

# Úvod do softwarového inženýrství

## IUS

### 2024/2025

#### 3. přednáška

Ing. Radek Kočí, Ph.D.  
Ing. Bohuslav Křena, Ph.D.  
*doc. Ing. Jaroslav Zendulka, CSc.*

30. září a 4. října 2024

# Téma přednášky

- Strukturovaná analýza a návrh
- Entity Relationship Diagrams (ERD)

# Přístupy k analýze a návrhu

Základní přístupy k analýze a návrhu:

- **Strukturovaný**  
Systém je chápán jako kolekce funkcí (procesů) operujících nad daty.
- **Objektově orientovaný**  
Systém je chápán jako kolekce vzájemně komunikujících objektů.

# Strukturovaný přístup k analýze a návrhu

## Konceptuální model

- vyjadřuje podstatu systému
- říká, co má systém dělat
- obsahuje sémantický model dat
- vymezuje, co budeme sledovat, ne jak to budeme realizovat

## Logický model

- definuje, jak bude konceptuální struktura dat implementována
- modely: lineární, síťový, relační, objektově orientovaný, ...

## Fyzický model

- model fyzického uspořádání dat (soubory, ...)

# Základní konceptuální modely

## Funkční (procesní) modelování

- základní model strukturované analýzy
- ukazuje funkce systému, toky dat mezi systémem a okolím a mezi funkcemi, data ukládaná v systému
- diagram datových toků (Data Flow Diagram – DFD)

## Minispecifikace

- popis funkcí (procesů) – co dělají

## Datové modelování

- ukazuje entity aplikační domény zpracovávané systémem a statické vztahy mezi nimi (typicky perzistentní data ukládaná v databázi)
- důležitý model datově intenzivních aplikací
- zásadní význam pro návrh databáze
- diagram entit a vztahů (Entity Relationship Diagram – ERD)

# Základní konceptuální modely

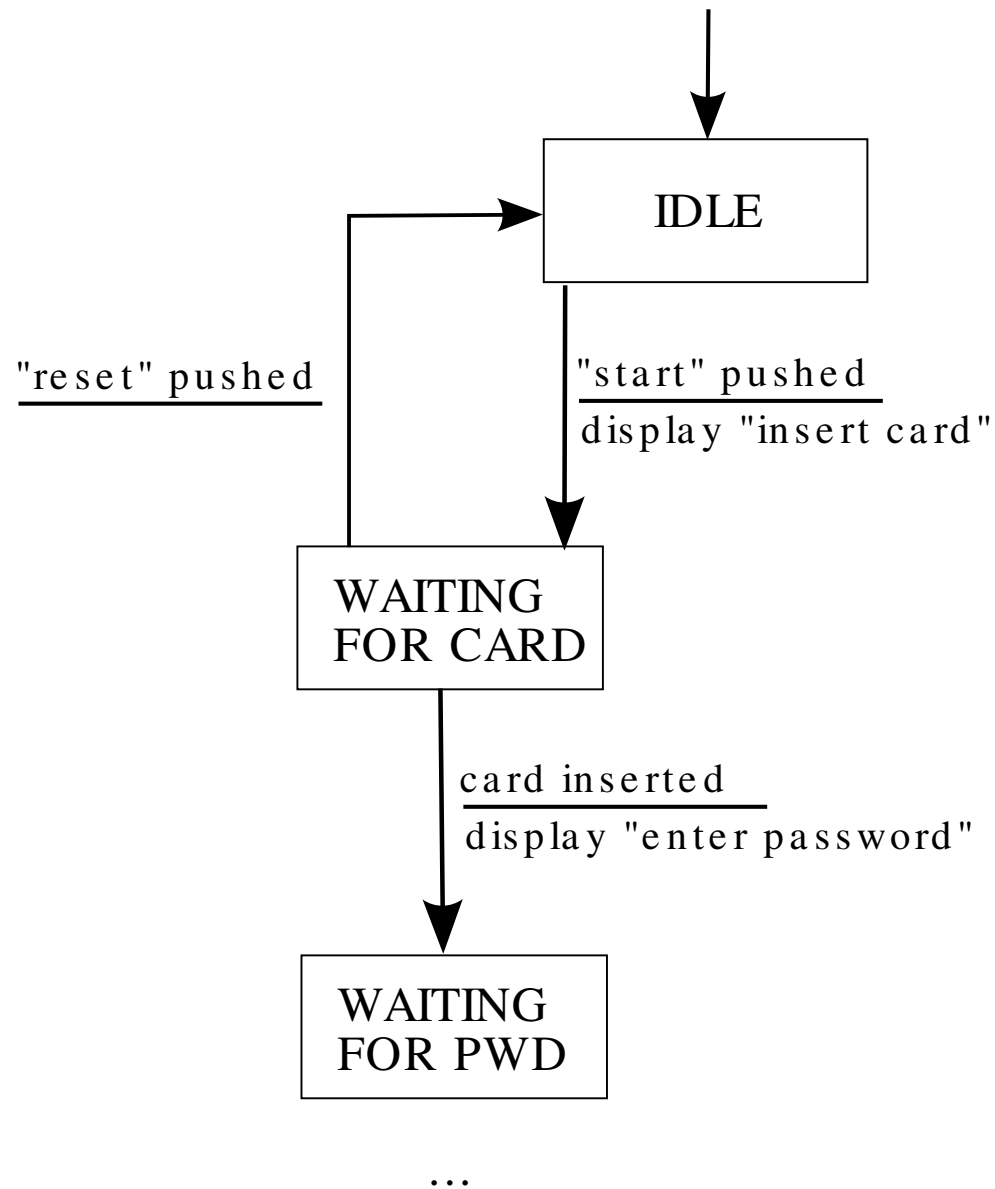
## Datový slovník

- obsahuje specifikace prvků modelů
- notace pro specifikaci informačního obsahu prvků DFD a ERD

## Stavový diagram (State-Transition Diagram – STD)

- Modeluje dynamické chování systému nebo jeho části.
  - stavy – zachycují určitou situaci (počáteční, koncové)
  - přechody – změny stavů
  - podmínky – externí události ovlivňující proveditelnost přechodů
  - akce – události jako komunikace, výpočet, ...
  -
- Teoretický koncept: konečný automat
- UML: stavový diagram

# Stavový diagram



# Data Flow Diagram (DFD)

## Data Flow Diagram

- je technika používaná při strukturované analýze a návrhu pro specifikaci chování systému,
- je hierarchický model, který ukazuje funkce systému, toky dat mezi systémem a okolím a toky dat mezi funkcemi a datovými sklady,
- je tedy blíže návrhu,
- je doplněn minispecifikacemi.

