

# Návrhové třídy a přiřazení zodpovědností

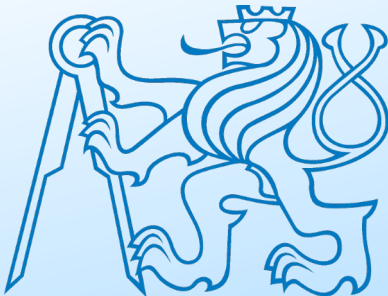
Ing. Jiří Mlejnek

Katedra softwarového inženýrství  
Fakulta informačních technologií  
České vysoké učení technické v Praze

© Jiří Mlejnek, 2011

[jiri.mlejnek@fit.cvut.cz](mailto:jiri.mlejnek@fit.cvut.cz)

Softwarové inženýrství BI-SI1  
ZS 2015/ Před. 6



Evropský sociální fond  
Praha & EU: Investujeme do vaší budoucnosti



# Obsah

- Návrhový model tříd
- Spolupráce objektů
- Databázový model
- Vzory GRASP

# Návrhový model tříd

- Cílem je správně přiřadit zodpovědnosti třídám
  - Navrhnout systém tak, aby byl dlouhodobě udržitelný a rozšiřitelný
    - Rozšíření bude zřejmě provádět jiný tým
    - Není možné se spolehnout na znalosti autorů
  - Snadno lokalizovatelné dopady změn
    - Snadné odhady pracnosti požadovaných rozšíření
  - Minimalizace dopadů při provádění změn
    - Změna v jedné části systému by neměla ovlivňovat jinou část

# Návrhový model tříd

- Použití
  - Dokumentace použitých principů a vzorů
  - Generování zdrojových kódů

# Návrhový model tříd

- Závislý na:
  - Zvolené technologii
  - Zvolené architektuře
  - Použitých vzorech
    - Architektonické vzory:
      - Fowler, Martin. Patterns of Enterprise Application Architecture
    - GRASP vzory – na konci přednášky
    - GoF vzory – příští přednáška

# Návrhový model tříd

- Může vycházet z doménového modelu
  - Atributy – upřesnění datových typů atributů
  - Upřesnění relací
    - Určení směru asociace
    - Názvy konců asociací
    - **Nové softwarové třídy**

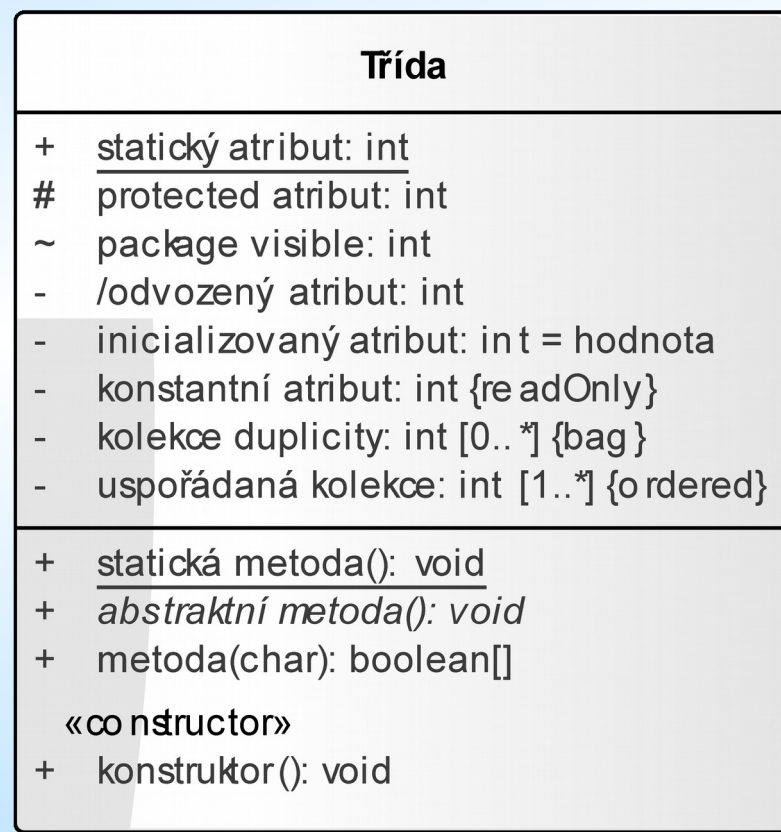
# Návrhový model tříd

- Přiřazení zodpovědností
  - Realizace scénářů případů užití
  - Přiřazení metod jednotlivým třídám
    - Vstupní / výstupní parametry
    - Viditelnost
    - Datové typy
  - Jak to udělat správně?

# Návrhový model tříd

- Diagram tříd –  
pokročilá notace
  - Statické
  - Abstraktní
  - Viditelnost
  - Konstruktor
  - Inicializace
  - Kolekce
  - Závislosti

