

# Databázové systémy

Dátové modelovanie

Q&A?

sli.do#DB01

alebo

**nahlas**

# Review: Čo nám poskytuje DBMS

- Efektívne
- Spoľahlivé
- Vhodné/pohodlné (angl. convenient)
- Bezpečné
- Viac-používateľske
- Ukladanie a prístup k veľkému množstvu perzistentných dát

# Review: Čo nám poskytuje DBMS

- Efektívne
- Spoľahlivé
- Vhodné/pohodlné (angl. convenient)
- Bezpečné
- Viac-používateľske
- Ukladanie a prístup k veľkému množstvu perzistentných dát

# Review: Čo nám poskytuje DBMS

- Efektívne
- Spoľahlivé
- Vhodné/pohodlné (angl. convenient)
  - Nezaujíma nás fyzická reprezentácia dát
  - Deklaratívne dopytovanie/manipulácia s dátami
  - Vysoká úroveň abstrakcie ==> dobrý dátový model
- Bezpečné
- Viac-používateľske
- Ukladanie a prístup k veľkému množstvu perzistentných dát

# Úrovne abstrakcie modelov

- Vysoko-úrovňové (konceptuálne) modely
  - Koncepty tak, ako ich vnímajú ľudia
- Nízko-úrovňové modely (fyzické) modely
  - Podrobnosti o tom, ako sú dáta uložené v počítači

# Úrovne abstrakcie modelov

- Vysoko-úrovňové (konceptuálne) modely
  - Koncepty tak, ako ich vnímajú ľudia
- Prezentačné (implementačné) modely
  - Ľudia ešte dokážu pochopiť
  - Dajú sa jednoducho implementovať v počítači
- Nízko-úrovňové modely (fyzické) modely
  - Podrobnosti o tom, ako sú dáta uložené v počítači

# Úrovne abstrakcie modelov

- Vysoko-úrovňové (konceptuálne) modely
  - Entitno-relačný model + ďalšie features
- Prezentačné (implementačné) modely
  - Relačný dátový model
- Nízko-úrovňové modely (fyzické) modely
  - Ehm...who cares? :) (o tomto teraz nie)



# Úrovne abstrakcie modelov

- **Vysoko-úrovňové (konceptuálne) modely**
  - **Entitno-relačný model**
- **Prezentačné (implementačné) modely**
  - **Relačný dátový model**
- **Nízko-úrovňové modely (fyzické) modely**
  - Ehm...who cares? :) (o tomto teraz nie)

# Konceptuálny dátový model

- Koncepty
  - Entity
    - Zamestnanec, oddelenie, miestnosť
  - Atribúty
    - meno, označenie, poschodie
  - Vzťahy
    - Zamestnanec *je zaradený* na oddelenie
    - Zamestnanec *sedí* v miestnosti

# Konceptuálny dátový model

- Koncepty
  - Entity
    - Zamestnanec, oddelenie, miestnosť, **sedenie**
  - Atribúty
    - meno, označenie, poschodie, **od, do**
  - Vzťahy
    - Zamestnanec *je zaradený* na oddelenie
    - ~~Zamestnanec *sedí* v miestnosti~~

# Konceptuálny dátový model

- Koncepty
  - Entity
    - Zamestnanec, oddelenie, miestnosť, **sedenie**
  - Atribúty
    - meno, označenie, poschodie, **od, do**
  - Vzťahy
    - Zamestnanec *je zaradený* na oddelenie
    - ~~Zamestnanec sedí v miestnosti~~
- V skutočnosti sú to typy entít a typy vzťahov

Schéma, triedy – nie inštancie

# dátový model vs. objektový model

- Anglicky: Impedance mismatch
- Software engineering disclaimer: Pod objektovým modelom tu myslíme dátovú štruktúru v pamäti
- Dátový model môže mať na rozdiel od objektového modelu len jednoduché atribúty
  - Žiadne vnorené štruktúry
  - Žiadne zoznamy hodnôt
  - V súčasnosti to už neplatí

# Pravidlá modelovania

- Entity – navzájom odlišiteľné (majúce identitu) objekty reálneho sveta s nezávislou existenciou
- Vzťahy medzi entitami
  - Majú kardinalitu
- Iba atomické atribúty
  - Čiže žiadne štruktúry, zoznamy hodnôt
    - Hoci to nie je v dnešnom svete úplne pravda
- Snažíme sa zbytočne neduplikovať údaje
- Myslíme na vývoj dát v čase
- Schému meníme vtedy, keď sa mení doména
  - Nie vtedy, keď pribúda inštancia

# Príklad - Fakulta

- Má zamestnancov
  - O tých si chceme viesť klasické záznamy  
meno, rodné číslo, adresu, vek
- Každý je na nejakom ústave (iba jednom)
- Každý ústav má svoj názov a riaditeľa
  - Chceme vedieť odkedy je riaditeľ

# Príklad - Fakulta

- Zamestnanci pracujú na projektoch
  - Nezávisle od ústavu
  - najviac na troch projektoch

Chceme vedieť koľko odrobil na ktorom projekte



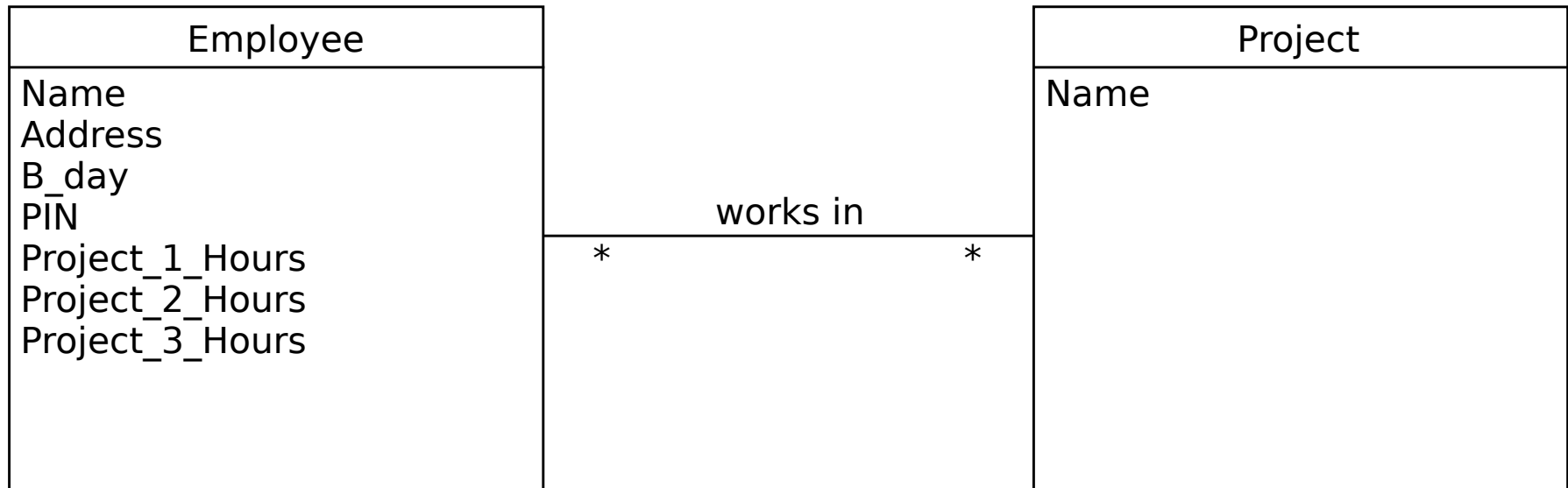
# Prvý pokus

Employee
Name
Address
Age
PIN
Project_1
Project_1_Hours
Project_2
Project_2_Hours
Project_3
Project_3_Hours

