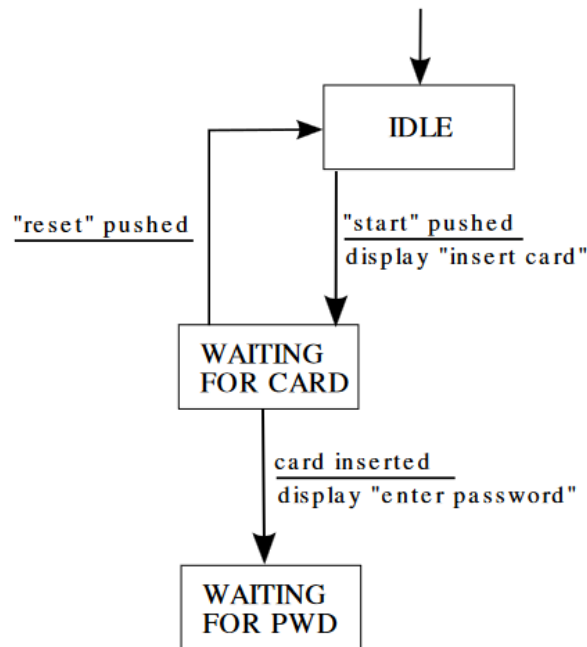


# Strukturovaná analýza a návrh

- **Základní přístupy k analýze a návrhu:**
  - o **Strukturovaný**
    - **Systém je chápán jako kolekce funkcí (procesů) operujících nad daty.**
  - o Objektově orientovaný
    - Systém je chápán jako kolekce vzájemně komunikujících objektů
- **Strukturovaný přístup**
  - o **Konceptuální model**
    - **Vyjadřuje podstatu systému**
    - **Říká, co má systém dělat**
    - **Obsahuje sémantický model dat**
    - **Vymezuje, co budeme sledovat, ne jak to budeme realizovat**
  - o Logický model
    - Definuje, jak bude konceptuální struktura dat implementována
    - \*modely: lineární, síťový, relační, objektově orientovaný, . . .
  - o Fyzický model
    - Model fyzického uspořádání dat (soubory, . . .)
- **Základní konceptuální modely**
  - o 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)**
  - o Minispecifikace
    - **Popis funkcí (procesů) – co dělají**
  - o **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)**
  - o Datový slovník
    - • obsahuje specifikace prvků modelů
    - • notace pro specifikaci informačního obsahu prvků DFD a ERD
  - o **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**

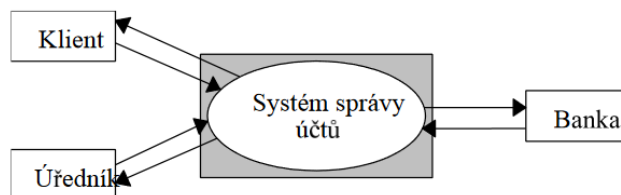
- **Příklad:**

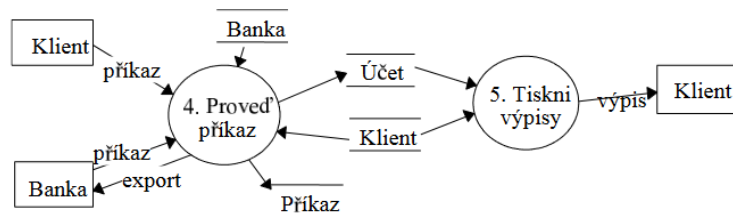
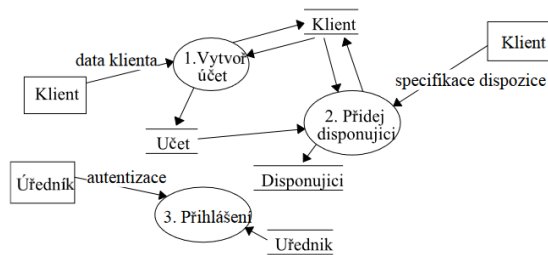


...

- Data Flow Diagram (DFD)

- 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:
  - 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.