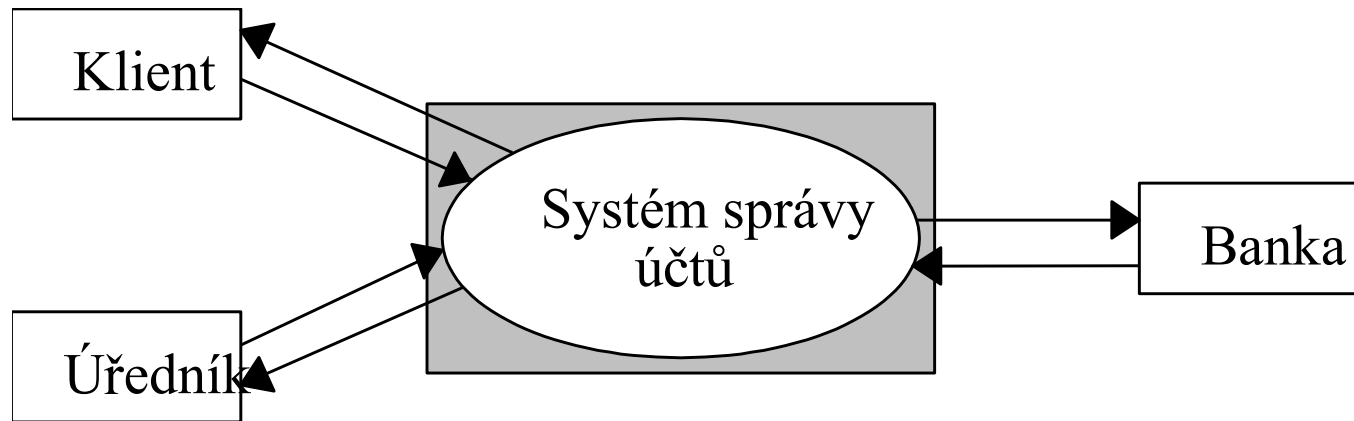


# Příklad pro DFD – systém správy účtů

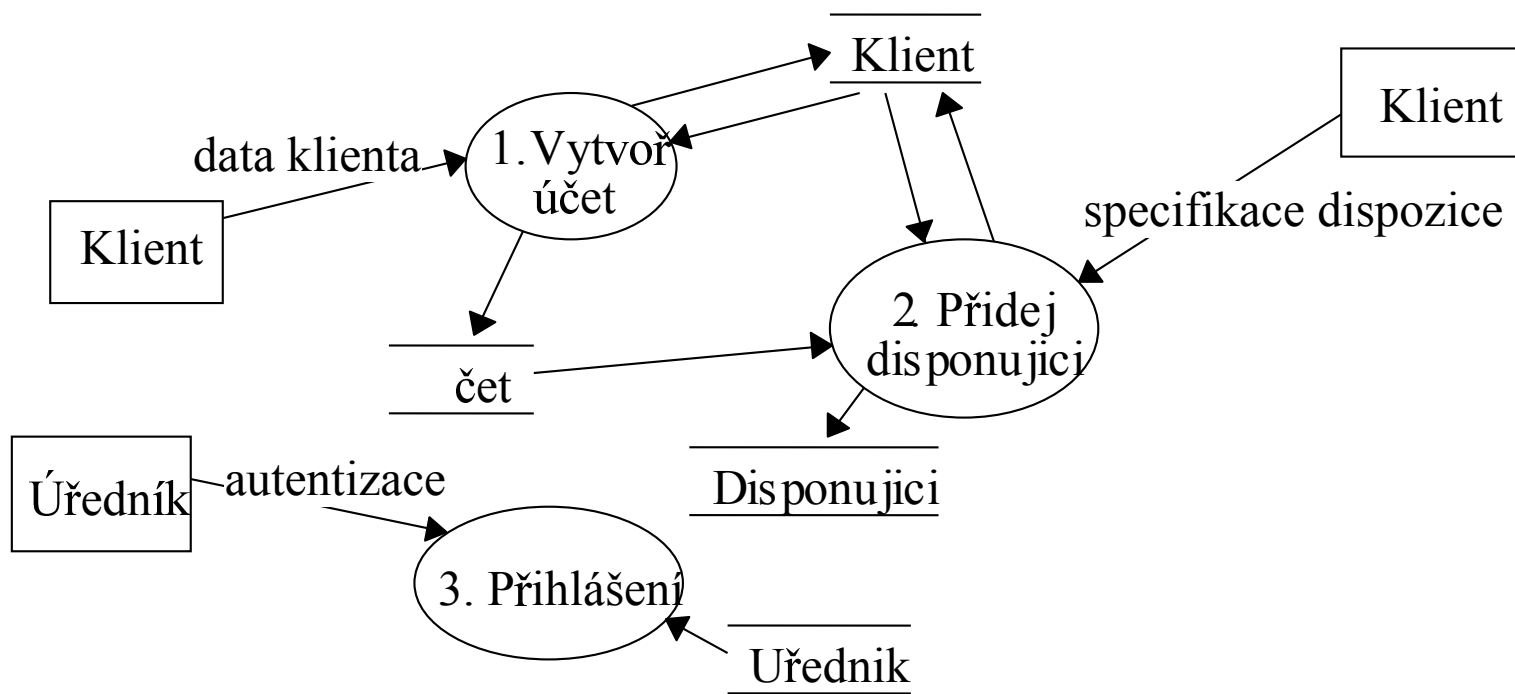
Provádíme analýzu systému správy účtů banky. Každý účet má jednoznačné číslo, dále je potřeba znát jméno a adresu majitele účtu. Kromě majitele mohou s účtem disponovat i další jím určené osoby. O těch je třeba znát stejné údaje jako o majiteli. Každá z disponujících osob může mít stanoven limit pro výběr z daného účtu. S účty manipuluje úředník banky na základě příkazu osoby oprávněné s účtem disponovat.

Na účet lze provádět vklad, z účtu lze provádět výběr a lze převádět částky na jiné účty v téže nebo jiné bance. Musí být k dispozici informace, kdo příkaz zadal a který úředník ho provedl. Systém musí poskytovat prostředky pro správu informací o klientech banky, musí umožňovat vytvářet a rušit účty, zadávat příkazy, importovat příkazy pro převody z jiných bank a naopak exportovat příkazy pro převody na účty v jiných bankách. Systém musí být schopen tisknout měsíční výpisy z účtů a řadu dalších tiskových sestav.

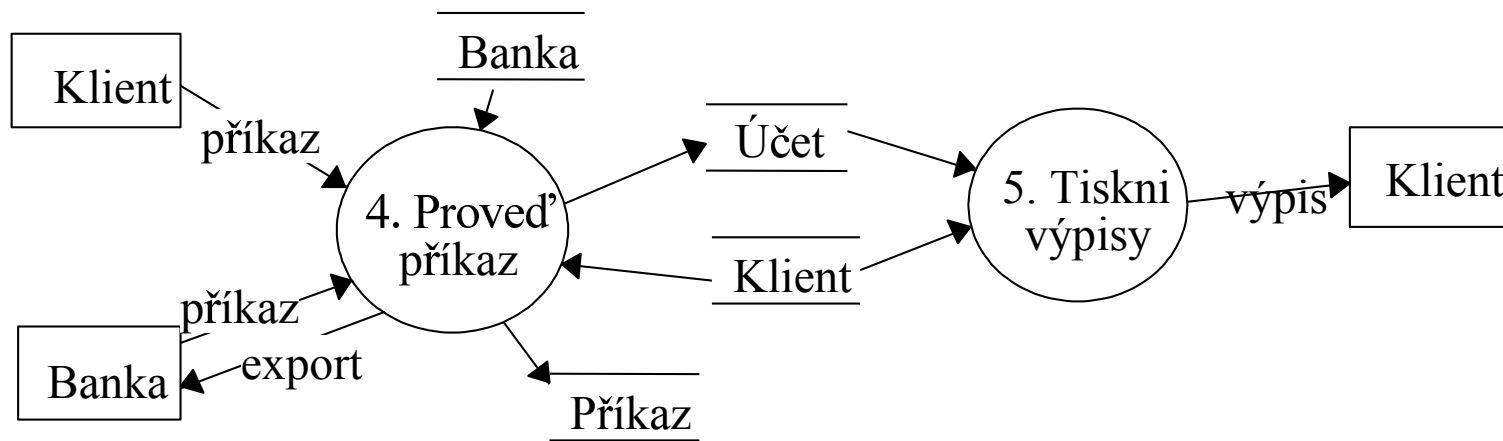
# Příklad DFD (1/3)



# Příklad DFD (2/3)



# Příklad DFD (3/3)



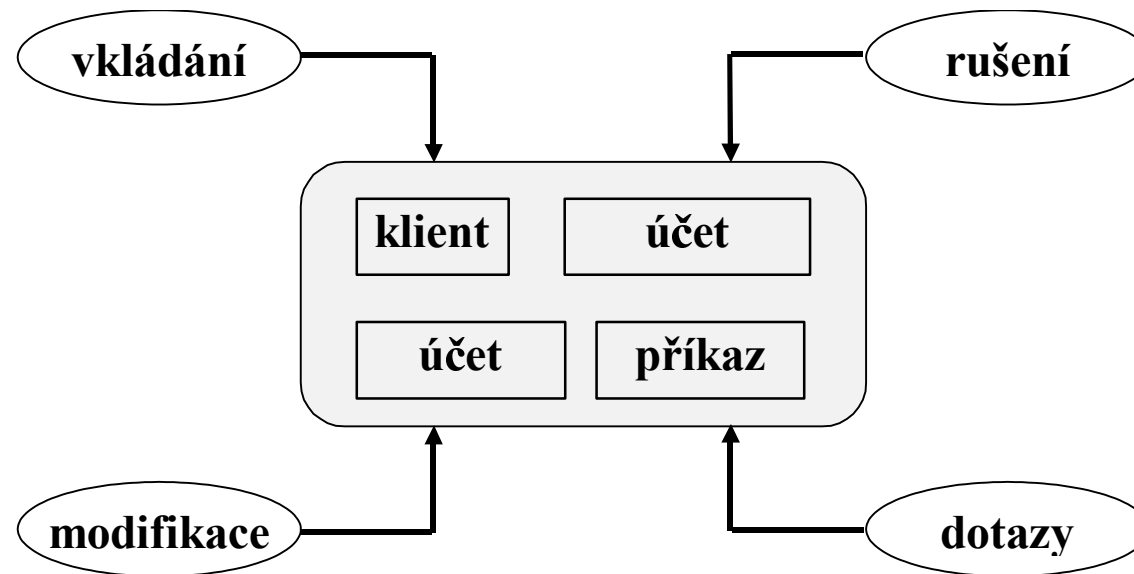
# Datové modelování

## Cíle návrhu datově intenzivních systémů

- mít v systému všechna potřebná data
- nemít v systému žádná nepotřebná data
- vyjádřit vztahy mezi daty
- popsat transformaci dat v systému

# Datové modelování – ER model

- Slouží k modelování dat aplikační domény a jejich vztahů „v klidu“.

