

Zapisi $\{ l_1 = v_1; \dots; l_n = v_n \}$
 $\{ re = 3.7; im = 2.5 \}$

Tip zapisa $\{ l_1 : \tau_1; \dots; l_n : \tau_n \}$
 $\{ re : float; im : float \}$

Podtipi: $A \leq B$ "A je podtip B"
STRUKTURNI
izraze tipa A lahko uporabljamo kot da bi bili tipa B

Podtipi zapisov: $\{ l_1 : \tau_1; \dots; l_n : \tau_n \} \leq \{ k_1 : \sigma_1; \dots; k_m : \sigma_m \}$

kadar za vsak $j \leq m$ obstaja $i \leq n$, da je $l_i = k_j$ in $\tau_i \leq \sigma_j$.
(širina & globina)

Pozor: zgornje velja za zapise z nespremenljivimi polji.
(read-only)

Kaj so objekti?

Enkapsulirano stanje + funkcionalost
(metode)



```
public class Complex {
    float re;
    float im;
    public Complex add(...) { ... }
}
```

$\{ re : float; im : float \}$
atributi ← polja
metode
this

Objekt = rekurzivni zapis
(se sklicuje sam nase z 'this')

Objekti & podtipi:

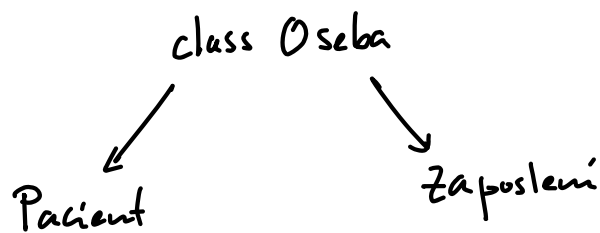
① Strukturne podtipi: (glej zgoraj)

② Nominalni podtipi:

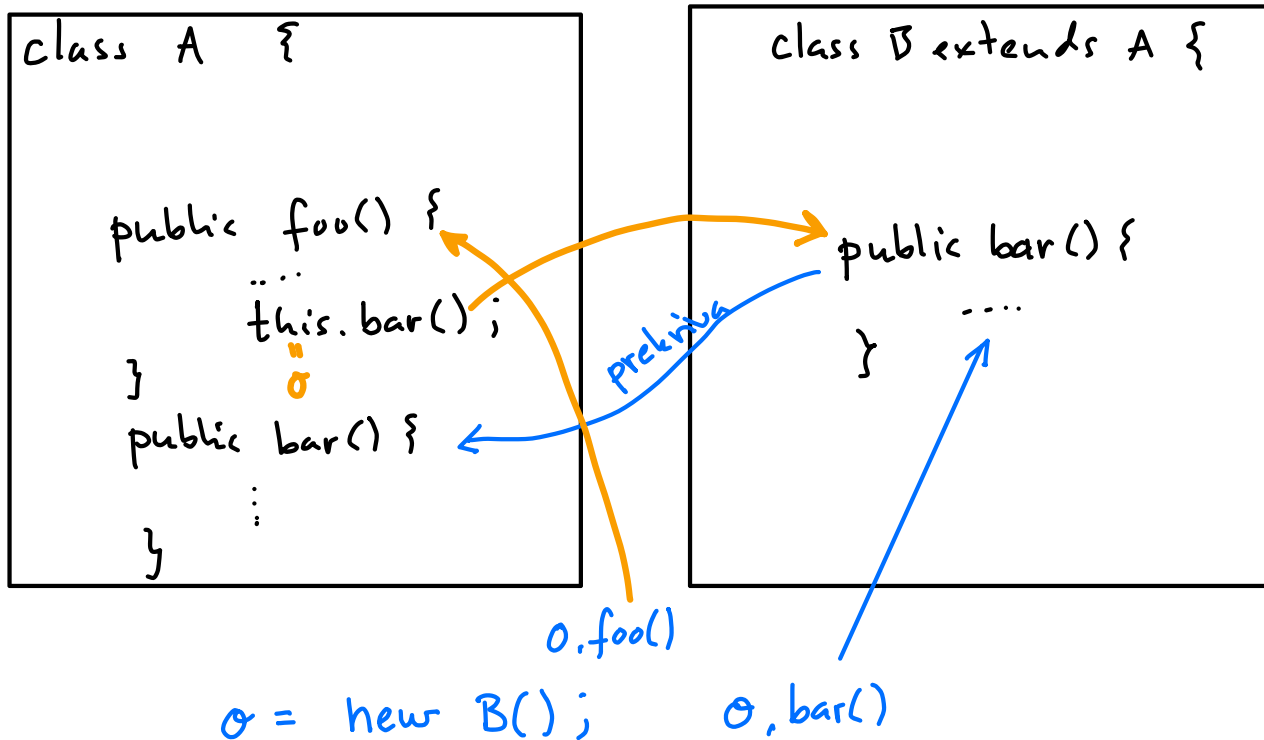
Podtipi/podrazredi so določeni s hierarhijo razredov, ki jo definira programer
(public class A extends B }

