

# SOFTVÉROVÉ MODELOVANIE

## 4.

*Ján Ružbarský*  
*Marek Tavač*

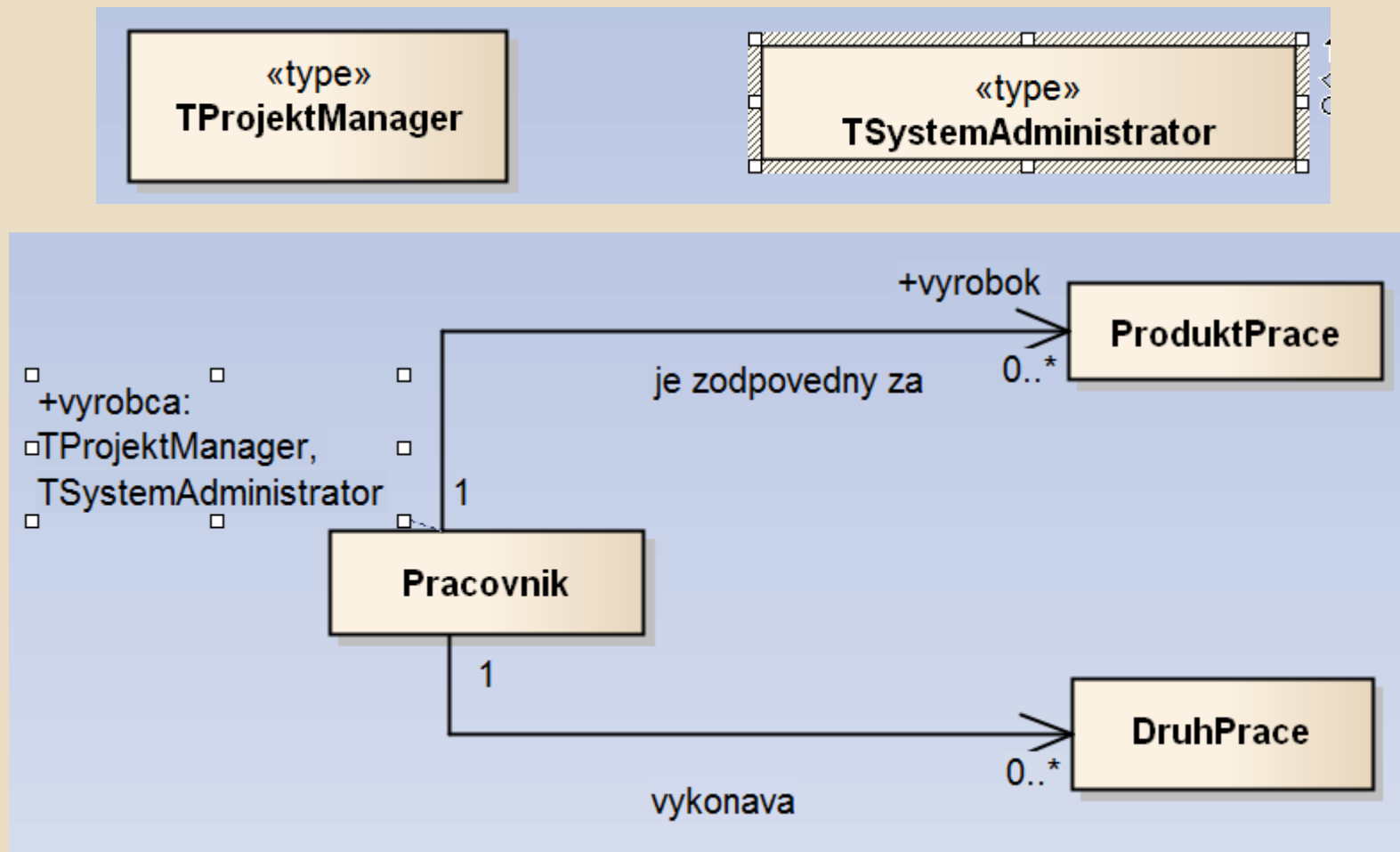
# Obsah 4. prednášky

- *Opakovanie*
- *Diagram prípadov použitia - Use Case diagrams*
  - Elementy
    - Aktor
    - Use case
    - Vzťahy
    - Generalizácie
  - Pravidlá tvorby
- *Vaše otázky*

- *Zodpovednosť (responsibilities) tried*
- *Odvodene (derived) property*
- *Šablóny – parametrizované triedy*
- *Abstraktné triedy*
- *Vymenované typy - enumerations*

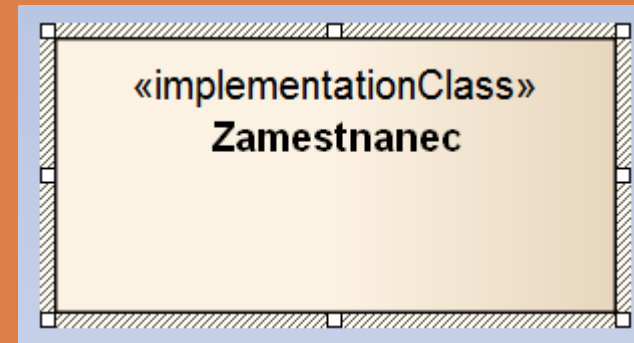
# Opakovanie – Typové triedy

- *definuje rolu objektu vo vzťahu k iným objektom*



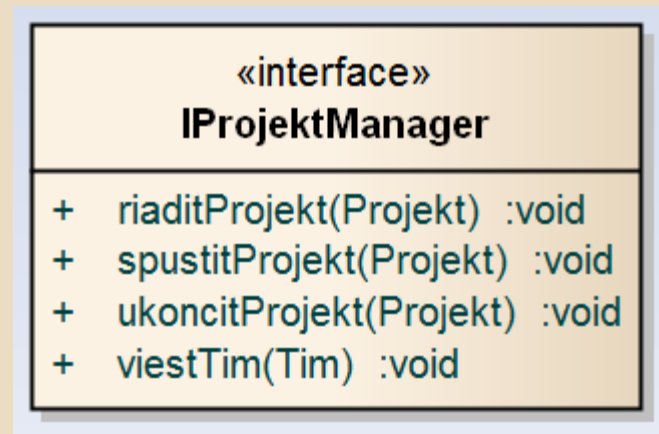
# Opakovanie – Implementačné triedy

- *Definuje fyzickú reprezentáciu objektov triedy*
- *V neskorej fáze návrhu a počas implementácie*

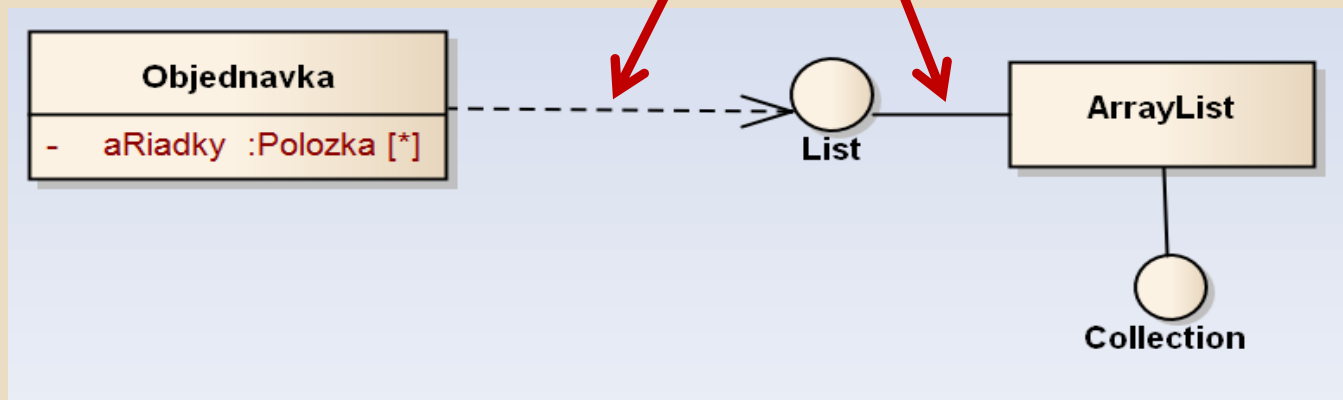


# Opakovanie – Interface

- *Môžu mať iba operácie*



- *Poskytovanie rozhrania - realizácia*
- *Požadovanie rozhrania - závislosť na rozhraní*

