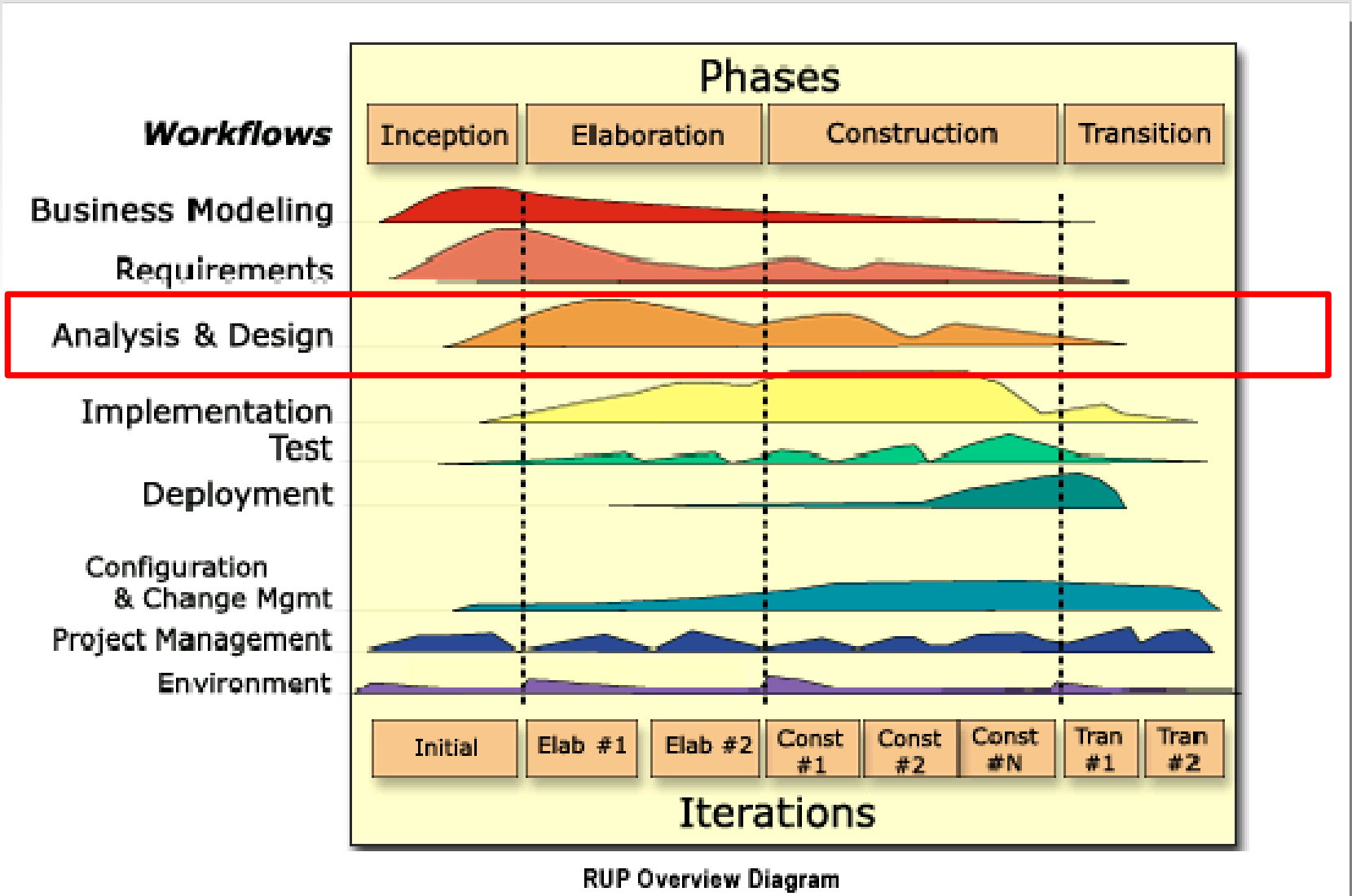
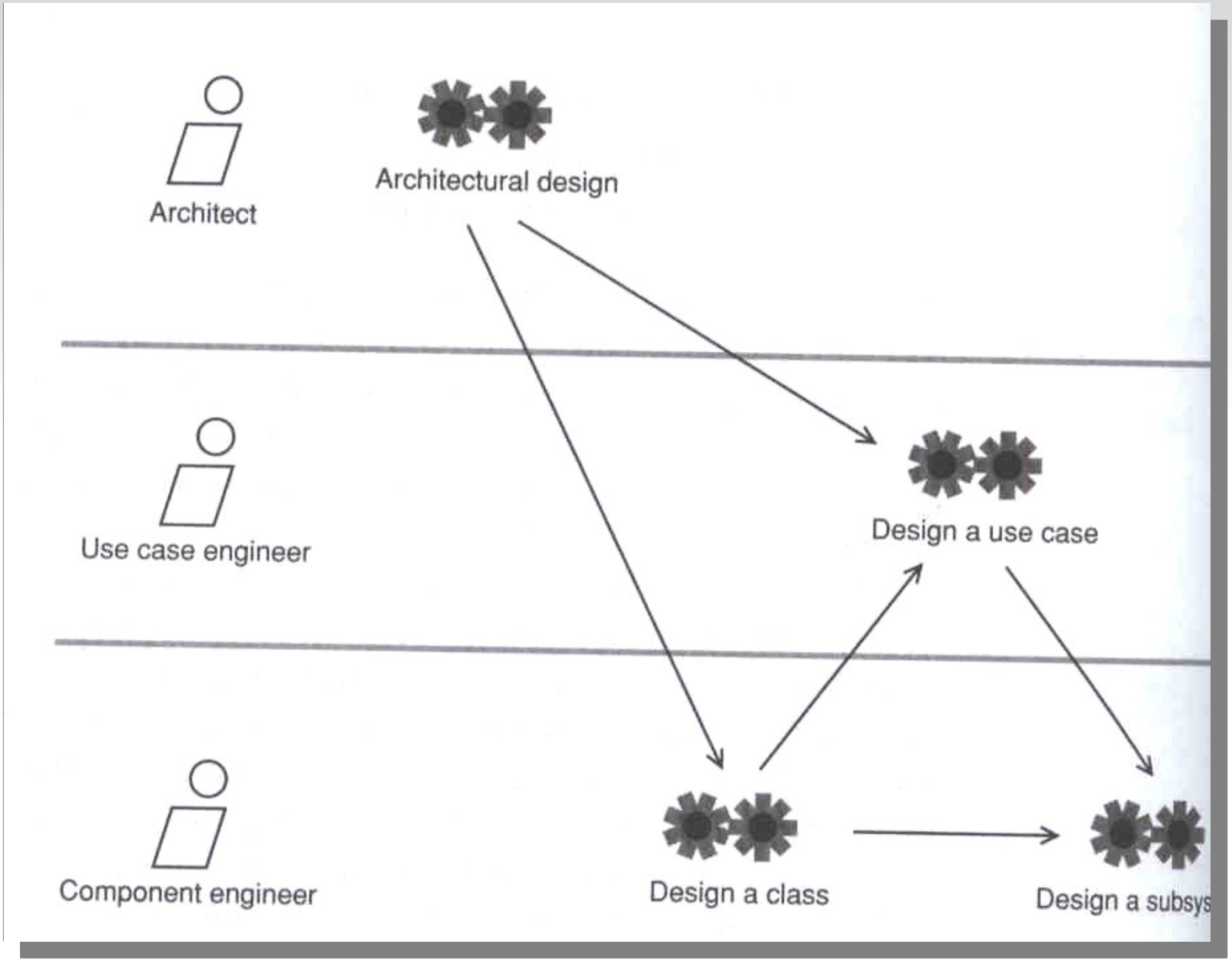
Návrh