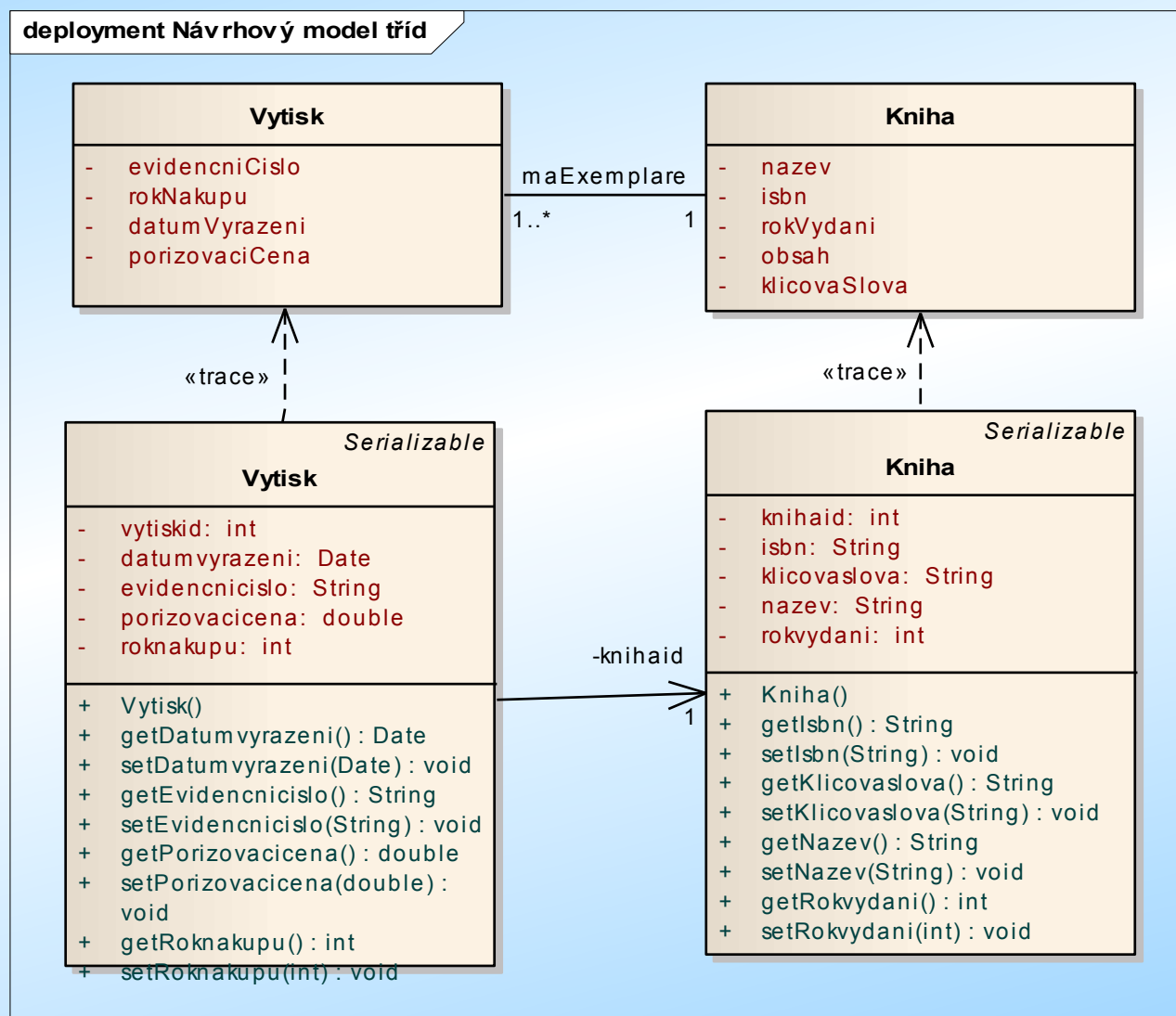
class Návrhový model tříd



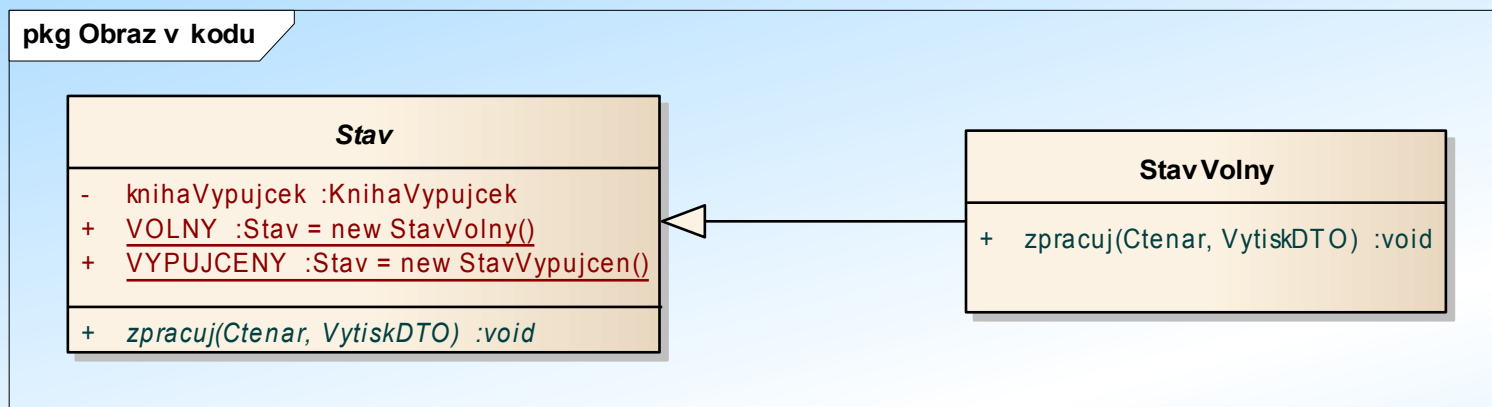


# Návrhový model tříd

- Java



# Návrhový model tříd

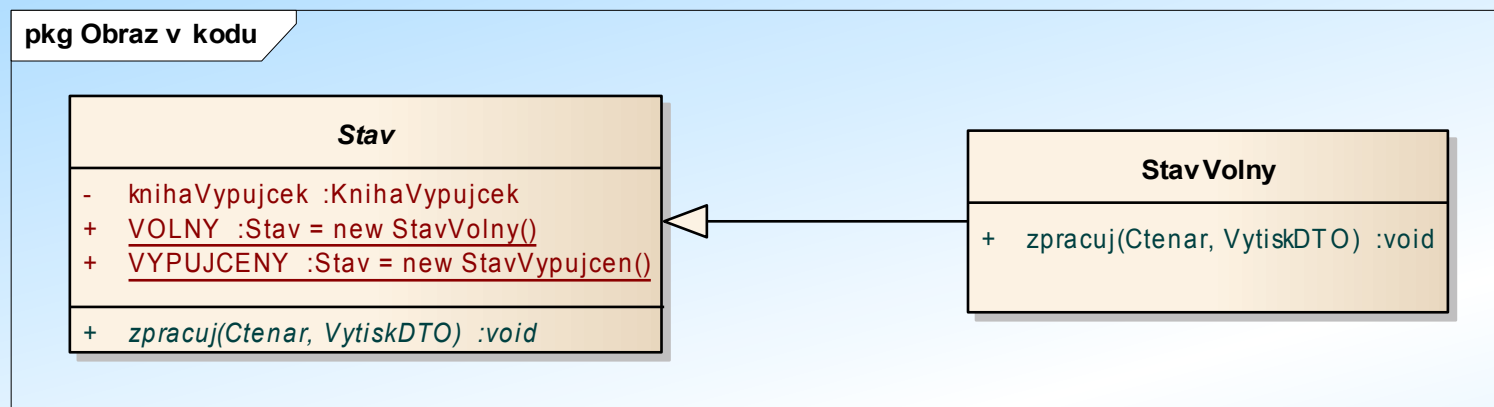


```
public abstract class Stav {
    private KnihaVypujcek knihaVypujcek;
    private Stav VOLNY = new StavVolny();
    public static Stav VYPUJCENY = new StavVypujcen();

    public abstract void zpracuj(Ctenar ctenar, VytiskDTO vytisk);
}
```

```
public class StavVolny extends Stav {
    public void zpracuj(Ctenar ctenar, VytiskDTO vytisk){
        ...
    }
}
```

# Návrhový model tříd



```
public abstract class Stav {
    private KnihaVypujcek knihaVypujcek;
    private public static Stav VOLNY = new StavVolny();
    public static Stav VYPUJCENY = new StavVypujcen();

    public abstract void zpracuj(Ctenar ctenar, VytiskDTO vytisk);
}
```

```
public class StavVolny extends Stav {
    public void zpracuj(Ctenar ctenar, VytiskDTO vytisk){
        ...
    }
}
```

# Návrhový model tříd

- Doporučení - důležitá je přehlednost a přínos
  - Nezobrazovat názvy asociací
  - Nezobrazovat get a set metody
  - Většinou není nutné zachytit všechny softwarové třídy
    - Důležité je popsat principy a pravidla, která mají být během implementace dodržována
    - Tyto principy lze zobecňovat i na další realizované funkčnosti

# Návrhový model tříd

Dotazy?

# Spolupráce objektů

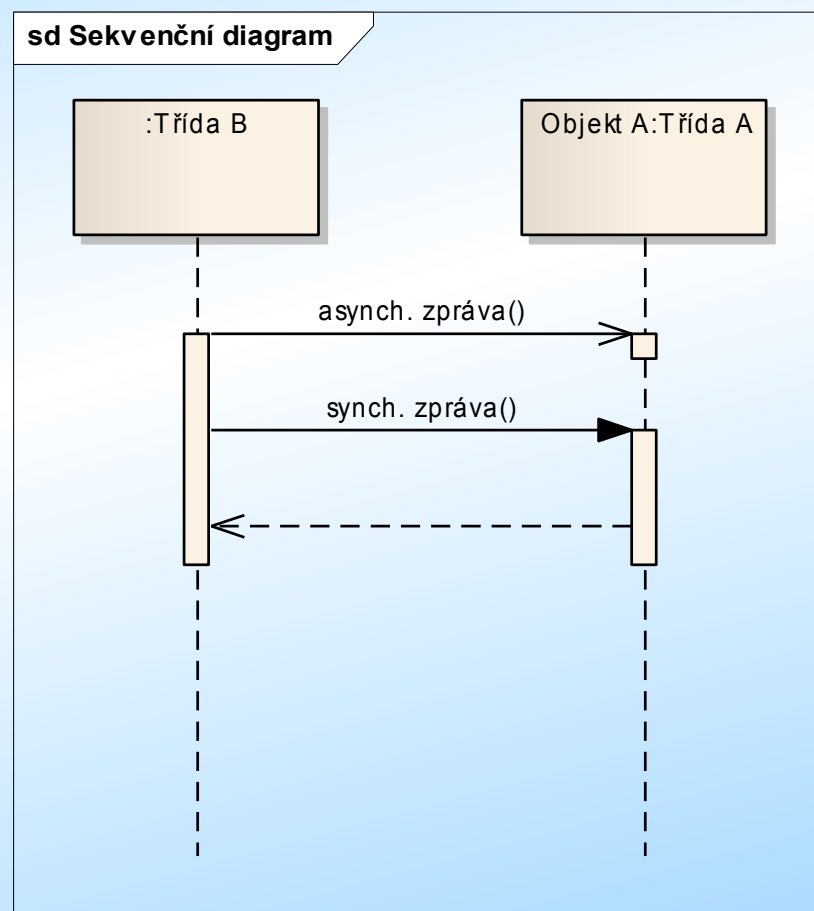
- Popis realizace scénářů případů užití
- Zachycení komunikace spolupracujících objektů (instancí návrhových tříd)
- Přiřazení zodpovědnosti třídám