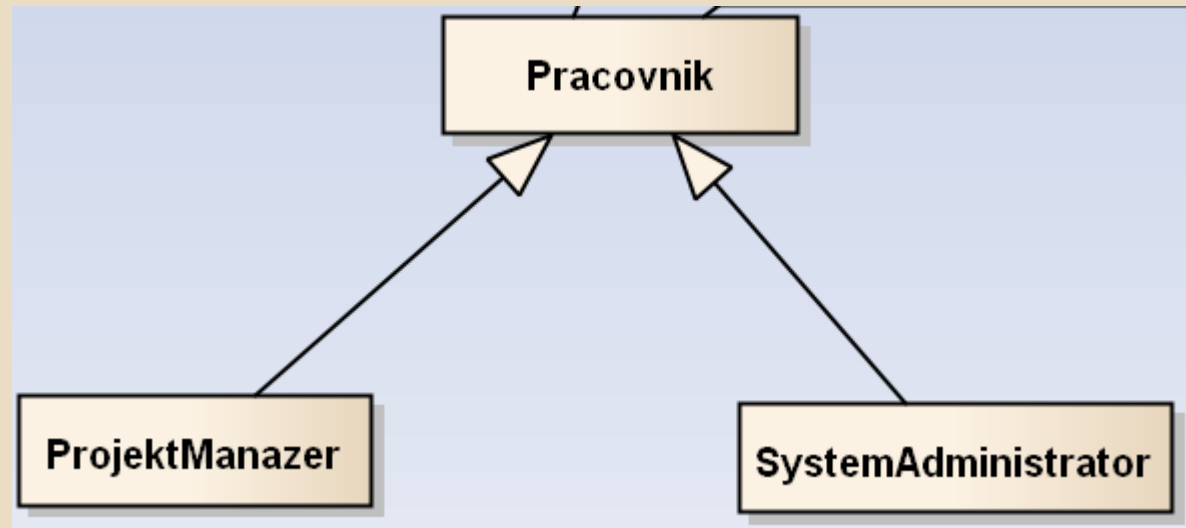


# Opakovanie – vzťah Generalizácia

➤ *prvky musia byť rovnakého druhu*

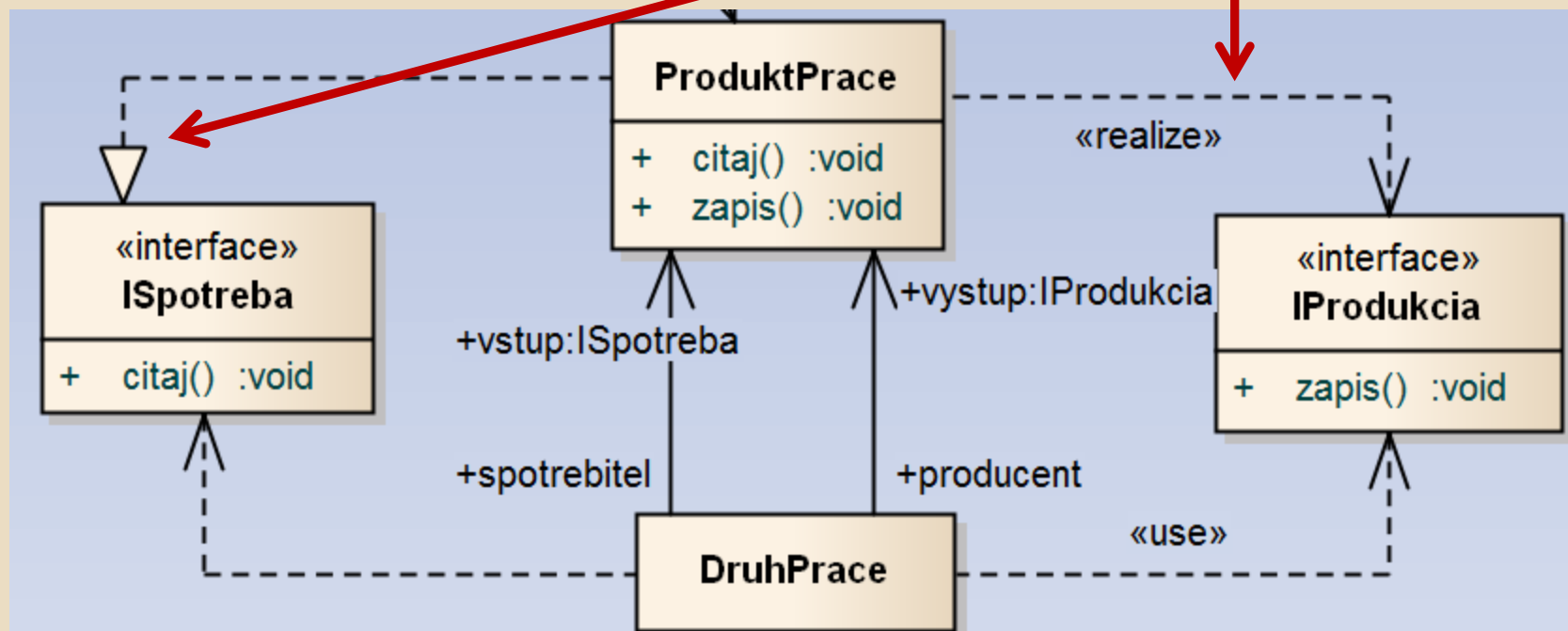
## ➤ **Nahradiťnosť - Subtyping**

- Subclassing – použitie dedičnosti
- implementácia rozhraní
- ...



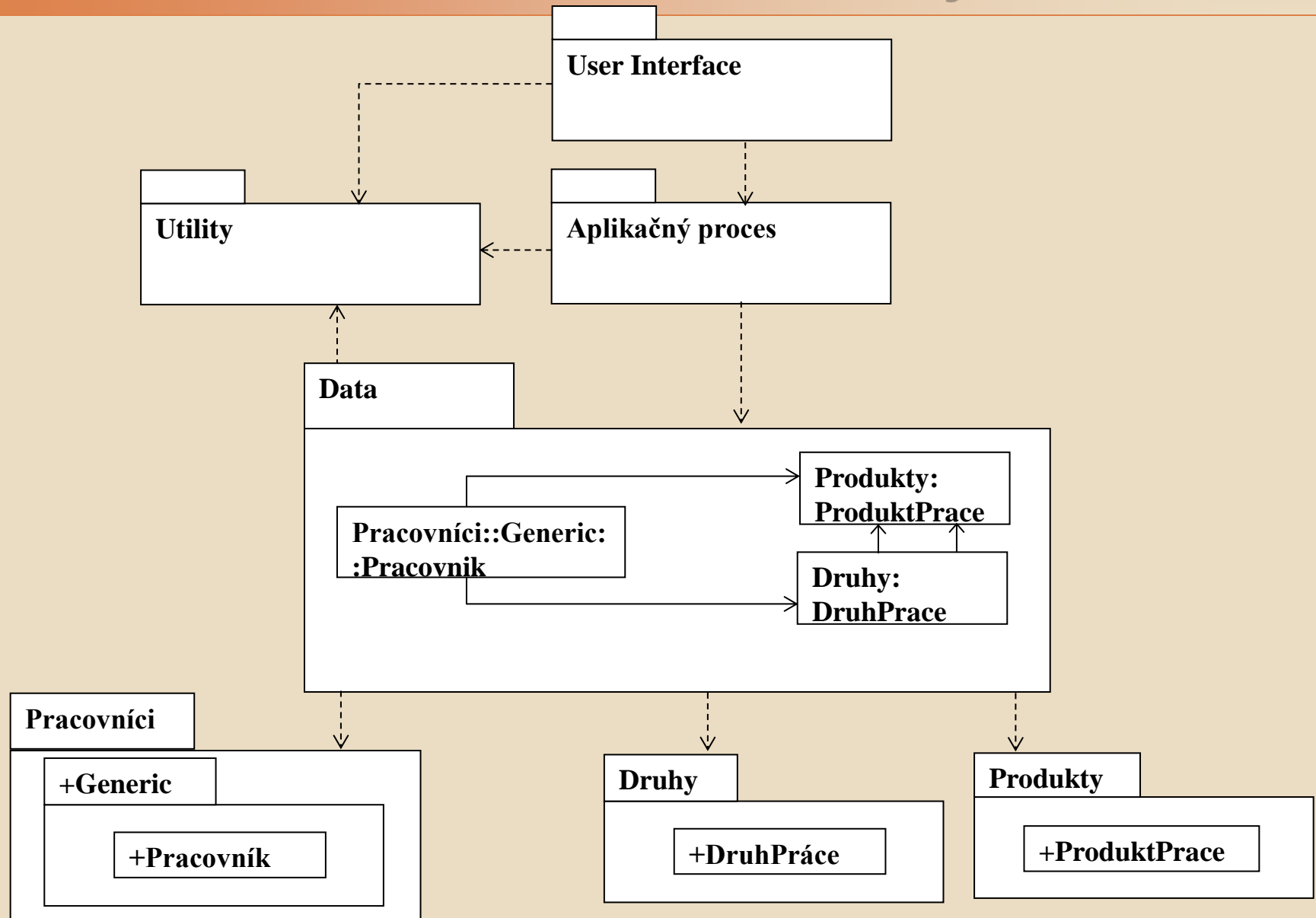
# Opakovanie-vzťah Závislosť (dependency)

