

## Nejdůležitější funkce

- **all::(a→Bool)→[a]→Bool**  
*all even [2,4,6] = True*  
*all even [1,4,6] = False*  
 $\Rightarrow \text{and } (\text{map } f [x])$
- **and::[Bool]→Bool**  
*and [False, True, True] = False*  
stačí jednou False
- **any::(a→Bool)→[a]→Bool**  
*any even [1,3,4] = True*  
 $\Rightarrow \text{or } (\text{map } f [x])$
- **concat::[[a]]→[a]**  
*concat [[6],[4]] = [6,4]*  
 $\Rightarrow \text{concat } [x] \Leftrightarrow \text{foldr } (++) [] [x]$
- **const::a→b→a**  
*const x y = x*
- **curry::(a,b)→c)→a→b→c**  
*curry f x y = f (x,y)*
- **cycle::[a]→[b]**  
*cycle [1,2] = [1,2,1,2,...]*
- **div::Integer→Integer→Integer**  
*div 9 2 = 4*  
celočíslné dělení
- **drop::Int→[a]→[a]**  
*drop 2 [3,8,9] = [9]*
- **dropWhile::(a→Bool)→[a]→[a]**  
*dropWhile odd [1,3,2,4] = [2,4]*
- **elem::Eq a⇒a→[a]→Bool**  
*elem 1 [1,2,6] = True*  
*elem 1 [2,9,8] = False*
- **even::Integer→Bool**  
Vrací True pro sudá čísla
- **filter::(a→Bool)→[a]→[a]**  
*filter odd [1,1,2] = [1,1]*
- **flip::(a→b→c)→b→a→c**  
*flip f x y = f y x*
- **foldr::(a→b→b)→b→[a]→b**  
*foldr:: (\*) 3 [1,2,3] = 18*  
Výpočet:  $(1 * (2 * (3 * 3)))$
- **foldr1::(a→a→a)→[a]→a**  
*foldr1 (\*) [2,3,4] = 24*
- **foldl::(a→b→a)→a→[b]→a**  
*foldl (+) [2,3,4] = 24*  
Výpočet:  $((3 + 1) + 2) + 3$
- **foldl1::(a→a→a)→[a]→a**  
*foldl1 (+) [1,2,3] = 6*
- **fst::(a,b)→a**  
*fst (a,b) = a*
- **gcd::Integer→Integer→Integer**  
*gcd x y = největší společný dělitel*
- **head::[a]→a**  
*head [x...] = x*  
head [] není definováno
- **id::a→a**  
*id x = x*
- **init::[a]→[a]**  
*init [...a,b,c] = [...a,b]*
- **iterate::(a→a)→a→[a]**  
*(3+) 2 = [2,5,8,11,...]*
- **last::[a]→a**  
*last [...a,b,c] = [c]*
- **lcm::Integer→Integer→Integer**  
nejmenší společný násobek
- **length::[a]→Int**  
*length [] = 0*