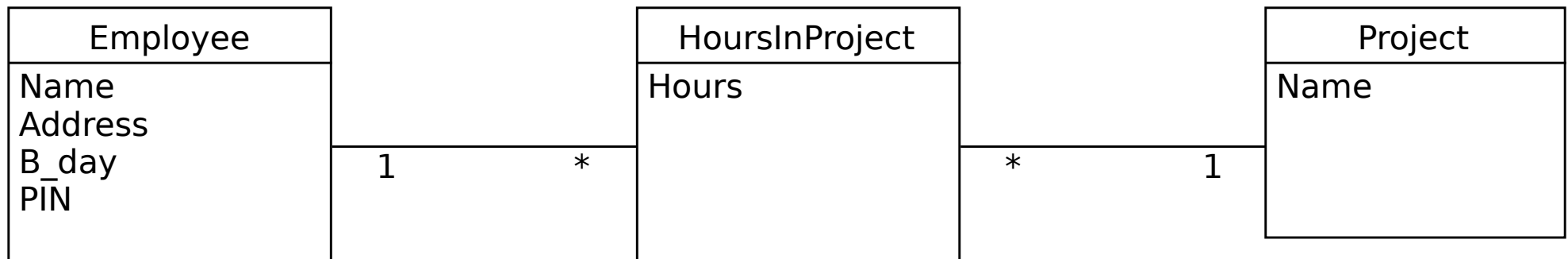
# Vek

Employee
Name
Address
B_day
PIN
Project_1
Project_1_Hours
Project_2
Project_2_Hours
Project_3
Project_3_Hours