© Radek Ošlejšek
Fakulta informatiky MU
oslejsek@fi.muni.cz

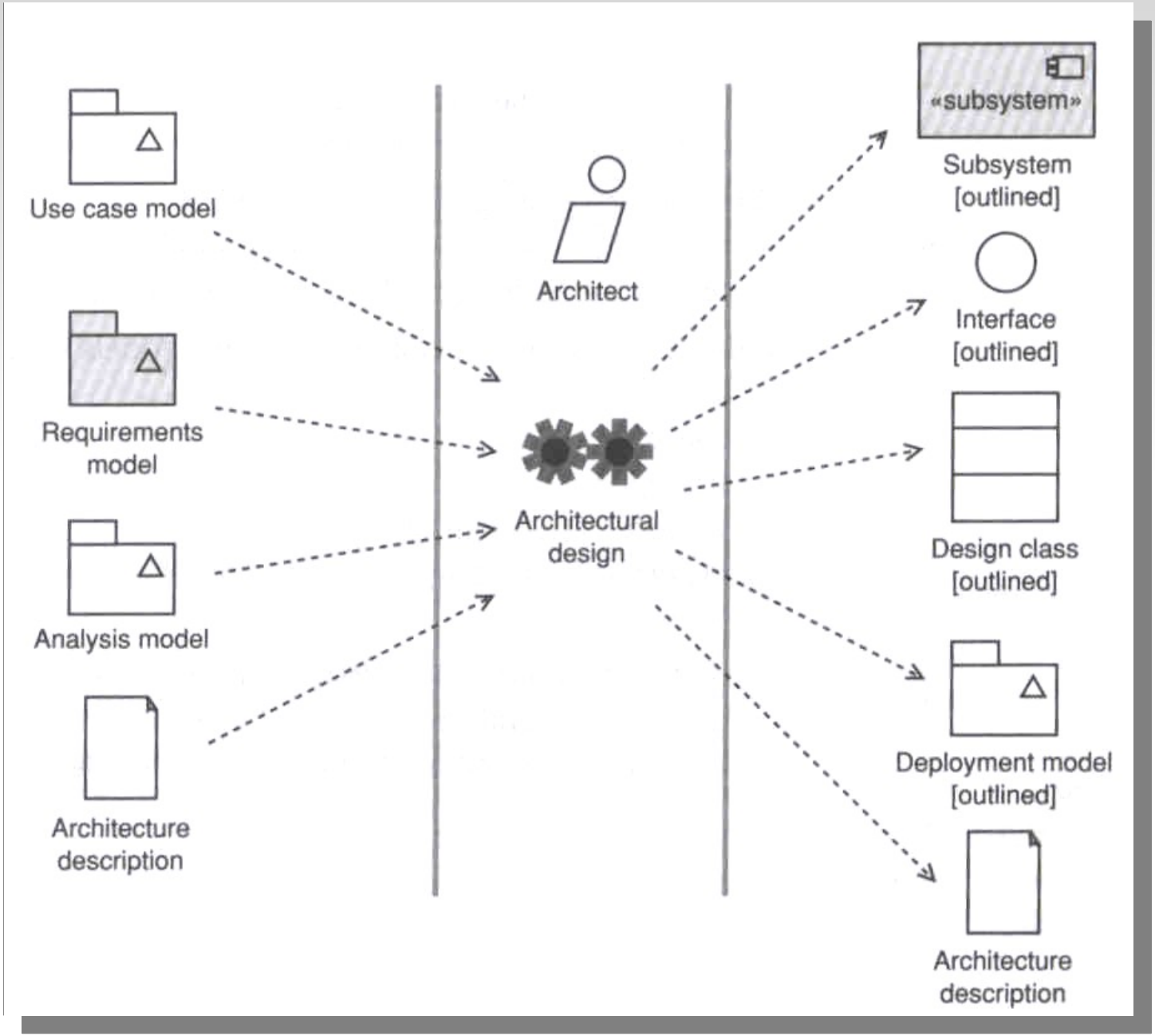
Design workflow



Design workflow detail

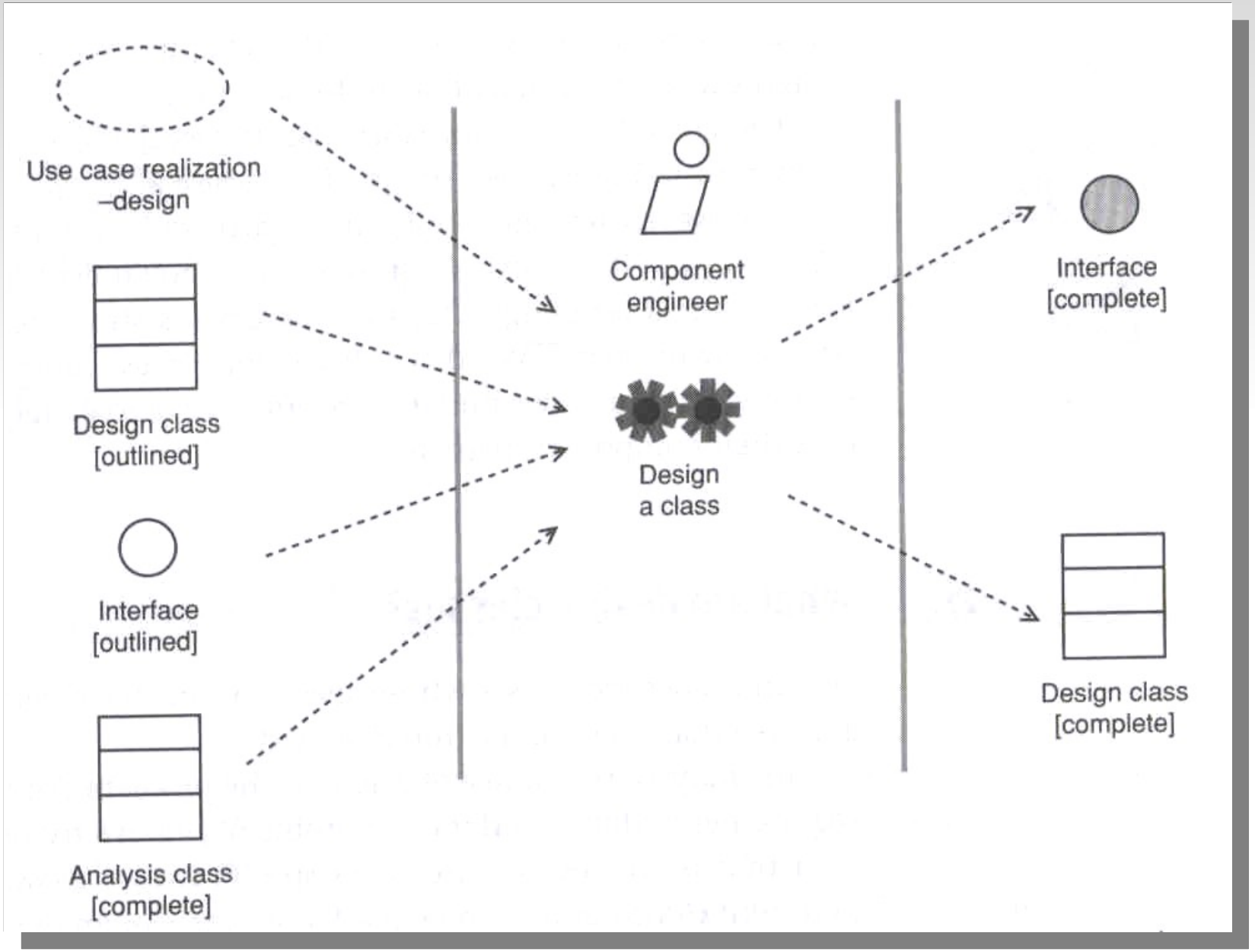


UP aktivita: Architectural design



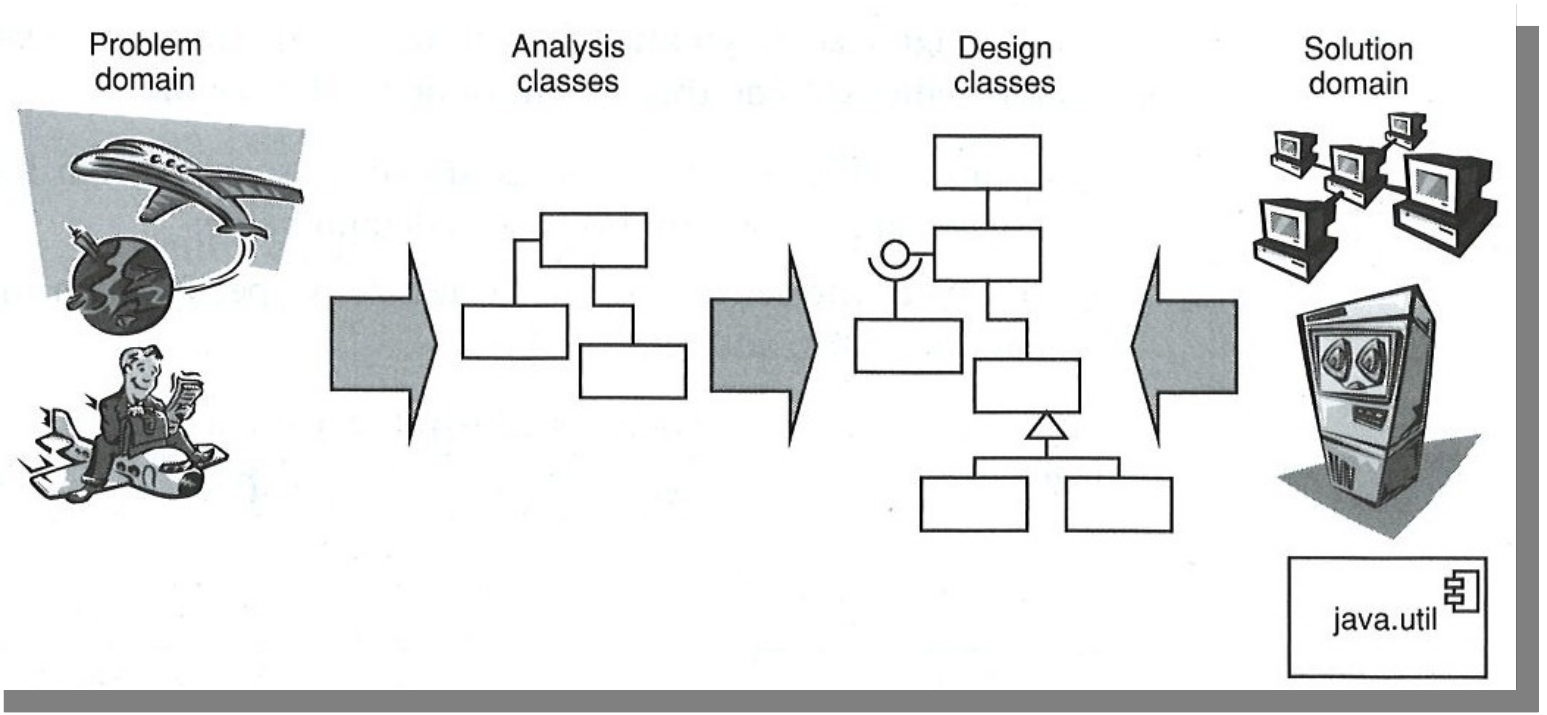
Návrhové třídy
(Design classes)