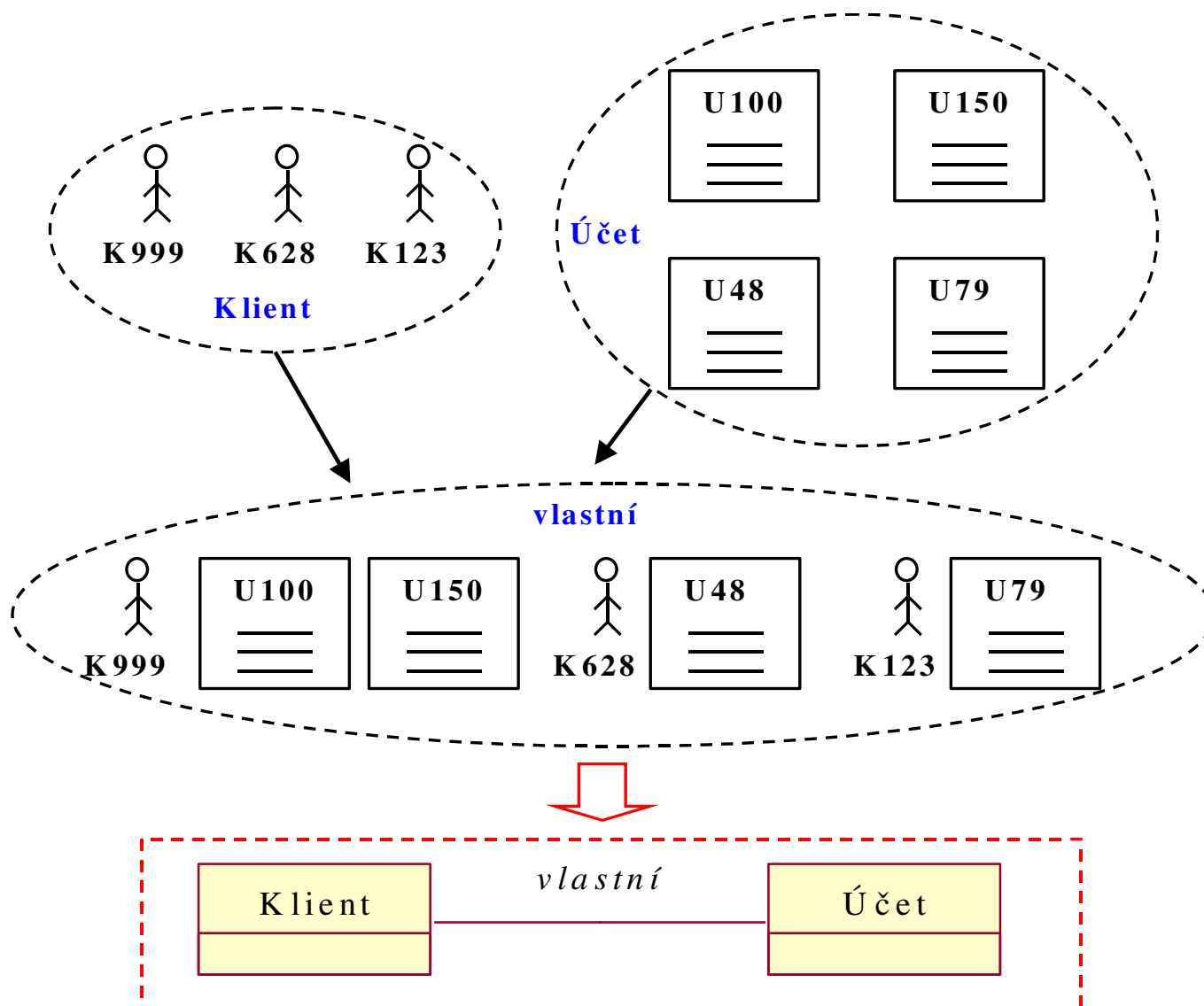


- Která data potřebujeme v systému uchovávat?
- Jaké jsou mezi nimi vztahy?

# Datové modelování – základní pojmy

- **Entita** – „věc“ reálného světa (objekt) rozlišitelný od jiných objektů.  
Např. klient banky s identifikačním číslem K999, účet číslo U100
- **Entitní množina** – množina entit téhož typu, které sdílí tytéž vlastnosti (atributy).  
Např. Klient, Účet
- **Atribut** – vlastnost entity, která nás v kontextu daného problému zajímá.  
Např. Klient: čísloKlienta, jméno, příjmení, adresa, ...
- **Vztah** – asociace mezi několika entitami.  
Např. klient s číslem klienta K999 vlastní účet s číslem účtu U100.
- **Vztahová množina** – množina vztahů téhož typu, které sdílí tytéž vlastnosti.  
Např. Klient vlastní Účet – pro vztah mezi entitami typu Klient a Účet

# Tvorba ER diagramu





# Typy atributů

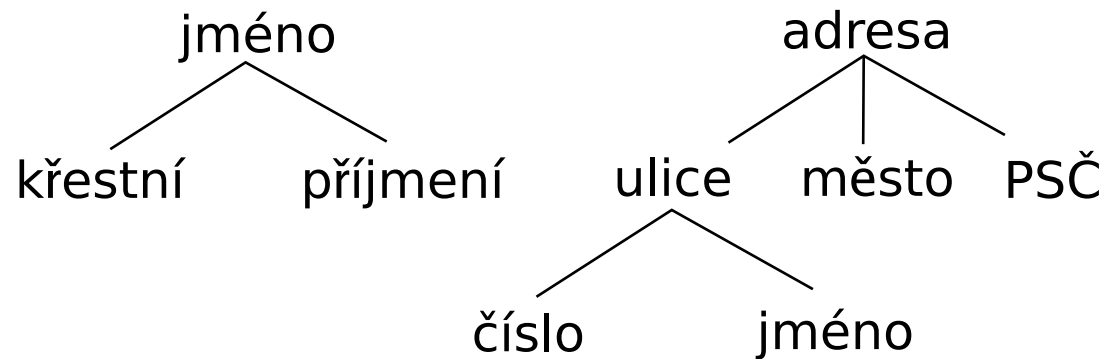
- jednoduché (simple) a složené (composite) atributy