- *vzťah dvoch prvkov, keď jeden prvok používa informáciu alebo službu iného prvku*
- *vždy jednosmerná*
- *Stereotypy - <<use>>, <<refine>>, <<realize>>, ...*





# Opakovanie – závislosti, balíčky



- ***Pojmová perspektíva***
  - Nezávislá na programovacom jazyku
  - Pojmy, pri štúdiu problému
- ***Špecifikačná perspektíva (špecifikácia)***
  - Zaoberá sa rozhraniami systému
  - Často prehliadaná v praxi
- ***Implementačná perspektíva (implementácia)***
  - Triedy a ich implementácia
  - Najčastejšie používaná

# Diagramy prípadov použitia – UCD

- **Popisujú funkčnosť systému, jeho použitie**
- **Prečo ?**
  - zapojiť používateľa do analýzy a návrhu systému
  - systém bude po ukončení prospešný pre tých, pre ktorých prospešný byť má
- **Kedy ?**
  - začína v počiatočných fázach projektu - počas činnosti zbierania požiadaviek na zachytenie požiadaviek, ktoré definujú čo má systém robiť
  - pokračuje počas celého procesu vývoja systému

# Diagramy prípadov použitia – UCD

## ➤ *Prvky*

- **aktor** alebo **aktér** (actor - participant)
- **use case** – prípad použitia

## ➤ *Vzt'ahy*

- medzi aktormi
- medzi prípadmi použitia
- navzájom

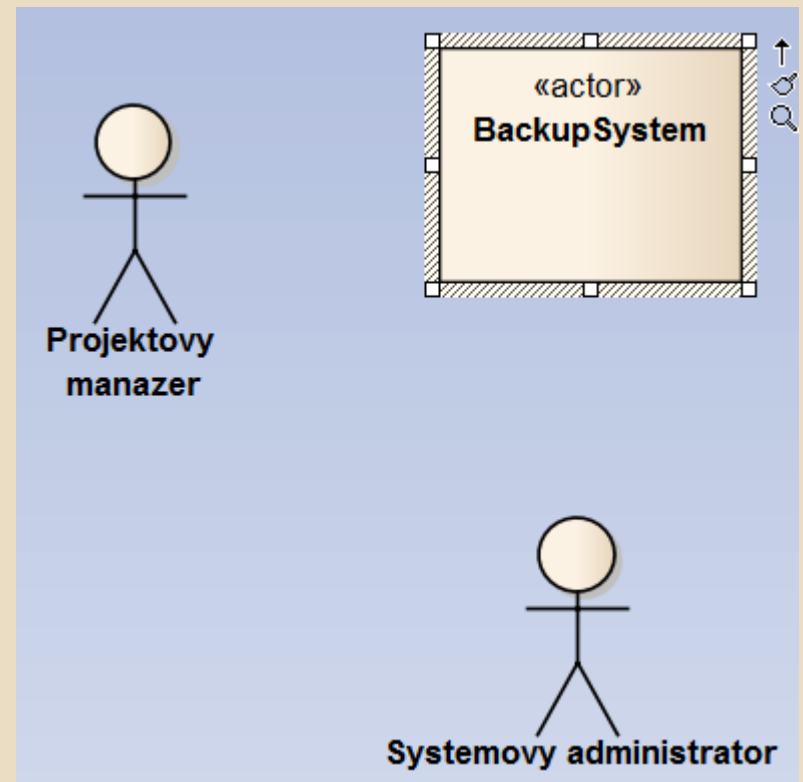
## ➤ UML – žiadny návod ako zachytiť požiadavky

