

Funkcionális programozás 9. gyakorlat

Magasabbrendű függvények

- Függvény a paraméter
- Függvény a visszatérési érték

Curry-zés

- Minden függvény tulajdonképpen 1 paraméteres
- Pl.:

```
max 1 5 == (max 1) 5
--      ~~~~~
```

1. Mi a típusa a (max 1) kifejezésnek?

```
add3 :: (Int -> (Int -> (Int -> (Int))))    -- :: Int -> Int -> Int -> Int
add3 a b c = a+b+c
```

1. Mi a típusa az add3 1 kifejezésnek?

Szeletek

- Operátorokból képzett (egyparaméteres) függvények
 - Pl.: (+3), (<2), (*10)
 - Magasabb rendű függvények paraméterezésére használhatóak
1. Definiáld az apply :: Num a => (a -> a) -> a -> a függvényt, amely alkalmazza a paraméterül kapott szeletet a második paraméterére!
 2. Definiáld az applyTwice :: Num a => (a -> a) -> a -> a függvényt, amely kétszer alkalmazza a paraméterül kapott szeletet a második paraméterére!

Elemenként feldolgozás

1. Definiáljuk újra az elemenként feldolgozás függvényét!

```
map :: (a -> b) -> [a] -> [b]
map (+10) [1..10] == [11,12,13,14,15,16,17,18,19,20]
```

Szűrés

1. Definiáljuk újra a szűrés feldolgozás függvényét!

```
filter :: (a -> Bool) -> [a] -> [a]
filter (>5) [1..10] == [6,7,8,9,10]
```