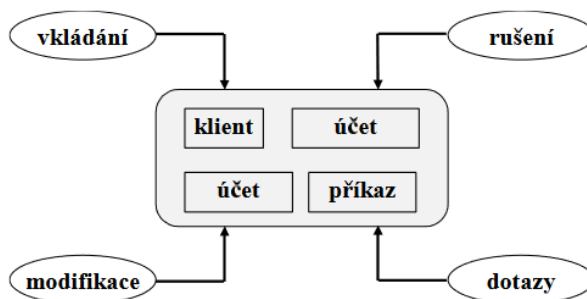
○ Datové modelování

▪ 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

▪ ER Model

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



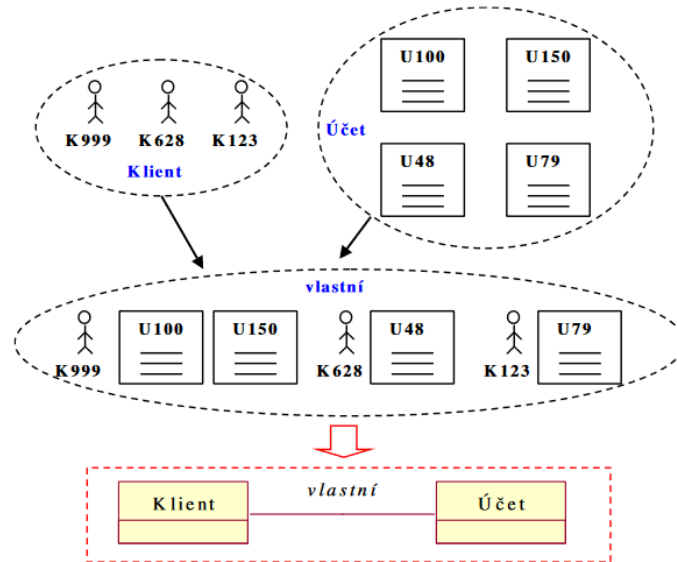
- Která data potřebujeme v systému uchovávat?

- Jaké jsou mezi nimi vztahy?

• **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

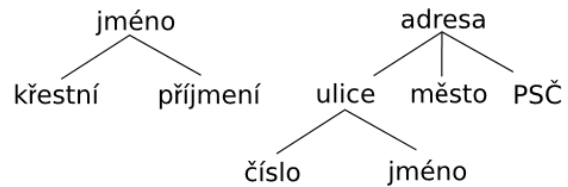
- Jednoduché (simple) a složené (composite) atributy

Entitní množina

Klient

Složené atributy

Složky atributu



○

- Jednohodnotové (single-valued) a vícehodnotové (double-valued) atributy

- Např. Telefon – může být více čísel
- Lze omezit minimální a maximální počet hodnot

- Prázdné (NULL) atributy

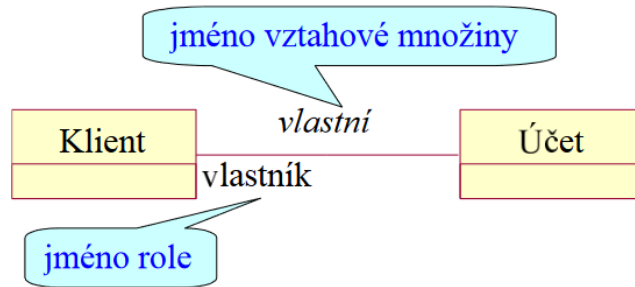
- Mohou nabývat speciální hodnoty NULL
- Může zastupovat chybějící hodnotu – existuje, ale neznáme ji
- Může zastupovat neznámou hodnotu – nevíme, zda existuje

- Odvozené atributy

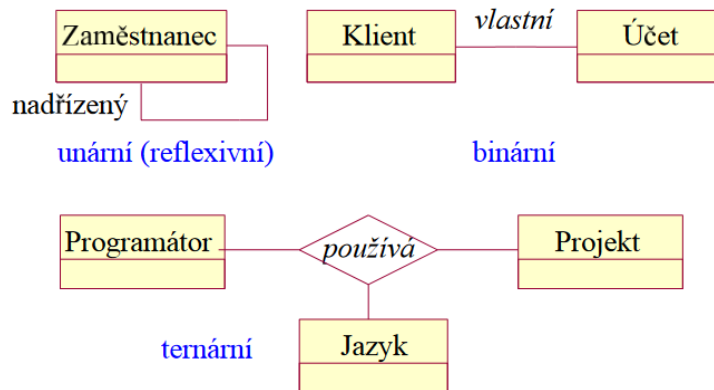
- Hodnotu lze odvodit od jiných atributů nebo entit
- Např. Datum narození ⇒ věk

- Parametry vztahů

- Jméno vztahové množiny i jméno role vyjadřuje význam vztahu.