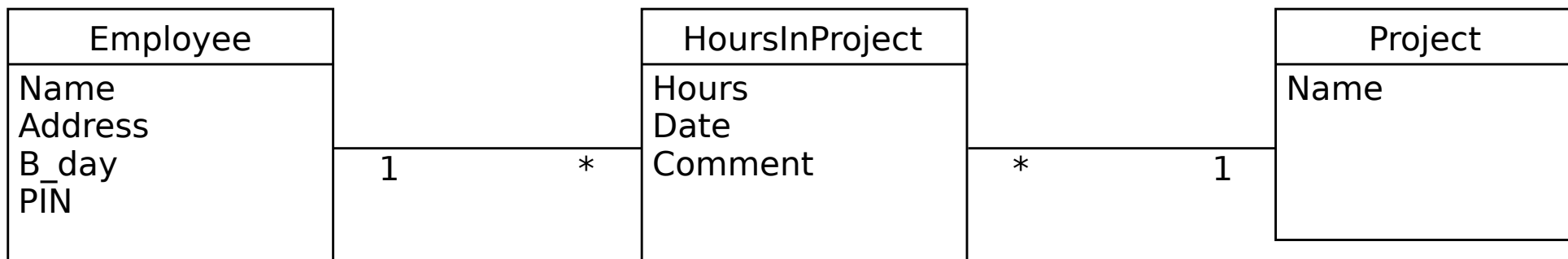
# Projekt



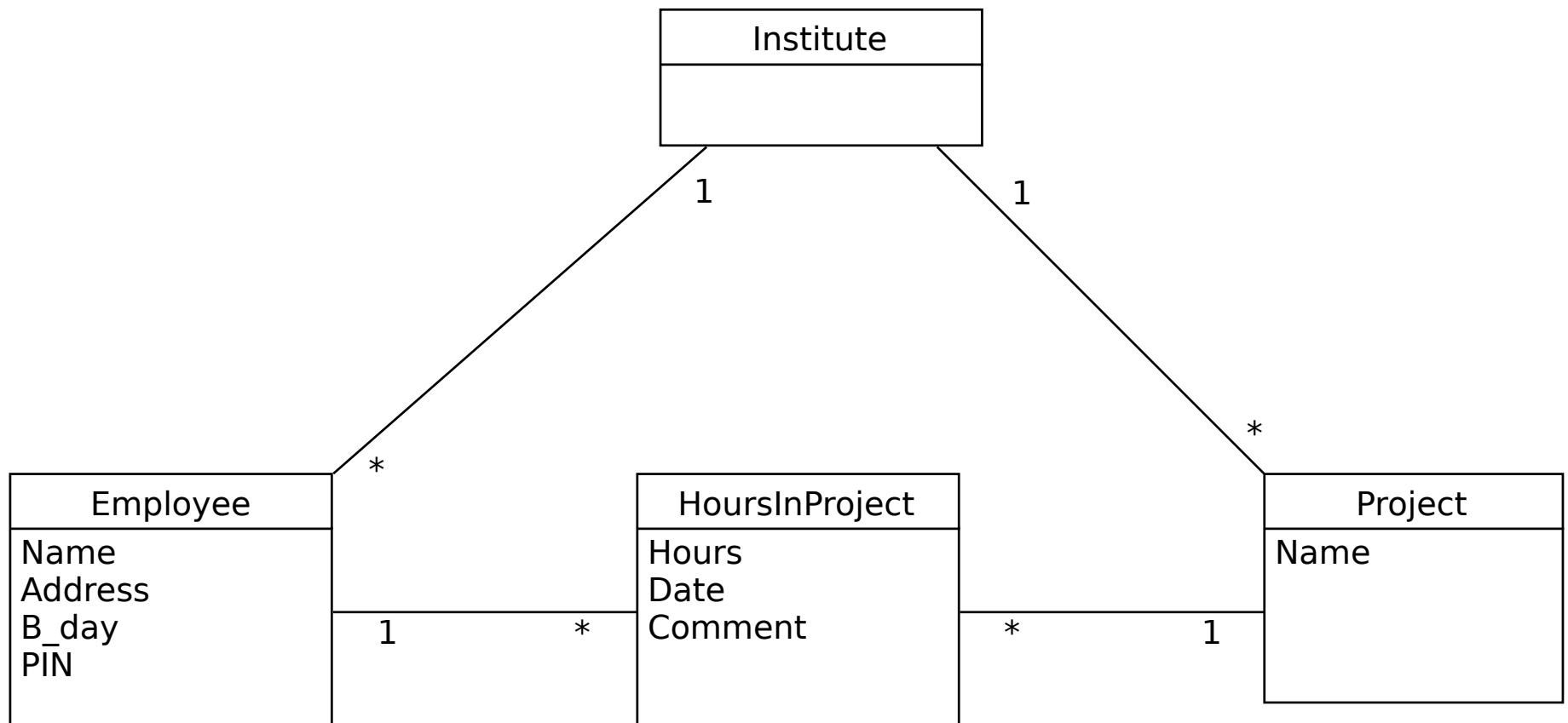
# Práca na projekte



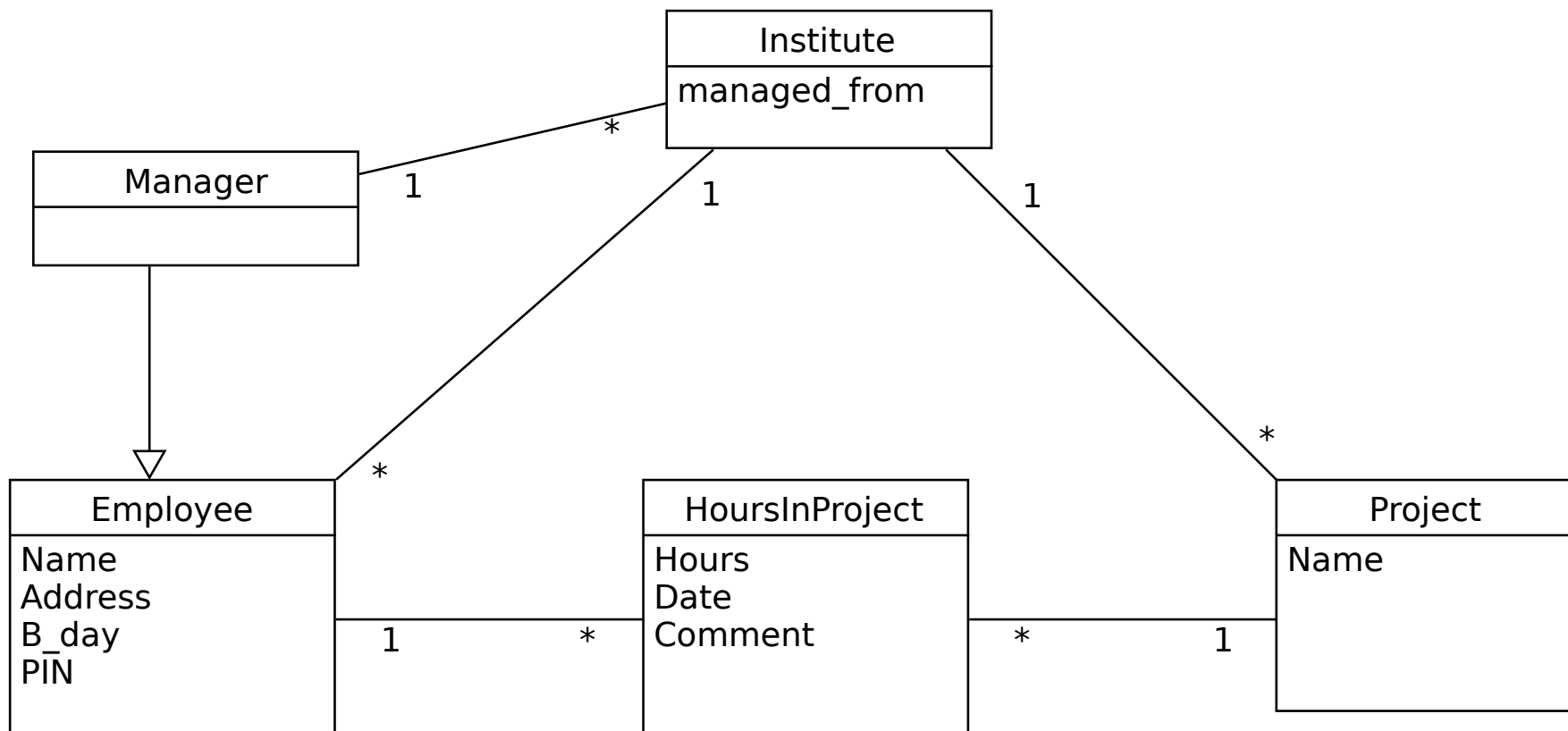
# Ďalšie metadáta



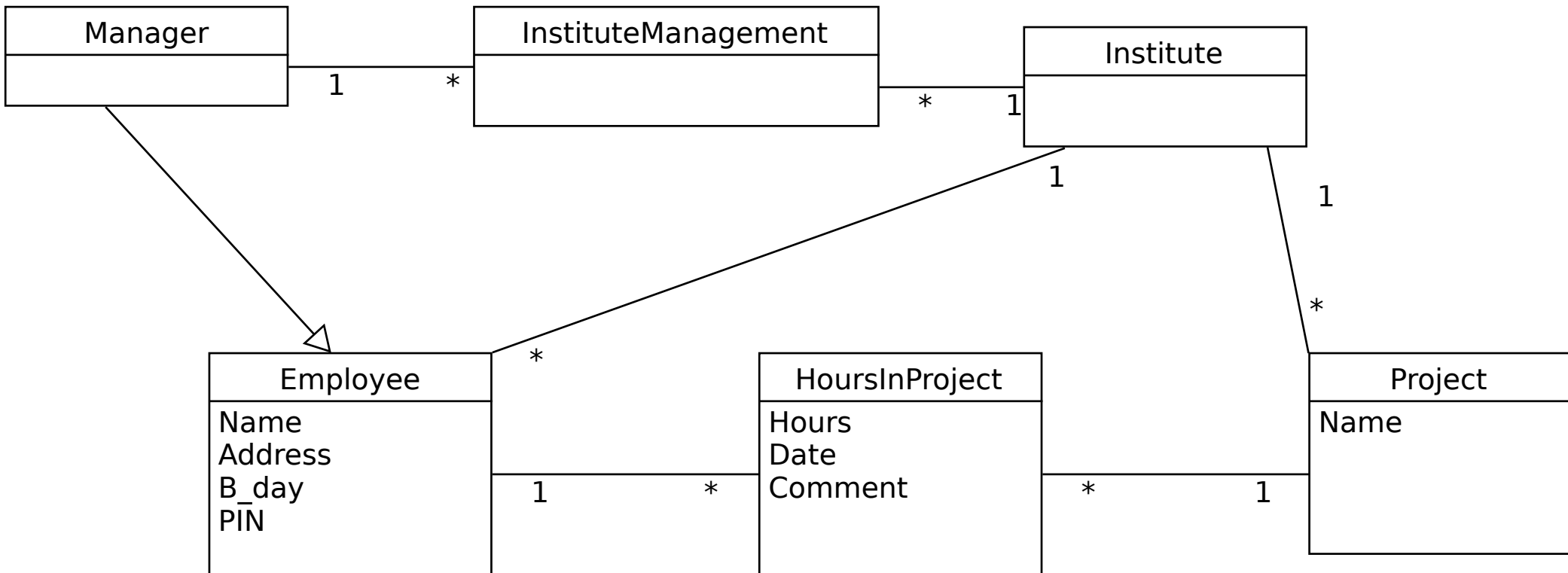
# Ústav



# Riaditeľ ústavu



# Odkedy je riaditeľom?





Prečo v modeli nemám  
samotnú fakultu?