# Aktor - Aktér

## ➤ *Rola používateľa systému*

- človek
- externý systém
- čas
- iný vonkajší prvok

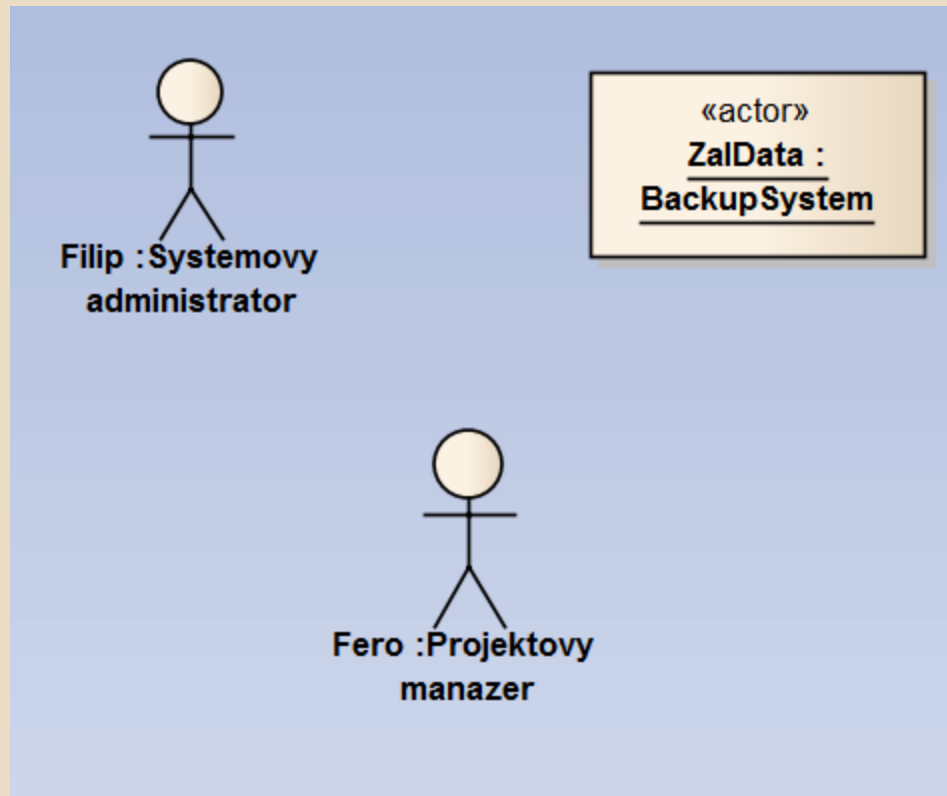
- *ikona figúry*
- *štvoruholník*



# Aktor – príklad

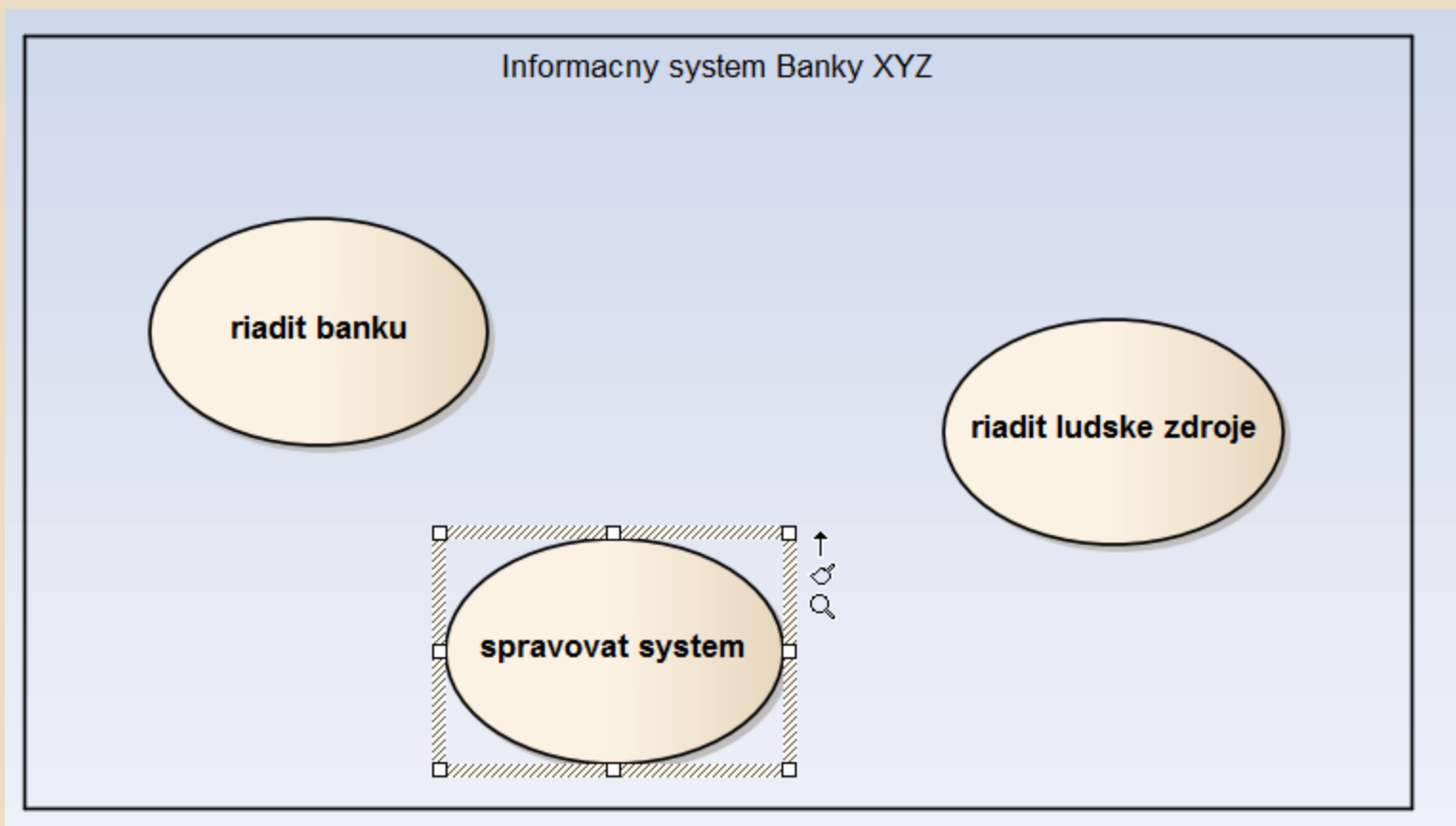
- **projektový manažér** – je zodpovedný za zaistenie kvality produktu v špecifikovanom čase a nákladoch pri špecifikovaných základných obmedzeniach
- **manažér zdrojov** – je zodpovedný za zaistenie toho, že tréňovanosť a zručnosť ľudských zdrojov je dostatočná pre daný projekt
- **ľudské zdroje** – je zodpovedný za zaistenie, že zručnosť pracovníkov je priebežne udržiavaná na požadovanej úrovni a že kvalita práce je dostatočná pre projekt
- **systémový administrátor** – je zodpovedný za zaistenie toho, že systém riadenia projektu bude prístupný pre projekt
- **zálohovací systém** – je zodpovedný za zálohovanie dát pre systém riadenia projektu

- *Inštancie aktora – konkrétny používateľ, systém*



# Prípad použitia – use case

- 1 use case = 1 funkčná požiadavka
- požiadavky zapísané ako postupnosť krokov





- ***Scenár***
  - konkrétne vykonávanie use case
  - nie je bežne zobrazovaný v use case diagramoch
  - slúži väčšinou na diskusiu
  
- ***Sekvencie správania(krokov)***
  - časť use case
  - postupnosť krokov

# Sekvencia správania

- *system ponúka používateľské rozhranie na získanie informácie o prihlásení*
- *používateľ zadá používateľské meno*
- *používateľ zadá heslo*
- *system skontroluje používateľské meno a heslo*
- *system primerane reaguje – buď dovolí používateľovi pokračovať alebo odmietne používateľa, záleží to od toho, či sú používateľské meno a heslo platné*

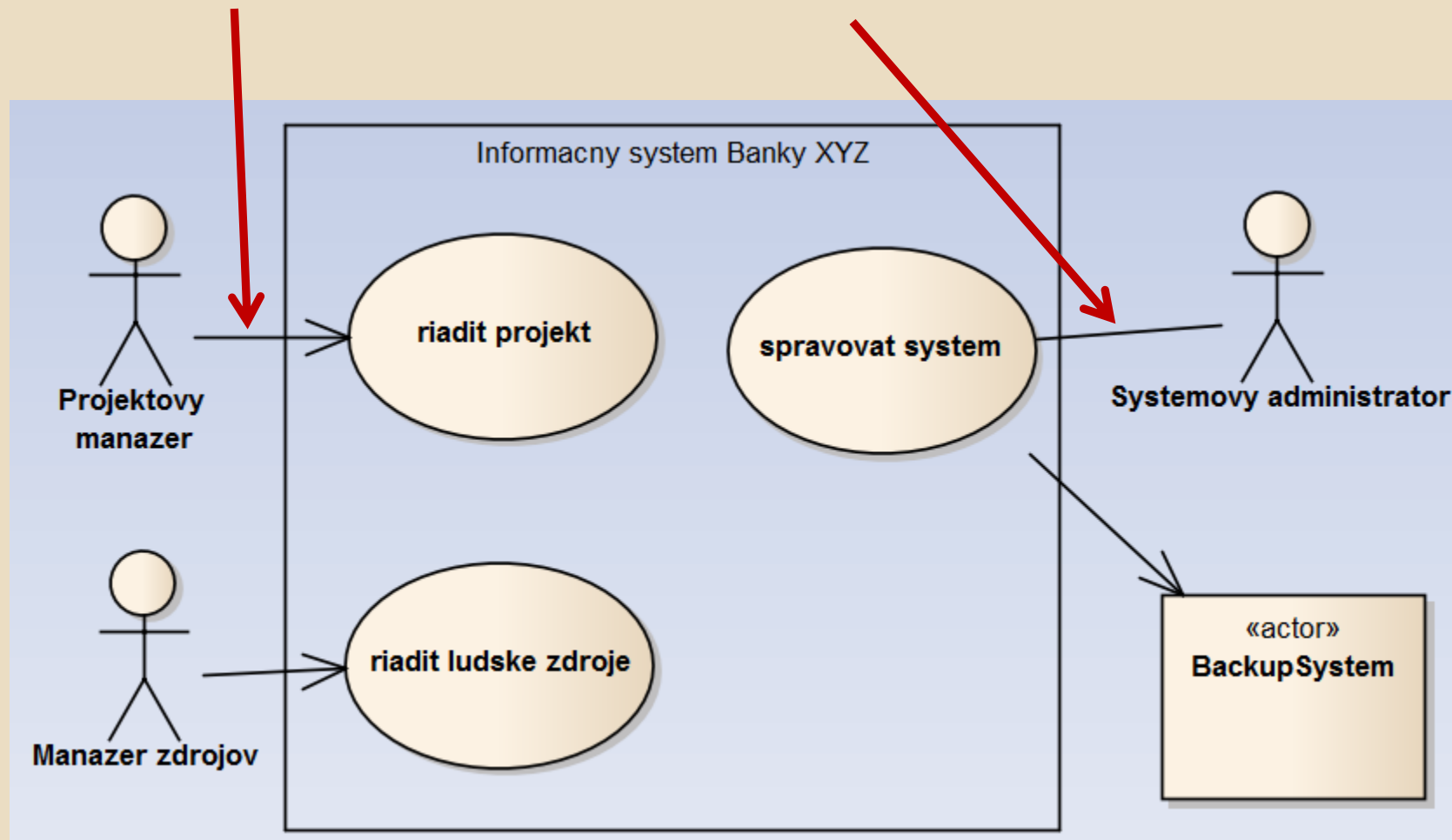
## ➤ Use case *Riadiť projekt*

1. pre riadenie projektu zahrňujúceho iba zamestnancov
2. pre riadenie projektu zahrňujúceho iba konzultantov
3. pre riadenie projektu zahrňujúceho aj zamestnancov aj konzultantov