# Spolupráce objektů

- Notace – interakční diagramy UML
  - Sekvenční diagram
  - Diagram komunikace
- EA: UML Behavioral - Sequence

# Spolupráce objektů

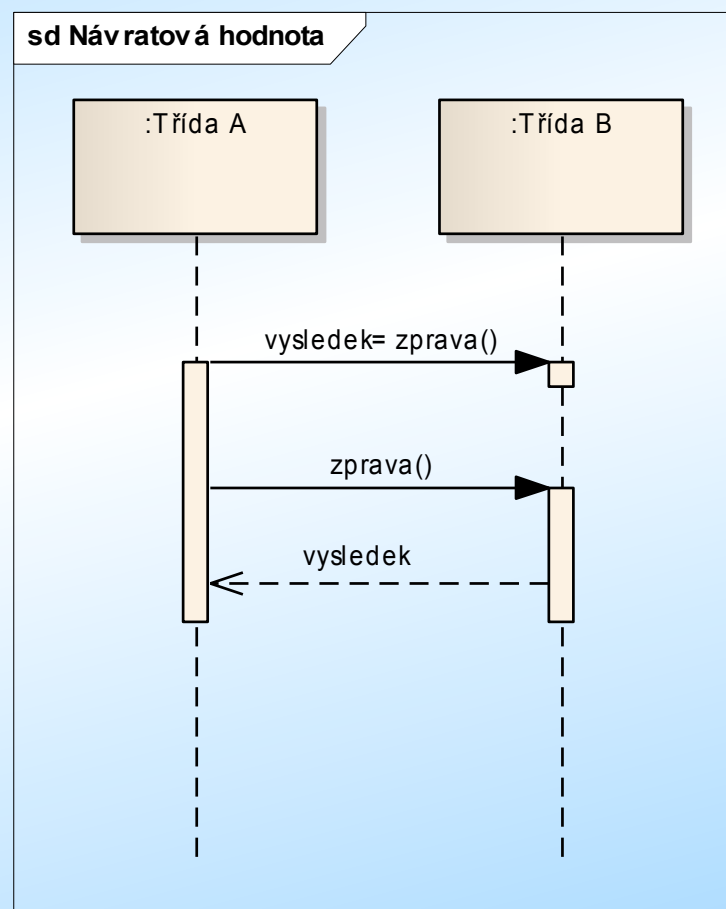
- Sekvenční diagram
  - Objekt
    - Pojmenovaný
    - Nepojmenovaný
  - Třída
    - Statická metoda
  - Zpráva
    - Asynchronní
    - Synchronní