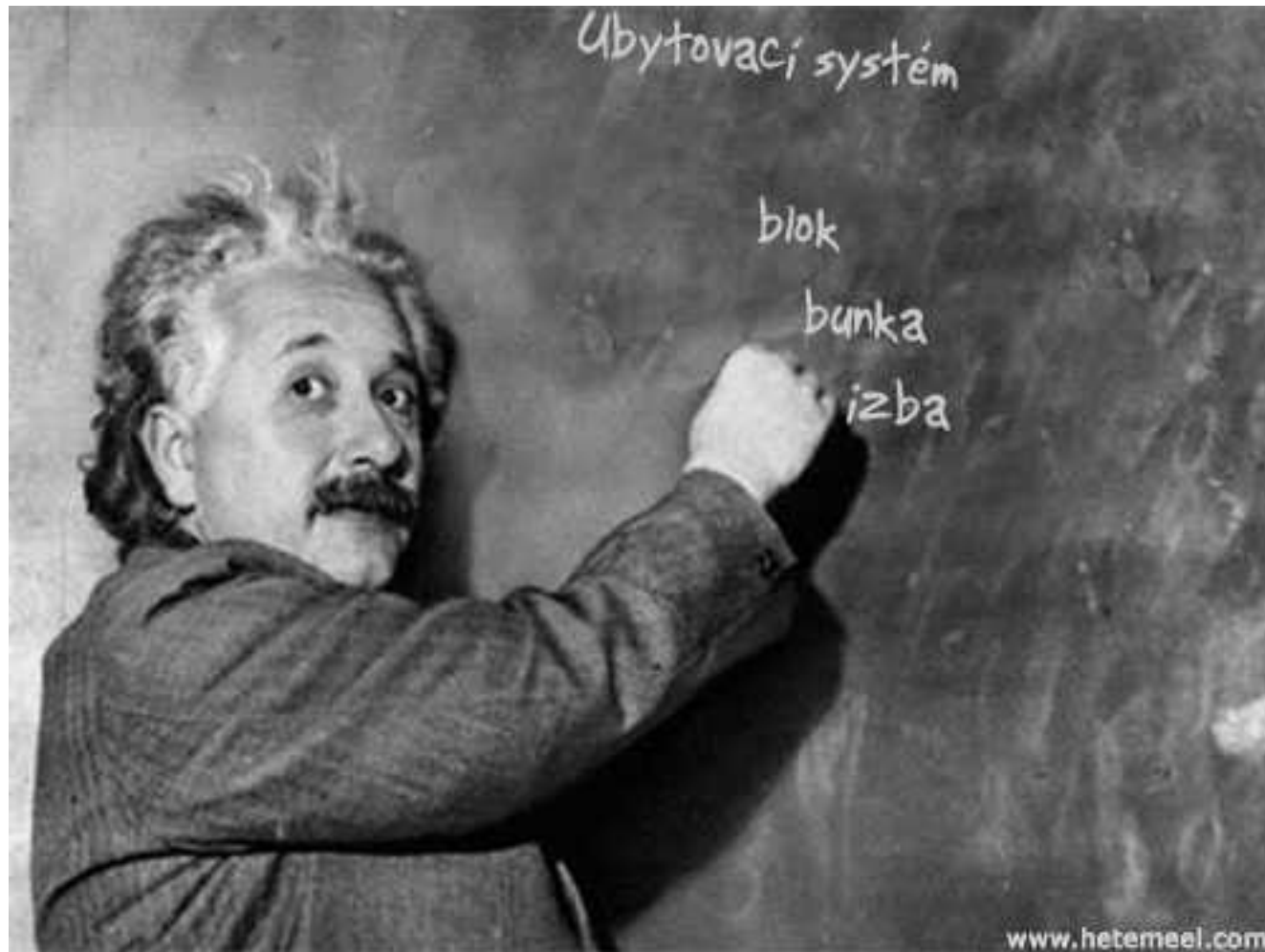
# Príklad: Ubytovací systém

Chceme ubytovací systém pre náš internát, ktorý nahradí papierové záznamy a pomôže zamestnancom internátu v ich každodennej agende. Študenti bývajú na izbách zariadených inventarizovaným nábytkom. Ten je už staršieho dáta a naši údržbári na ňom evidujú viacero poškodení. Izby sú organizované v bunkách, bunky v blokoch.