Entitní množina

Klient

Složené  
atributy

Složky  
atributu

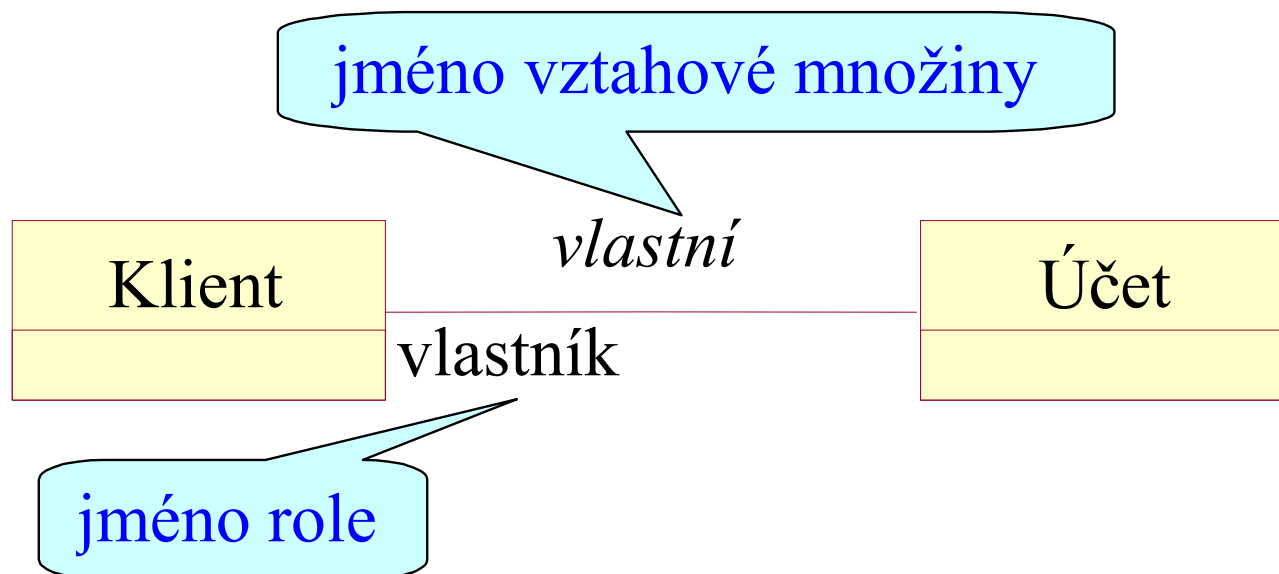


# Typy atributů

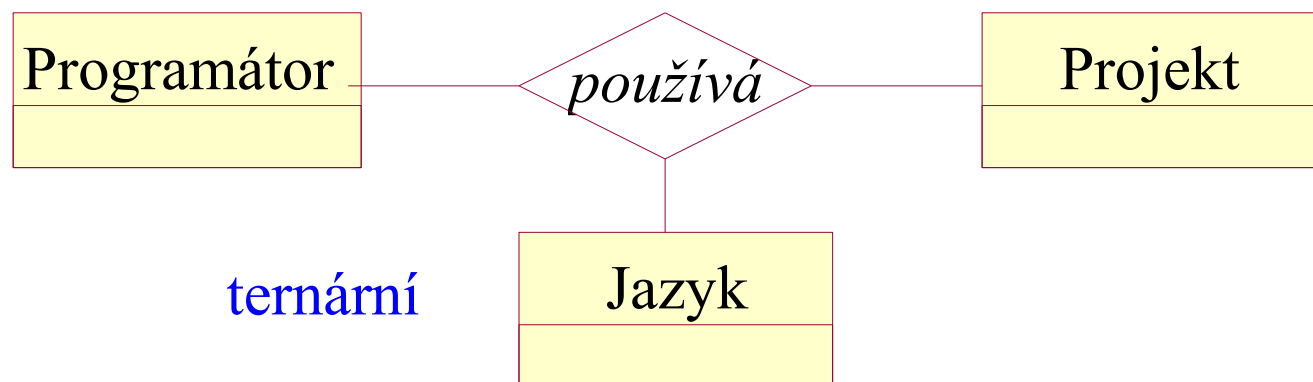
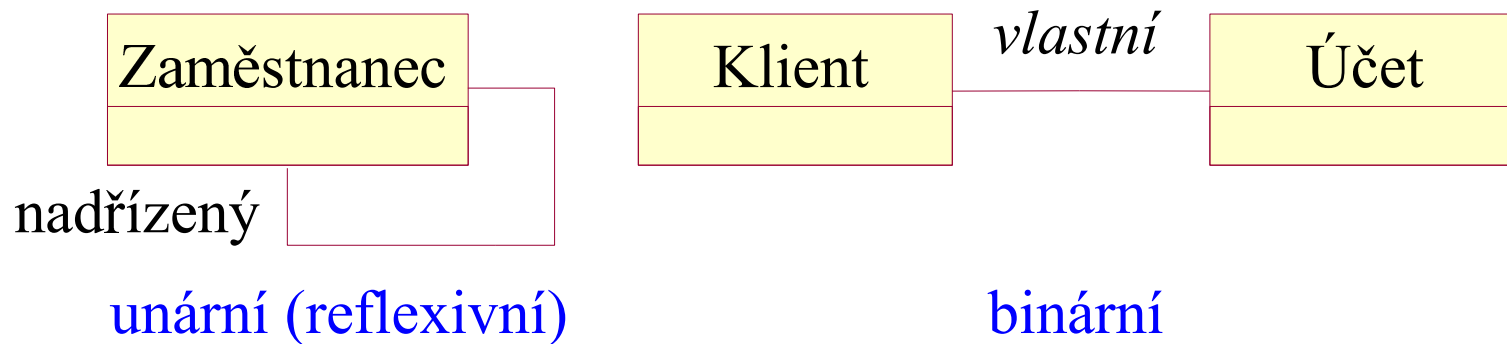
- jednohodnotové (single-valued) a vícehodnotové (double-valued) atributy
  - např. telefon – může být více čísel
  - lze omezit minimální a maximální počet hodnot
- prázdné (NULL) atributy
  - mohou nabývat speciální hodnoty NULL
  - může zastupovat chybějící hodnotu – existuje, ale neznáme ji
  - může zastupovat neznámou hodnotu – nevíme, zda existuje
- odvozené atributy
  - hodnotu lze odvodit od jiných atributů nebo entit
  - např. datumNarození  $\Rightarrow$  věk

# Parametry vztahů

Jméno vztahové množiny i jméno role vyjadřuje význam vztahu.



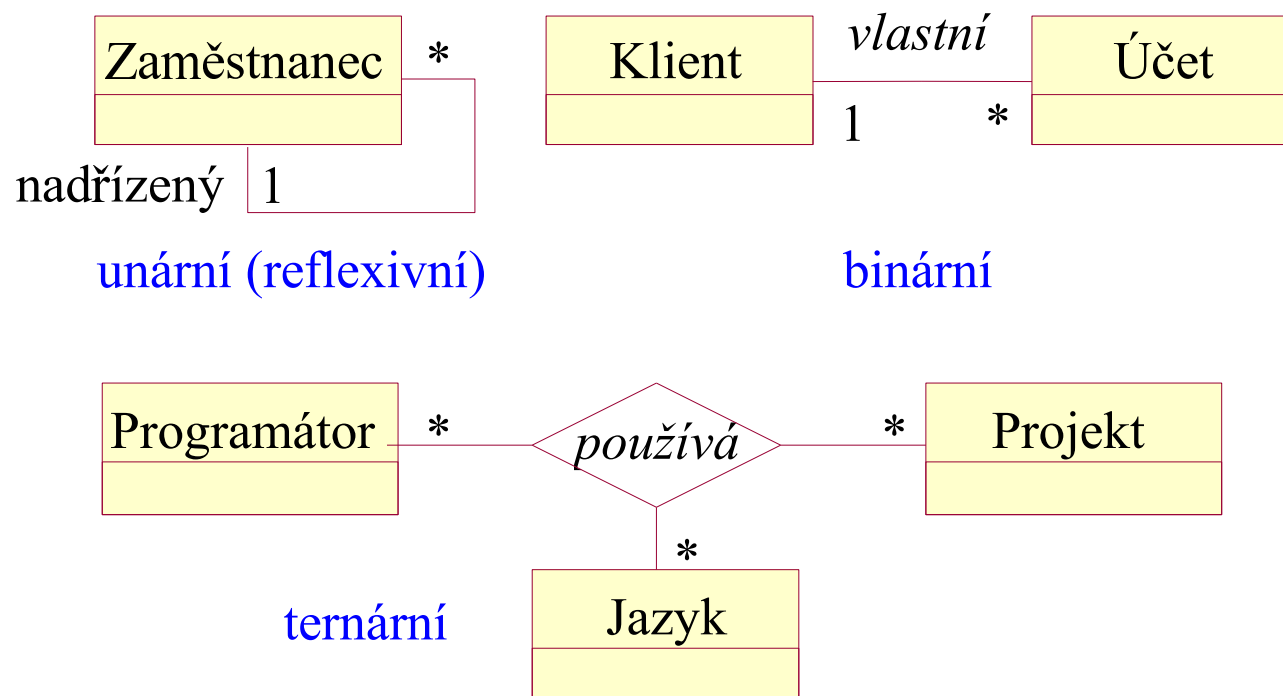
# Stupeň



# Kardinalita

Kardinalita (*cardinality*) je maximální počet vztahů daného typu (vztahové množiny), ve kterých může participovat jedna entita.

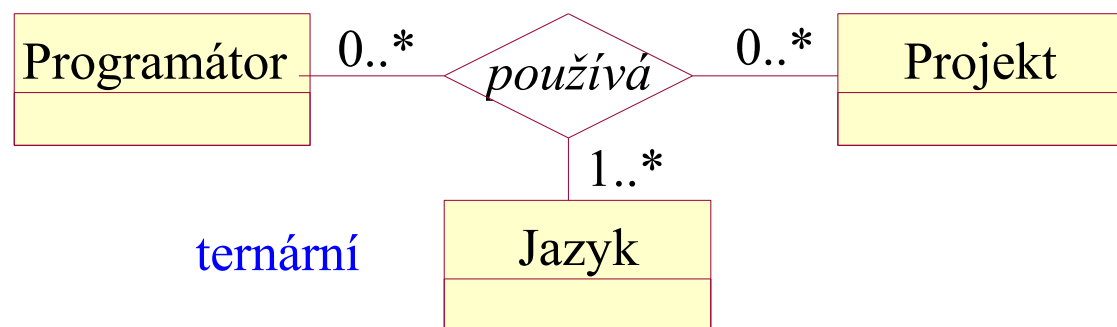
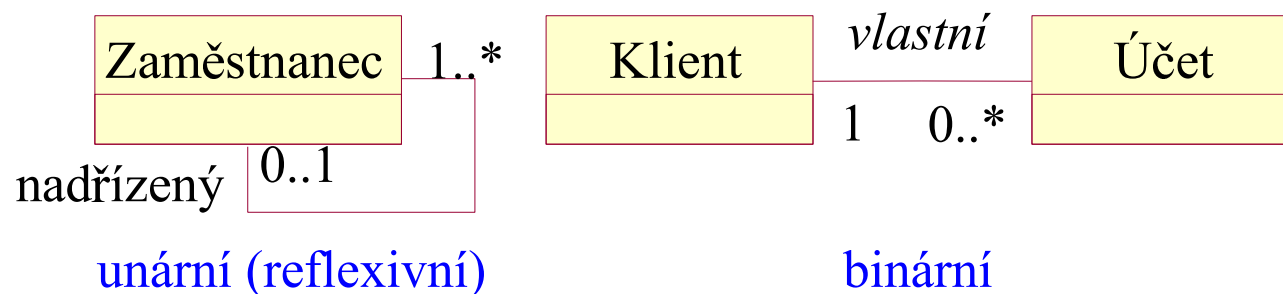
Typické hodnoty: 1, M, případně přesněji



