

# Monade

**prejšnjič...**

# Spoznali smo jezik **Haskell**



# Ad-hoc polimorfizem smo opisali z **razredi tipov**

## definicija razreda

```
class Num a where
  (+), (-), (*) :: a -> a -> a
  negate       :: a -> a
  abs          :: a -> a
  signum       :: a -> a
  fromInteger  :: Integer -> a

  x - y        = x + negate y
  negate x     = 0 - x
```

```
instance Num a => Num (Fraction a) where
  Frac a b + Frac c d =
    Frac (a * d + c * b) (b * d)
  Frac a b * Frac c d =
    Frac (a * c) (b * d)
  abs (Frac a b) =
    Frac (abs a) (abs b)
  signum (Frac a b) =
    Frac (signum a * signum b) 1
  fromInteger n =
    Frac (fromInteger n) 1
  negate (Frac a b) =
    Frac (negate a) b
```

## definicija pripadnika



razredi  
konstruktorjev  
tipov



Razrede lahko priredimo tudi **konstruktorjem tipov**

In [1]: `:t (+)`

```
(+) :: forall a. Num a => a -> a -> a
```

In [2]: `:t sum`

```
sum :: forall (t :: * -> *) a. (Foldable t, Num a) => t a -> a
```

In [3]: `:t elem`

```
elem :: forall (t :: * -> *) a. (Foldable t, Eq a) => a -> t a ->  
Bool
```

## Primer konstruktorja za **dvojiška drevesa**

```
In [4]: data Drevo a
        = Prazno
        | Sestavljeno a (Drevo a) (Drevo a)
        deriving Show
```

```
In [8]: 4 `elem` mojeDrevo
```

```
<interactive>:1:3: error: [GHC-39999]
    • No instance for 'Foldable Drevo' arising from a use of 'e
      lem'
    • In the expression: 4 `elem` mojeDrevo
      In an equation for 'it': it = 4 `elem` mojeDrevo
```

```
In [5]: mojeDrevo =
        Sestavljeno 1
          (Sestavljeno 2
            (Sestavljeno 3 Prazno Prazno)
            Prazno
          )
        (Sestavljeno 4 Prazno Prazno)
```

Razredu `Functor` pripadajo **zbirke podatkov**

```
class Functor f where
    fmap :: (a -> b) -> f a -> f b
```

```
In [9]: instance Functor Drevo where
        fmap _ Prazno = Prazno
        fmap f (Sestavljeno x l d) =
            Sestavljeno (f x) (fmap f l) (fmap f d)
```

```
In [10]: fmap odd mojeDrevo
```

```
Sestavljeno True (Sestavljeno False (Sestavljeno True Prazno Pr
azno) Prazno) (Sestavljeno False Prazno Prazno)
```

```
In [ ]: fmap odd [1, 2, 3]
```

```
[True,False,True]
```



Razredu `Foldable` pripadajo zbirke **z vrstnim redom**

```
class Foldable t where
    foldr :: (a -> b -> b) -> b -> t a -> b
```

```
In [16]: instance Foldable Drevo where
        foldr f z Prazno = z
        foldr f z (Sestavljeno x l d) = foldr f (f x (foldr f z d)) l
```

```
In [17]: sum mojeDrevo
```

10

```
In [18]: 3 `elem` mojeDrevo
```

True

```
In [ ]: any (> 10) mojeDrevo
```

False

**Računski učinki**

# Računalniški programi **niso samo** funkcije

program  
=  
funkcija + **učinki**



# Kam slikajo programi, ki sprožajo učinke?

učinki	<i>int</i>	<i>float</i>
brez	$\mathbb{Z}$	$\mathbb{R}$
izjeme	$\mathbb{Z} + E$	$\mathbb{R} + E$
nedeterminizem	$\mathcal{P}(\mathbb{Z})$	$\mathcal{P}(\mathbb{R})$
izpis na izhod	$\mathbb{Z} \times \Sigma^*$	$\mathbb{R} \times \Sigma^*$
branje vhoda	$\Sigma^* \rightarrow \mathbb{Z}$	$\Sigma^* \rightarrow \mathbb{R}$
pomnilnik	$S \rightarrow \mathbb{Z} \times S$	$S \rightarrow \mathbb{R} \times S$

# Monada je podana s tremi komponentami

## funktor

za vsako množico  $A$  imamo množico **programov**  $TA$ ,  
ki **sprožajo učinke** in **vračajo vrednosti** v  $A$

## enota

za vsak  $v \in A$  imamo **trivialni** program  $\text{return } v : TA$

## veriženje

vsak  $c \in TA$  lahko **združimo z nadaljevanjem**  
 $f : A \rightarrow TB$  v program  $(c \gg f) \in TB$

Eugenio Moggi, 1960–



Veriženje je tudi v Haskellovem **logotipu**





# primeri monad v Haskellu



Philip Wadler, 1956–



# Za lažje pisanje Haskell podpira zapis do

pisanje z  $\gg$

```
get >>= \x ->
set (x + 30) >>
get >>= \y ->
set (x + y) >>
return "foo"
```

pisanje z **do**

```
do x <- get
    set (x + 30)
    y <- get
    set (x + y)
    return "foo"
```

# Monade zadoščajo **trem zakonom**

$$\text{return } v \gg f = f(v)$$

$$c \gg (x \mapsto \text{return } x) = c$$

$$(c \gg f) \gg g = c \gg (x \mapsto f(x) \gg g)$$



prost  
monada



**naslednjič...**

To so bila **zadnja** predavanja

