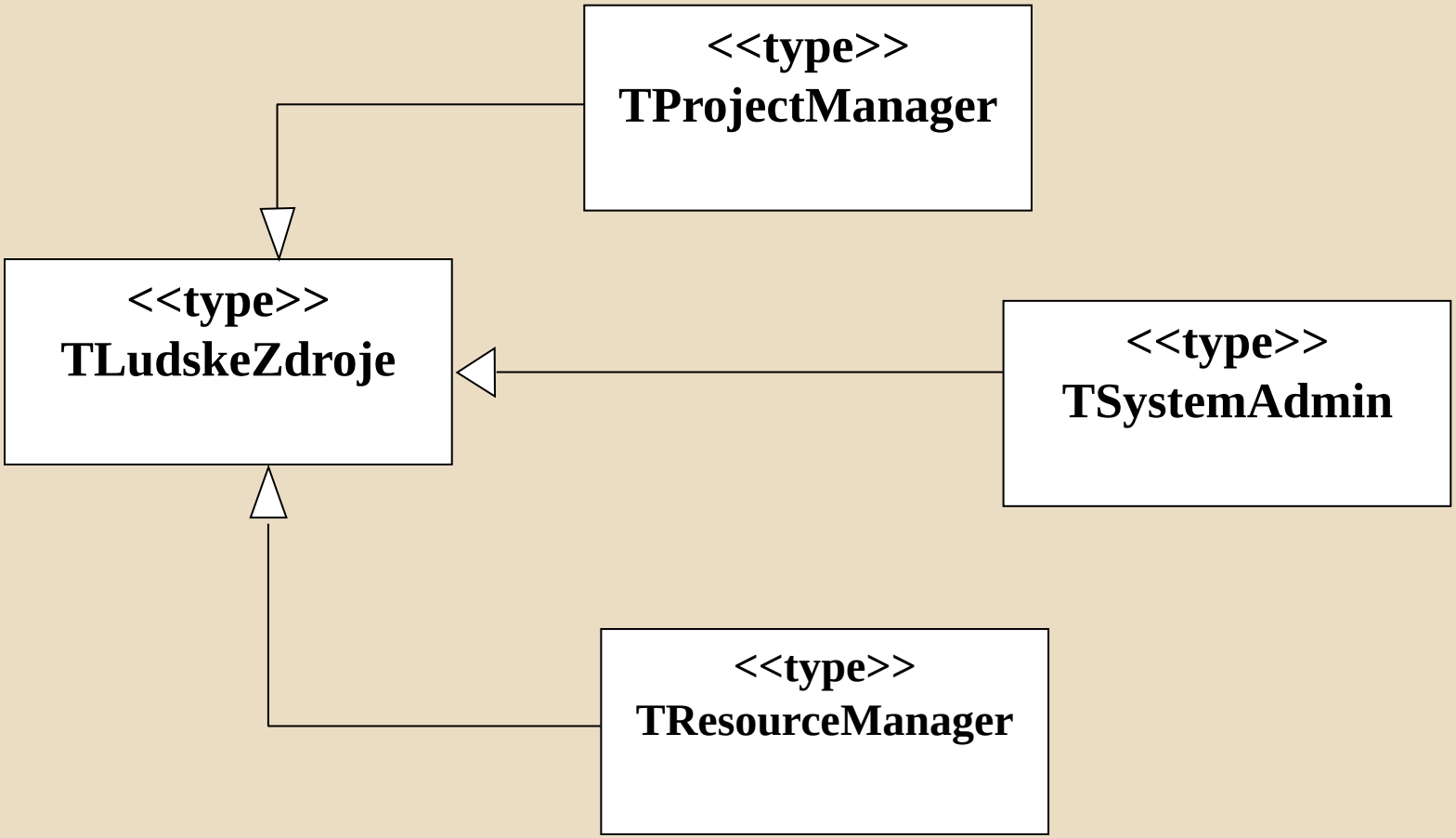
Generalizácie – Nediferencované triedy

- Rodič (nadtrieda)
- Dieťa (subtrieda)