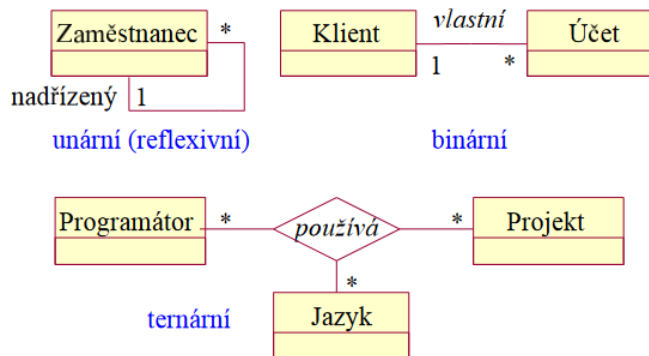


- 
- Jméno vztahové množiny se víc používá
- Jméno role centrujeme k entitě, která danou roli má
- **Stupeň vztahu**



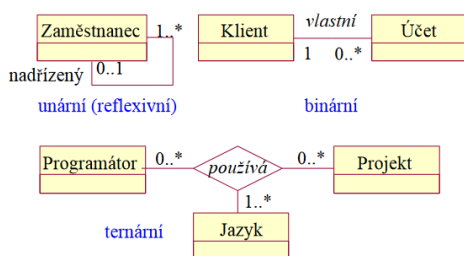
- **Kardinalita**

- Kardinalita (cardinality) je maximální počet vztahů daného typu (vztahové množiny), ve kterých může participovat jedna entita.
- Typické hodnoty: 1, M/\*, případně přesněji



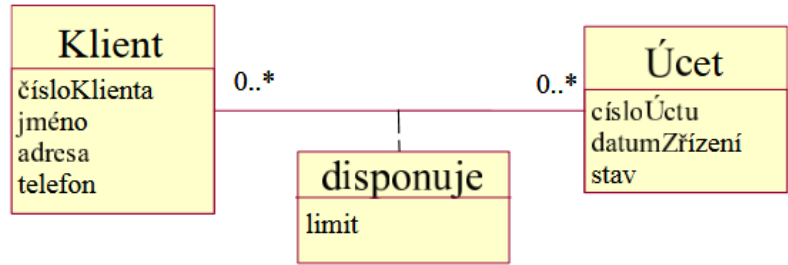
- **Členství / účast**

- Členství (membership) / účast (participation) je minimální počet vztahů daného typu (vztahové množiny), ve kterých musí participovat jedna entita.
- Typické hodnoty: 0 – volitelné, 1 – povinné



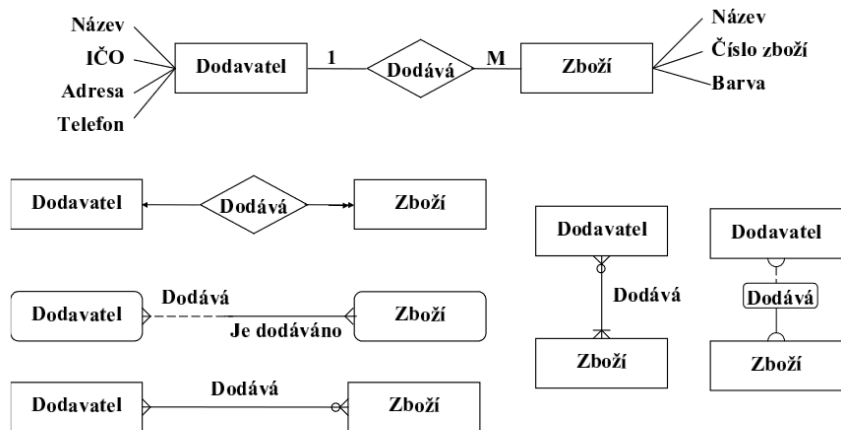
- **Atributy vztahu**

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



- Disponuje limitem, protože klient může mít více účtů s jinými limitami, ale jeden účet může mít podle pozice klienta jiné limity (zaměstnanec, generální ředitel)

• **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 Předmět, který má zapsaný Student
- Učitel učí Student ⇒ redundantní

○ **Pozor na entity**

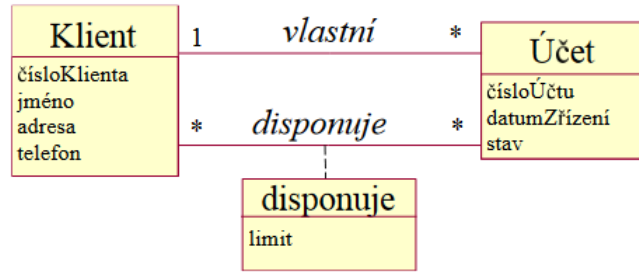
- Bez atributů
- Mající pouze identifikátor
- Mající pouze jeden výskyt
- Obsahující atributy patří jiným entitám (cizí atributy)