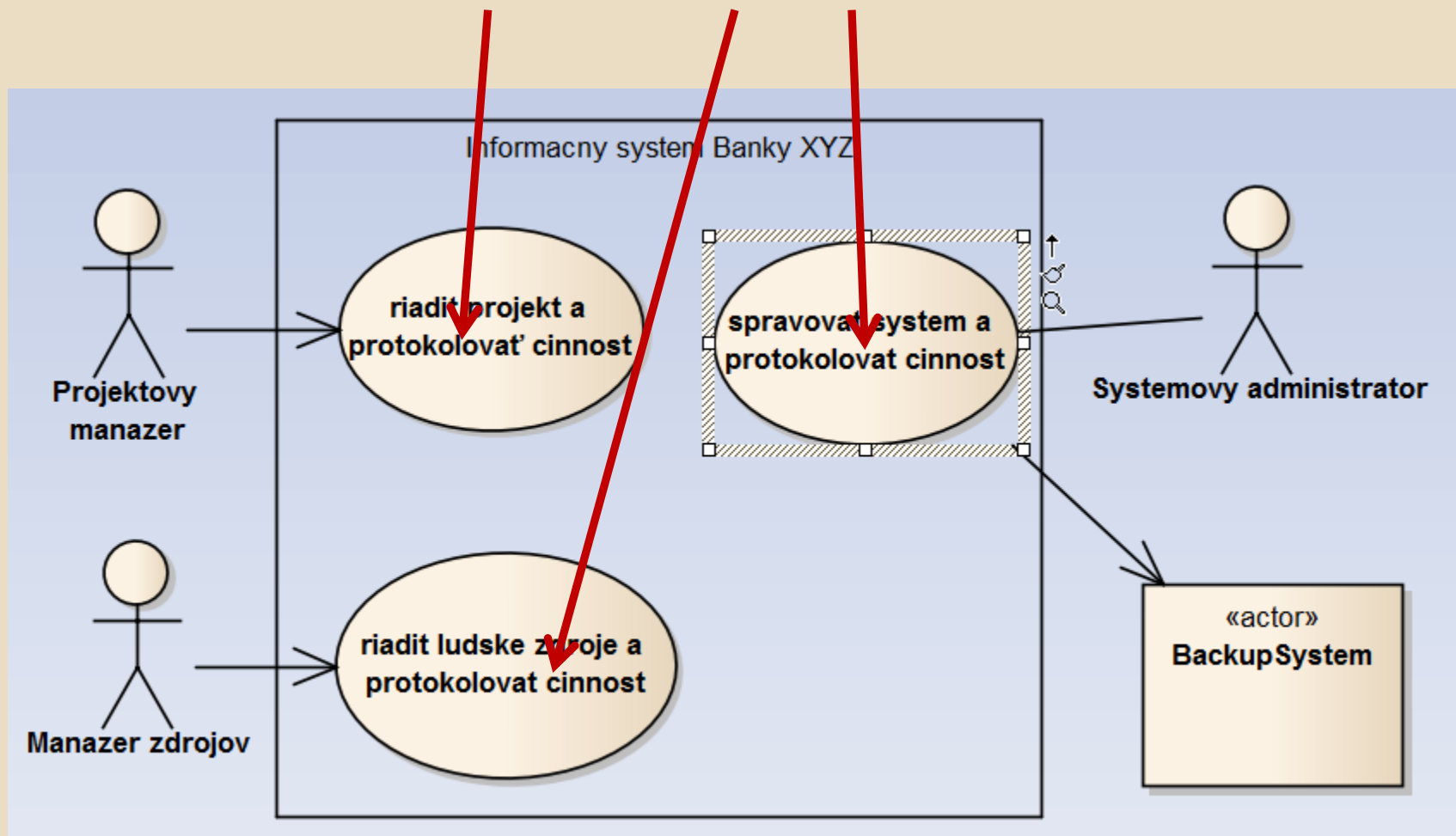
# Vzťahy v diagrame prípadov použitia

- ***Komunikačné asociácie***
- ***Vzťah vkladania – include***
- ***Vzťah rozšírenia – extend***
- ***Generalizácia***
  - Aktorov
  - Prípadov použitia