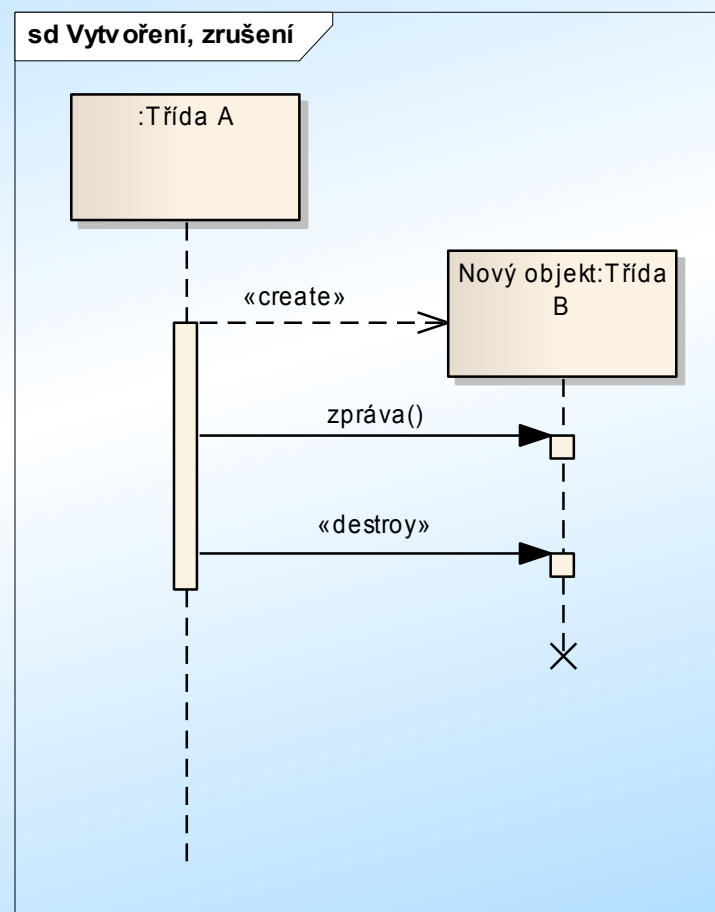
# Spolupráce objektů

- Návratová hodnota
  - 2 různé způsoby



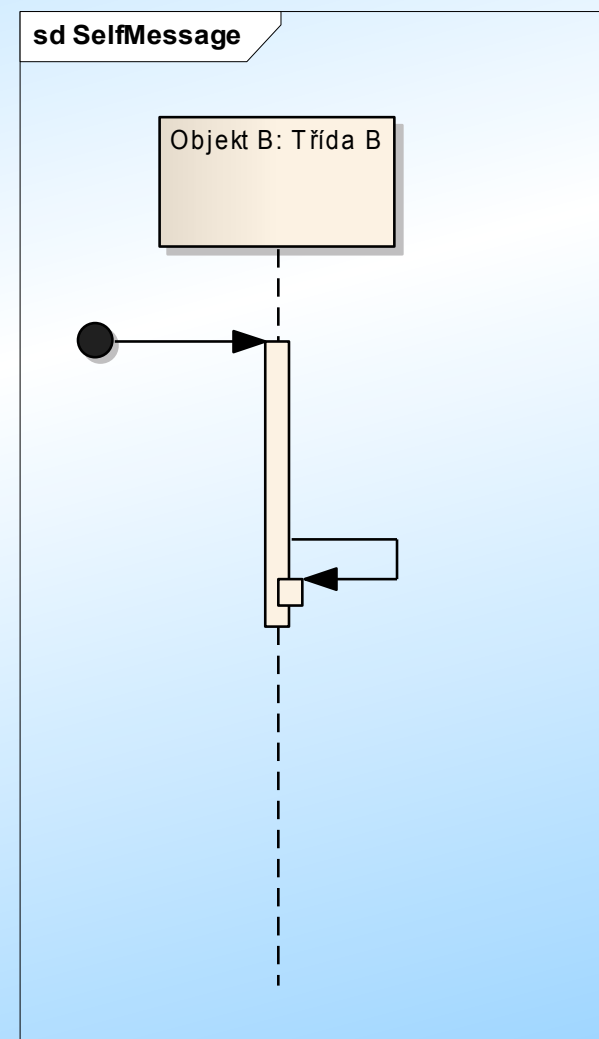
# Spolupráce objektů

- Vytvoření objektu
- Zrušení objektu



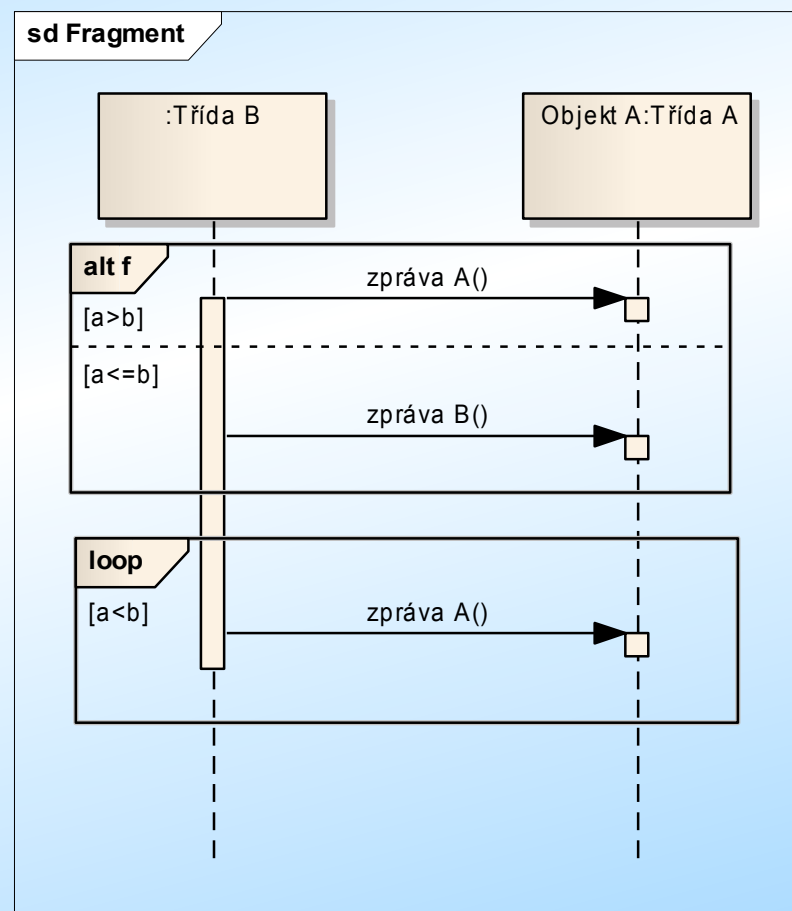
# Spolupráce objektů

- Sekvenční diagram
  - Nalezená zpráva (Found Message/Endpoint)
  - Zaslání zprávy sám sobě (Self Message)



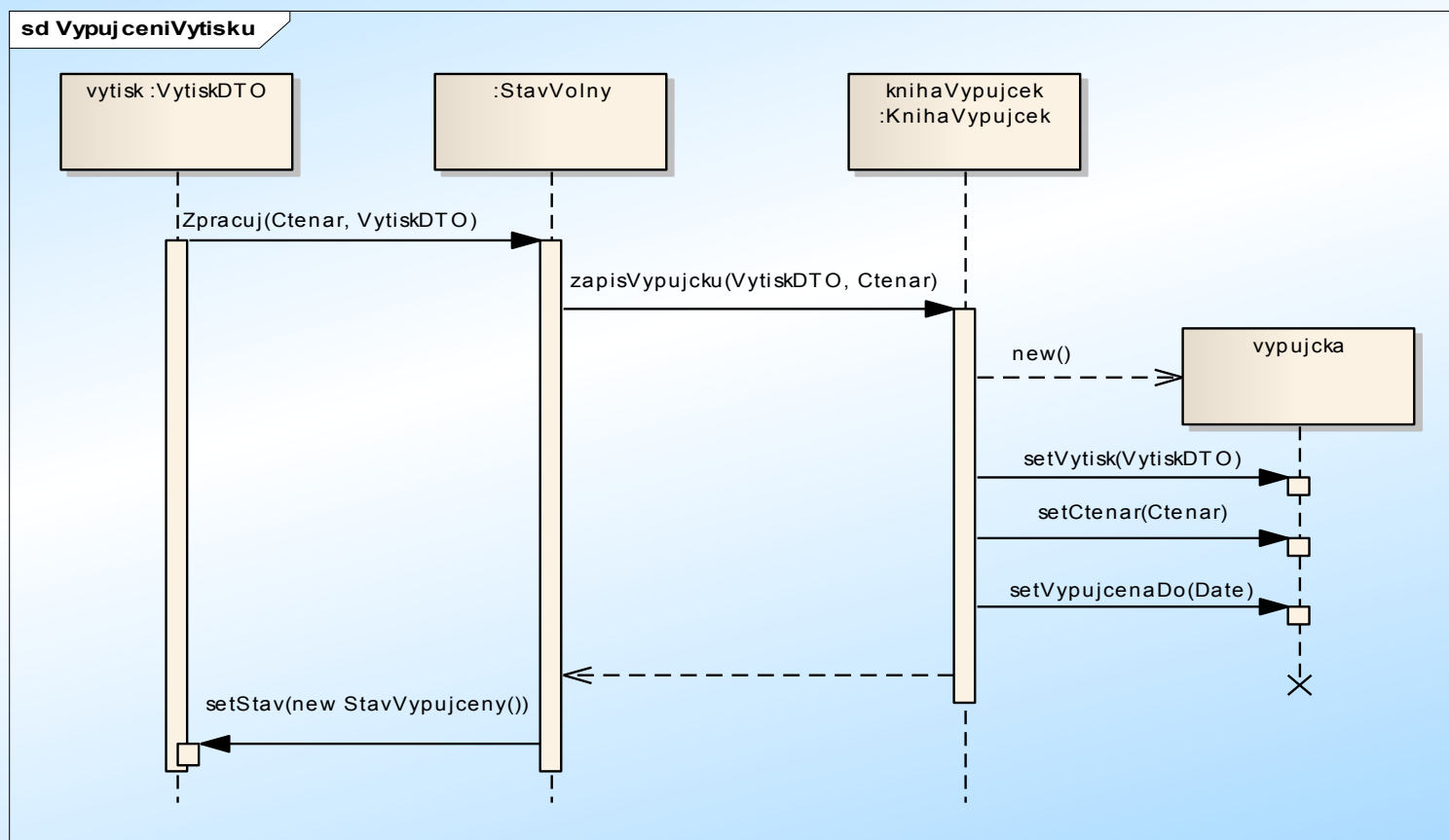
# Spolupráce objektů

- Sekvenční diagram
  - Fragment
    - Větvení
    - Cyklus
    - Další (opt, par)