UP aktivita: Design a class



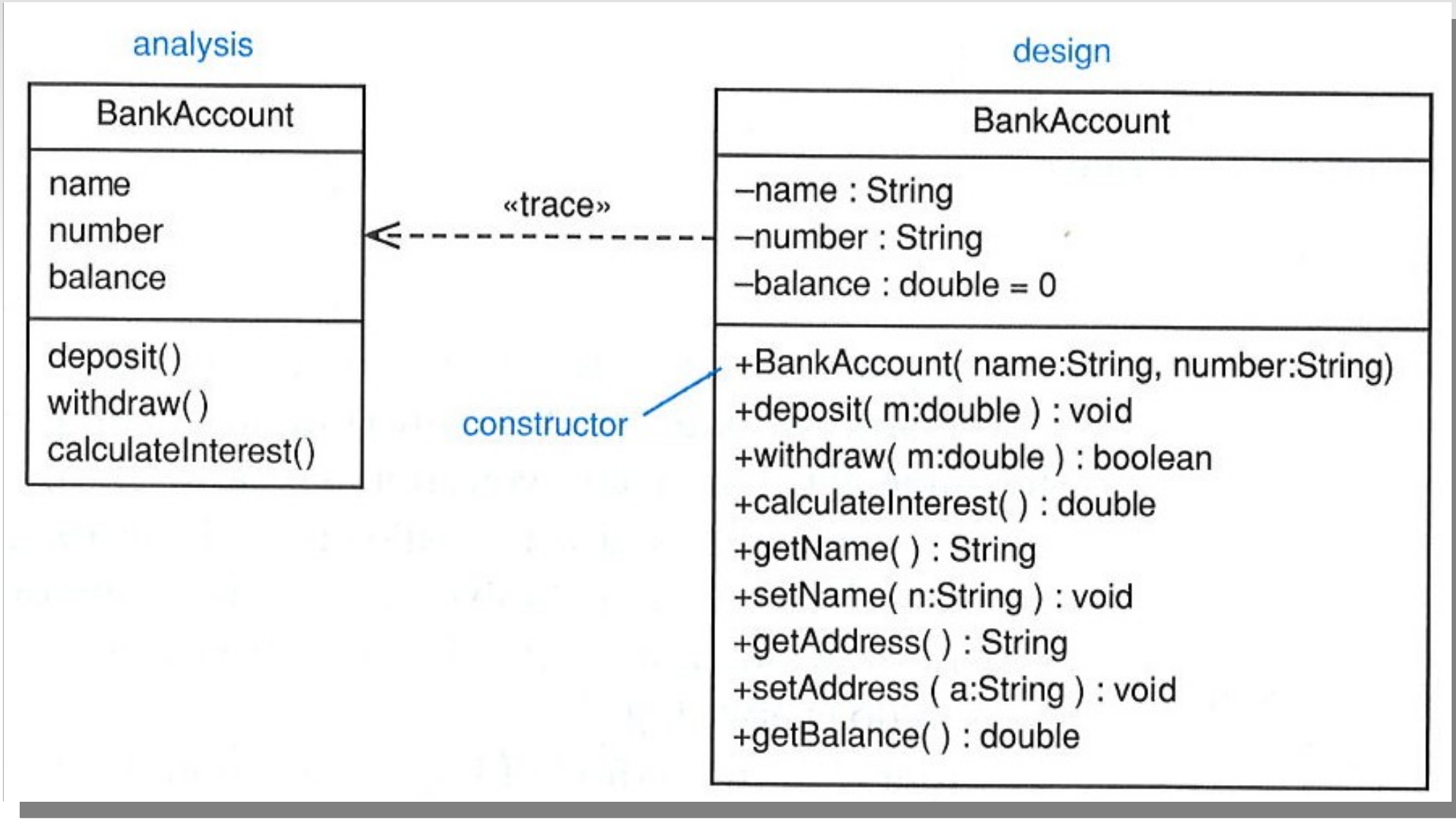
Návrhové třídy – vycházejí z

- „problem domain:
  - rozpracování analytických tříd
  - přidání implementačních tříd
  - rozdělení velkých analytických tříd na menší s jasnější zodpovědností
- „solution domain“
  - zapojení existujících knihoven, middlewaru, GUI a obecných znovupoužitelných elementů (*String*, *Time, Date*, kolekce apod.)



Vlastnosti návrhových tříd

- Obsahují všechny nezbytné detaily



Vlastnosti návrhových tříd (pokr.)

- Úplnost
  - veřejné operace definují kontrakt mezi třídou a jejími klienty
  - musí poskytovat všechny nezbytné operace
  - neměla by poskytovat zbytečné operace, o které nikdo nestojí
- Jednoduchost
  - operace by měly být jednoduché, atomické a unikátní (nemělo by existovat více způsobů, jak dosáhnout stejného cíle)
  - porušuje se v odůvodněných případech, např. z důvodu efektivity kódu
- Vysoká soudržnost (koheze, zodpovědnost)
  - každá třída by měla mít jednu dobře definovanou abstrakci a co nejméně vlastností
- Co nejmenší propojení s ostatními třídami (coupling)

Má smysl udržovat dva modely?

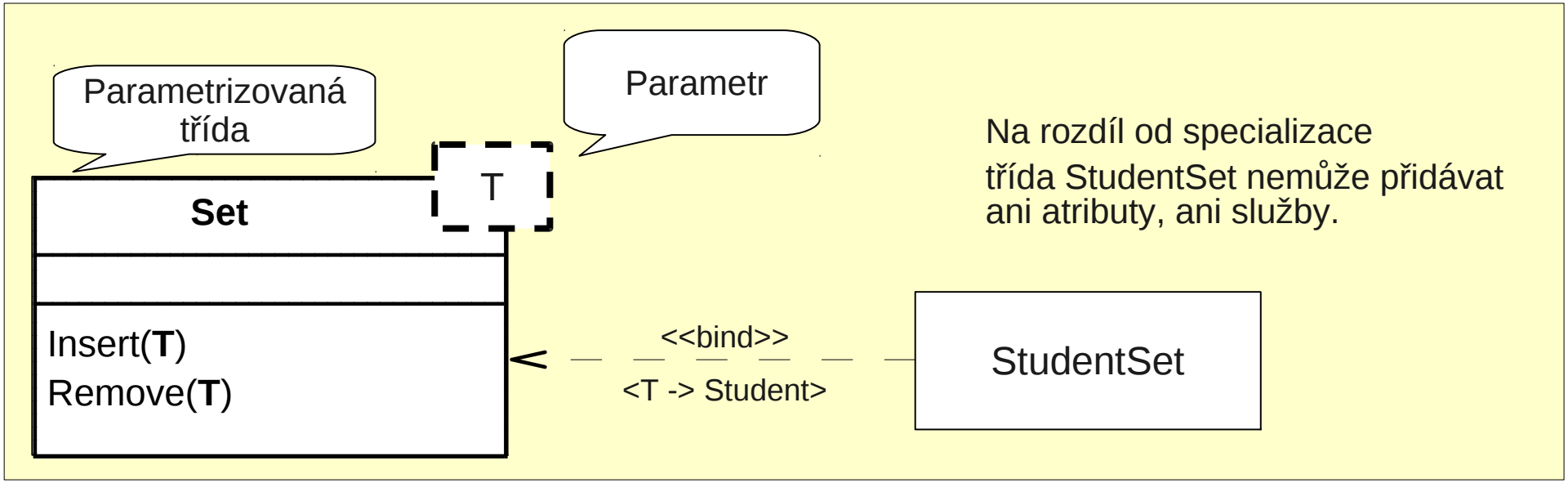
- Strategie, jak přejít od analýzy k návrhu:
  - Postupně přepracujte analytický model na návrhový
    - Pracujete pouze s jediným modelem, ztrácíte analytický pohled
  - Analytický model přepracujte na návrhový a pak použijte CASE nástroje pro vytvoření “analytického pohledu”
    - Pracujete pouze s jediným modelem, ale analytický model vytvořený CASE systémem nemusí být uspokojivý
  - Zamkněte analytický model ve vhodném bodě vývoje a přepracujte *kopii* analytického modelu na návrhový model
    - Máte k dipozici oba modely, mohou se ale rozcházet
  - Spravujte dva oddělené modely, analytický a návrhový
    - Máte k dipozici oba modely které se navíc schodují, správa je ale komplikovaná

Parametrizované třídy, šablony

Parametrizovaná třída definuje *rodinu* tříd.

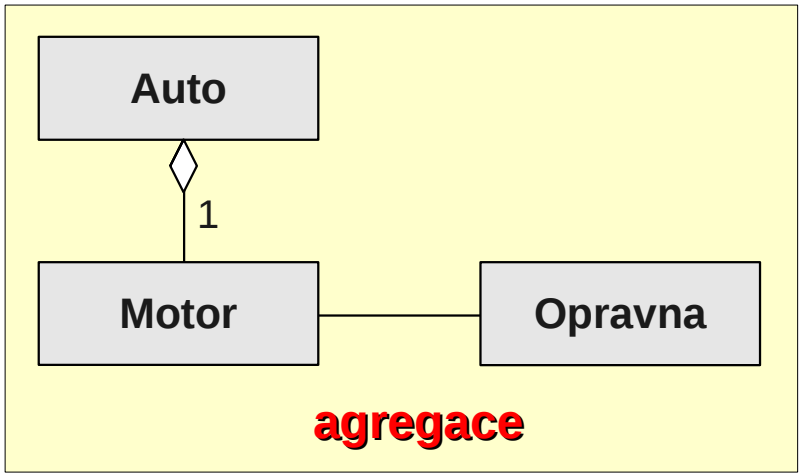
Nelze ji přímo použít, je nutné ji nejprve instanciovat přiřazením potřebných typů. Instanciace teprve vytvoří použitelnou třídu.

Příklad: třída **Set** – reprezentuje kolekci objektů nějaké *třídy.* Tato třída se stává parametrem parametrizované třídy *např. množina **mincí**, množina **studentů**, množina <T>*



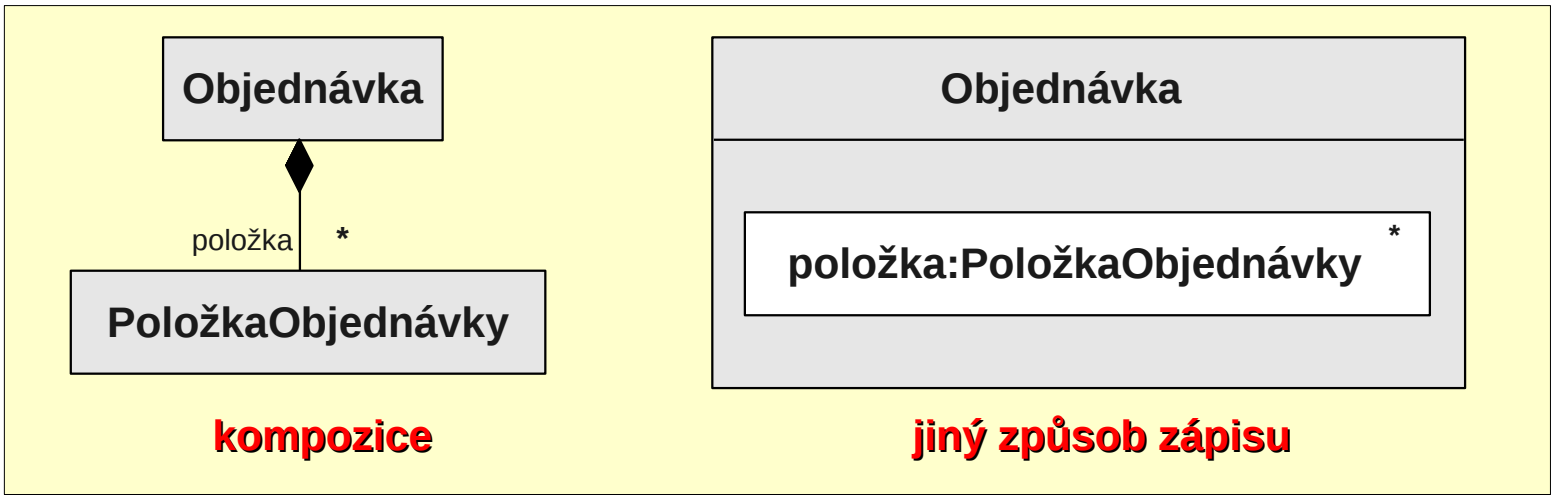
Agregace

- Angl. *Aggregation*
- Speciální případ asociace pro **vztah celek-část**
- Symbol prázdného diamantu
- Tranzitivní
  - pokud je píst součást motoru a motor součástí auta, pak je i píst součástí auta
- Asymetrická
  - pokud je motor součástí auta, pak auto není součástí motoru
  - diagram objektů musí být acyklický
- Sémantika:
  - totéž co asociace, pouze zdůrazňujeme vztah celek-část
  - součásti mohou existovat nezávisle na celku
  - součást může být sdílena více celky