# Členství / účast

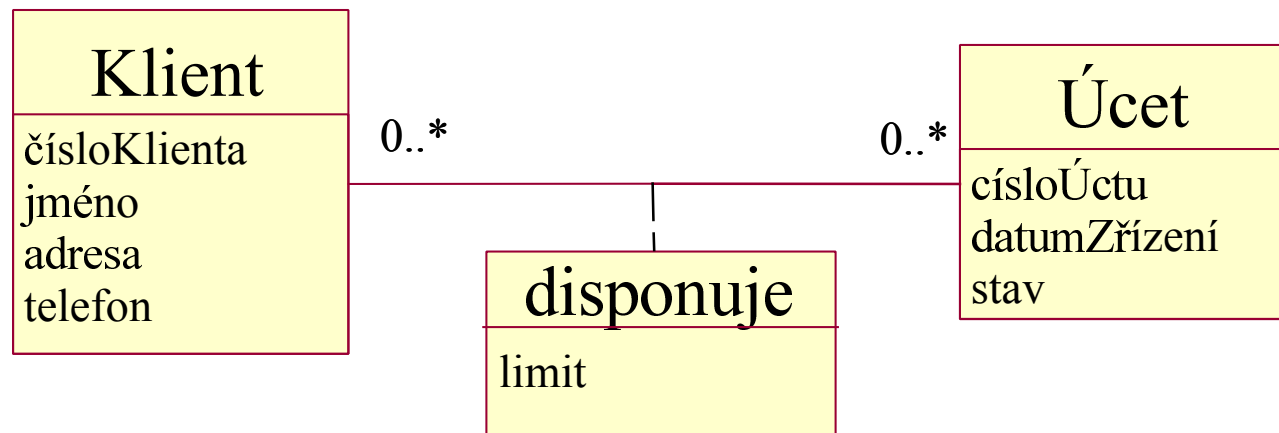
Členství (*membership*) / účast (*participation*) je minimální počet vztahů daného typu (vztahové množiny), ve kterých musí participovat jedna entita.

Typické hodnoty: 0 – volitelné, 1 – povinné

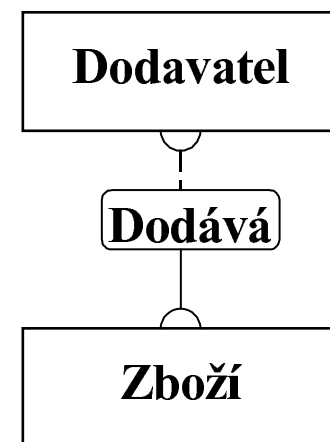
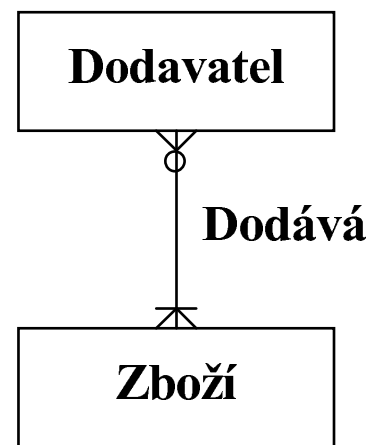
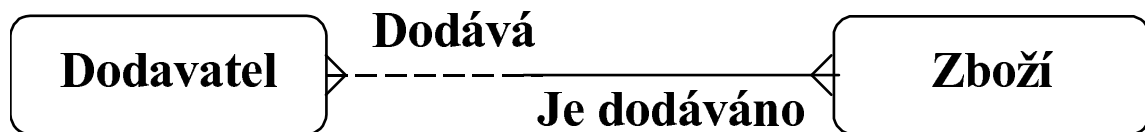
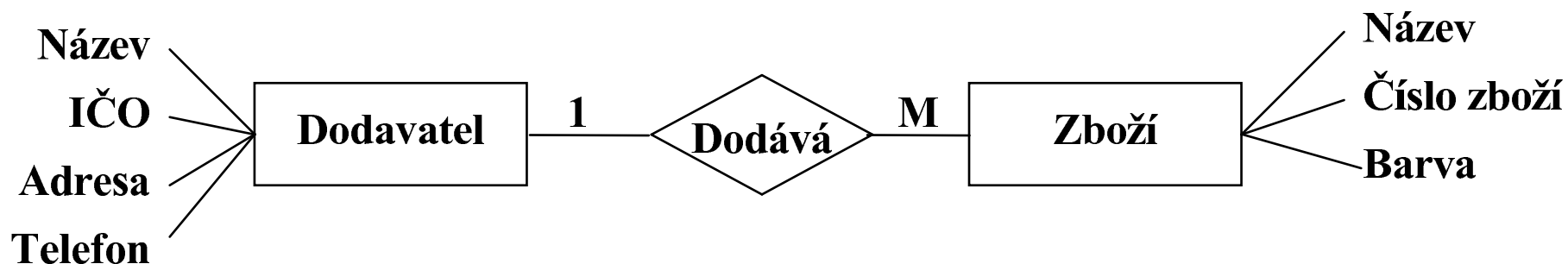


# Atributy vztahu

- Použijeme tehdy, když atribut nelze přiřadit ani jedné z entit.
- Jedná se o vztah povýšený na entitu.



# Alternativní notace ERD





# Pravidla návrhu ERD

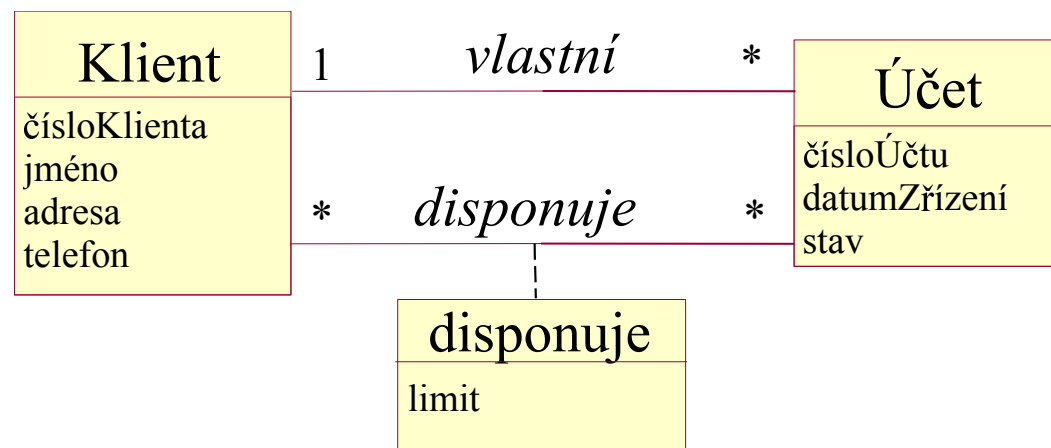
- zobrazujeme pouze data a jejich vztahy, žádné procesy
- každý atribut zobrazujeme pouze jednou
- seskupujeme data pro účely databáze, ne výstupních sestav
- zobrazujeme pouze perzistentní datové objekty
- zobrazujeme pouze nezbytně nutné vztahy
  - Učitel učí Predmet, který má zapsaný Student
  - Učitel učí Student  $\Rightarrow$  redundantní

## Pozor na entity

- bez atributů
- mají pouze identifikátor
- mají pouze jeden výskyt
- obsahující atributy patřící jiným entitám (cizí atributy)

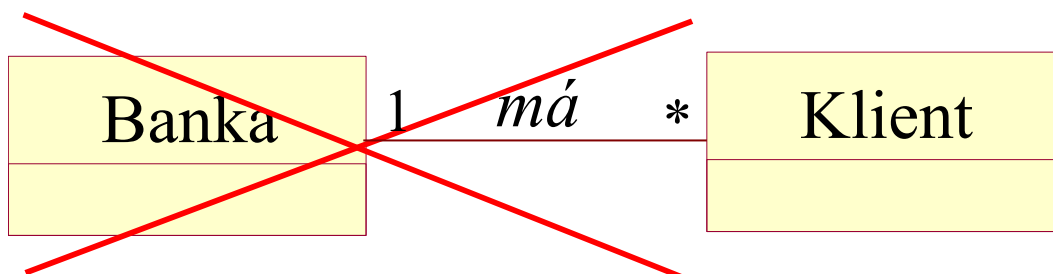
# Doporučení pro tvorbu ERD (1.)