- **map::**( $a \rightarrow b$ )  $\rightarrow [a] \rightarrow [b]$   
*map square [1,3,5] = [1,9,25]*
- **max::ord**  $a \Rightarrow a \rightarrow a \rightarrow a$   
*max 9 8 = 9*  
*max 's' 'b' = 's'*
- **maximum::ord**  $a \Rightarrow [a] \rightarrow a$   
*maximum [2,9,5,4] = 9*
- **min::ord**  $a \Rightarrow a \rightarrow a \rightarrow a$   
*min 7 2 = 2*  
*min 'a' 'm' = 'a'*
- **minimum::ord**  $a \Rightarrow [a] \rightarrow a$   
*minimum [9,2,1,8] = 1*
- **mod::Integer**  $\rightarrow$  **Integer**  $\rightarrow$  **Integer**  
*mod 9 2 = 1*  
Zbytek po celočíselném dělení.
- **not::Bool**  $\rightarrow$  **Bool**  
*not False = True*
- **notElem::Eq**  $a \Rightarrow a \rightarrow [a] \rightarrow$  **Bool**  
*notElem 1 [2,9,6] = True*  
*notElem 1 [9,8,1] = False*
- **null::[a]**  $\rightarrow$  **Bool**  
*null [] = True*
- **odd::Integer**  $\rightarrow$  **Bool**  
vrací *True* pro lichá čísla.
- **or::[Bool]**  $\rightarrow$  **Bool**  
*or [False,False,True] = True*
- **product::[Integer]**  $\rightarrow$  **Integer**  
*product [1,2,3,4] = 24*
- **repeat::a**  $\rightarrow [a]$   
*repeat 3 = [3,3,..]*
- **replicate::Int**  $\rightarrow a \rightarrow [a]$   
*replicate 5 3 = [3,3,3,3,3]*
- **reverse::[a]**  $\rightarrow [a]$   
*reverse [1,2,3] = [3,2,1]*
- **show::a**  $\rightarrow$  **String**  
*show 56 = "56"*
- **snd::(a,b)**  $\rightarrow b$   
*snd (1,2) = 2*
- **signum::Integer**  $\rightarrow$  **Integer**  
*signum (-56) = -1*  
*signum (56) = 1*
- **sum::[Integer]**  $\rightarrow$  **Integer**  
*sum [1,2,3] = 6*
- **tail::[a]**  $\rightarrow [a]$   
*tail [y,x...x] = [x...x]*  
tail [] není definováno
- **take::Int**  $\rightarrow [a] \rightarrow [a]$   
*take 2 [3,4,5] = [3,4]*
- **takeWhile::(a  $\rightarrow$  Bool)**  $\rightarrow [a] \rightarrow [a]$   
*takeWhile odd [1,2,3,4] = [1,3]*
- **toLower::Char**  $\rightarrow$  **Char**  
*toLower 'M' = 'm'*
- **toUpper::Char**  $\rightarrow$  **Char**  
*toUpper 'd' = 'D'*
- **uncurry::(a  $\rightarrow$  b  $\rightarrow$  c)**  $\rightarrow (a,b) \rightarrow c$   
*uncurry f (x,y) = f x y*
- **unzip::(a,b)**  $\rightarrow ([a],[b])$   
*unzip [('a',1),('b',2)] = ("ab",[1,2])*
- **zip::[a]**  $\rightarrow [b] \rightarrow [(a,b)]$   
*zip "abc" [1,2,3] = [('a',1),('b',2),('c',3)]*
- **zipWith::(a  $\rightarrow$  b  $\rightarrow$  c)**  $\rightarrow [a] \rightarrow [b] \rightarrow [c]$   
*zipWith (\*) [3,4] [5,6] = [15,24]*
- **(,)::a**  $\rightarrow b \rightarrow (a,b)$   
*(,) x y = (x,y)*
- **(!!)::[a]**  $\rightarrow$  **Int**  $\rightarrow a$   
*[3,1,7] !! 2 = 7*

- $(:)\colon\mathbf{a}\rightarrow[\mathbf{a}]\rightarrow[\mathbf{a}]$   $True \parallel False = True$   
 $(:) \ 1 \ [2,3] = [1,2,3]$
- $(++)\colon[\mathbf{a}]\rightarrow[\mathbf{a}]\rightarrow[\mathbf{a}]$  •  $(\&\&\colon\mathbf{Bool}\rightarrow\mathbf{Bool}\rightarrow\mathbf{Bool})$   
 $True \ \&\& \ False = False$   
 $(++) \ [1,2] \ [3,4] = [1,2,3,4]$
- $(\parallel)\colon\mathbf{Bool}\rightarrow\mathbf{Bool}\rightarrow\mathbf{Bool}$  •  $[\ ]\colon[\mathbf{a}]$   
prazdný seznam

## Funkce pro stromy

- $preorder \ Empty = [ ]$   
 $preorder \ (Node \ v \ l \ r) = v:preorder \ l ++ preorder \ r$
- $postorder \ Empty = [ ]$   
 $postorder \ (Node \ v \ l \ r) = postorder \ l ++ postorder \ r ++ [v]$
- $inorder \ Empty = [ ]$   
 $inorder \ (Node \ v \ l \ r) = inorder \ l ++ [v] ++ inorder \ r$

## Monadické funkce

- **return** - vrací svůj argument jako výsledek akce
- **getLine** - načítá řádek ze standardního vstupu
- **getChar** - načítá znak ze stand. vstupu
- **putStr** - vypisuje argument na standardní výstup
- **putStrLn** - totéž co putStr, ale pokračuje na novém řádku
- $>>$
- $>>=$