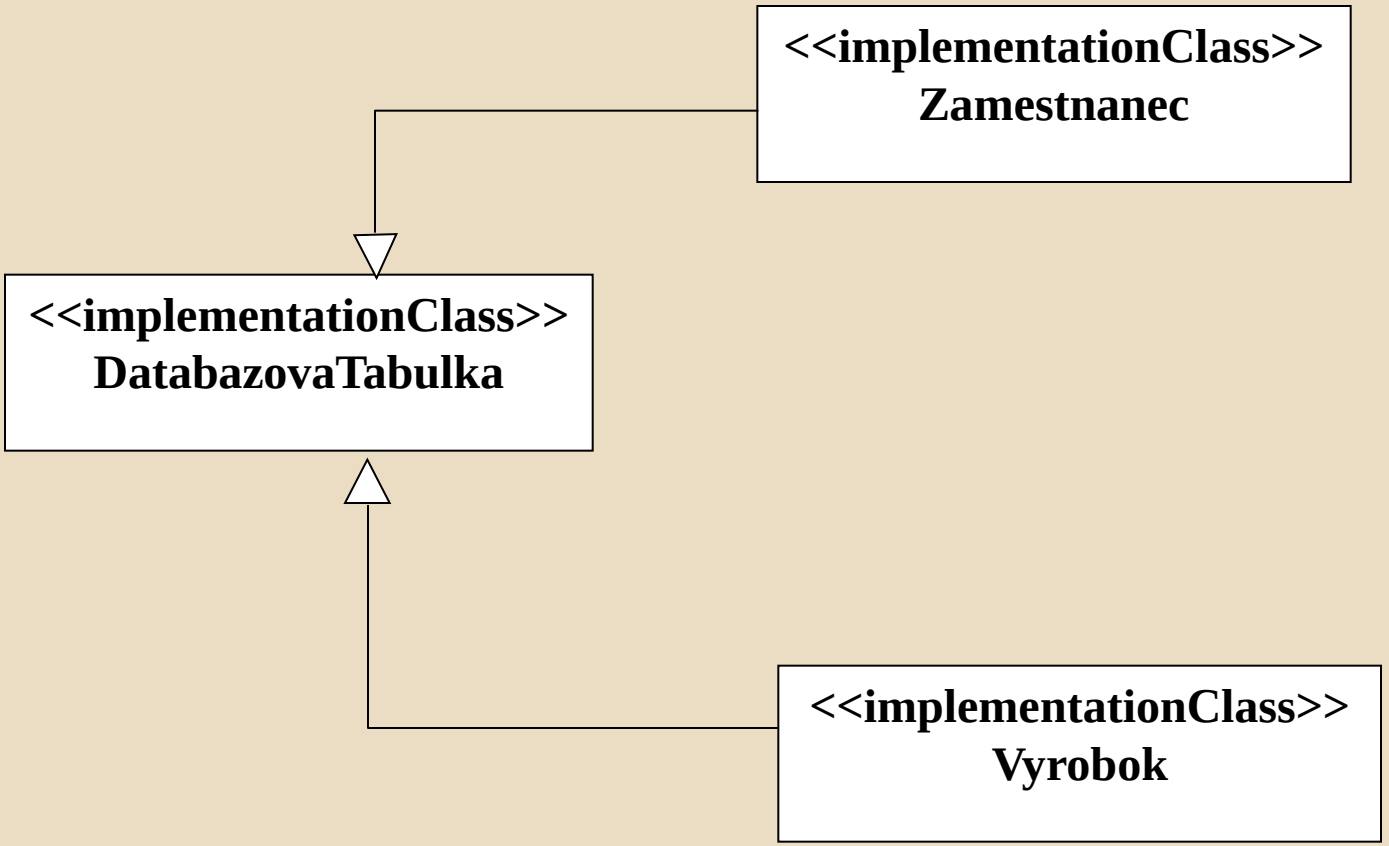
- Predok
- Potomok



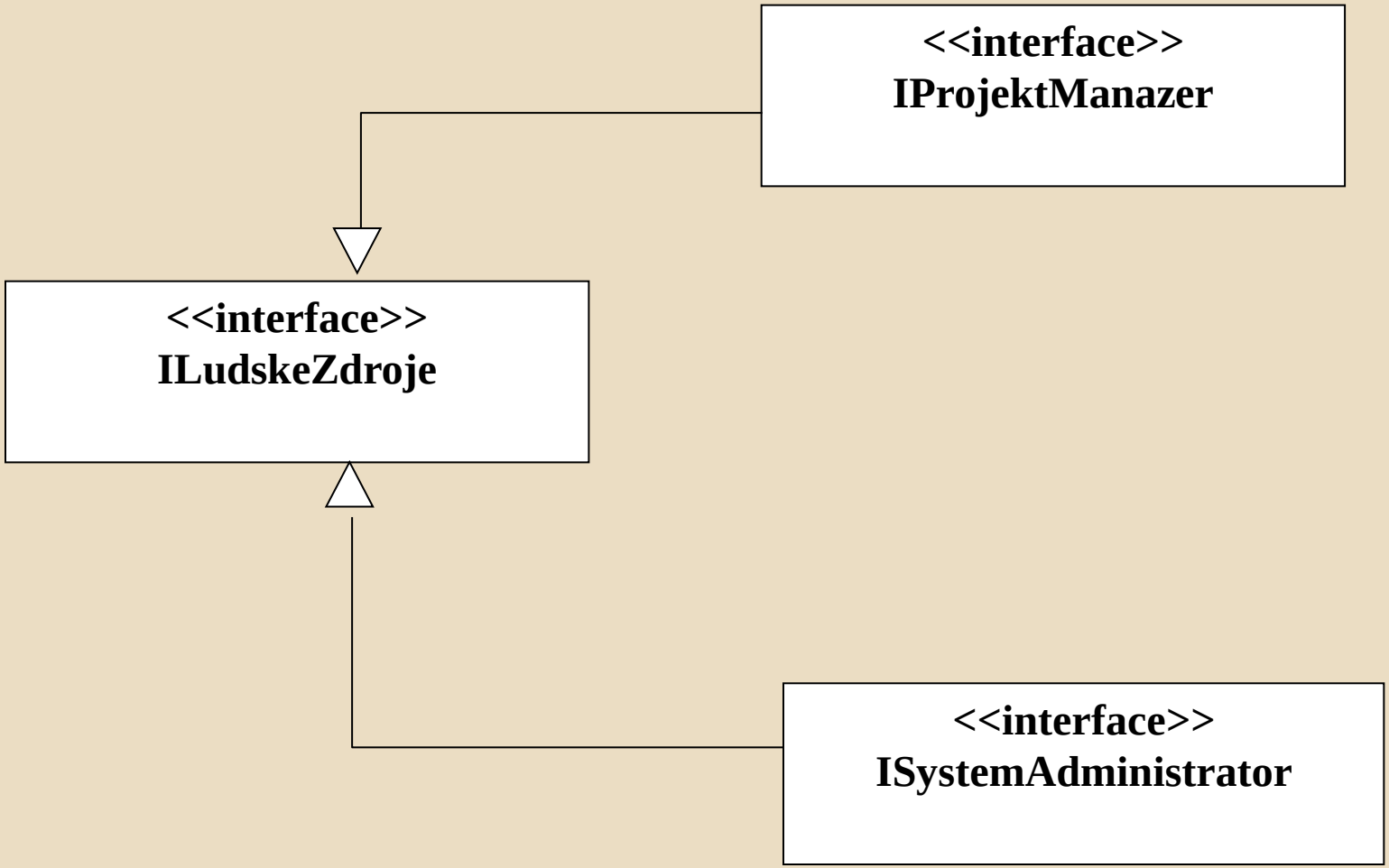
Generalizácie - Typy



Generalizácie - Implementačné triedy



Generalizácie - Rozhrania



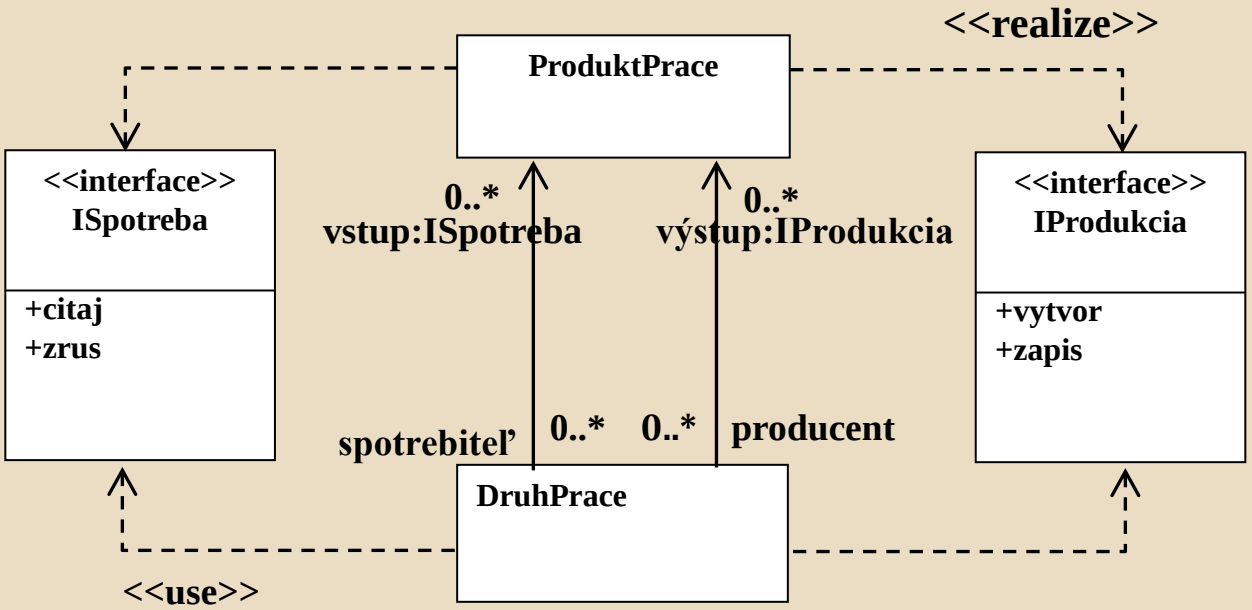
Závislosti - dependency

- *vzťah dvoch prvkov, keď jeden prvok používa informáciu alebo službu iného prvku*
- ***vždy jednosmerná***
- ***čiarkovaná čiara ukončená šípkou*** *smerom od závislého prvku*
- *Možnosti:*
  - Trieda posiela správu inej triede
  - Trieda používa druhú ako parameter svojej operácie

Závislosti - dependency

- *Pomenovanie závislosti – málokedy*
- *Skôr* ***stereotypy s kľúčovými slovami:***
  - **<<call>>**
  - **<<derive>>**
  - **<<realize>>**
  - **<<refine>>**
  - **<<substitute>>**
  - **<<use>>**
  - **...**
- ***MINIMALIZÁCIA ZÁVISLOSTÍ***
- ***ZACYKLENIE***