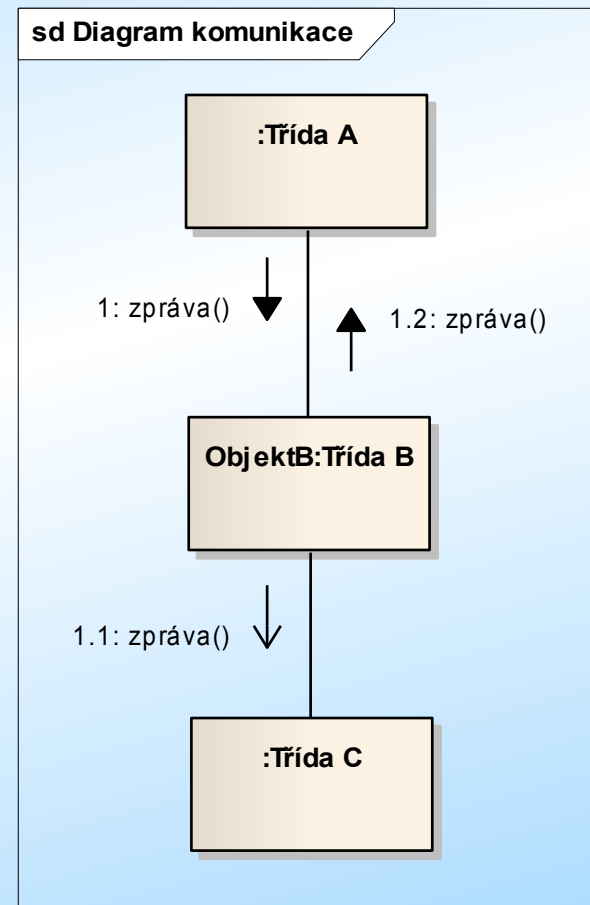
# Spolupráce objektů

- Sekvenční diagram



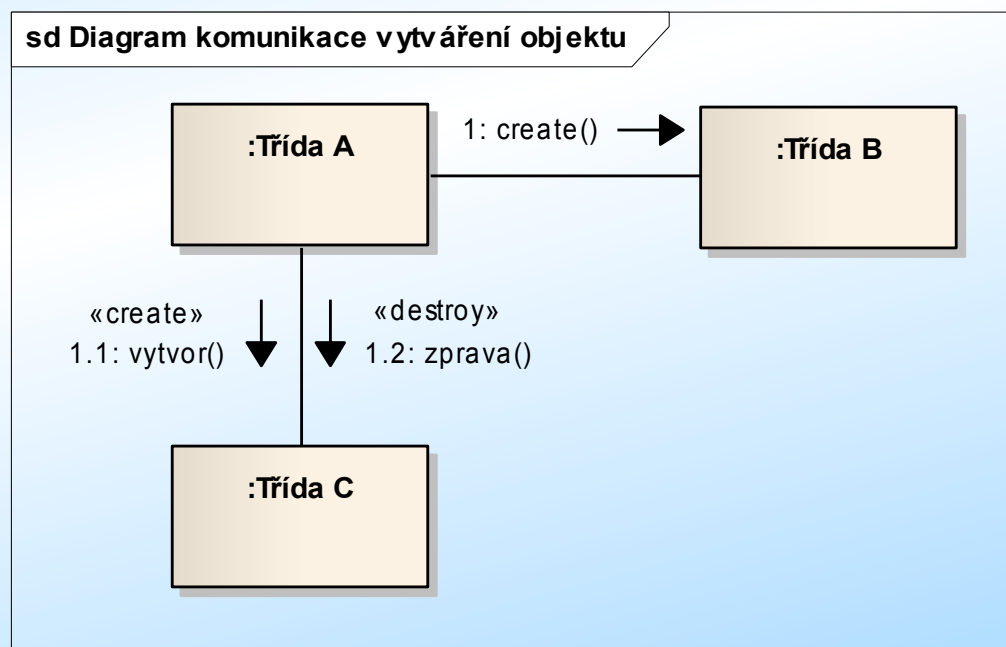
# Spolupráce objektů

- Diagram komunikace
  - Objekt
    - Pojmenovaný
    - Nepojmenovaný
  - Třída
    - Statická metoda
  - Zpráva
    - Synchronní
    - Asynchronní
- EA: UML Behavioral - Communication



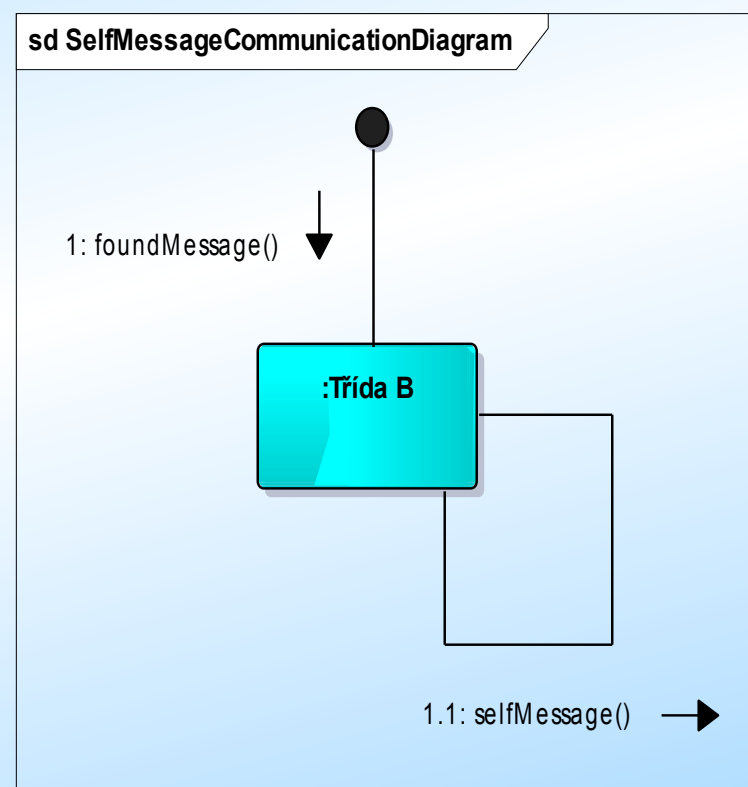
# Spolupráce objektů

- Diagram komunikace
  - Vytvoření objektu
  - Zrušení objektu



# Spolupráce objektů

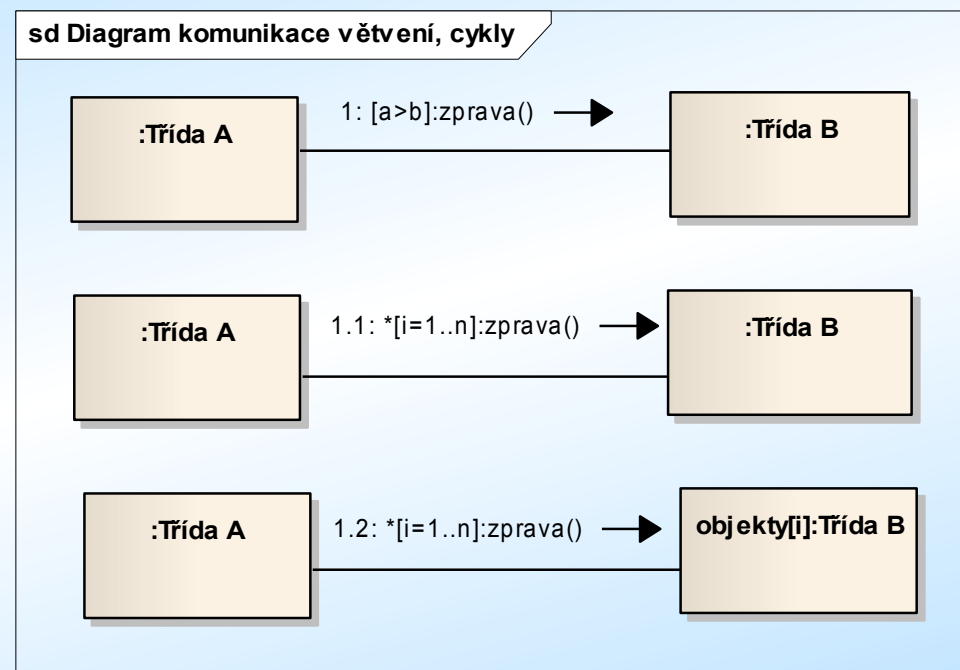
- Diagram komunikace
  - Nalezená zpráva (Found Message/Endpoint)
  - Zaslání zprávy sám sobě (Self Message)





