

Obseg (angl. scope)

ime x v kodu ima obseg = območje veljavnosti

Java: obseg = od deklaracije do konca bloka (lok. spremin.)

razred: public class \rightarrow veja posred
private class \rightarrow znotraj razreda

let $x = e_1$ in e_2
obseg x

```
{ int x = e1;  
  C }  
}
```

B where
 $\begin{bmatrix} x = \\ y = \\ z = \end{bmatrix}$

Haskell:

type $T = \dots$

T je deklaracija / sinonim

data $T = \dots$

definicija podatkovnega tipa
(nov tip)

Izpeljani seznam:

možica $\{ n^2 \mid n \in \{0, 1, \dots, 10\} \} = \{ 0, 1, 4, 9, \dots, 100 \}$
 $[n * n \mid n \leftarrow [0..10]]$

$f : A \times B \rightarrow C$

plus = + : $\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$

$\tilde{f} : A \rightarrow (B \rightarrow C)$

plũs : $\tilde{f} : \mathbb{R} \rightarrow \mathbb{R} \rightarrow \mathbb{R}$

operacija curry

plus (7, 5) \rightarrow sestoj 7 in 5

plũs 7 5

plũs 7 : $\mathbb{R} \rightarrow \mathbb{R}$

prĩstoj 7

Tipi:

- funkcijski tip

tip

$a \rightarrow b$

vrednost

$\lambda x. e$
 $\lambda x. e$

- produkt

(a, b)

(e_1, e_2)

kartezijski produkt $a \times b$

OCaml $a * b$

$(\text{Int}, \text{Bool})$

$(42, \text{False})$

$()$

$()$

OCaml: `unit`

Rant:

index x l ... indeks elementa x v seznamu l

index 5 $[6, 3, 5, 8] = 2$

index 4 $[6, 3, 5, 8] = ?$

open `"/foo/ne_obstaja.txt"` = ?

1) Vrneto posebno vrednost npr. `-1`

2) Sprožimo izjemo

3) Spremenimo tip rezultata:

index 5 $[6, 3, 5, 8] = \text{Just } 2$

OCaml

`Some 2`

index 4 $[6, 3, 5, 8] = \text{Nothing}$

`None`

Razredi tipov (type class)

≠ razredi iz objektnega programiranja

Bolj podobni so umesnikom:

type class: opisuje, kaj lahko pomenimo z vrednostmi nekega tipa

Primer:

$(\text{Sized } a, \text{ Sized } b) \Rightarrow \text{Sized } (a, b)$ omejitelj

$\text{sized } (\text{prod}(A, B)) :- \text{sized } A, \text{sized } B.$ prolog

$\forall a, b. \text{Sized } a \wedge \text{Sized } b \Rightarrow \text{Sized } (a * b).$ Hornova logika

Monoid:

→ množica S

→ operacija $* : S \times S \rightarrow S$

→ enota $e \in S$

asociativnost

$$(x * y) * z = x * (y * z)$$

$$e * x = x = x * e$$

Primeri: $(\mathbb{N}, +, 0)$

$(\mathbb{N}, \times, 1)$

$(\mathbb{N}, \max, 0)$

$(\text{List } a, ++, [])$

$$0 \in \mathbb{N}$$

$$\max(0, x) = x$$

$$\text{concat } [x_1, \dots, x_n] = x_1 * x_2 * \dots * x_n$$

$$x * y = \text{Concat } [x, y]$$

$$e = \text{Concat } []$$

*	0	m
0	0	m
m	m	m

$(\{0, 1\}, \max, 0)$
↓
OneShot
↘
multiShot