- Jména
  - Musí být srozumitelná a musí vyjadřovat význam entitních a vztahových množin.
  - entitní množiny: podstatná jména
  - vztahové množiny: slovesa, předložky
  - Je-li jméno vztahové množiny jasné ze jmen entitních množin, není nutné uvádět.
- Mezi stejnými entitními množinami může být několik vztahových množin.



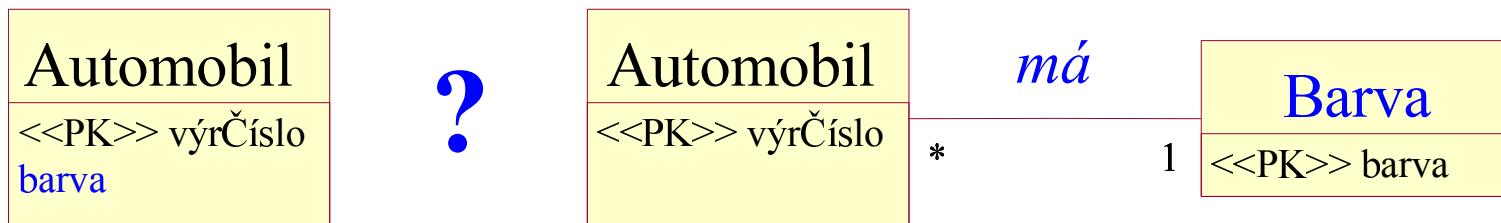
# Doporučení pro tvorbu ERD (2.)

- Identifikátor (klíč, primární klíč)
  - Entity a vztahy musí být identifikovatelné.
  - Hodnota identifikátoru musí být unikátní (a minimální).
  - Identifikátorem je jednoduchý nebo složený atribut.
  - Unikátnost hodnoty jen v rámci vyvíjeného systému (ne celého vesmíru).
- Celkový systém by neměl být zahrnut do ERD.



# Doporučení pro tvorbu ERD (3.)

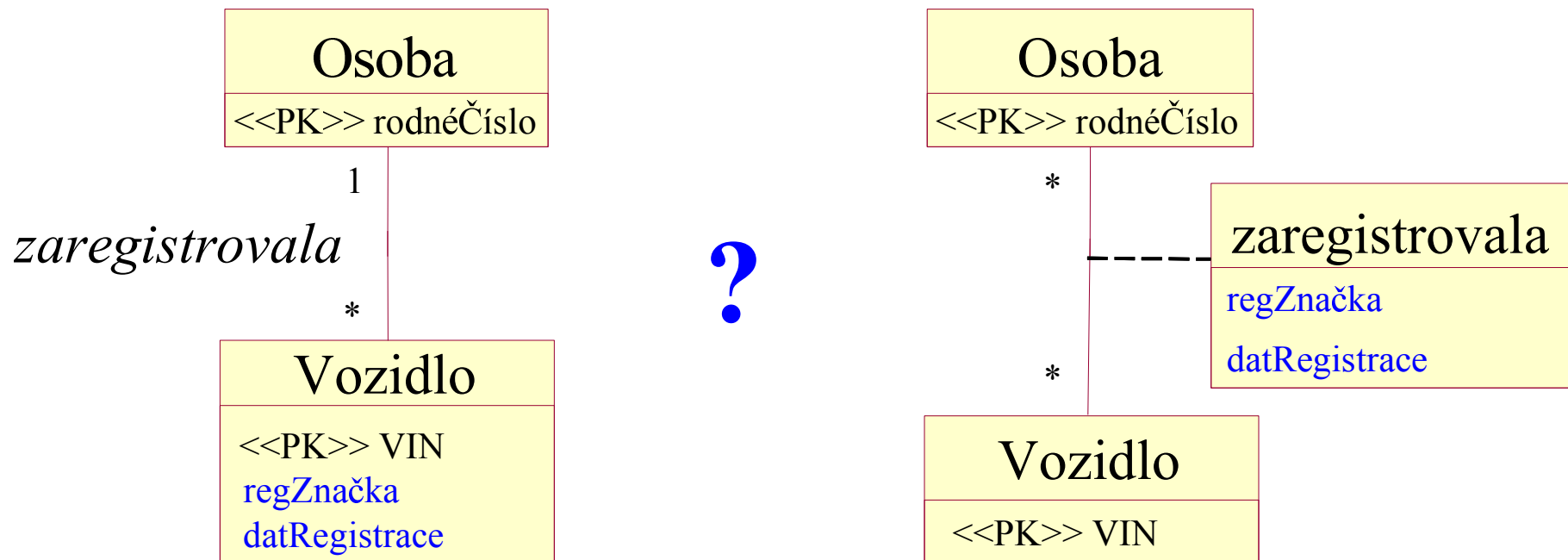
- Použít entitní množinu nebo atribut?



**Pravidlo:** Je-li hodnota atributu důležitá, i když neexistuje žádná entita s touto hodnotou jako vlastností, pak bychom ji měli modelovat jako entitu.

# Doporučení pro tvorbu ERD (4.)

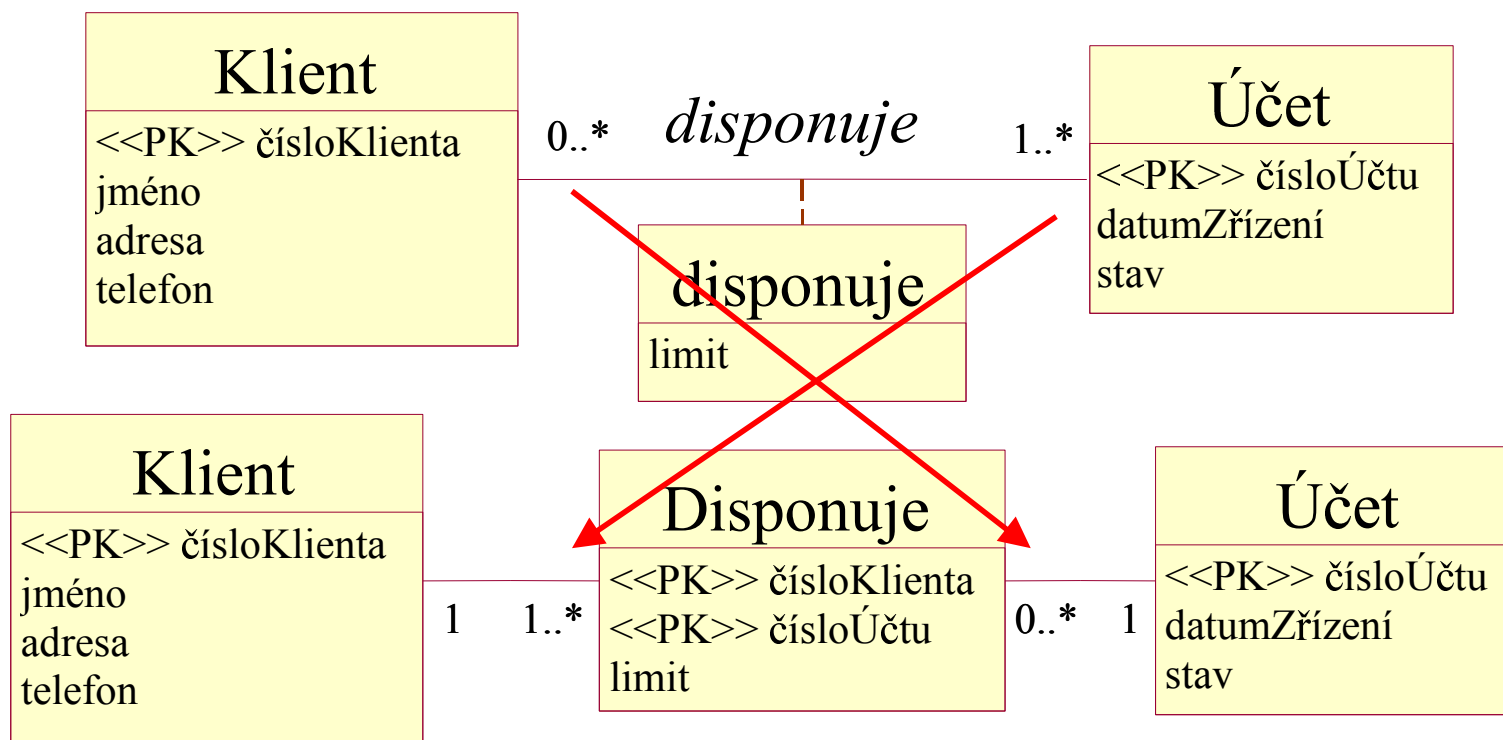
- Kardinalita a umístění atributů



Bude záležet na tom, zda budeme chtít uchovávat i historii registrací.