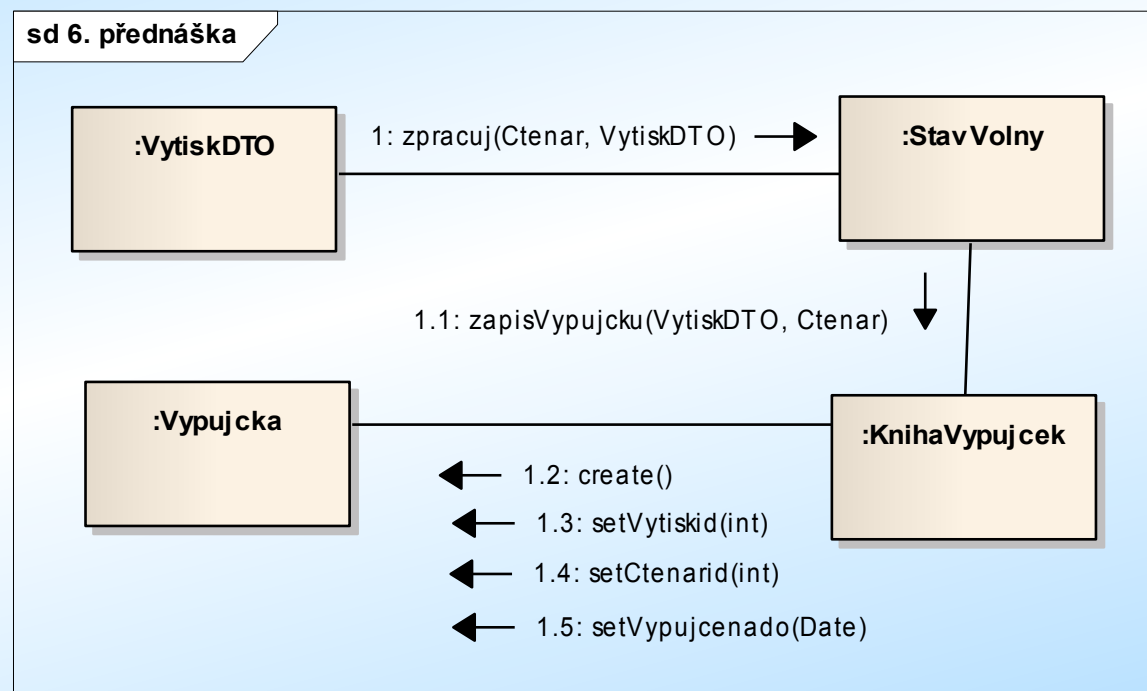
# Spolupráce objektů

- Diagram komunikace
  - Podmínky
  - Cykly
  - Iterace přes prvky kolekce



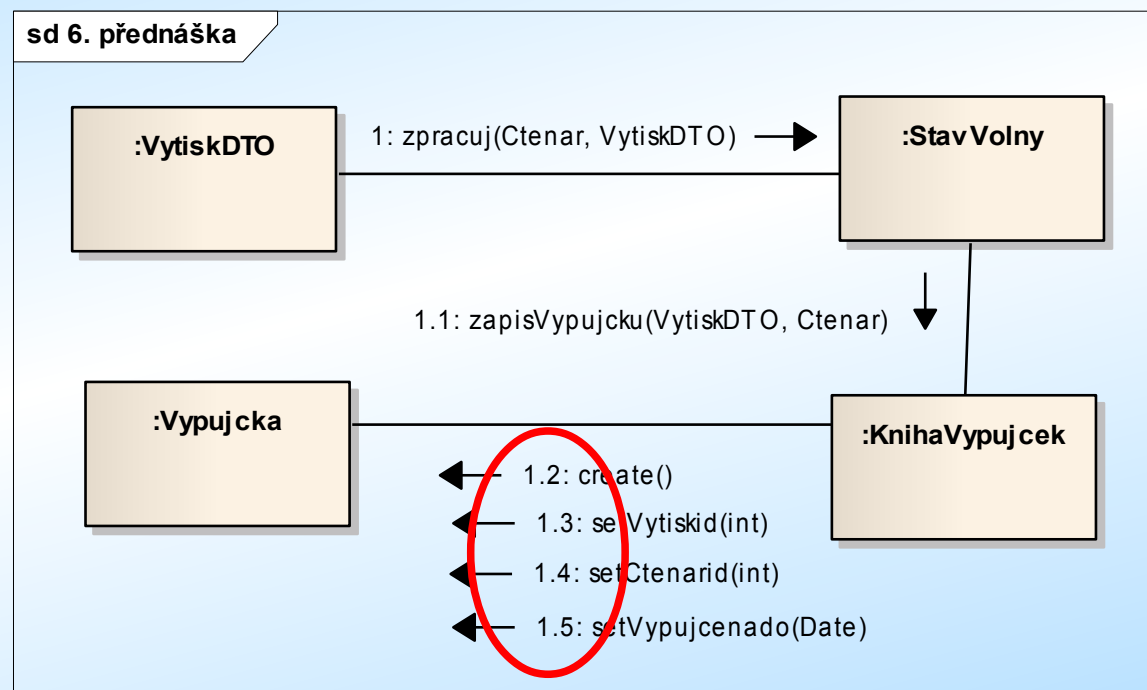
# Spolupráce objektů

- Diagram komunikace – co je špatně?



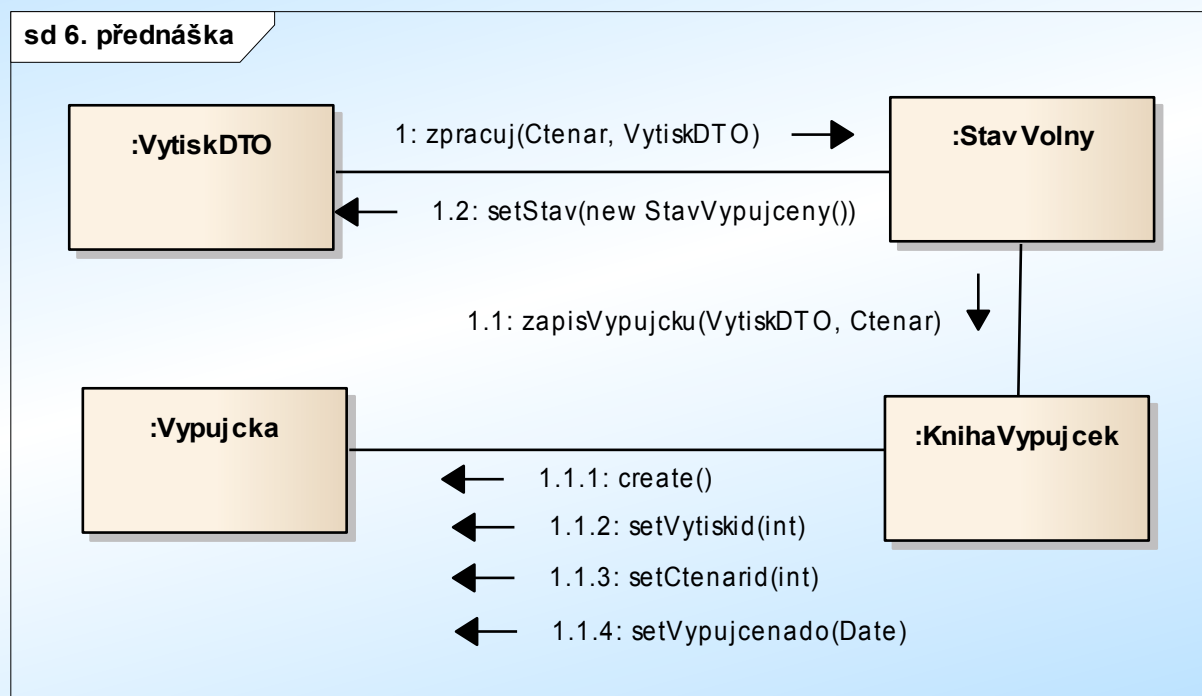
# Spolupráce objektů

- Diagram komunikace



# Spolupráce objektů

- Diagram komunikace – správné řešení



# Spolupráce objektů

- Sekvenční diagram
  - Výhody
    - Přehledné zachycení pořadí zasílání zpráv
    - Větší vyjadřovací schopnosti
  - Nevýhody
    - Velké nároky na prostor v horizontálním směru
    - Nevhodné pro zobrazení většího množství objektů

# Spolupráce objektů

- Diagram komunikace
  - Výhody
    - Úspornější na místo i při zobrazení více objektů
    - Možnost uspořádat objekty horizontálně i vertikálně
  - Nevýhody
    - Obtížnější sledování pořadí zasílání zpráv

# Spolupráce objektů

Dotazy?