• **Doporučení pro tvorbu ERD**

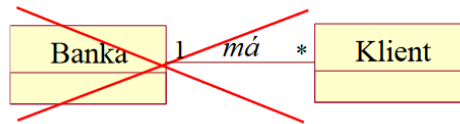
○ **Jména**

- Musí být srozumitelná a musí vyjadřovat význam entitních a vztahových množin.
- Entitní množiny: podstatná jména
- Vztahové množiny: slovesa, předložky

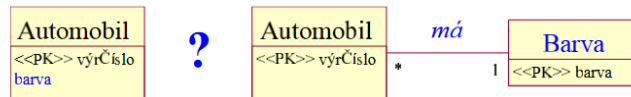
- Je-li jméno vztahové množiny jasné ze jmen entitních množin, není nutné uvádět.
- • Mezi stejnými entitními množinami může být několik vztahových množin



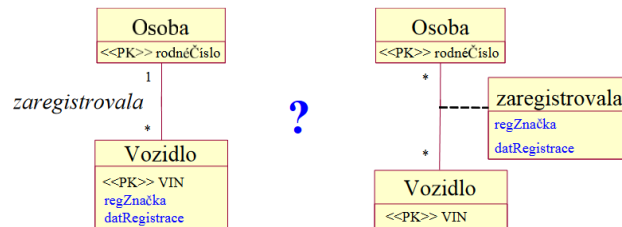
- 
- Identifikátor (klíč, primární klíč)
  - Entity a vztahy musí být identifikovatelné.
  - Hodnota identifikátoru musí být unikátní (a minimální).
  - Identifikátorem je jednoduchý nebo složený atribut.
  - Unikátnost hodnoty jen v rámci vyvíjeného systému (ne celého vesmíru).
- • Celkový systém by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



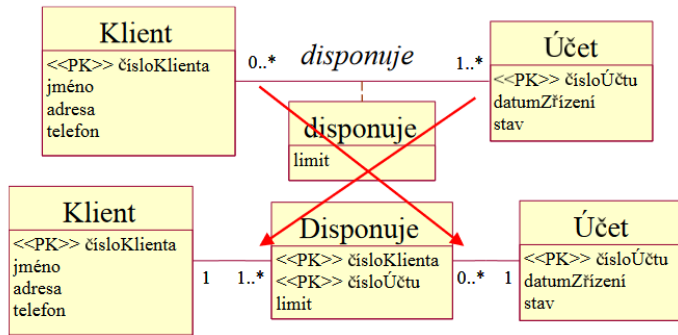
- Použít entitní množinu nebo atribut?



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

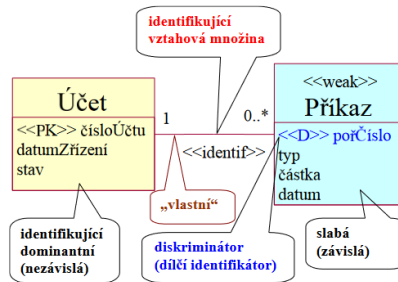


- Bude záležet na tom, zda budeme chtít uchovávat i historii registrací.
- Náhrada vztahů M:M vazební entitní množinou



- **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

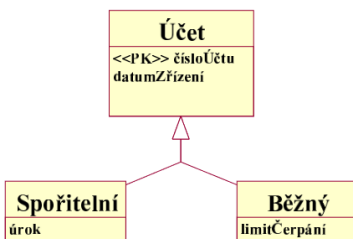
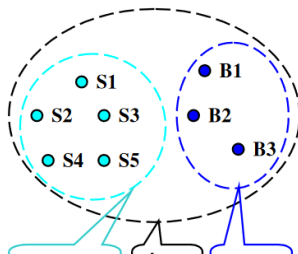
- **Rozšíření ER modelu**

- Enhanced Entity-Relationship (EER) Modeling

- Zobecnění množin (generalizace/specializace), vztah is-a
    - Kategorie (typ UNION)
    - Dědičnost atributů a vztah°

- **Zobecnění (Generalizace / Specializace) entit**

- Entity mající stejný základ (atributy) liší se v některých atributech
    - Př.: entity Spořitelní účet a Běžný účet jsou speciální variantou účtu



- Dědičnost atributů a účasti ve vztahových množinách
    - Hierarchie generalizace (podobně v OO přístupu)
      - Příklad: Spořitelní je (is a) účet
    - Identifikátor entitních množin nižší úrovně je stejný jako vyšší
    - **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. Provéřte atributy, zda je potřeba zaznamenat informace o některém z atributů v samostatné entitě
    - 4. Další entita ⇒ krok 1
    - 5. Vytvořte vztahy mezi entitami
    - 6. Určete, zda některé atributy potřebují být identifikovány pomocí více entit ⇒ 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