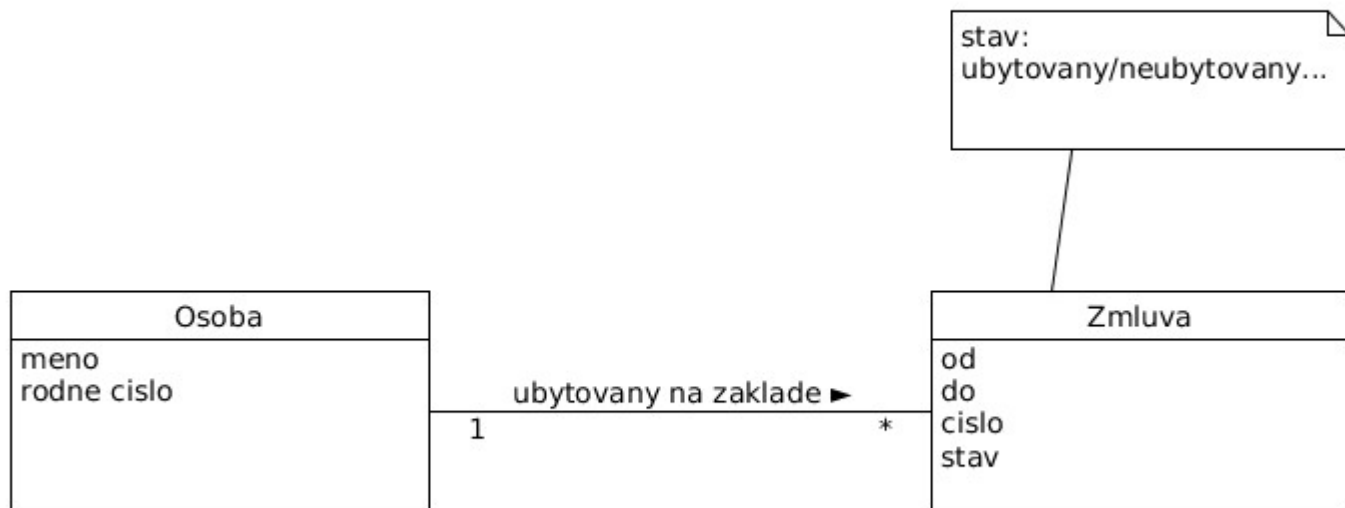
# Ubytovací systém

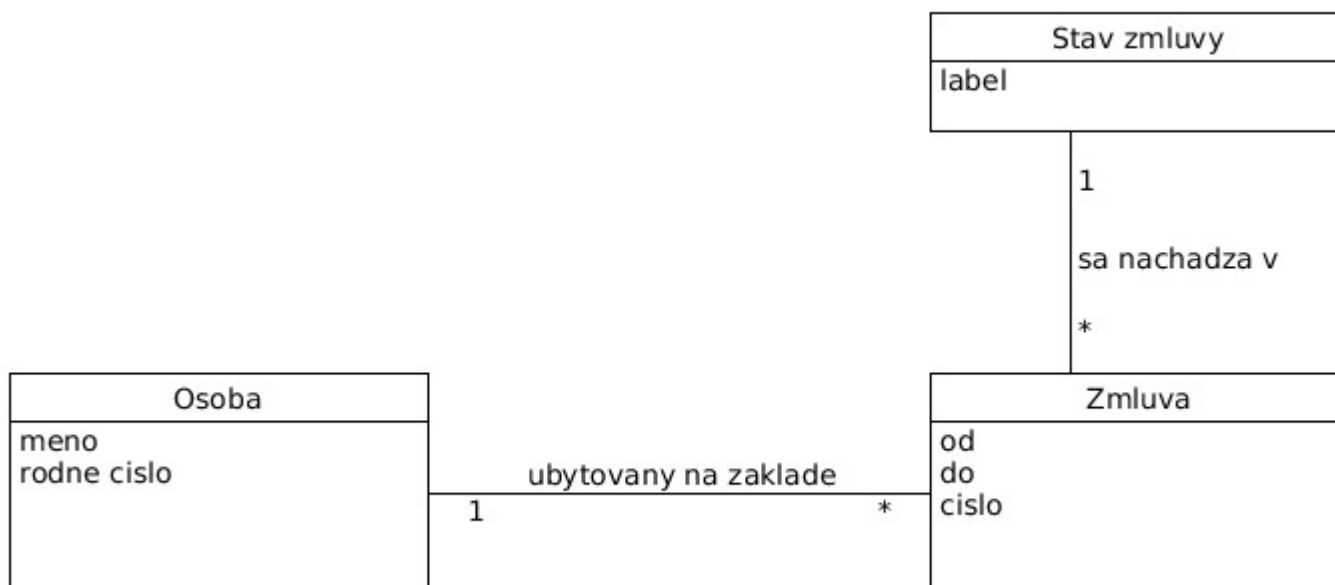
Študenti nám za ubytovanie platia každomesačné nájomné. Cena sa odvíja od toho, či študent býva na jednoposteľovej izbe alebo na trojposteľovej, pričom platbu identifikujeme podľa variabilného symbolu – čísla ubytovacej zmluvy. Chceme samozrejme vedieť, kto ešte nezaplatil.

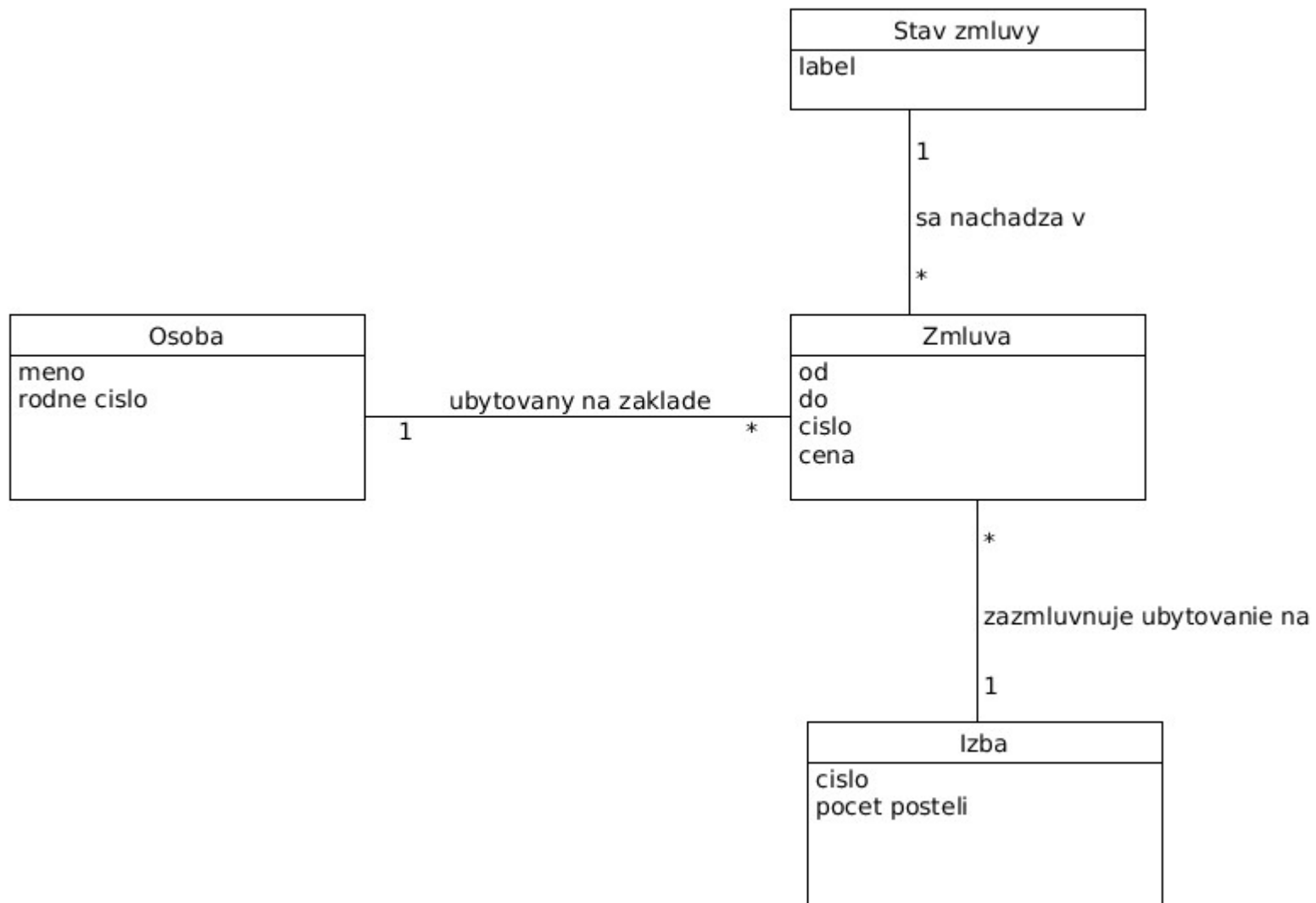
# Tak ideme na to...