# Komunikačné asociácie



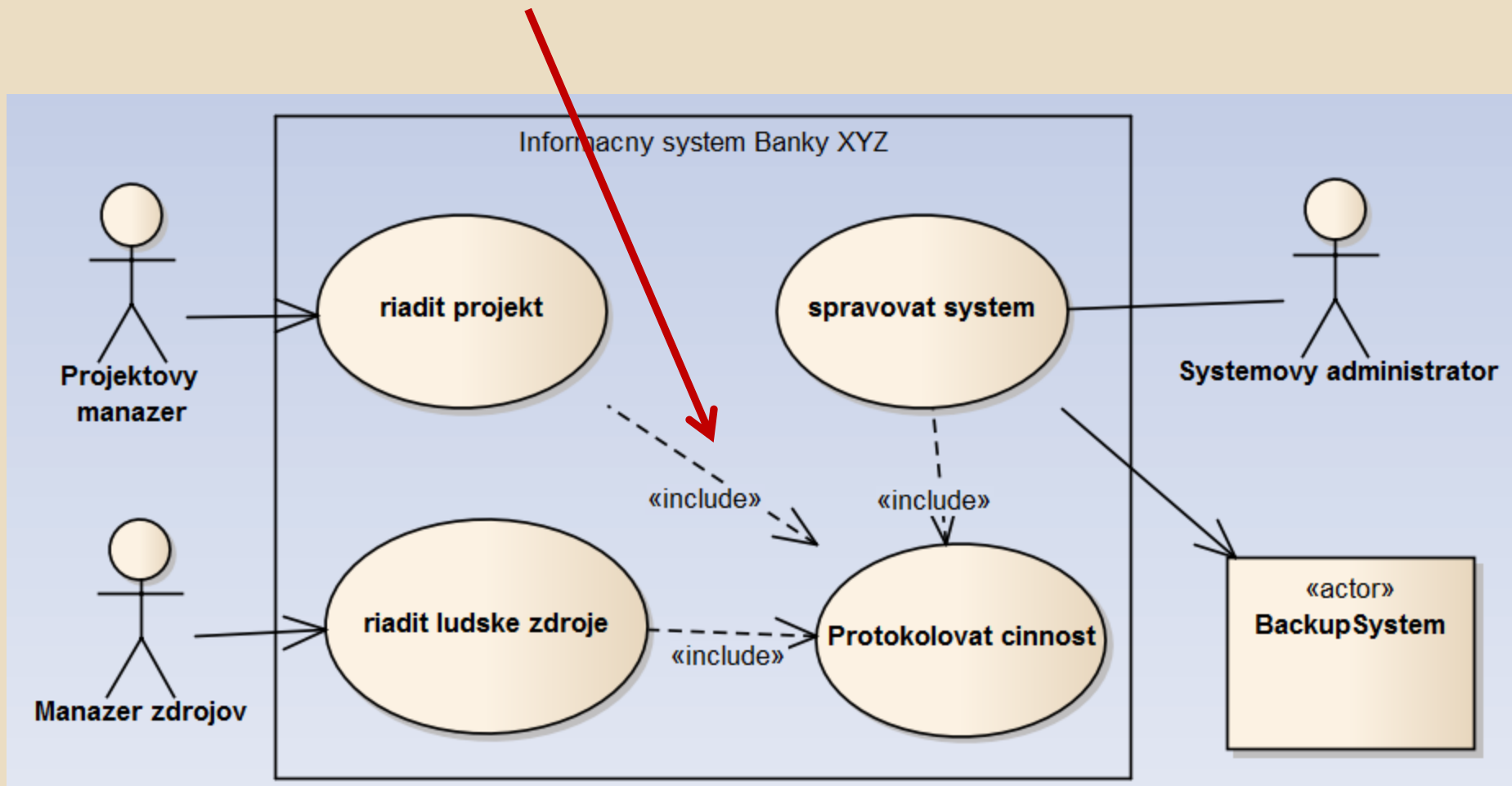
# Zahrňanie prípadov použitia



# Include závislosť

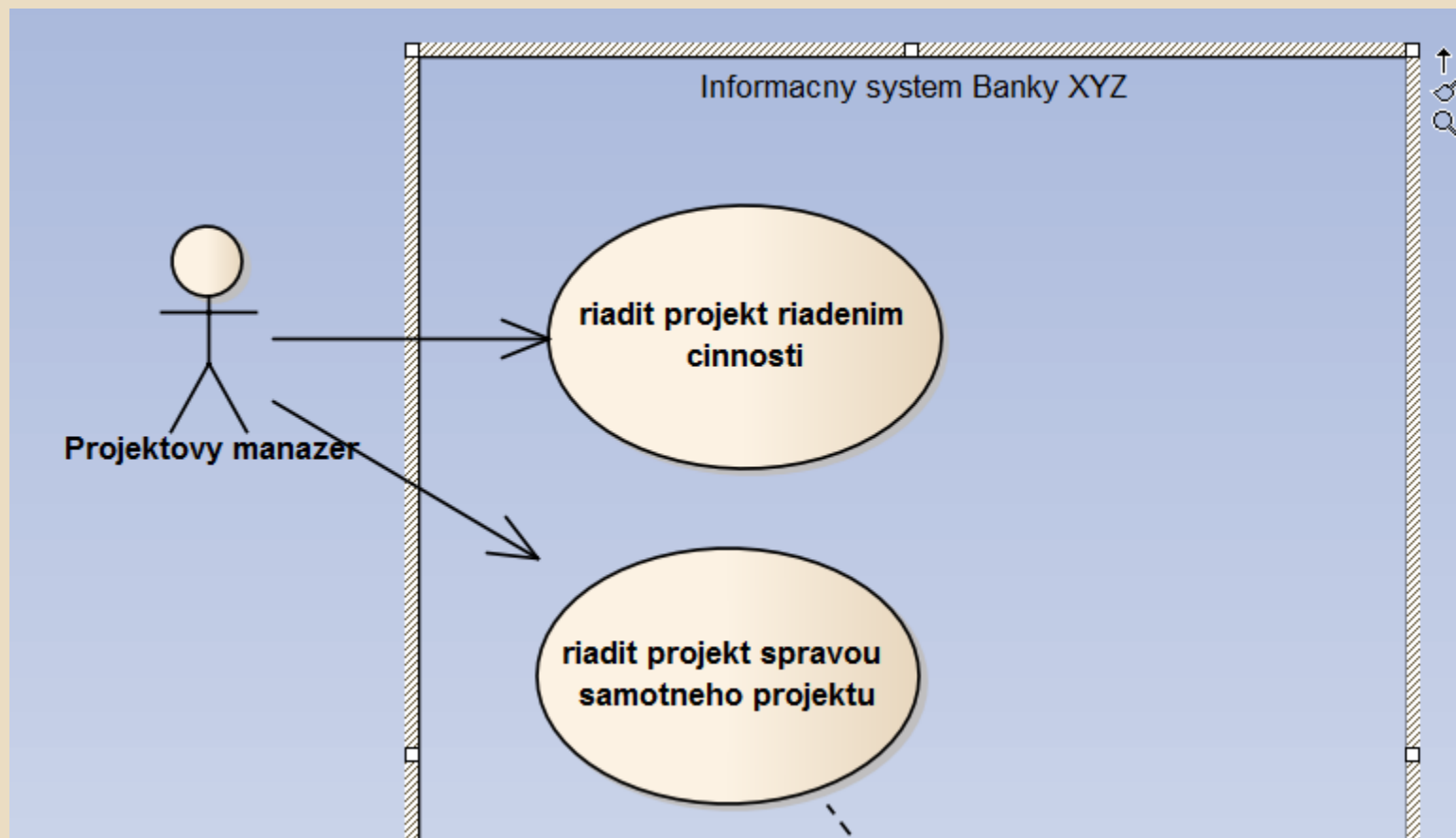
- *Zahrňňanie prípadov použitia*
- ***Základný (base) use case (Riadiť projekt)***
- ***Zahrnutý (inclusion) use case (Protokolovať činnosť)***
- *Prerušovaná čiara – šípka od základného k zahrnutému*
- *Stereotype*      ***<<include>>***

# Include závislosť





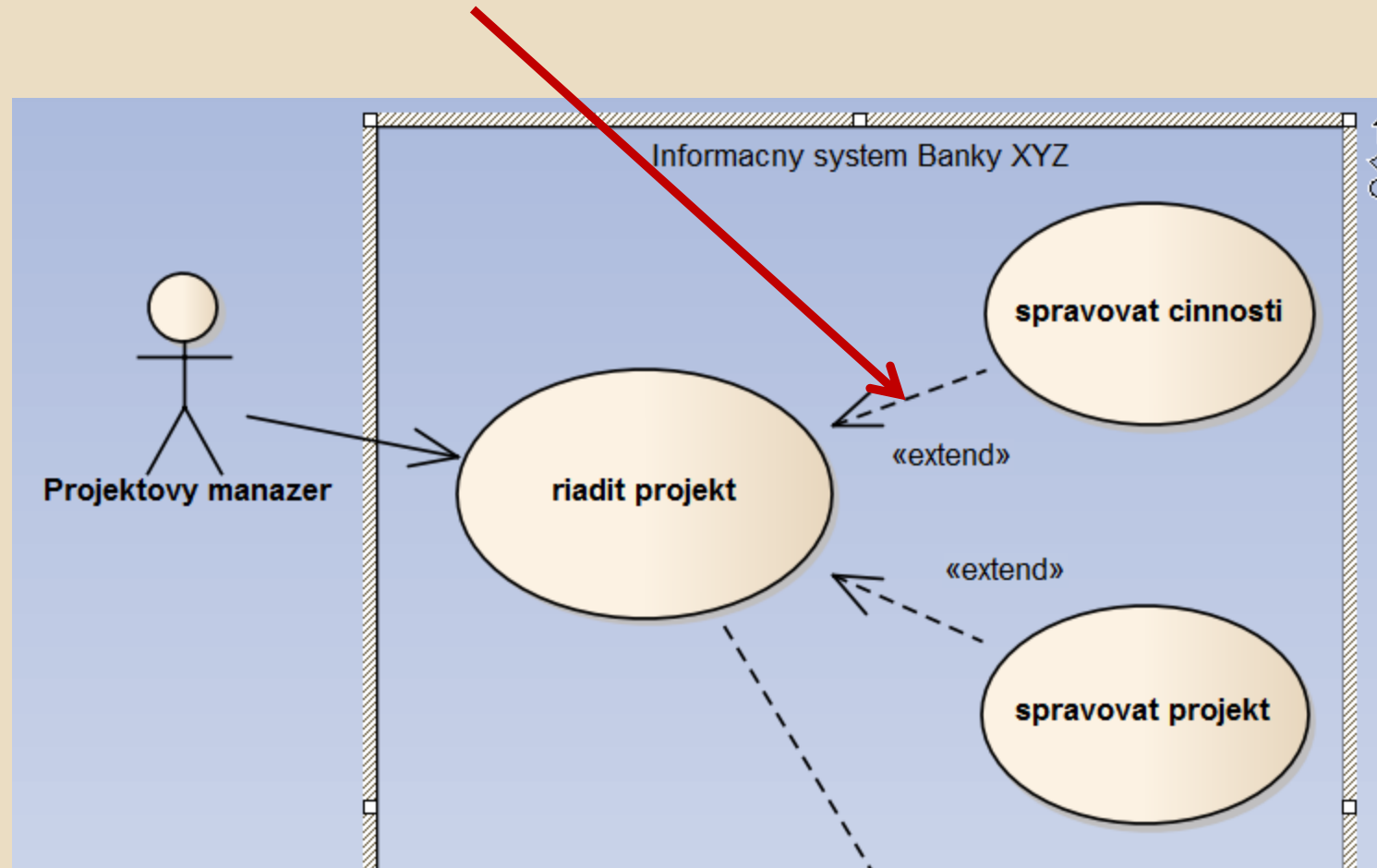
# Rozširovanie prípadov použitia



# Extend závislosť

- *Rozširovanie prípadov použitia*
- *Ďalší spôsob reuse pre use case*
- *Vzniká doplnením ďalších krokov*
- **Základný (base) use case** – *Riadiť projekt*
- **Rozšírený (extension) use case** – *Udržovať úlohy*
- *Prerušovaná čiara – šípka rozšíreného k základnému*
- *Stereotyp <<extend>>*

