## Klasa kao realizacija apstrakcije

- \* Banalno rešenje: preslikati dati opis u klase i raspored njihovih svojstava
- \* Problemi:
  - nepotrebno mnogostruko kopiranje podataka, jer jedna osoba ima mnogo izdatih dokumenata
  - ako je potrebno čuvati istoriju promena podataka, prave se kopije svih podataka, iako se menja samo manji deo njih; gde čuvati te promene?

    UML simbol za asocijaciju (association): relacija između klasa koja definiše skup veza (link) koje mogu postojati između

## Person

personalID : PersonalID
dateOfBirth : Date

placeOfBirth : Text

gender : Gender
firstName : Text
lastName : Text
address : Text
place : Text
photo : Binary

fingerprint : Binary

irisScan : Binary

•••

UML oznaka *multiplikativnosti* (*multiplicity*) kraja asocijacije: jedan objekat klase *Document* može biti vezan za tačno <u>1</u> objekat klase *Person* vezama ove asocijacije.

Jedan objekat klase *Person* može biti vezan za proizvoljno mnogo (<u>\*</u>) objekata klase *Document* vezama ove asocijacije.

## Document

dateOfBirth : Date
placeOfBirth : Text

gender : Gender
firstName : Text

lastName : Text
address : Text

place : Text

photo : Binary

fingerprint : Binary

irisScan : Binary

•••

## Klasa kao realizacija apstrakcije

- \* Ideja boljeg rešenja: uvesti koncept Personalnog zapisa (Personal Record) koji sadrži određene lične podatke:
  - umesto da sadrže kopije, *Osoba* i *Dokument* referenciraju (vezani su za) *Personalne zapise*: sve dok se podatak ne menja, ovi objekti vezani su za iste zapise
  - prilikom promene nekog podatka, pravi se nova kopija zapisa u kom se taj podatak menja
  - kako bi se dodatno smanjilo kopiranje, podaci su grupisani u različite specijalizacije zapisa, po tome koliko se često i da li se zajedno ili odvojeno menjaju
  - za *Osobu* se mogu čuvati i raniji zapisi, sa svim podacima koji su ikada bili registrovani za tu osobu (nezavisno od *Dokumenata* u kojima su upotrebljeni)