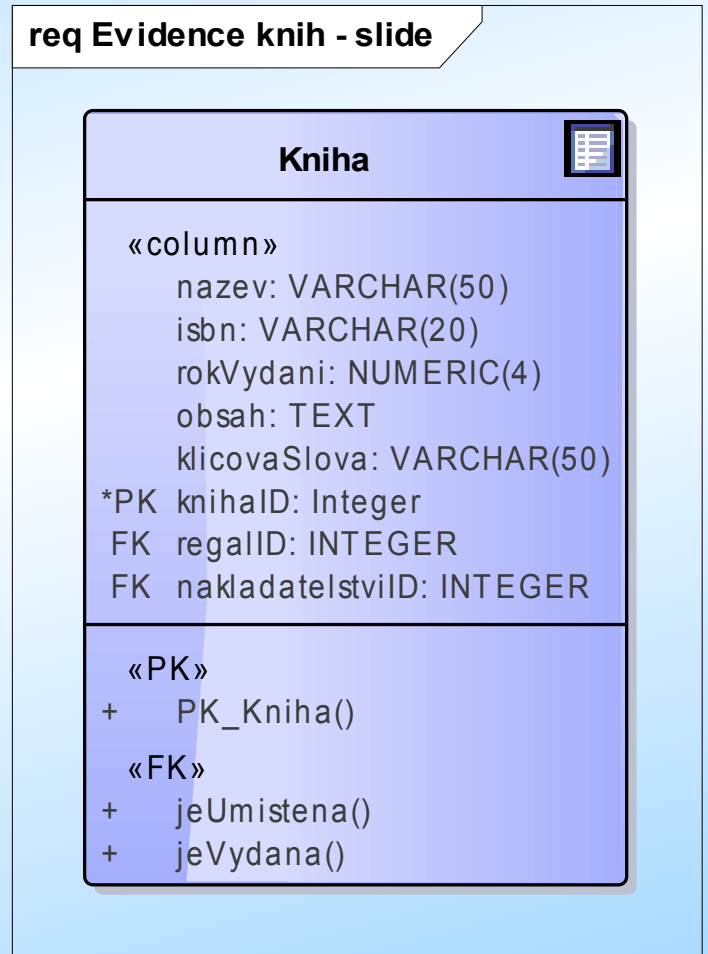
# Databázový model

- Popisuje uložení dat v relační databázi
- Generování základacích SQL skriptů
- Obsahuje datové typy zvolené databáze



# Databázový model

- Primární klíče
- Cizí klíče
- Dekompozice m:n vazeb
- Dědičnost
- Diagram v UML
  - rozšíření diagramu tříd
  - EA: UML Structural – Class
  - Ukázka bude na cvičení

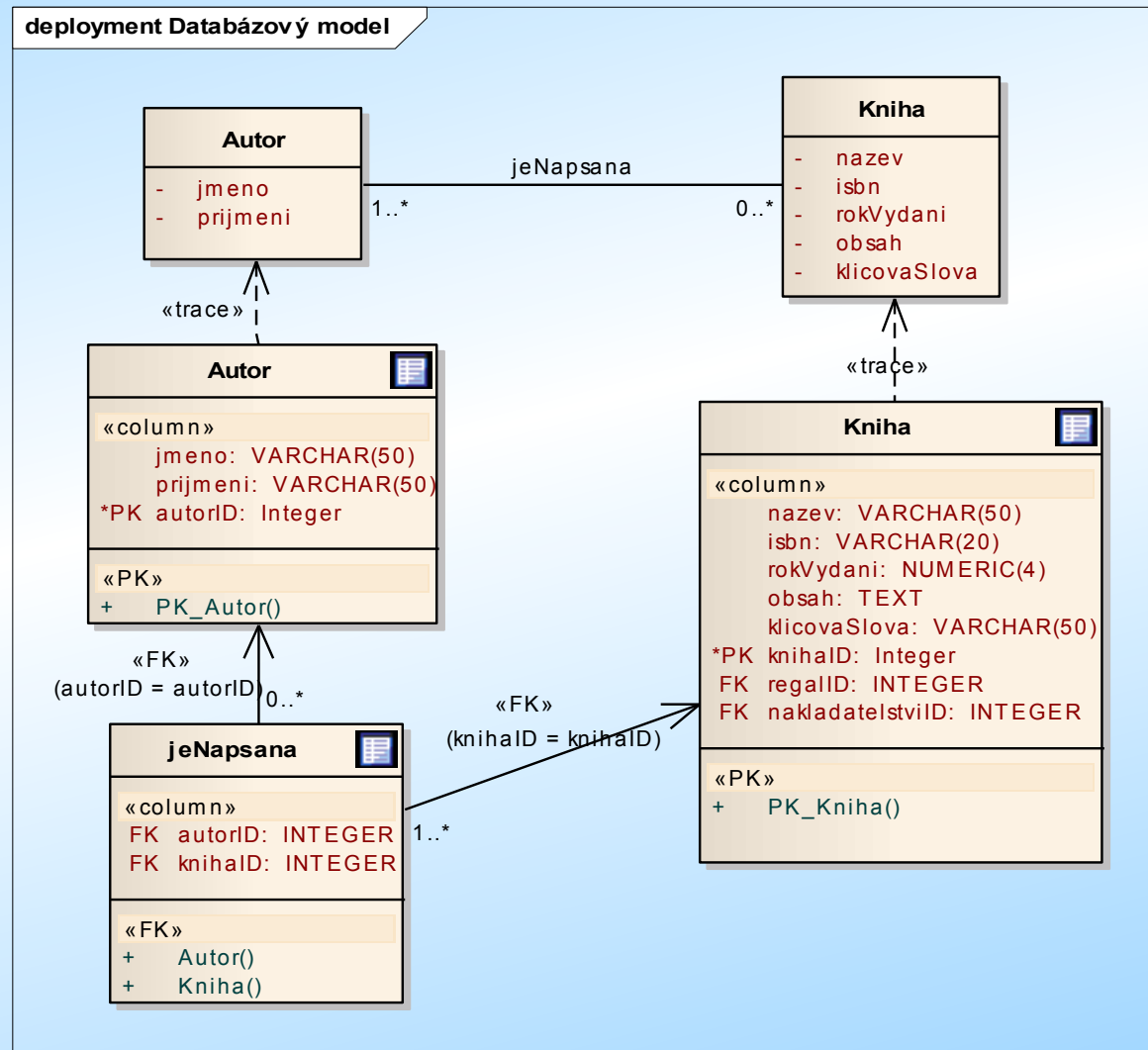


# Databázový model

- Dědičnost
  - 1 tabulka
  - 2 tabulky (počet potomků rodičovské třídy)
  - 3 tabulky (počet potomků rodičovské třídy + 1)
- Výhody a nevýhody?