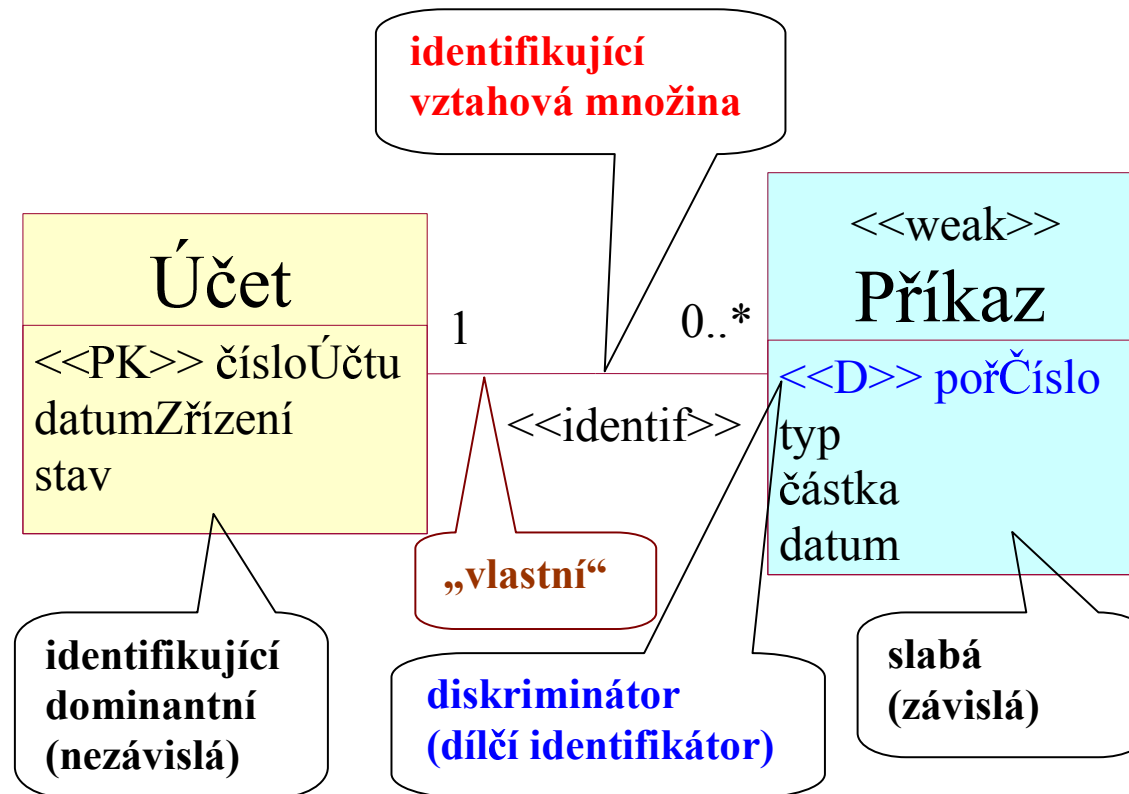
# Doporučení pro tvorbu ERD (5.)

- Náhrada vztahů M:M vazební entitní množinou



# Slabé (weak) entitní množiny

- Silná (strong) entitní množina **má** identifikátor tvořený vlastními atributy.
- Slabá entitní množina **nemá** identifikátor tvořený vlastními atributy.



# Identifikace slabé entitní množiny

- Rysy slabé entitní množiny:
  - identifikátor = identifikátor\_dominantní + diskriminátor
  - existenční závislost slabé na identifikující
- Slabá nebo silná entitní množina?
  1. Jako slabou modelovat tehdy, kdy entita kompletně zmizí při odstranění odpovídající identifikující entity.  
Příklad: Objednávka - PoložkaObjednávky
  2. Cokoliv s atributem, který je jednoznačný, by nemělo být modelováno jako slabá entitní množina.
  3. Jsme-li na pochybách, modelujeme jako silnou entitní množinu.



# Rozšíření ER modelu

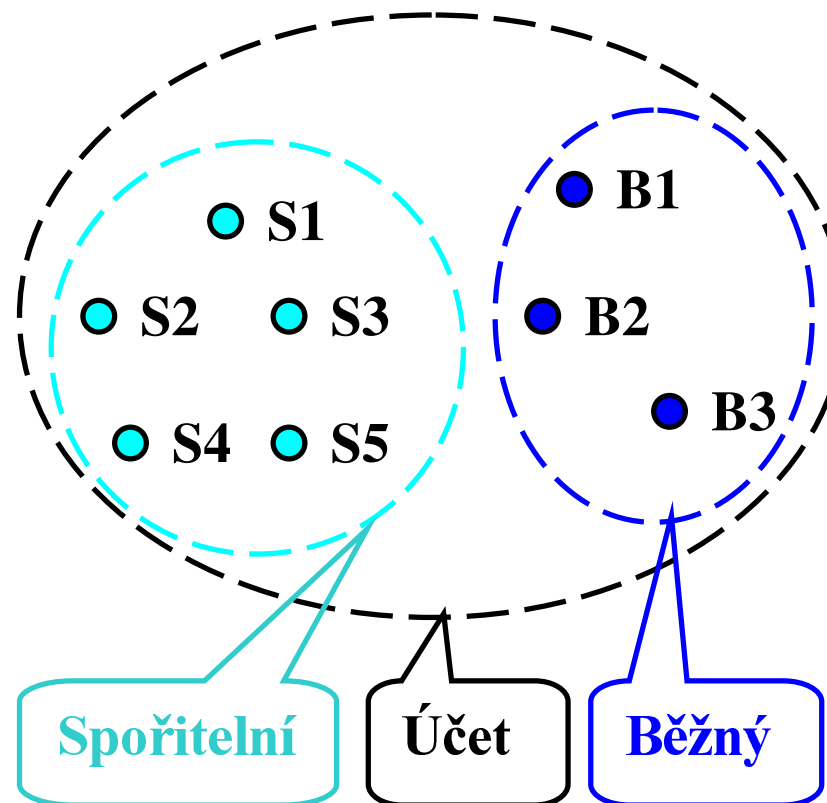
## Enhanced Entity-Relationship (EER) Modeling

- **Zobecnění množin** (generalizace/specializace), vztah **is-a**
- *kategorie (typ UNION)*
- *dědičnost atributů a vztahů*

# Zobecnění (generalizace/specializace)

## Zobecnění/Specializace entit

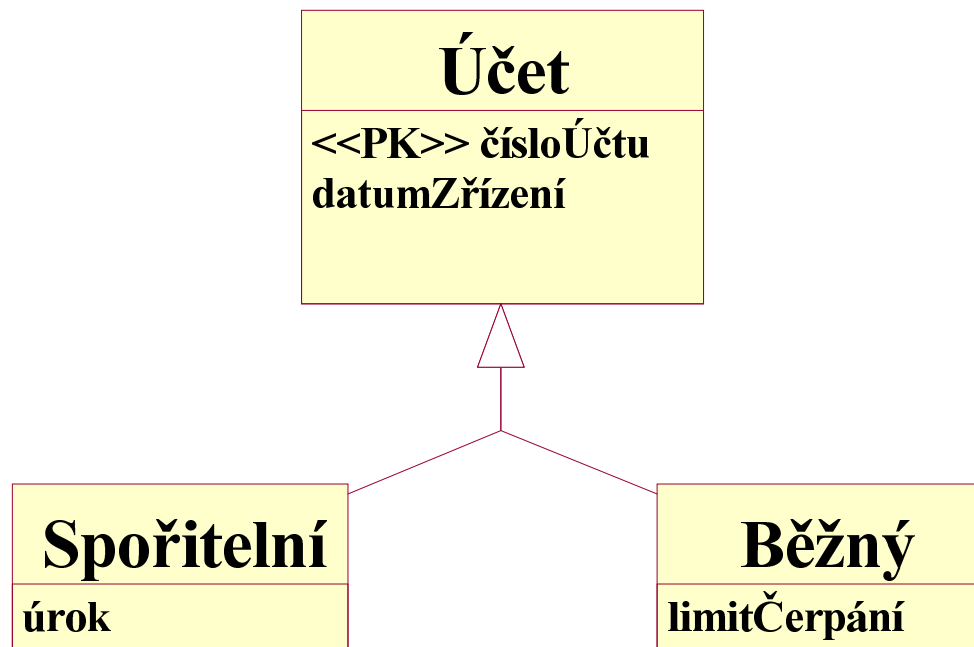
- entity mající stejný základ (atributy) lišící se v některých attributech
- Př.: entity *Spořitelní účet* a *Běžný účet* jsou speciální variantou *Účet*