# Extend závislosť

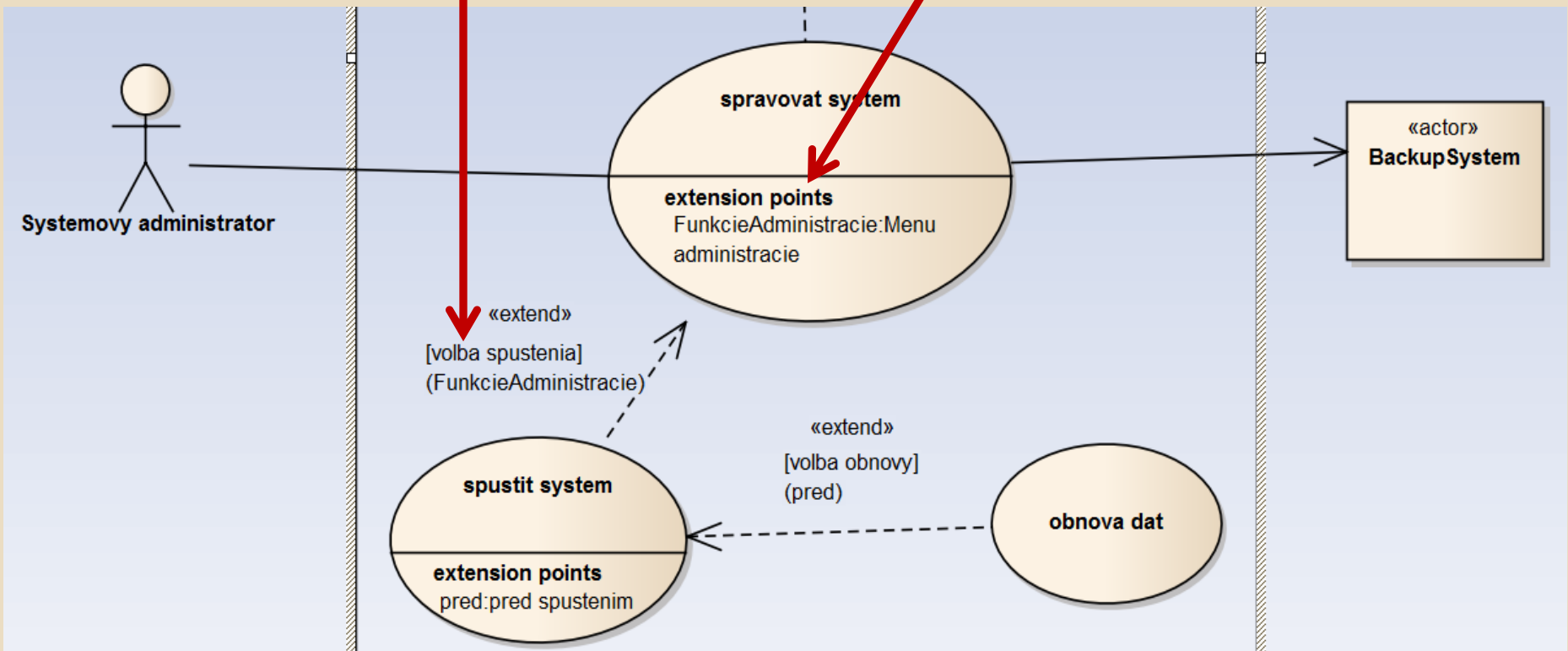


# Body rozšírenia

- *Extension points*
- Rozšírenie je možné urobiť len v týchto bodoch
- Aj niekoľko v jednom use case
- Nová časť v ikone pre use case
- Popis

# Body rozšírenia – extension points

- Definícia –
  - Použitie – [volba] (Meno)
- Meno : popis*

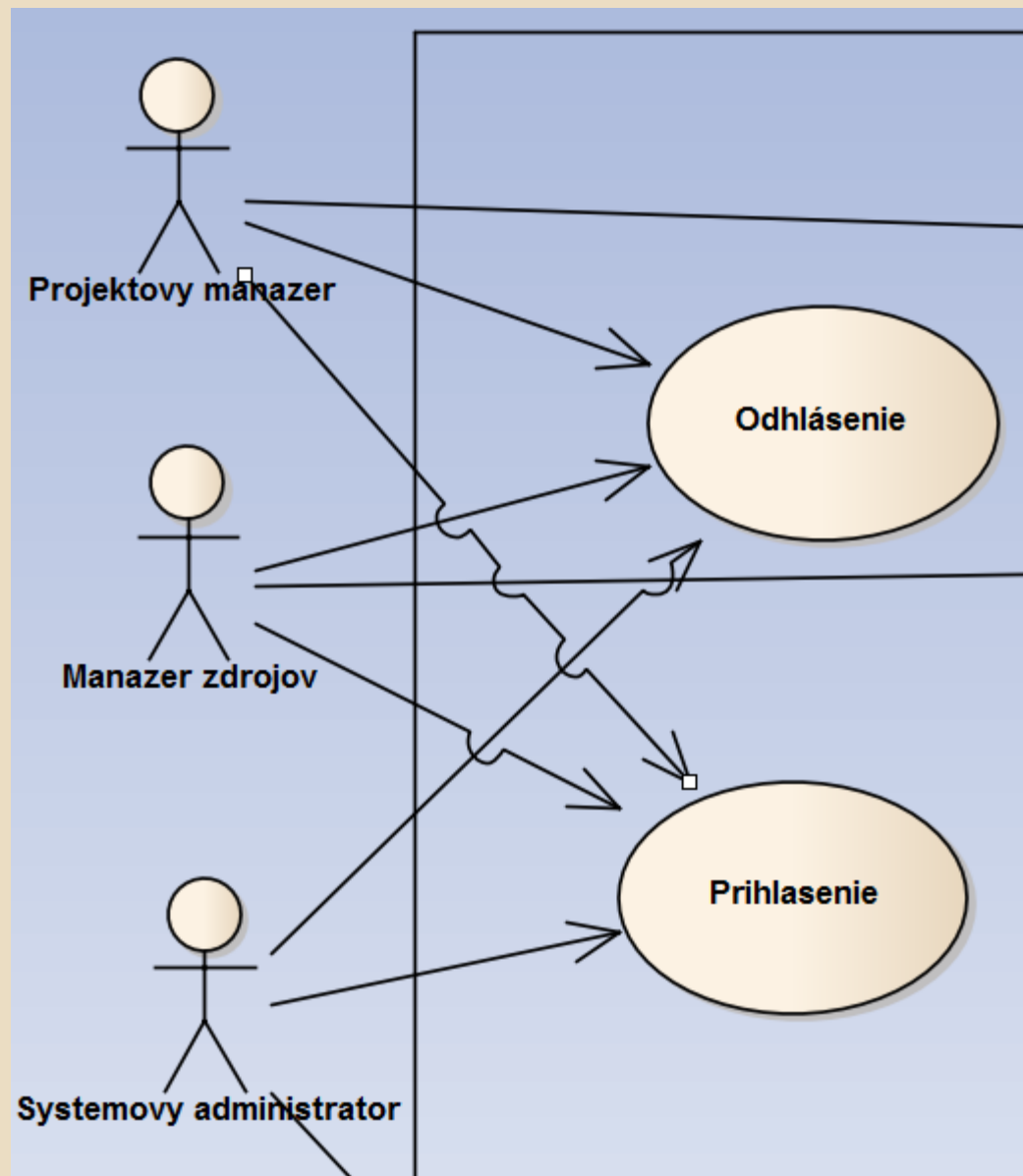


# Vzťah Generalizácia

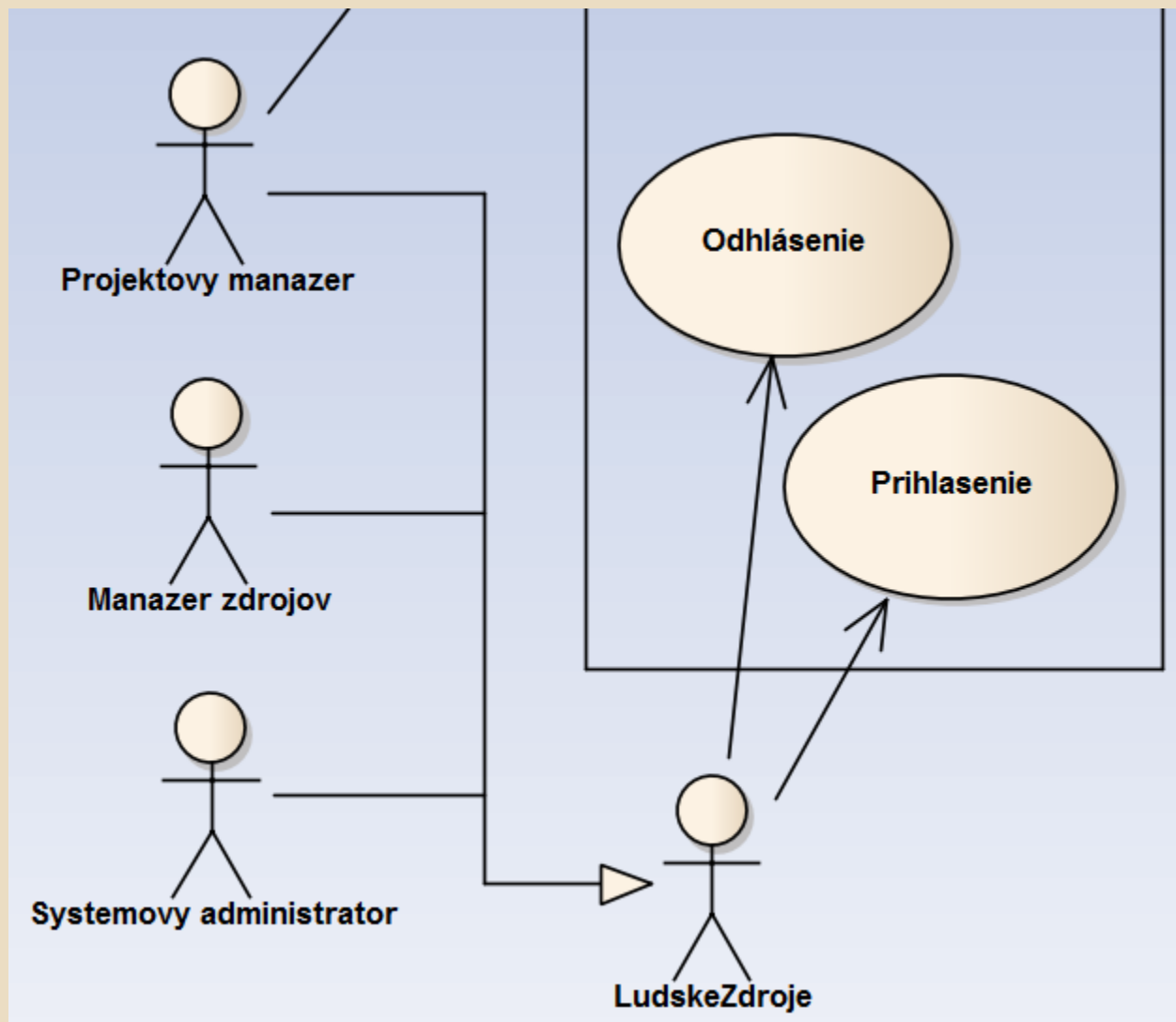
- vzťah medzi rovnakými prvkami
- **Aktorové** generalizácie
- Generalizácie **prípado**v použitia