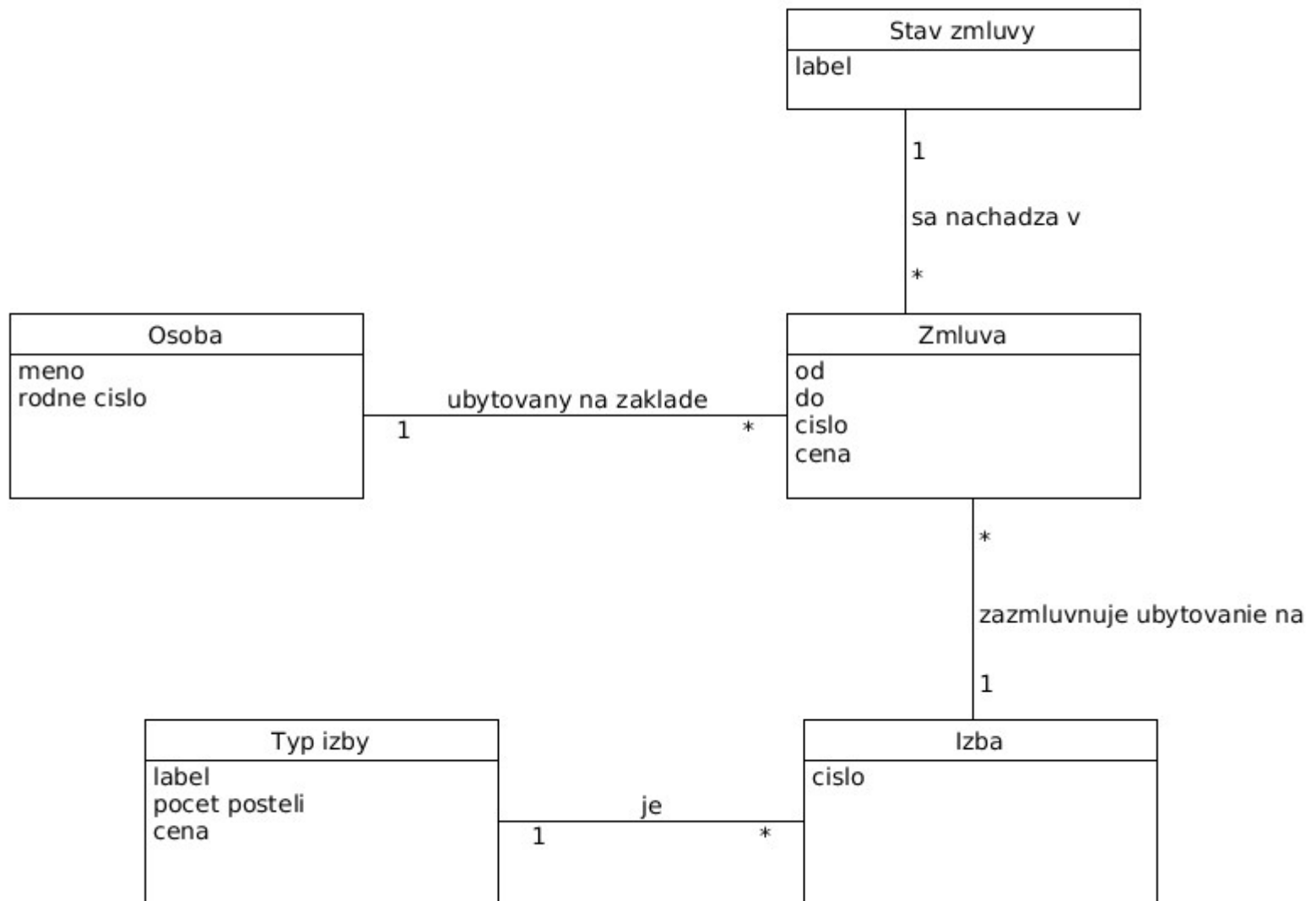


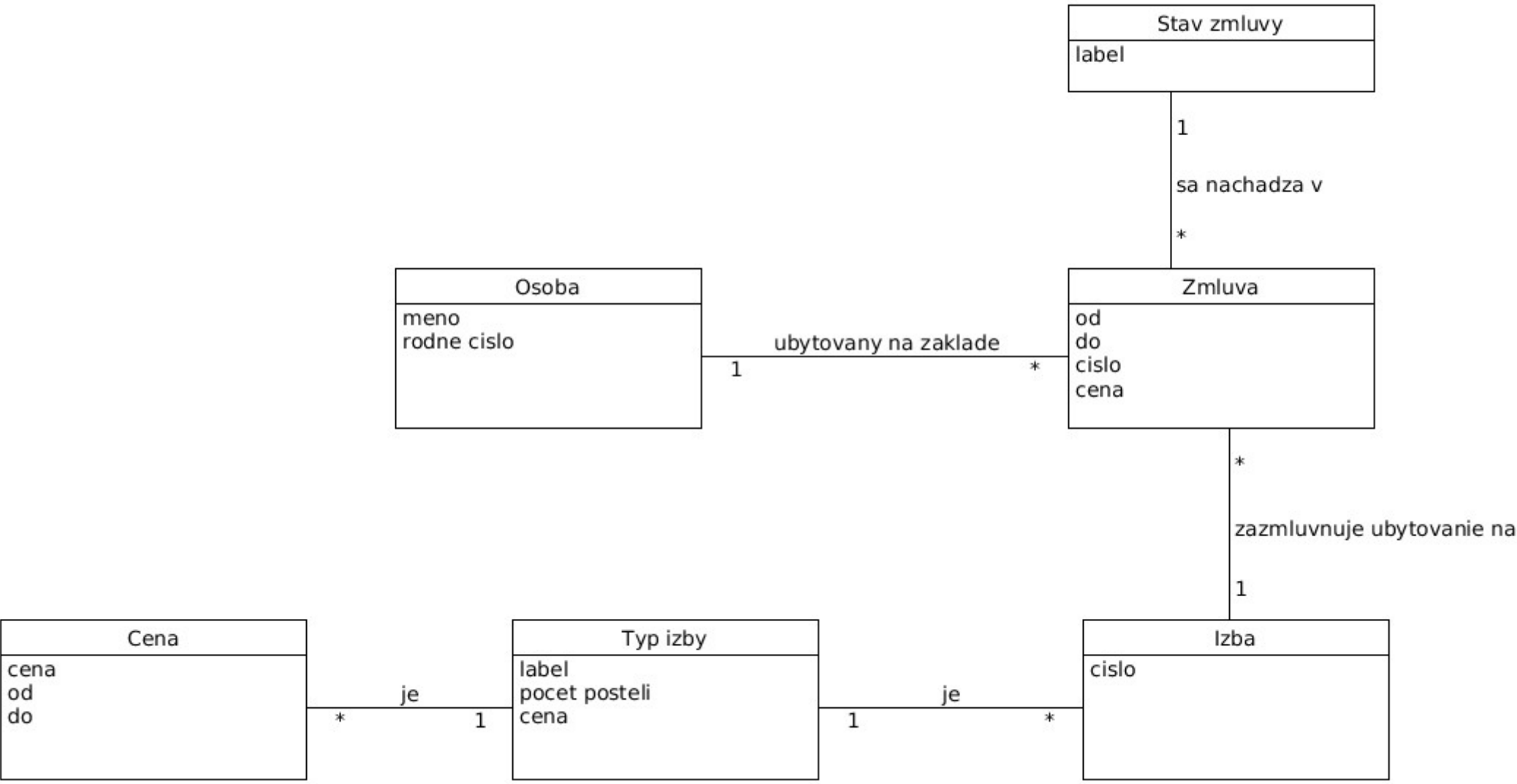


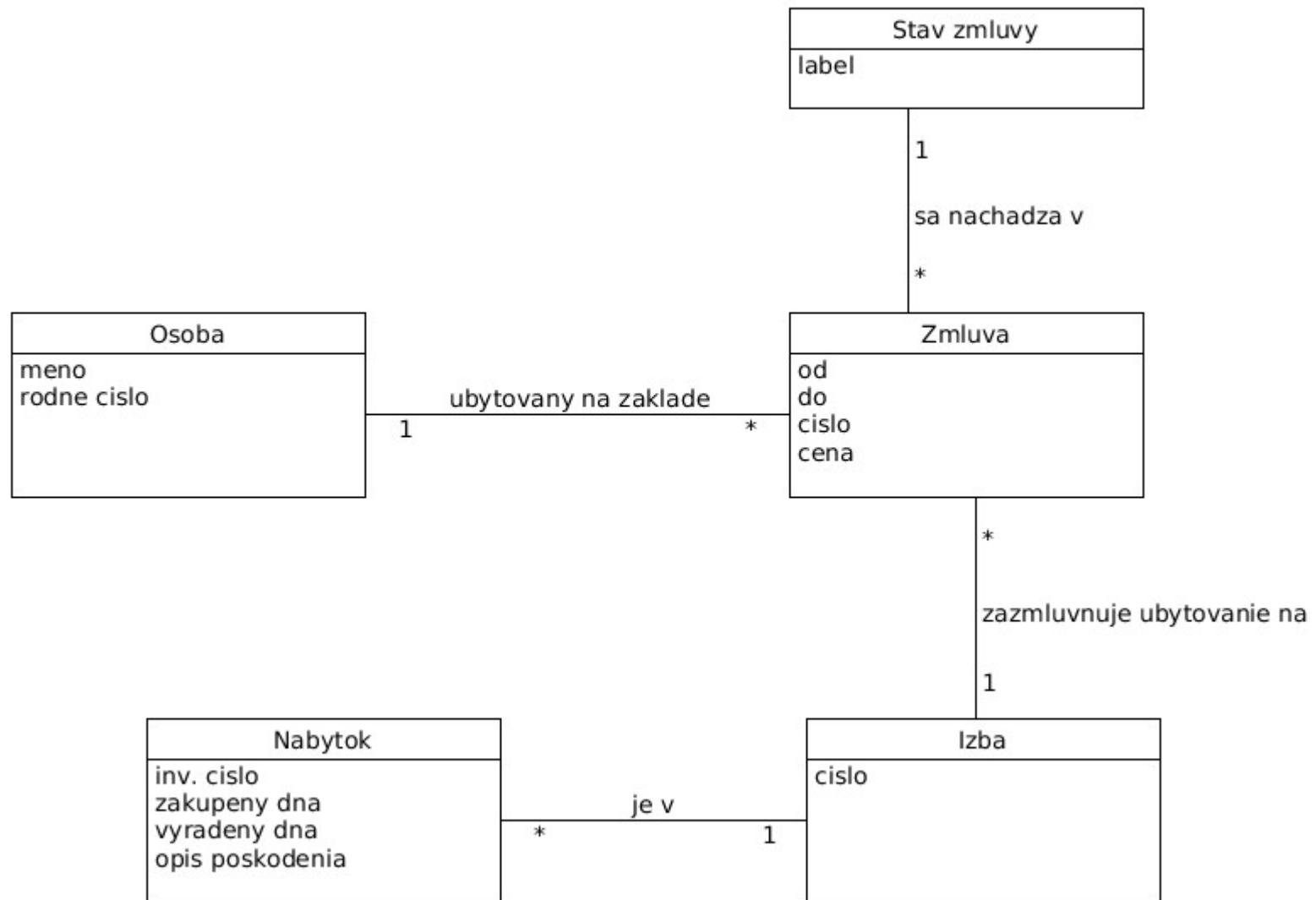


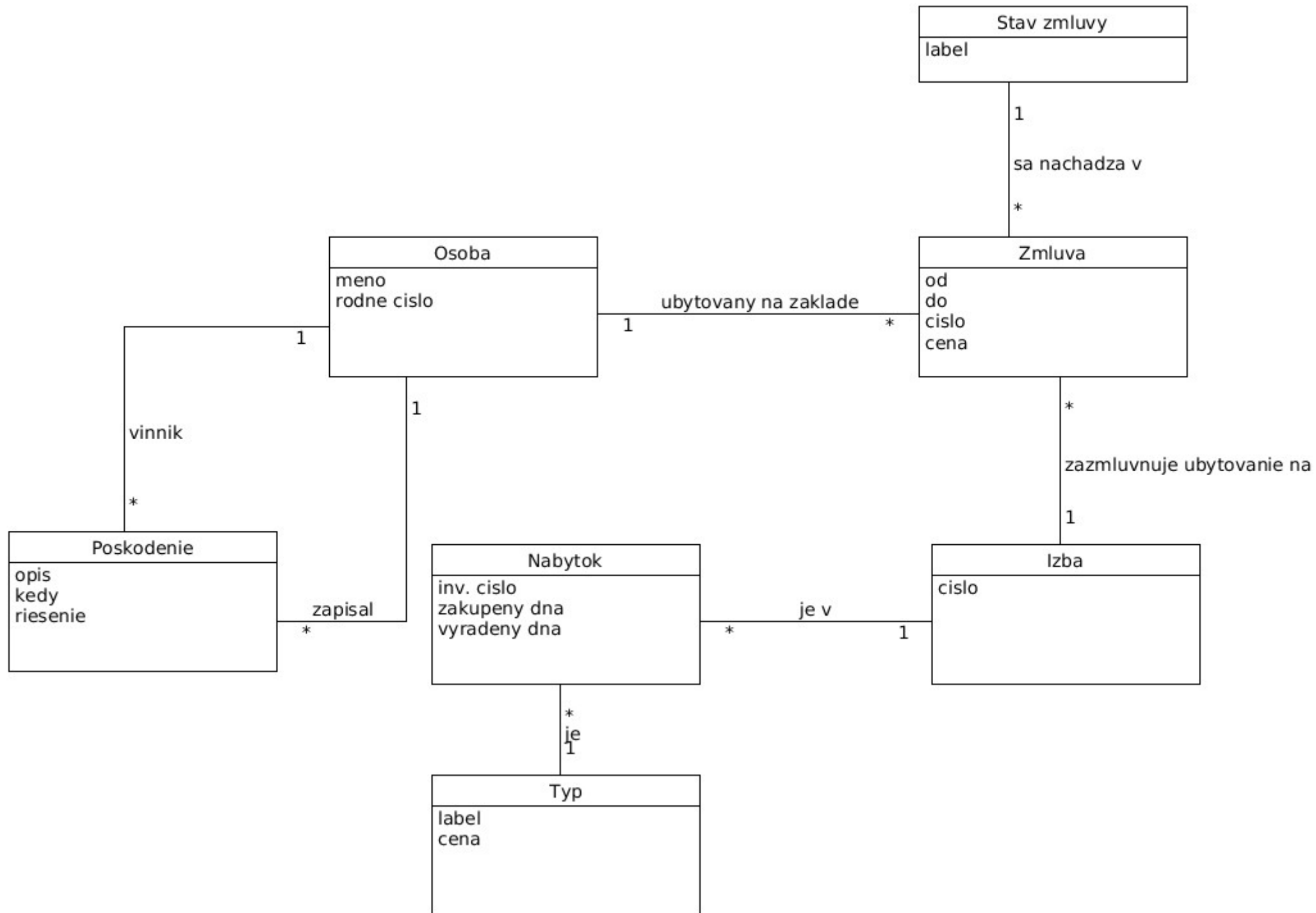


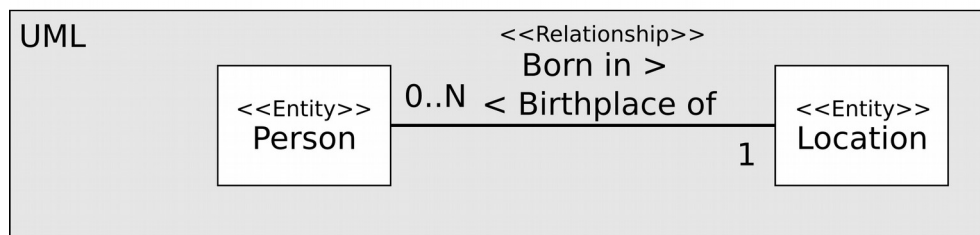
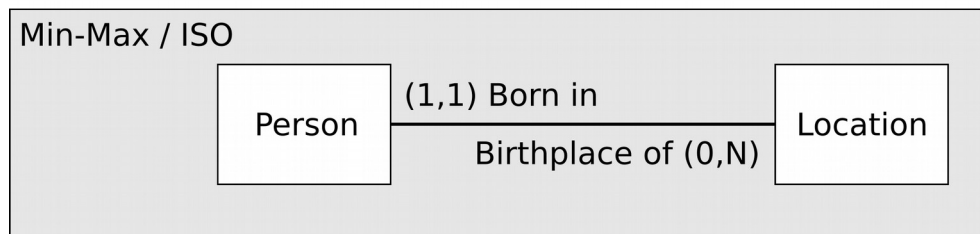
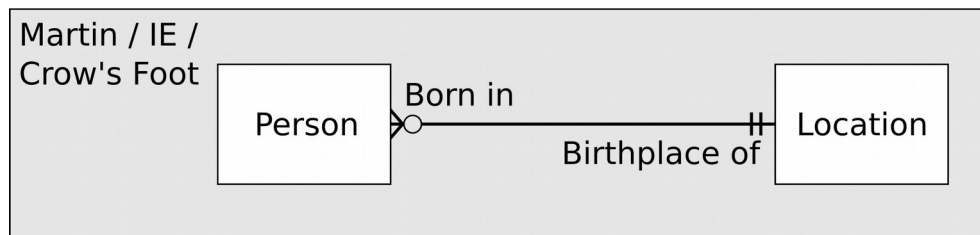
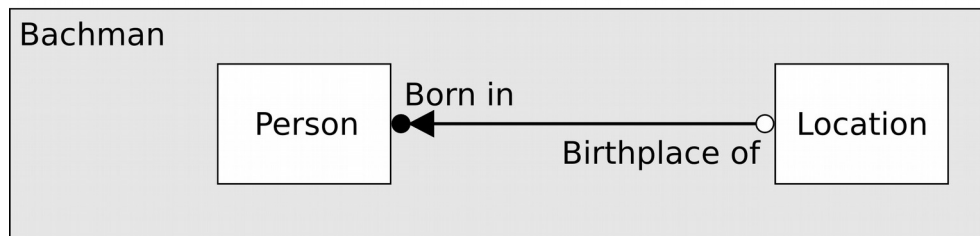
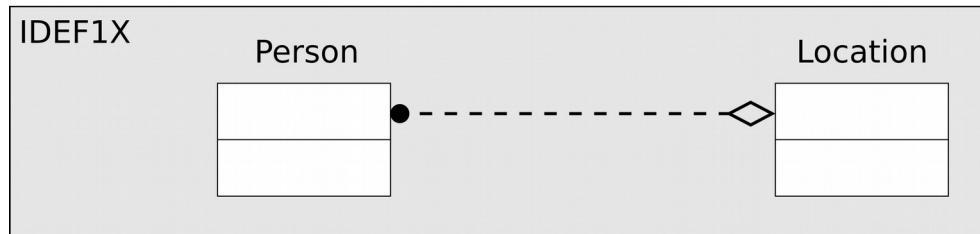