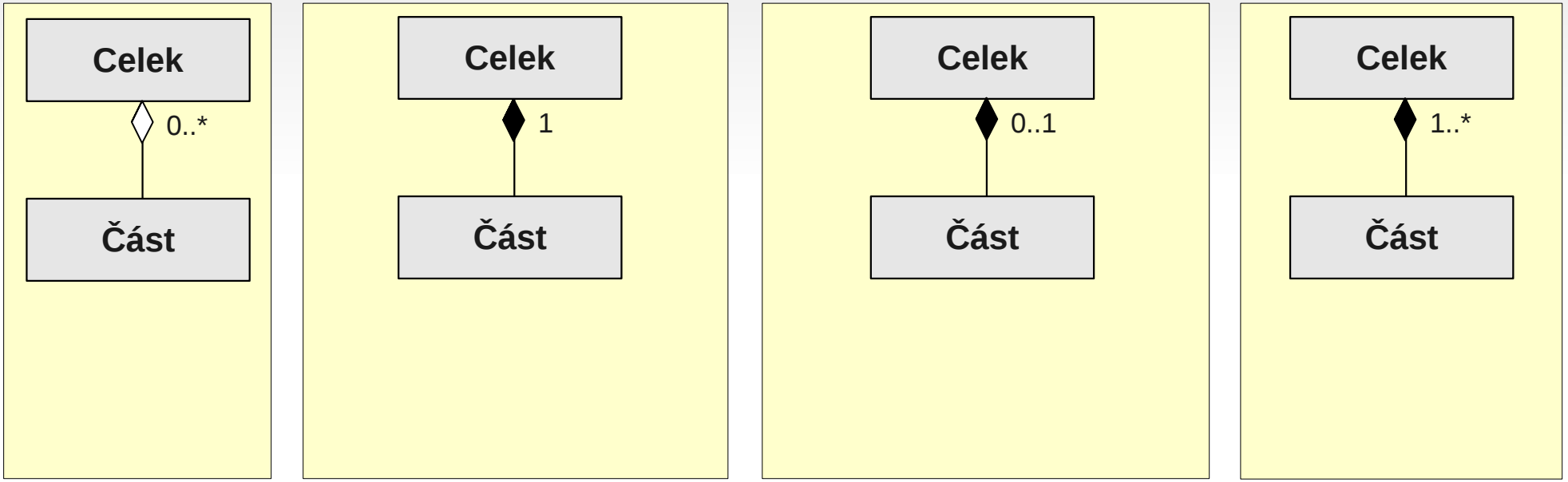
Kompozice

- Angl. *Composition*
- Silnější forma agregace
- Symbol plného diamantu
- Tranzitivní a asymetrická
- Sémantika:
  - na úrovní instancí (objektů) části neexistují vně celku
  - každá část patří pouze do jediného celku (části nelze sdílet)
  - je-li celek zničen, musí zničit i svoje součásti, nebo převést zodpovědnost za ně na jiný objekt



Vlastnosti agregace a kompozice (I)

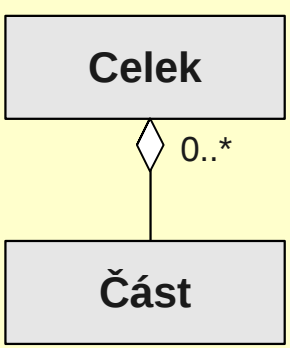
- Které modely jsou správně a které naopak špatně z důvodu vlastností agregace a kompozice?



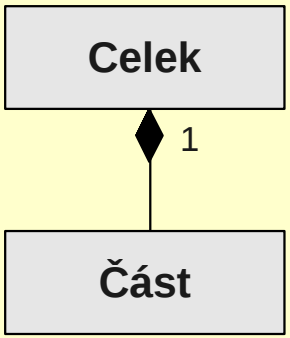
... odpověď na další obrazovce >>

Vlastnosti agregace a kompozice (II)

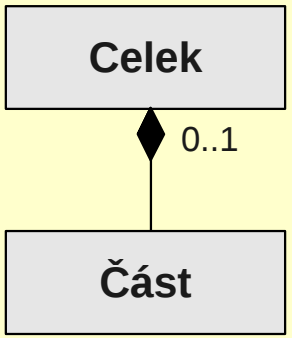
... odpovědi z předchozí stránky:



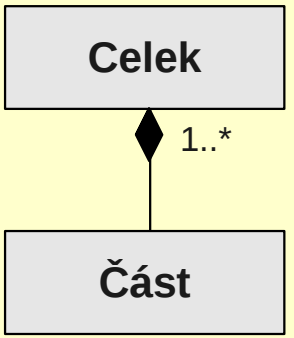
správně



správně
totéž co bez kardinality



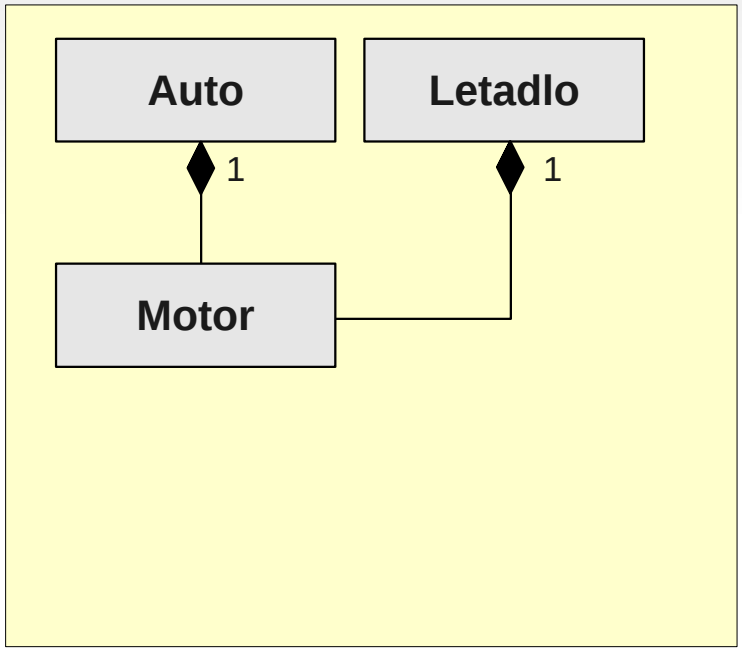
špatně
část neexistuje bez celku

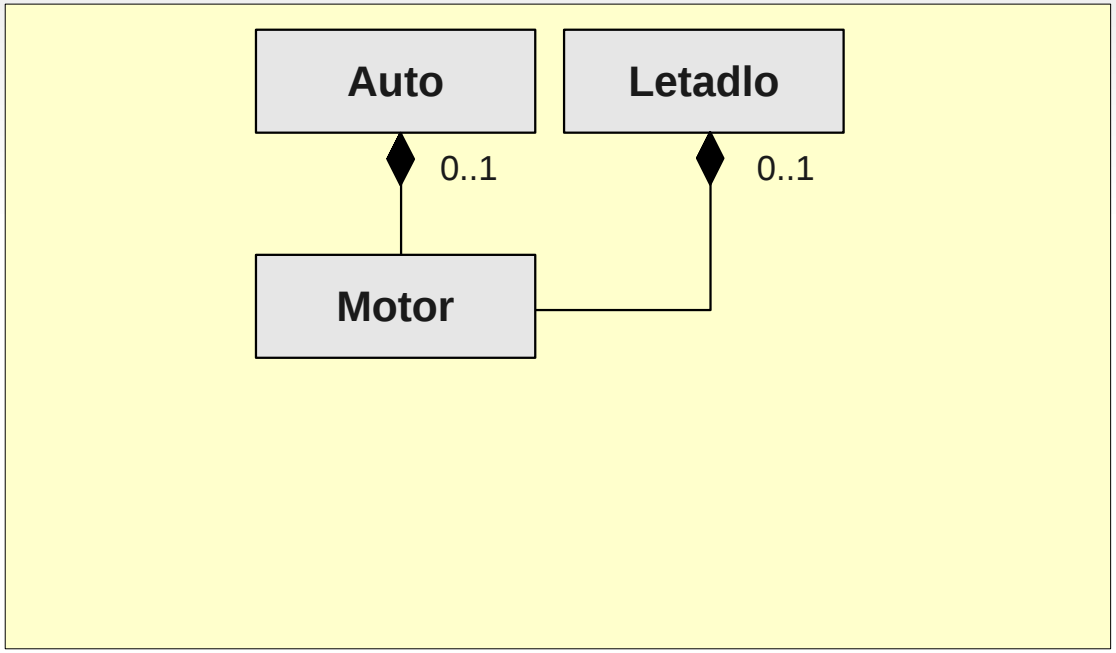


špatně
část je součástí jediného celku

Vlastnosti agregace a kompozice (IV)

- Které modely jsou správně a které naopak špatně z důvodu vlastností agregace a kompozice?