Primer:

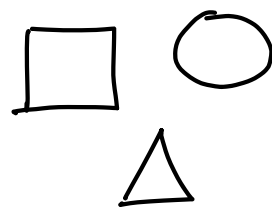
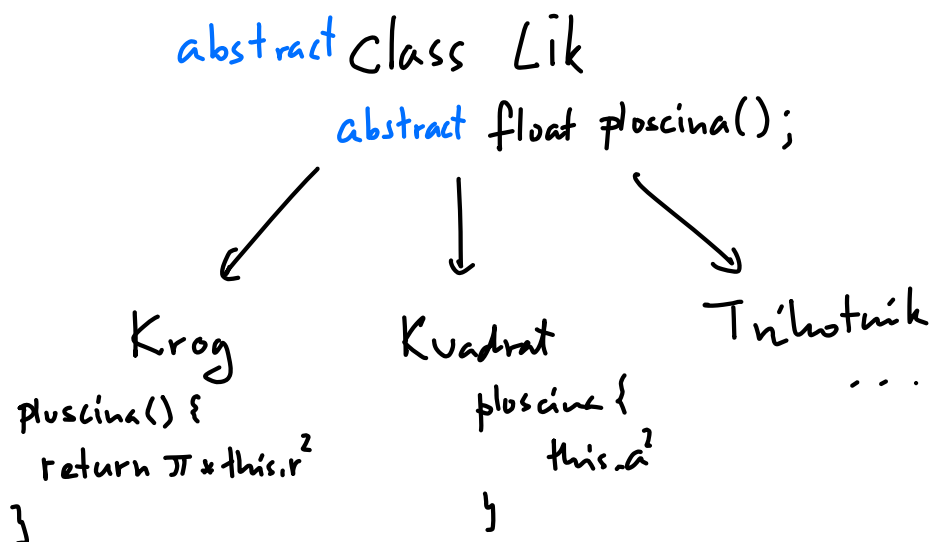
• class Pacient extends zaposleni { }



Prekrivanje:



Abstraktni / virtualni razredi & metode:



Lik[] ldi;

$$\int_0^1 \frac{ax}{\sqrt{1+x}} dx \quad \text{lejer je } a = \sqrt[3]{7} + \ln \dots$$

where

$$\{n^2 \mid n \in \mathbb{N}\} = \{0, 1, 4, 9, 16, \dots\}$$

Razredi tipov (Type classes)

① To nima veze z razredi iz objektnega programiranja.

Numerični tipi (Int, Float, Fraction, Complex, ...)
imajo skupno funkcionalnost:

+ * - ...

$$5 + 7$$

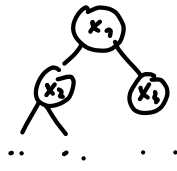
$$5.3 + 7.6$$

Container / collection, zbirha :

seznam

$[x_1, \dots, x_n]$

drevo



zgoščenjena tabela

tabela

iterator
add
find

Predstavitev s string :

42

"42"

(False, "foo")

type class = Skupna funkcionalnost
nekega razreda tipov

Num

numerični tipi

Show

tipi, predstavljeni na zaslonu (kot stringi)

⋮

Class Functor

$a \rightarrow b$

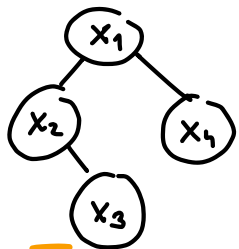
$[a]$

$[b]$

$\text{map } f [x_1, \dots, x_n] = [f x_1, \dots, f x_n]$

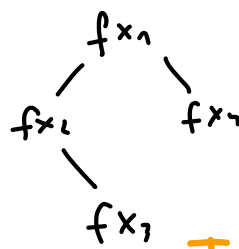
$\text{map } f$

$a \rightarrow b$



Tree a

=



Tree b

$\mathbb{N} : \text{Maybe } a$

$\text{map } f (\text{Just } x) = \text{Just } (f x)$
 $\text{map } f \text{ Nothing} = \text{Nothing}$

$a \rightarrow b \quad f a \rightarrow f b$

$fmap : (a \rightarrow b) \rightarrow (f a \rightarrow f b)$

Applicative :

a $\text{Maybe } a$
 x Nothing
 x $\text{Just } x$

a $\text{List } a$
 x $[]$
 x $[x]$
 $[x, y]$
 \vdots