# Aktorové generalizácie

## ➤ Podobnosť aktorov

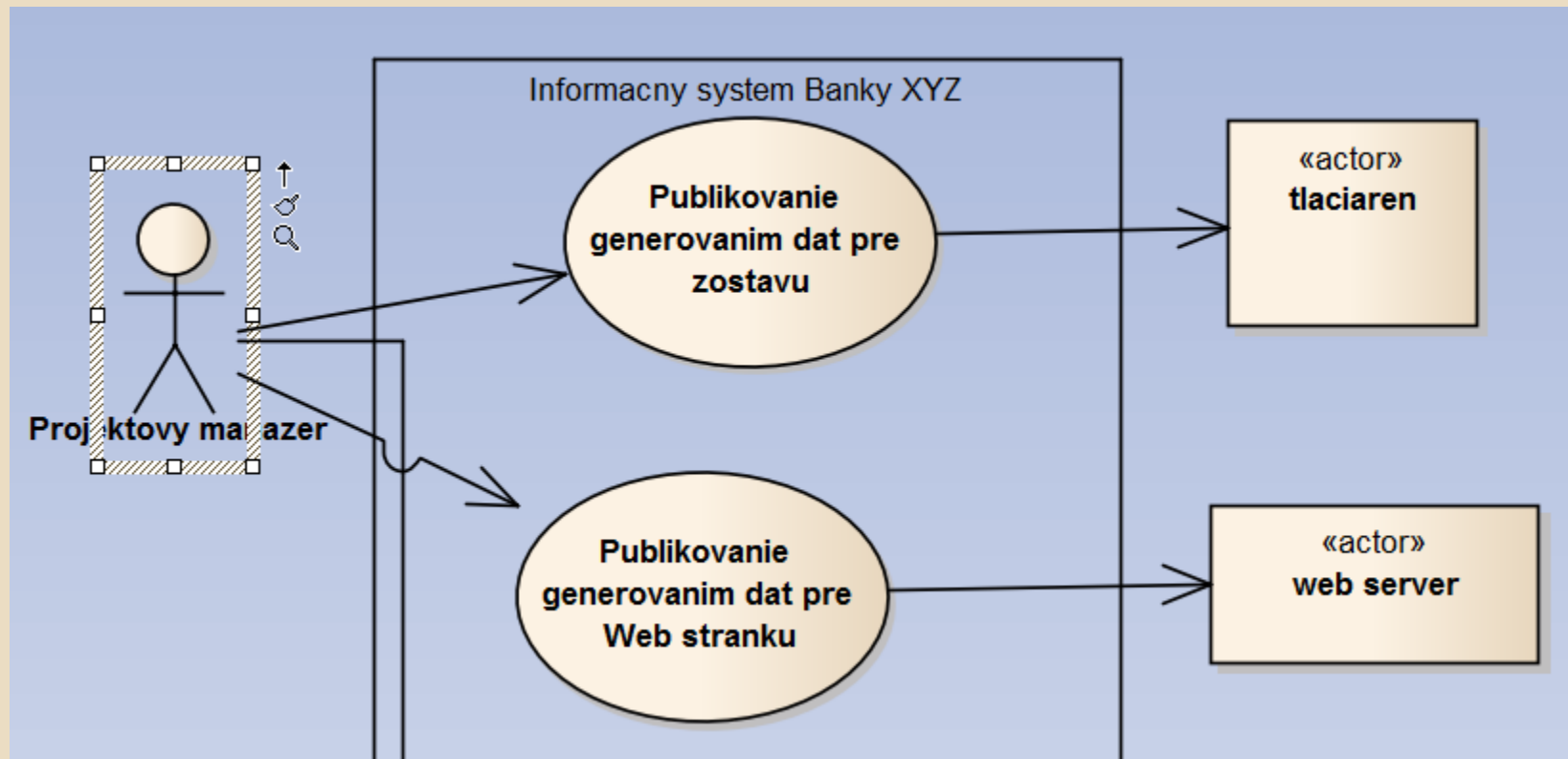


# Aktorové generalizácie

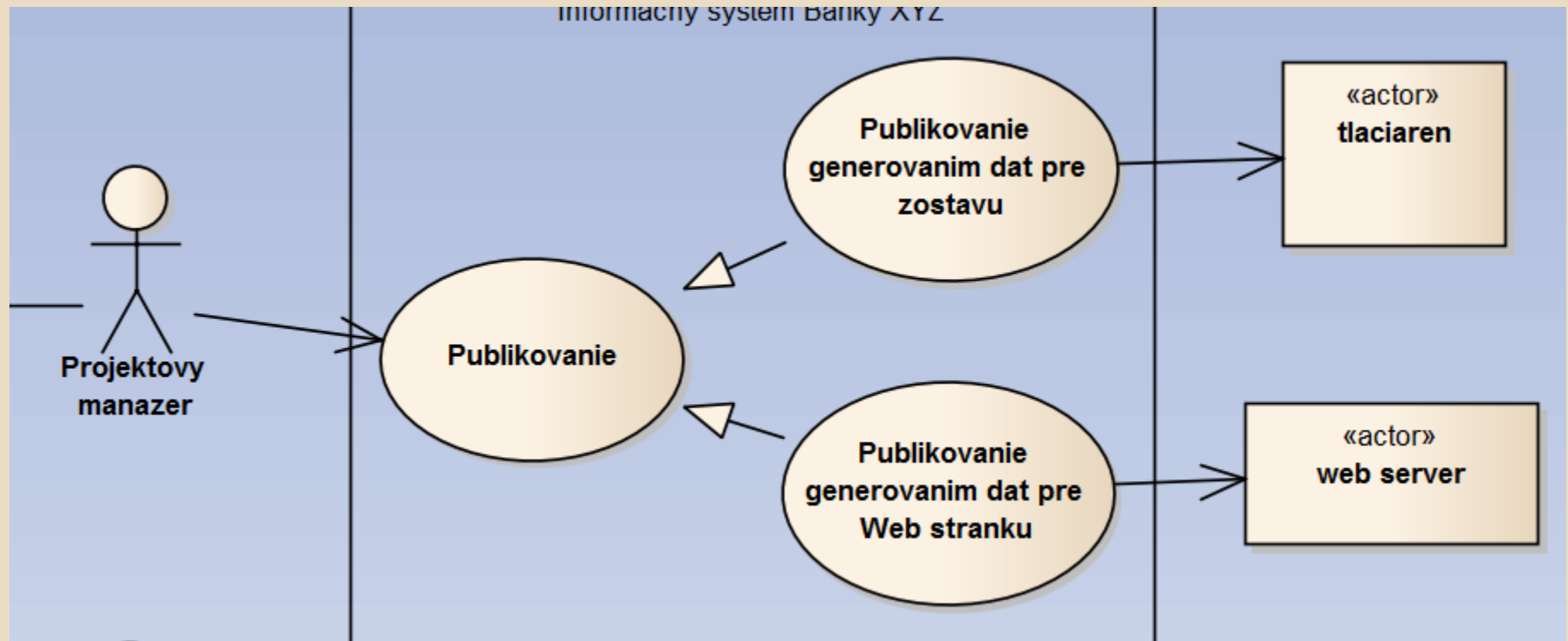




# Use case generalizácia



# Use case generalizácia



➤ *Generalizácia alebo extend ?*

# Hlavné zásady tvorby use case

- *Hlavný cieľ: nájsť všetky use case*
- *Postup zhora dolu*
- *Konzistencia modelu*
- *+ 1 nový use case = + množstvo ďalších*
- ***ZLATÉ** use case, obslužné use case*

# Hlavné zásady tvorby use case

- ***Aktéry nie sú najdôležitejší***
- ***Úplnosť modelu - najdôležitejšia***
  - Nechýba žiadny use case
  - Hocikto na základe use case pochopí a dokáže pokračovať
- ***Chyby pri tvorbe use case***
  - Málo informácií, príliš veľa informácií
  - Analytik viazaný na implementáciu
  - Časová závislosť

*Otázky ?*

*Ďakujem za pozornosť*