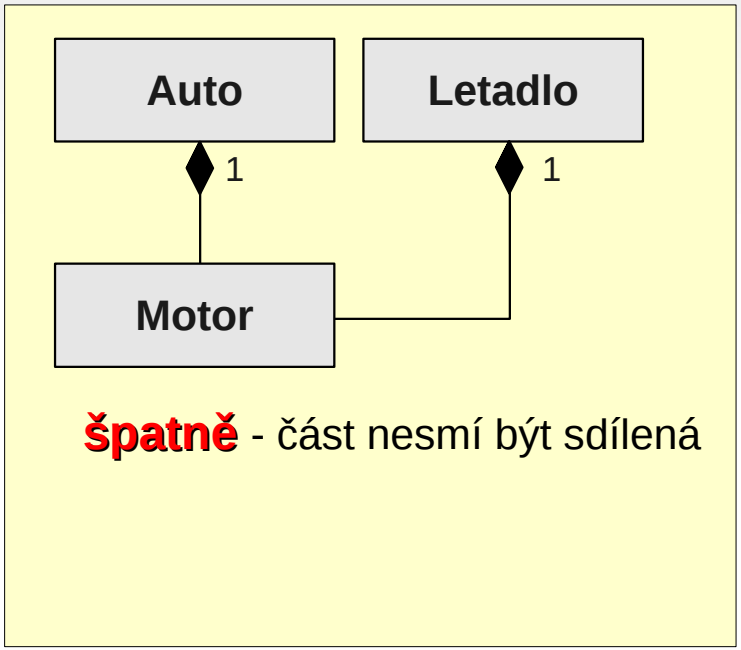




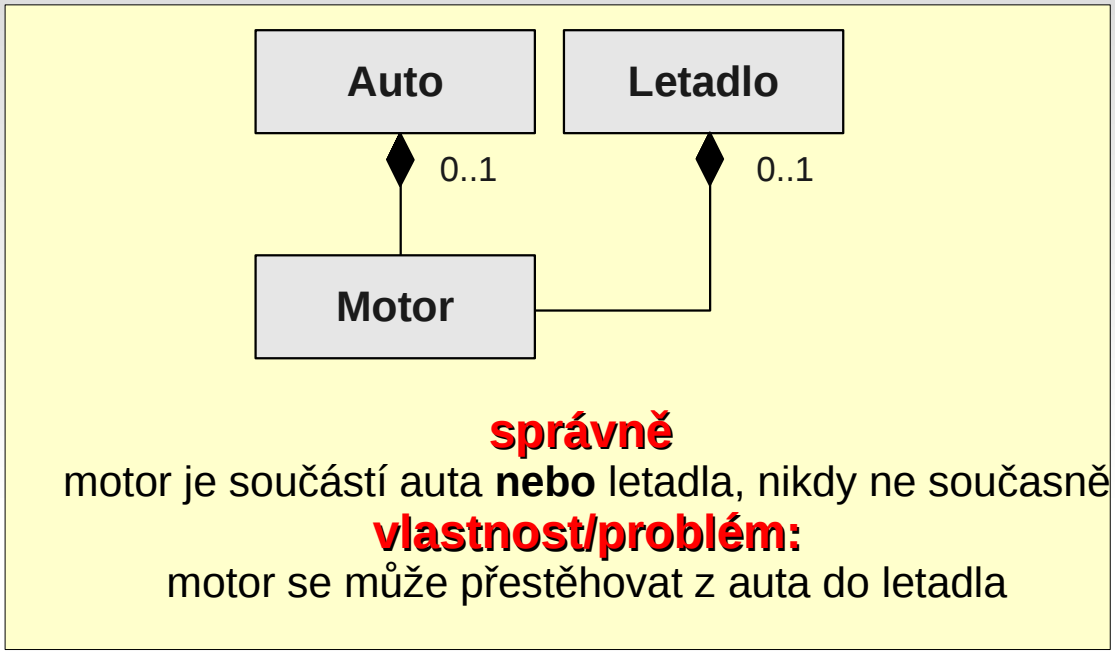
... odpověď na další obrazovce >>

Vlastnosti agregace a kompozice (V)

... odpovědi z předchozí stránky:



špatně - část nesmí být sdílená

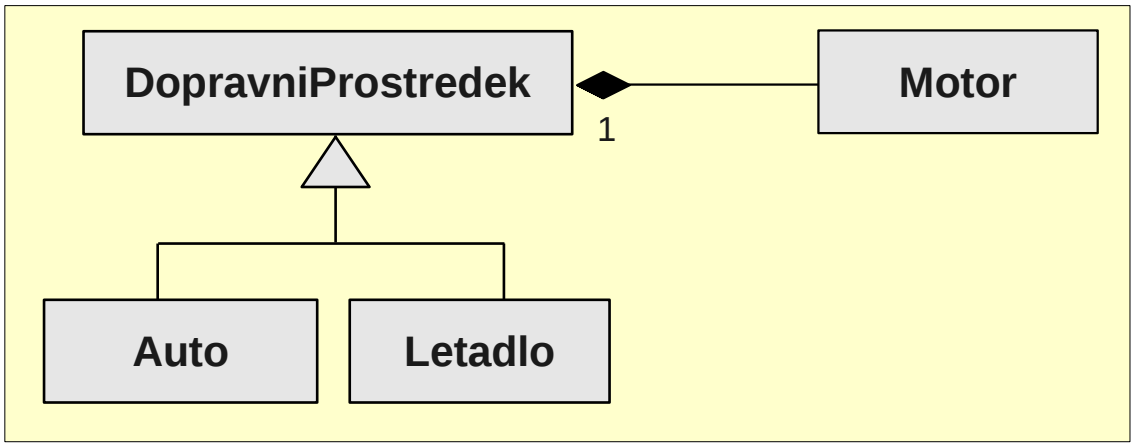


správně

motor je součástí auta **nebo** letadla, nikdy ne současně

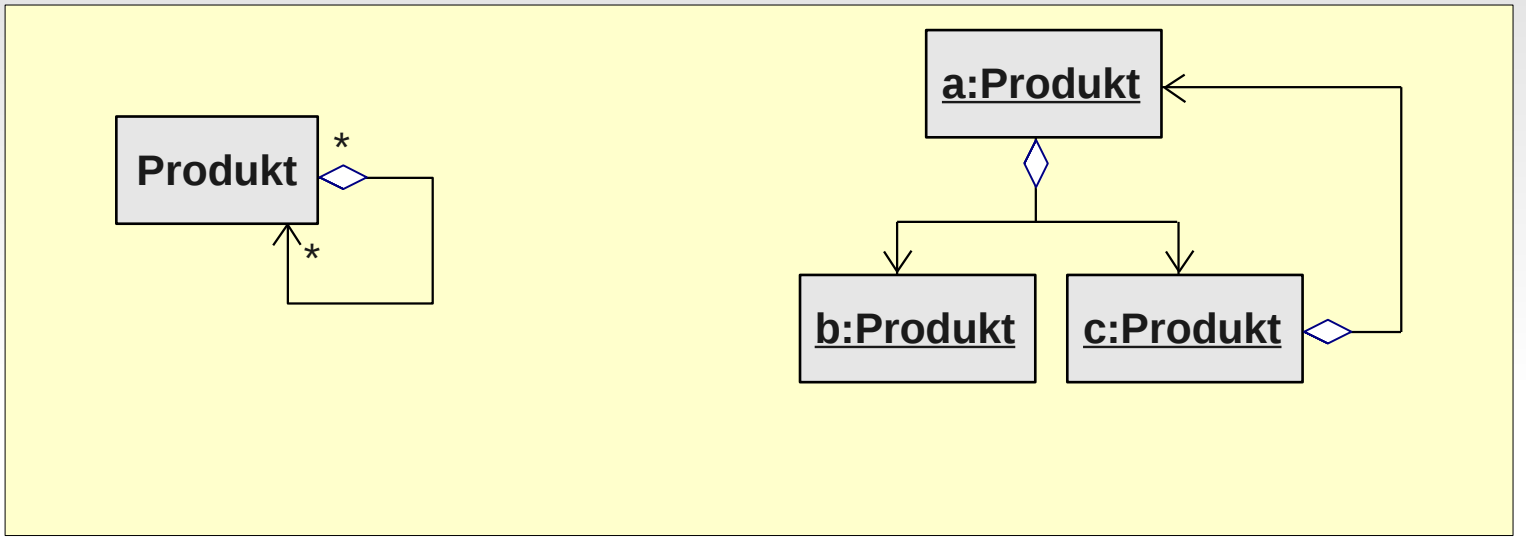
vlastnost/problém:

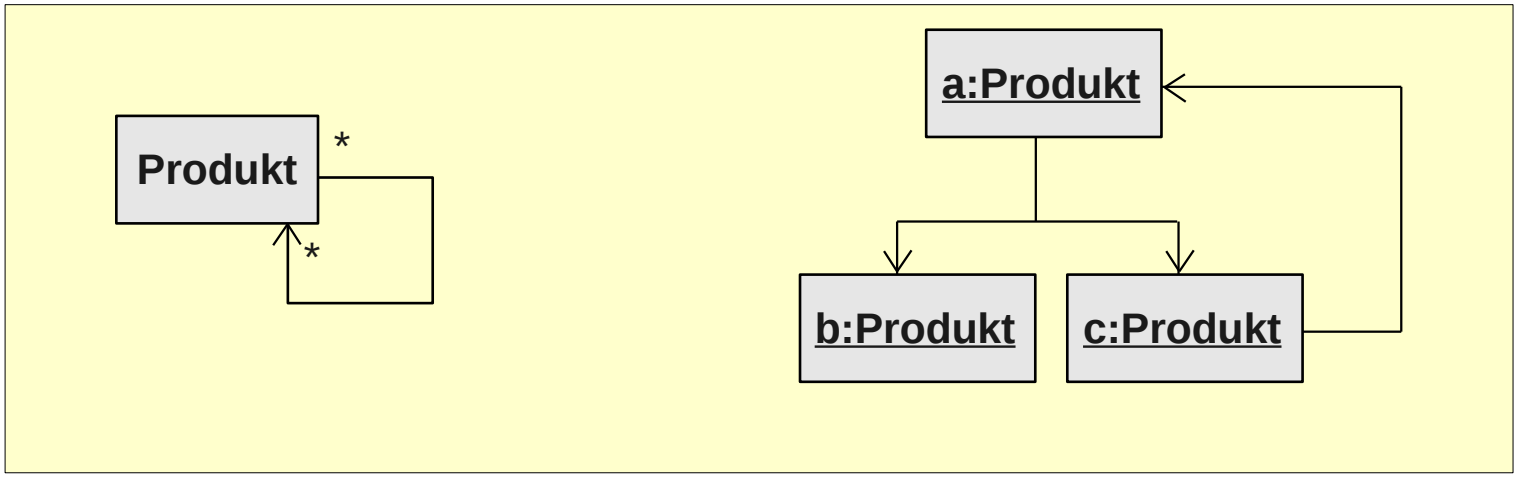
motor se může přestěhovat z auta do letadla



Vlastnosti agregace a kompozice (VI)