# Databázový model

- MySQL

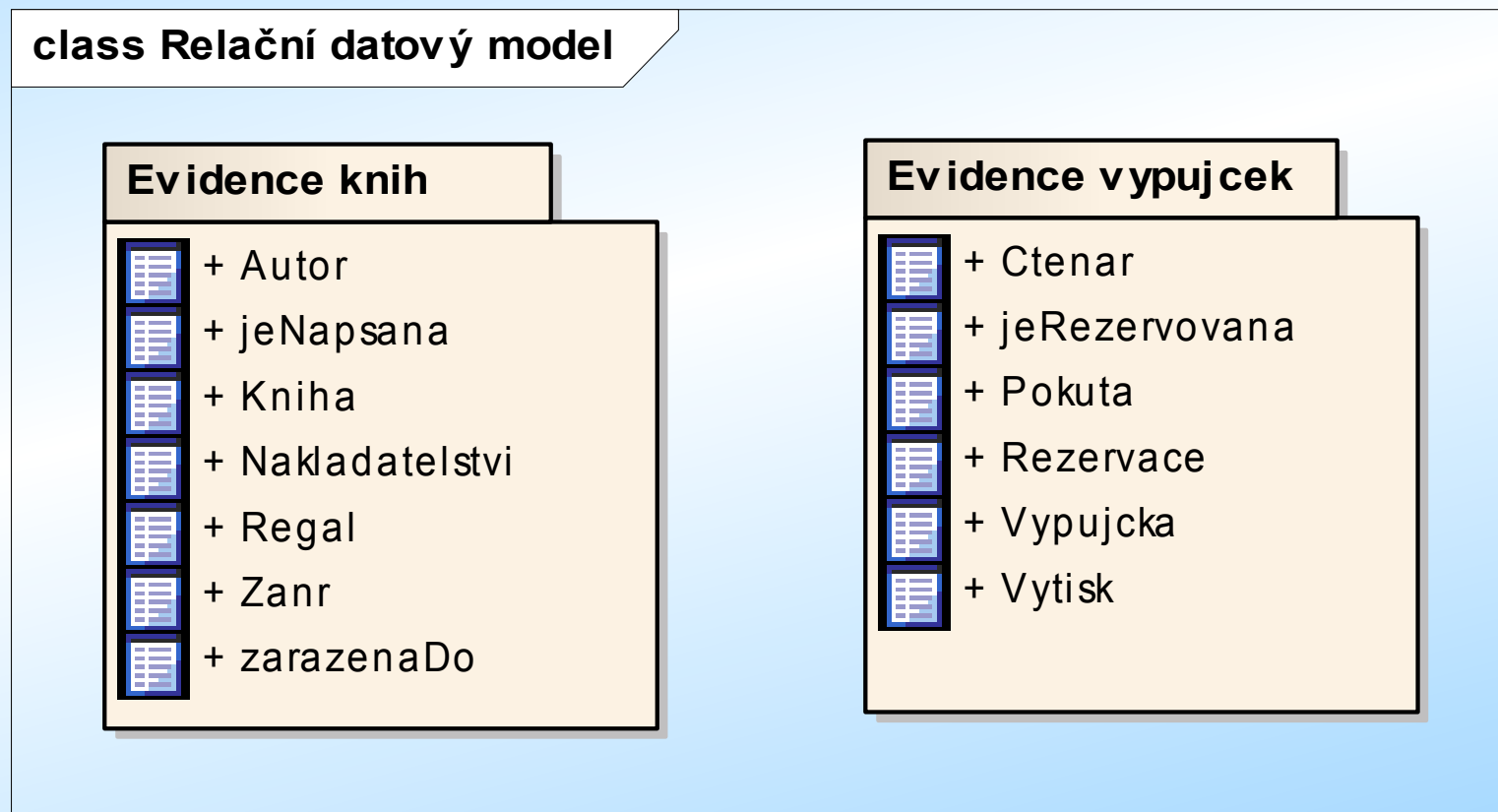


# Databázový model

- Doporučení - důležitá je přehlednost
  - Členění tabulek do více balíčků
    - Je vhodné zobrazovat i vztahy mezi tabulkami v různých balíčcích
    - Doporučení – při zobrazení tabulky z jiného balíčku skrýt názvy sloupečků a zobrazit pouze název
  - Skrývání popisů konců asociací (názvy PK, FK)

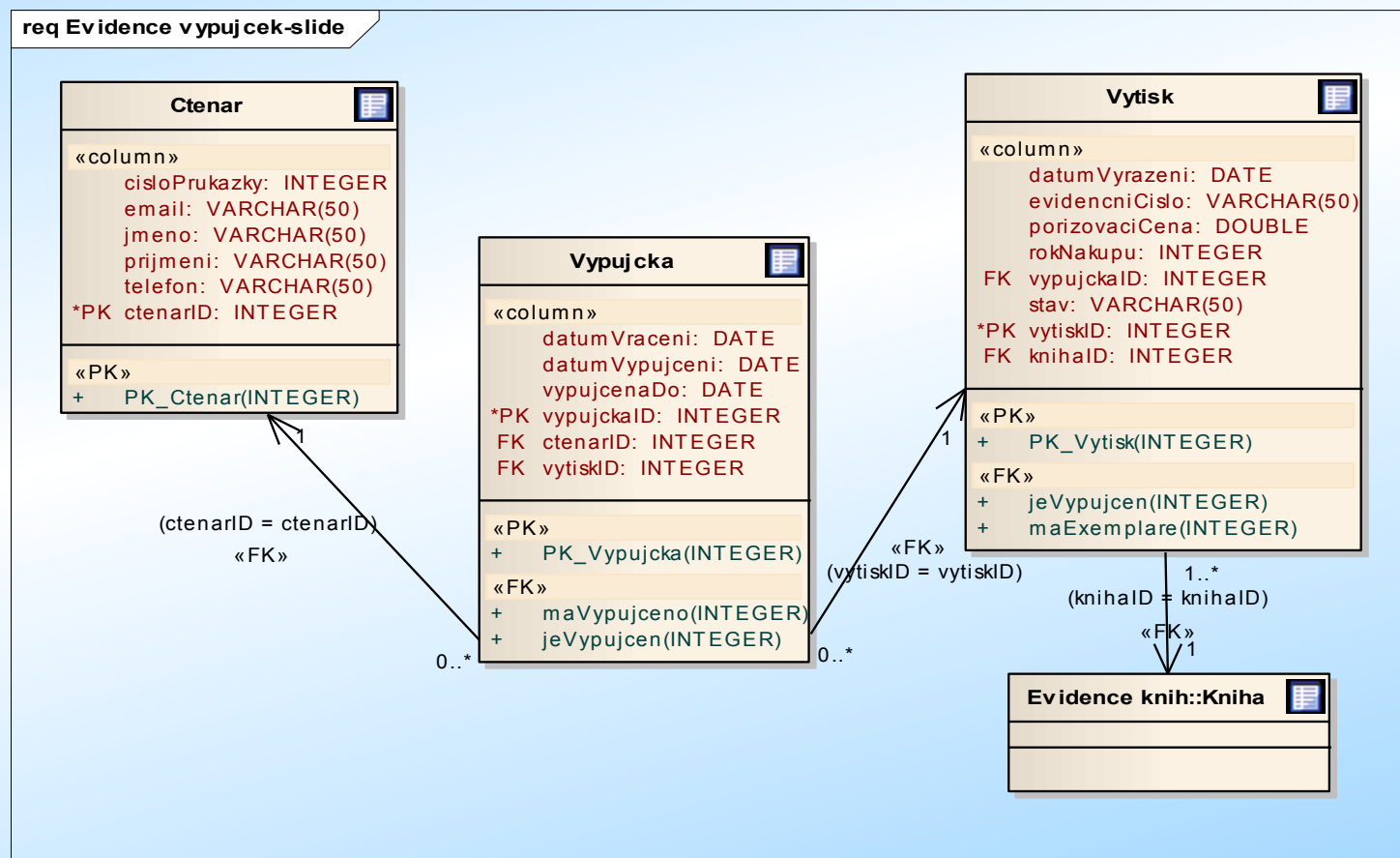
# Databázový model

- Členění do balíčků



# Databázový model

- Rozdělení na více diagramů



# Databázový model

Dotazy?

# GRASP

- Základní vzory / principy pro přiřazení zodpovědností třídám
- Zodpovědnost je úkol, který má třída řešit
- Existuje mnoho způsobů rozdělení úloh mezi třídy
- Neexistuje jediné správné řešení



# GRASP

- Informační expert (Information Expert)
- Nízká provázanost (Low Coupling)
- Vysoká soudržnost (High Cohesion)
- 6 dalších

# GRASP

- Informační expert
  - Základní princip přiřazení zodpovědnosti
  - Popis
    - Přiřadte zodpovědnost třídě, která má informace potřebné pro splnění této zodpovědnosti

# GRASP

- Nízká provázanost
  - Zmenšuje dopad při provádění změn
  - Popis
    - Přiřadte zodpovědnost tak, aby provázanost zůstala nízká
  - Každá třída by si měla vystačit při plnění úkolu sama a minimalizovat nutnost využití ostatních tříd
  - Zvyšuje možnost znovupoužití
  - Počet vazeb mezi třídami by měl být minimální

# GRASP

- Vysoká soudržnost
  - Podporuje nízkou provázanost
  - Popis
    - Přiřadte zodpovědnost tak, aby soudržnost zůstala vysoká
  - Každá třída by měl být zaměřena na jediný úkol
  - Zvyšuje srozumitelnost systému
  - Zodpovědnost třídy je snadno pochopitelná

# GRASP

- Jedna třída umí všechno
  - Nízká provázanost
  - Malá soudržnost
- Každá třída pouze jednu metodu
  - Vysoká provázanost
  - Velká soudržnost

# GRASP

- Další informace
  - Larman, Craig (2005). Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design and Iterative Development
  - [http://en.wikipedia.org/wiki/GRASP\\_\(object-oriented\\_design\)](http://en.wikipedia.org/wiki/GRASP_(object-oriented_design))

# GRASP

Dotazy?

# Děkuji za pozornost.