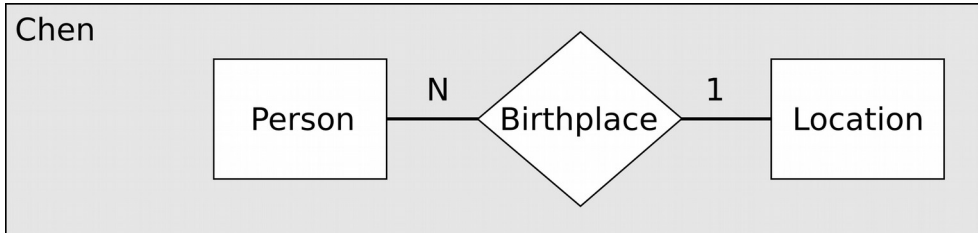












# E-R model vs. E-R diagram

- E-R model – množina entít, vzťahov a atribútov
- E-R diagram – vizualizácia modelu vo zvolenej notácii
- UML class diagram umožňuje zachytiť aj operácie nad objektami
- UML notáciu mám najradšej :)

# Zhrnutie

- Konceptuálny model je vhodný medzikrok medzi zadáním a databázovou schémou
  - Relačným modelom
- Konceptuálny model je “náš” (nás ľudí) a má čo najpresnejšie vyjadrovať to, čo si predstavujeme a ako chápeme problémovú oblasť
- Nesmieme zabudnúť aj na “systémové údaje”
  - tie nie sú čisto doménové
- Snažíme sa o čo najnižšiu redundanciu údajov
- Myslíme dopredu
  - ak život je zmena, tak softvér je niekedy až príliš živý
  - zákazník nevie čo chce