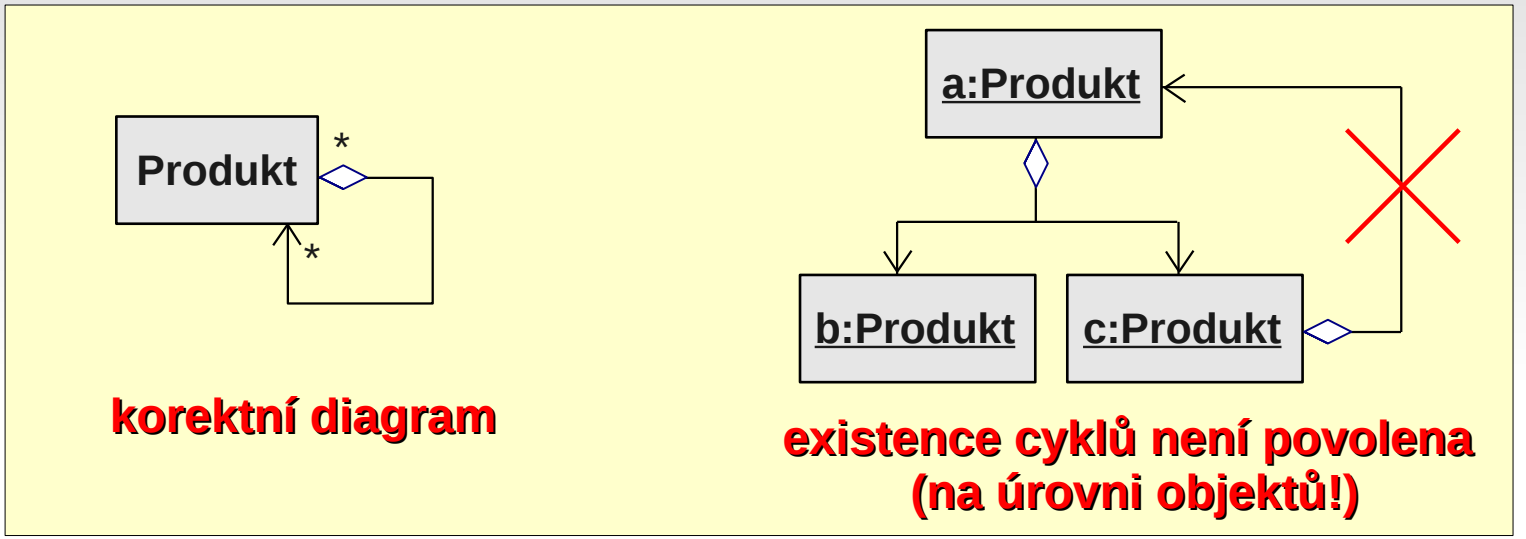
- Které modely jsou správně a které naopak špatně z důvodu asymetrie agregace?

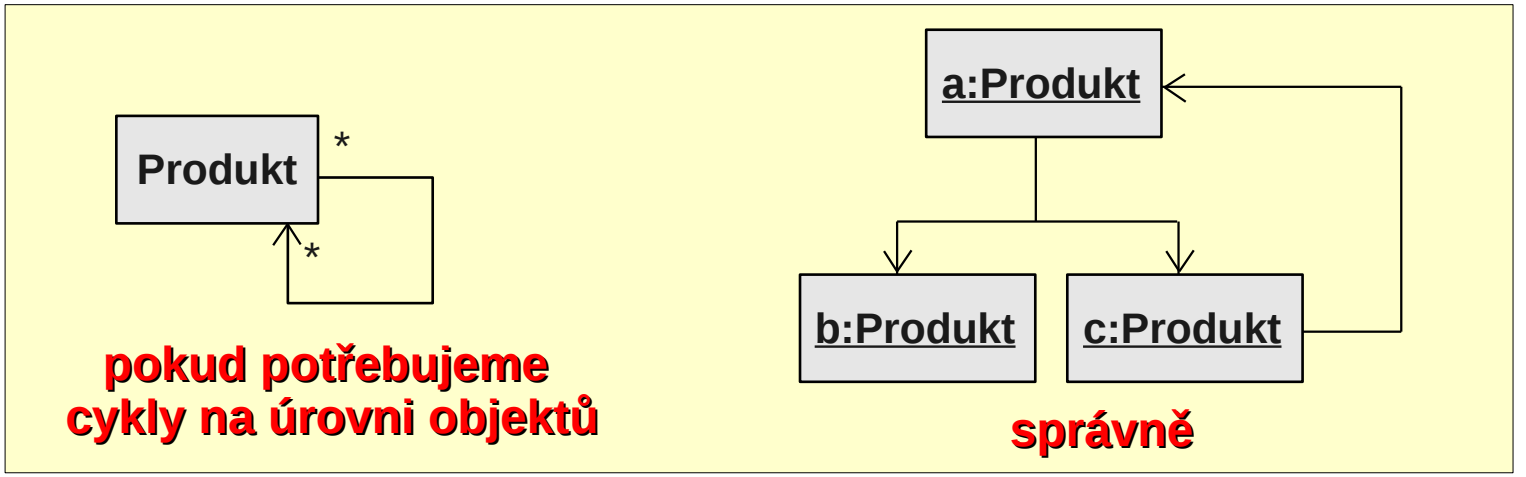




Vlastnosti agregace a kompozice (VII)

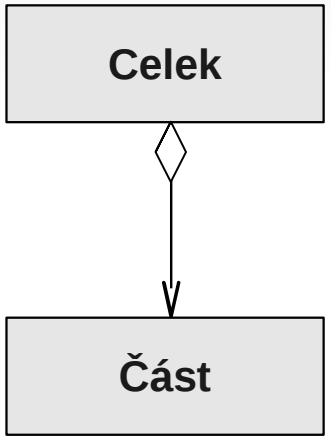
... odpovědi z předchozí stránky:

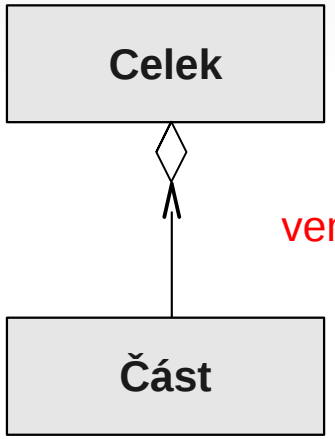




Agregace a směr řízení

- V agregace/kompozici jsou většinou části řízeny svými celky
- Většina asociací z analytického modelu přechází na agregace a kompozice v návrhu