Závislosti – dependency - príklad

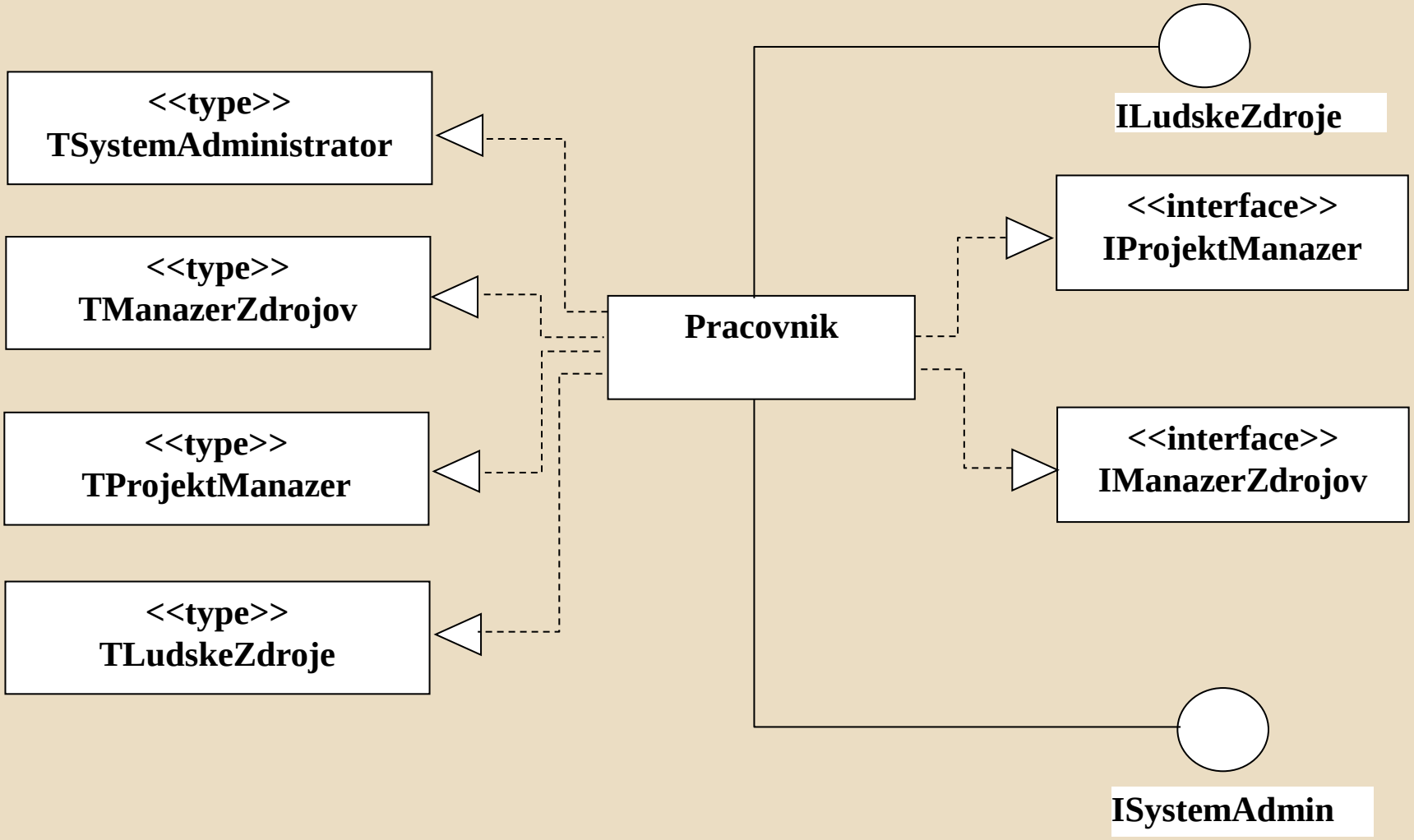


Realizácie

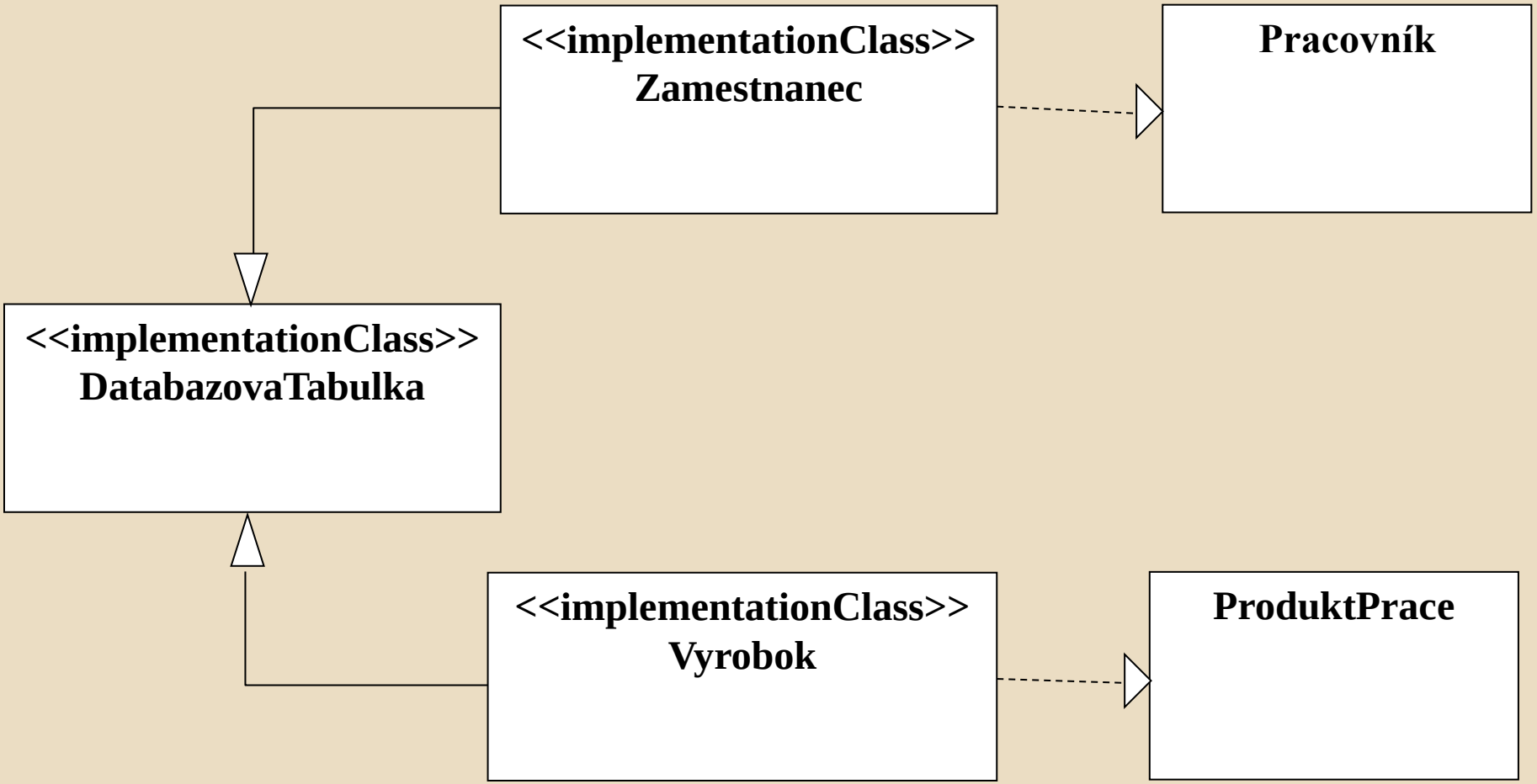
- *Zdrojový prvok sa realizuje do cieľového prvku*
  - Všetky **operácie** cieľového prvku
  - Bez nutnosti podpory atribútov, asociácií cieľového prvku