# Zobecnění (generalizace/specializace)

## Zobecnění/Specializace entitních množin

- dědičnost atributů a účasti ve vztahových množinách
- hierarchie generalizace (podobně v OO přístupu)  
Př.: *Spořitelní* je (is a) *Účet*
- identifikátor entitních množin nižší úrovně je stejný jako vyšší



# Zobecnění (generalizace/specializace)

## Omezení generalizace/specializace

- **příslušnost** – příslušnost entity do jedné nebo do více specializovaných entitních množin (EM)
  - disjunktí – entita může být součástí nejvýše jedné specializované EM (*účet může být buď spořitelní nebo běžný*)
  - překrývající se – entita může být součástí více specializovaných EM (*účet je spořitelní i běžný*)
- **úplnost** – zda každá entita z vyšší úrovně musí nebo nemusí patřit do jedné z EM na nižší úrovni
  - úplná specializace (obvyklá) – každá entita z obecné EM je součástí EM na nižší úrovni (*každý účet je vždy spořitelní nebo běžný*)
  - částečná specializace – entita z obecné EM nemusí patřit do žádné EM na nižší úrovni (*účet, spořitelní účet, běžný účet*)

## Převod na relace (tabulky)

- závisí na požadovaných vlastnostech a omezeních (1 / 2 / 3 relace)

# Postup při návrhu ERD

## Základní kroky

1. zvolte jednu entitu ze specifikace požadavků
2. určete atributy entity, označte kandidátní klíče
3. prověřte atributy, zda je potřeba zaznamenat informace o některém z atributů v samostatné entitě
4. další entita  $\Rightarrow$  krok 1
5. vytvořte vztahy mezi entitami
6. určete, zda některé atributy potřebují být identifikovány pomocí více entit  
 $\Rightarrow$  atribut přiřaďte vztahu, který spojuje příslušné entity
7. identifikujte a odstraňte redundantní vztahy

## Poznámka

1. entita se modeluje jako entitní množina
2. vztah se modeluje jako vztahová množina

# Příklad pro ERD – systém správy účtů

Provádíme analýzu systému správy účtů banky. Každý účet má jednoznačné číslo, dále je potřeba znát jméno a adresu majitele účtu. Kromě majitele mohou s účtem disponovat i další jím určené osoby. O těch je třeba znát stejné údaje jako o majiteli. Každá z disponujících osob může mít stanoven limit pro výběr z daného účtu. S účty manipuluje úředník banky na základě příkazu osoby oprávněné s účtem disponovat.

Na účet lze provádět vklad, z účtu lze provádět výběr a lze převádět částky na jiné účty v téže nebo jiné bance. Musí být k dispozici informace, kdo příkaz zadal a který úředník ho provedl. Systém musí poskytovat prostředky pro správu informací o klientech banky, musí umožňovat vytvářet a rušit účty, zadávat příkazy, importovat příkazy pro převody z jiných bank a naopak exportovat příkazy pro převody na účty v jiných bankách. Systém musí být schopen tisknout měsíční výpisy z účtů a řadu dalších tiskových sestav.

# Příklad pro ERD – systém správy účtů

# ERD – elektronická evidence diamantů (1/2)

*Zadání použité na zkoušce dne 12. ledna 2016 ve 14:00:*

*Toto zadání bylo inspirováno článkem PRAŠTĚNÁ POHÁDKA: Elektronická Evidence Trpaslíků, který Martin Jurica publikoval dne 31. 12. 2015 na Neviditelném psu. Sněhurka Vás požádala o vytvoření informačního systému pro elektronickou evidenci diamantů (EED), které vytěží trpaslíci. O každém trpaslíkovi EED eviduje jeho jméno (předpokládejte, že se žádní dva trpaslíci nejmenují stejně), datum narození, výšku a váhu, na které směny a na jaké pozici nastoupil (v rámci určité směny má trpaslík právě jednu pracovní pozici) a jaké pracovní nástroje má či v minulosti měl přiděleny. Každý pracovní nástroj má své unikátní číslo a v EED je veden jeho typ, hmotnost, datum nákupu, nákupní cena, aktuální stav nástroje případně datum vyřazení pro jeho nepoužitelnost. Pro jednoduchost předpokládejte, že jeden nástroj nebude stejnému trpaslíkovi přidělen opakovaně.*



# ERD – elektronická evidence diamantů (2/2)

Směna je určena datem a časem svého začátku (dvě směny nemohou začít ve stejném okamžiku) a v EED je evidována její délka, důl a patro, ve kterém probíhala. Směny a jejich místa (směna nikdy nebude zasahovat na více pater či dolů) jsou plánovány dopředu a trpaslíci a jejich pozice až v okamžiku nastoupení na směnu. Dále jsou evidovány všechny diamanty, které byly v rámci směny vytěženy, aby trpaslíci nemohli odnášet a prodávat diamanty bez vědomí Sněhurky. Každému vytěženému diamantu EED přidělí unikátní evidenční číslo a poté je zaznamenána jeho hmotnost, barva, čistota, tvar, odhad ceny i to, který trpaslík diamant vytěžil. EED bude dále uchovávat informace o prodeji diamantů, přičemž každý prodej bude mít své číslo a dále bude v EED uloženo datum prodeje, kupec (stačí jeho jméno), které diamanty koupil a za jakou cenu (celková cena nestačí, je třeba evidovat prodejní cenu každého diamantu), a který trpaslík za prodej odpovídal (aby Sněhurka mohla zkontrolovat, zda některý trpaslík neprodal diamanty nápadně nevýhodně).

*Zadání a (komentovaná) vzorová řešení ER diagramů ze zkoušek z let 2011/12, 2015/16 a 2017/18 najdete v Moodlu – Studijní materiály.*

# ERD – elektronická evidence diamantů

# Studijní koutek – Tituly a oslovení

## Akademické tituly

- Bc. – **bakalář** (angl. *bachelor*, lat. *baccalaureus* = vavřínem ověnčený)  
BcA. – bakalář umění (lat. *baccalaureus artis*)
- Ing. – **inženýr** (angl. *engineer* = strojník)  
Ing. arch. – inženýr architekt  
Mgr. – magistr (doslovně učitel)  
MgA. – magistr umění
- RNDr. – doktor přírodních věd (*rerum naturalium doctor*)  
MUDr. – doktor všeobecného lékařství (*medicinae universae doctor*)  
JUDr. – doktor práv (*juris utriusque doctor*)  
MDDr., MVDr., PhDr., PaedDr., PharmDr., ThDr., ThLic., RSDr., RTDr., ...  
⇒ asistent, lektor
- MBA – (angl. *Master of Business Administration*)  
navazující studium zaměřené na management
- LL.M. – (angl. *Master of Law*, lat. *Legum Magister*)

# Studijní koutek – Tituly a oslovení

## Vědecké hodnosti

- Ph.D. – **doktor** (lat. *philosophiae doctor*)  
Th.D. – (lat. *theologiae doctor*)  
Dr. – doktor = učený  
CSc. – kandidát věd (*candidatus scientiarum*)  
⇒ odborný asistent
- DrSc. / DSc. – doktor věd (lat. *doctor scientiarum*)
- akad. – akademik, tj. řádný člen ČSAV

## Vědecko-pedagogické hodnosti

- doc. – **docent**
- prof. – **profesor**  
Učitelům na střední škole se říká profesor, přestože titul prof. nemají.

## Čestná hodnost

- dr. h. c. – doctor honoris causa

# Studijní koutek – Tituly a oslovení

## Oslovení při běžných příležitostech

- Nejvyšším dosaženým titulem nebo významnou funkcí
- rektore, prorektore, děkane, proděkane, řediteli, ...
- profesore, docente, doktore, inženýre, magistře, bakaláři, ...

## Oslovení při akademických obřadech

- rektor – *Vaše Magnificence* (vznešenosti)
- děkan – *Spectabilis* (slovutný), množné číslo *Spectabiles*
- prorektor, proděkan, promotor – *Honorabilis* (ctihodný), množné číslo *Honorabiles*
- Prorektor zastupující rektora je oslovován jako rektor.
- Proděkan zastupující děkana je oslovován jako děkan.