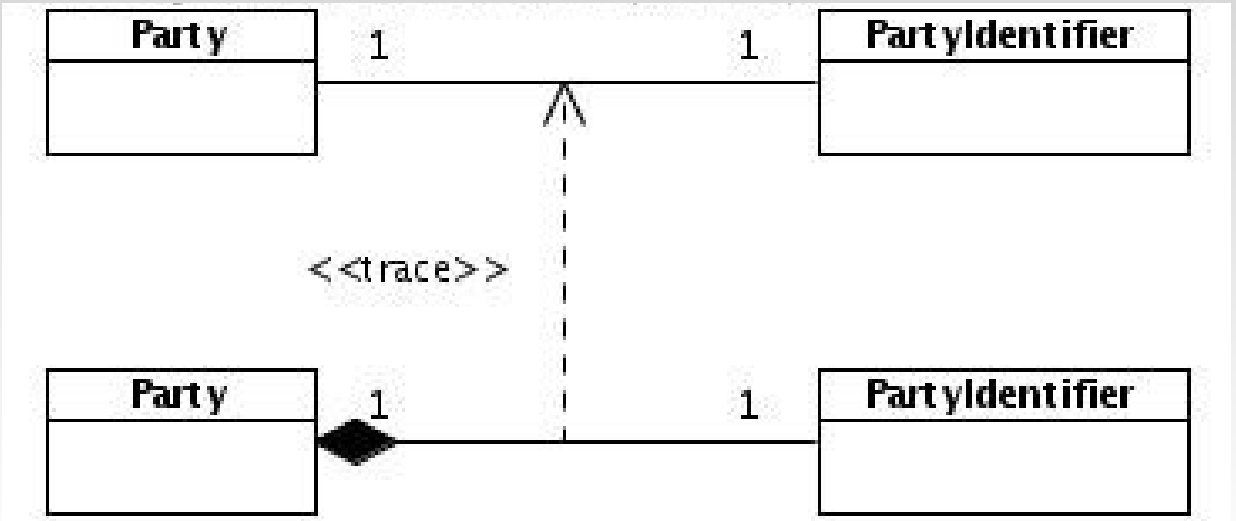




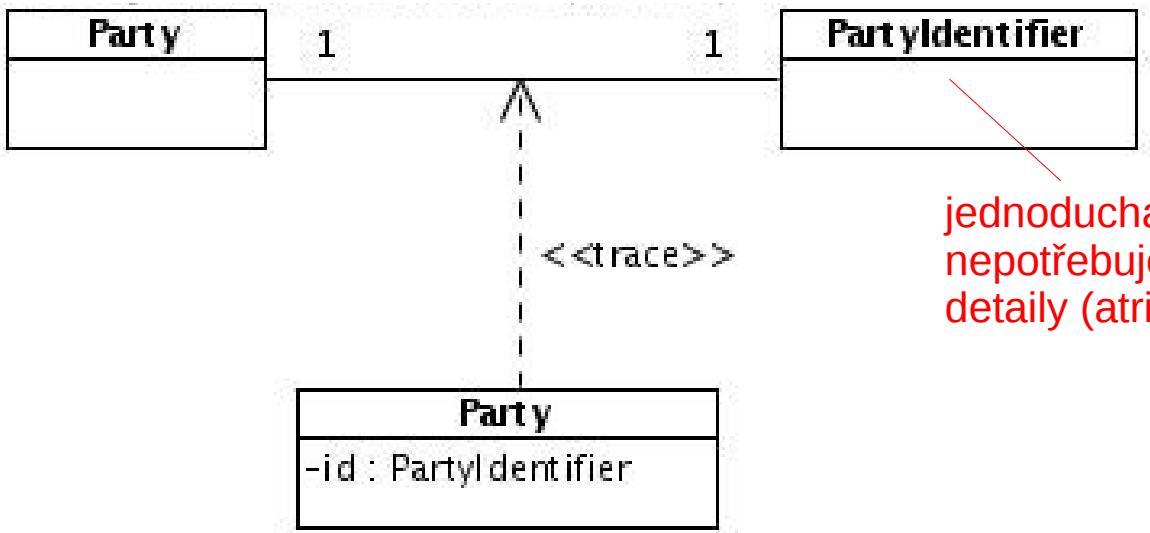
vemi neobvyklé

Návrhové třídy: Asociace 1:1

analýza



analýza



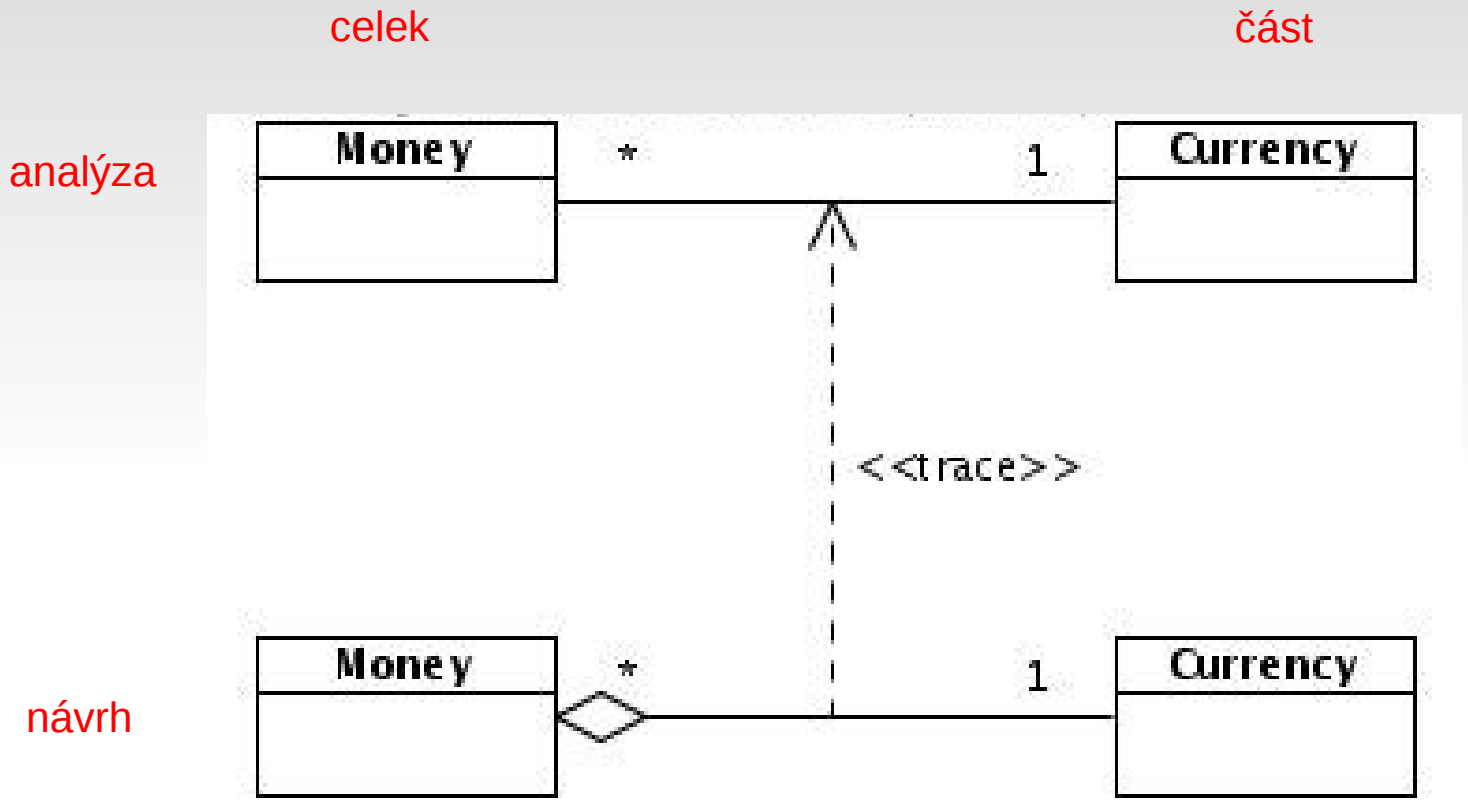
jednoduchá třída,
nepotřebujeme zobrazovat
detaily (atributy, ...)

návrh

návrh

Asociace 1:1 často vedou na kompozici

Návrhové třídy: Asociace M:1

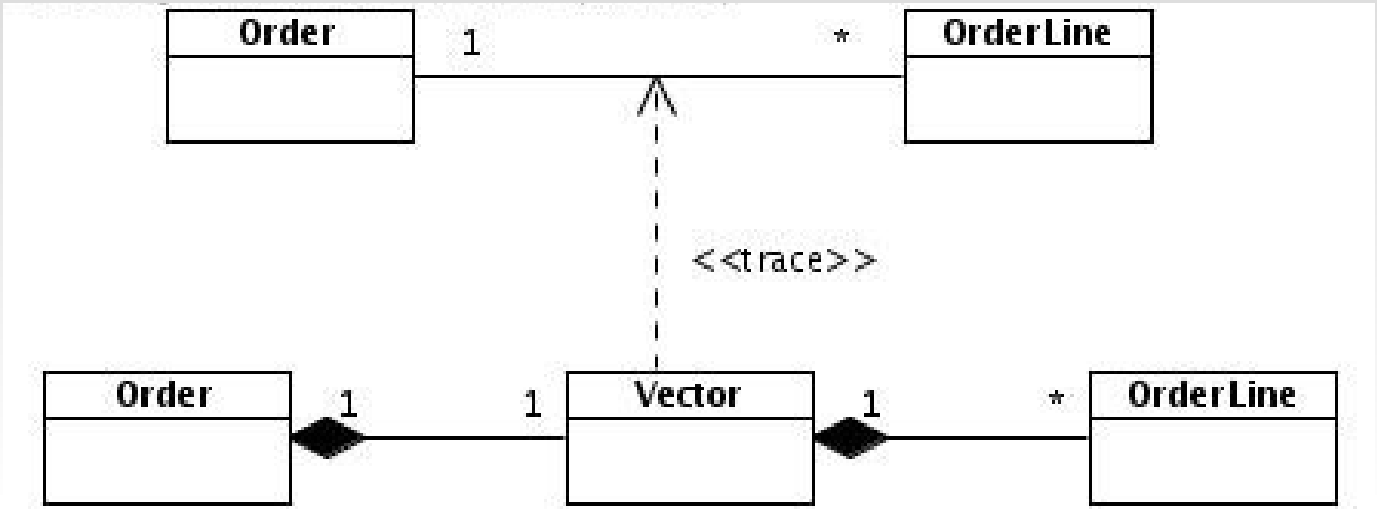


Asociace M:1 často vedou na agregaci, pokud není potřeba dělat cykly na úrovni objektů

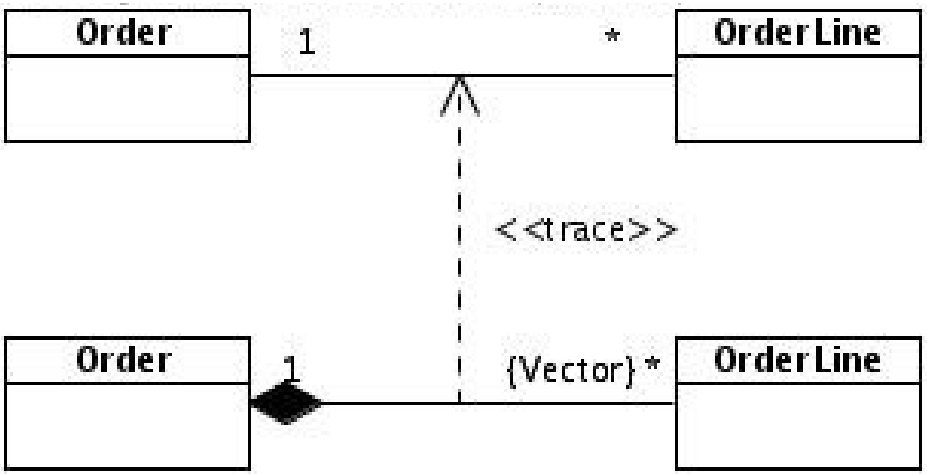
Návrhové třídy: Asociace 1:M (= kolekce)