- ***Čiarkovaná čiara ukončená prázdnym trojuholníkom** pri cieľovom prvku*
- ***Čiarkovaná čiara so stereotypom <<realize>>***

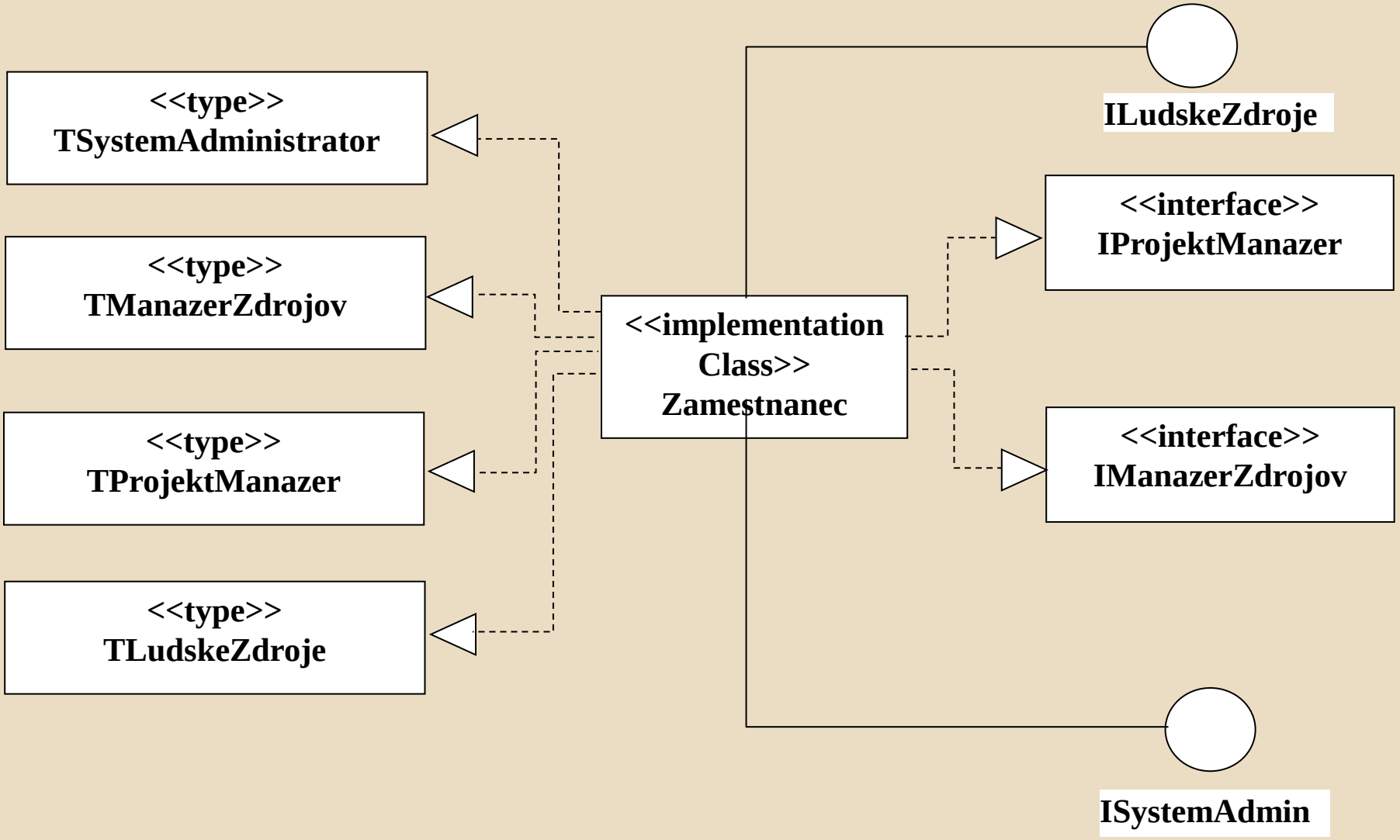
Realizácie - nediferencované triedy



Realizácie - implementačné triedy

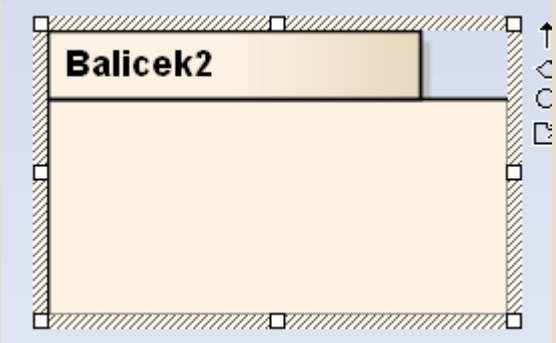


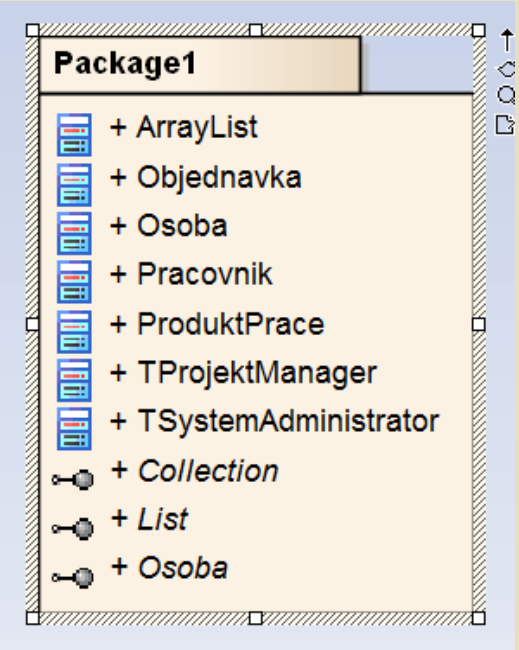
Realizácie - príklad



Balíčky

- *Zoskupenie prvkov*

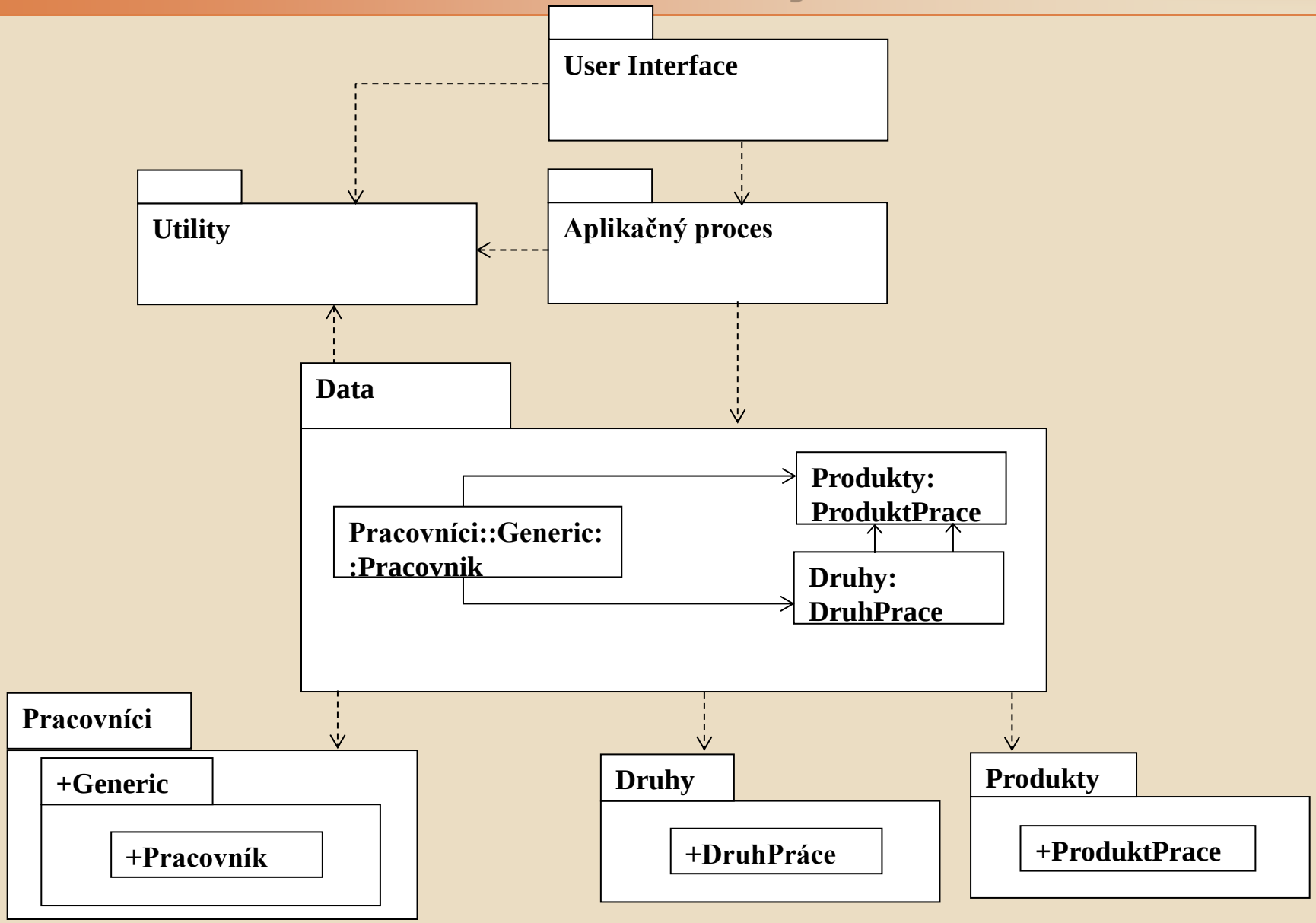




- *Rozdeľujú systém do logických skupín*

- *Definuje „namespace“, kde musia byť unikátne, jedinečné pomenovania prvkov systému*

Príklad – závislosti, balíčky



Balíčky

- *CRP (**common reuse principle**)*
  - v balíčku triedy, ktoré sú spolu znovupoužiteľné
- *CCP (**Common closure principle**)*
  - v balíčku triedy, ktoré sa menia z rovnakého dôvodu
- *ACP (**Acyclic dependencies principle**)*
  - závislosť medzi balíčkami musí byť acyklický digraf

Perspektívy používania diagramov tried

- ***Pojmová perspektíva***
  - Pojmy - štúdium problému
  - Jazykovo nezávislá
- ***Špecifikácia***
  - Rozhrania
  - Často prehliadaná v praxi
- ***Implementácia***
  - Triedy a ich implementácia
  - Najčastejšie používaná

Rady

- *Nesnažte sa vytvoriť hneď kompletný popis problému*
  - najprv triedy, asociácie, atribúty, generalizácie
  - zvyšok možno neskôr
- *Koncentrujte sa na kľúčové veci !!!*
- *Nezanedbávajte modelovanie správania !!!*

Ďakujem za pozornosť

Otázky?