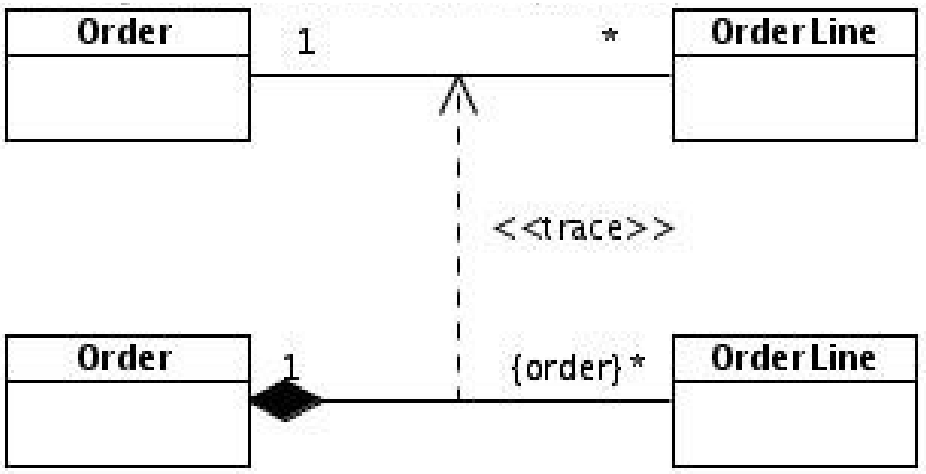
celek část

analýza



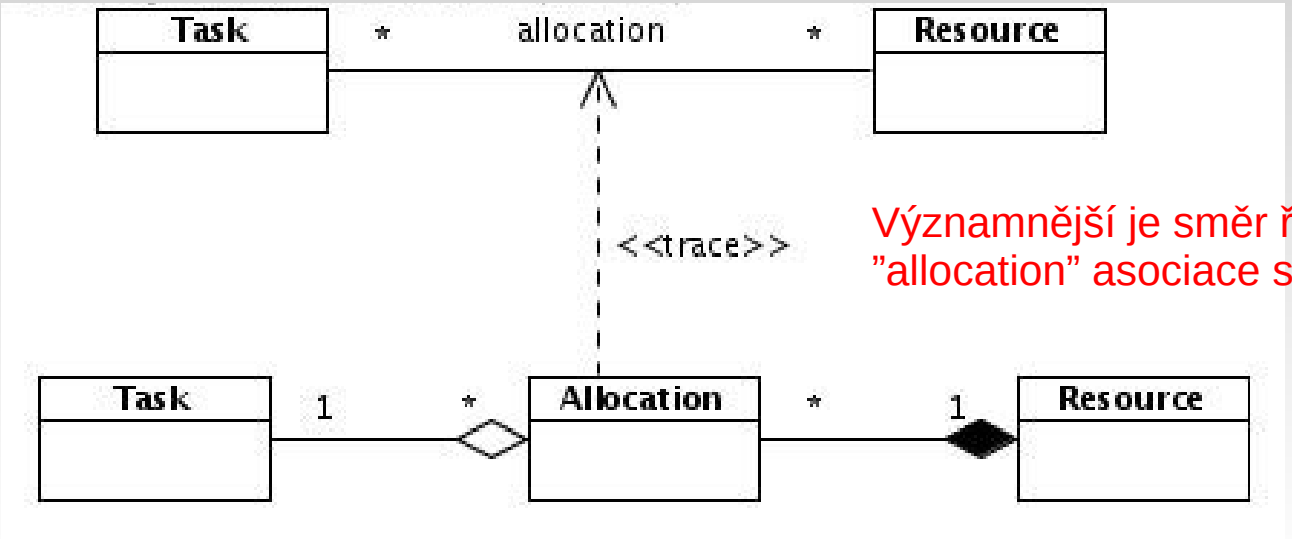
návrh





Návrhové třídy: Asociace M:N

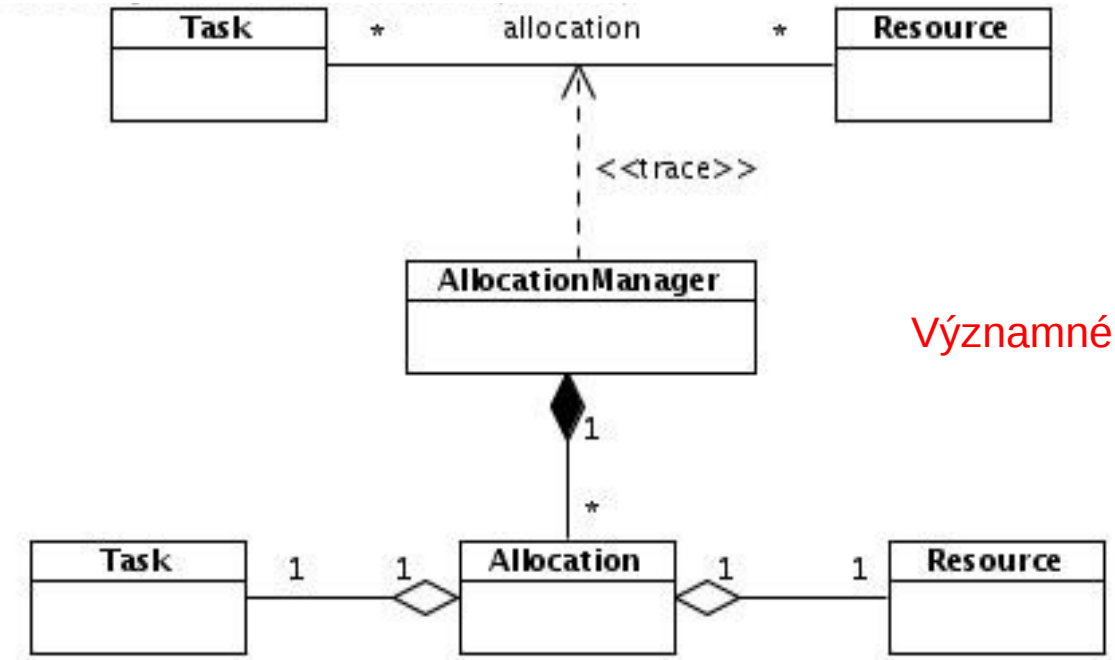
analýza



Významnější je směr řízení je Resource -> Task, ”allocation” asociace se změní ve třídu „Allocation”

návrh

analýza

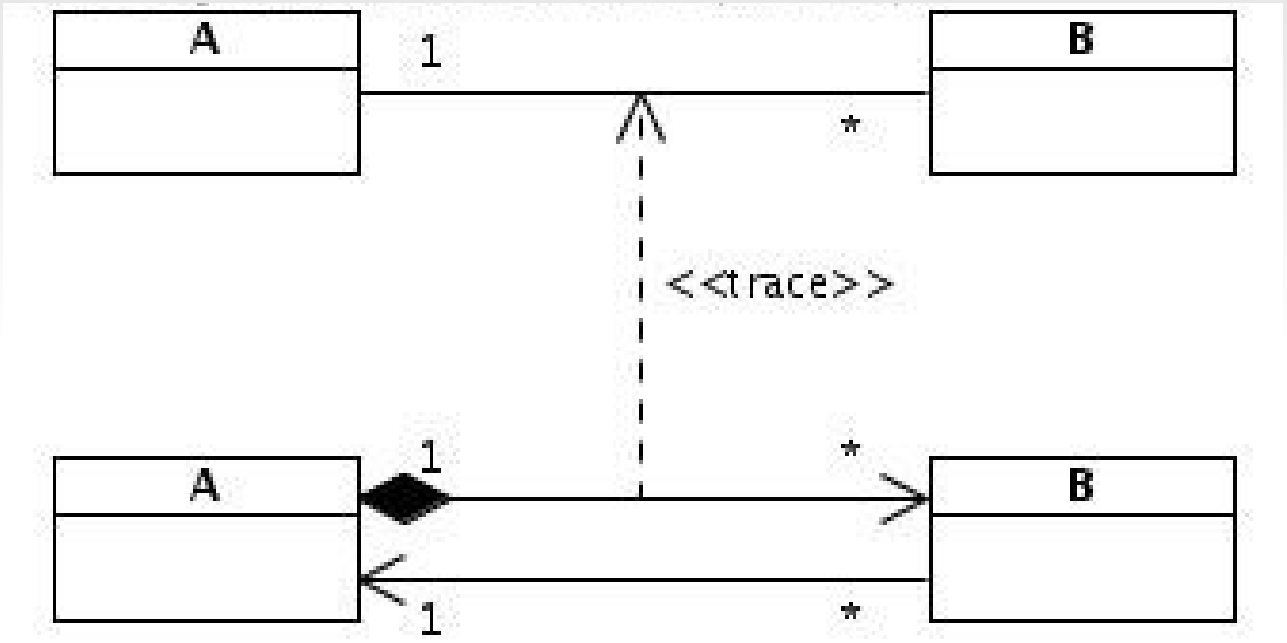


návrh

Významné jsou oba směry řízení (přístupu)

Návrhové třídy: Obousměrné asociace

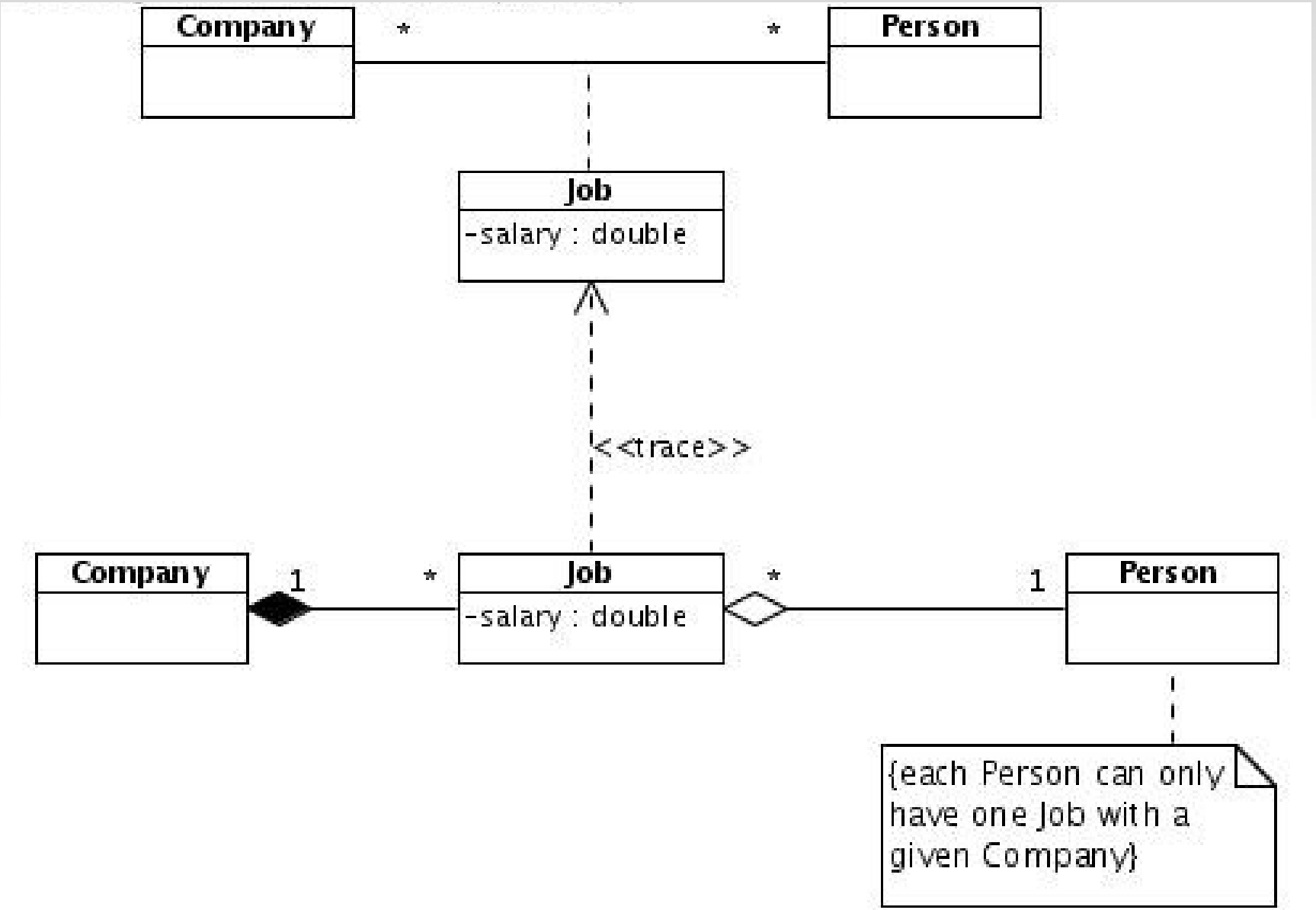
analýza



návrh

Návrhové třídy: Asociační třídy

analýza



návrh