

Haskell ve Fonksiyonel Programlama

Tuan Susam

Giris

- Fonksiyonel Programlama?

Temel Kavramlar

- Tipler & Tip Siniflari
- Listeler
- Fonksiyonlar
- Ifadeler

Uygulamali

- GHCi, Prelude ve Temel Fonksiyonlar
- Lambda Fonksiyonlari
- Katlanabilirler (foldables) Giris

- Fonksiyonlar ve bu fonksiyonların uygulamaları üzerine kurulu bir programlama yöntemidir.
- Kullanılan matematiksel notasyon problemlerin açıkça ve farkındalıkla belirtilmesini sağlar.
- Programların özellikleri hakkında denklemsel sebep verebilmeye olanak sağlayan matematiksel bir temeli vardır.

Neden Fonksiyonel?

Declarative Programming

Fonksiyonel programlama, hesaplama kavramini matematiksel fonksiyonların hesaplanması olarak görür. Diğer dillerde sıkça karşılaştığımız *changing-state* ve *mutable data*'dan kaçınır.

Declarative programming, ifadeler ve beyanlarla (**expressions & declarations**) yapılır, *statement* kavramı burada geçerli değildir. Hesaplamanın *control flow*'unu anlatmadan mantığını anlatarak yapılan bir programlamadır

Örnekleri

Common Lisp (1980s), Scheme (1970s), Clojure (2007), Wolfram Language (1988/2019), Racket (1995/2019), Erlang (1986/2019), OCaml (1996/2018), Haskell (1990/2010), F# (2005/2019)

Adini Amerikalı mantıkçi Haskell Brooks Curry'den alıyor

- Tembel Fonksiyonel Dil

Sektörde Haskell

- Facebook anti-spam programları
- Swift Navigation
- Github
- Cryptool

Web Frameworks

- Yesod
- Snap

Sayılar

- Int
- Integer
- Rational
- Float
- Double
- Complex (Data.Complex)

Bool

Boolean tipidir, *True* veya *False* değerlerinden birini alır

Char

Bir karakteri temsil eder 'a' gibi, karakterler listesi ise *string*'dir

Tip Siniflari

Eq

Bir fonksiyonun tip degiskeni icin **Eq** sinifi belirteci varsa fonksiyonumuz taniminda `==` veya `/=` icermektedir.

5 'elem' [1..10]

Ord

Ord butun standart kiyaslama fonksiyonlarini icerir. Ornegin `>`, `<`, `>=`, `<=`. *Kiyaslama* fonksiyonu iki **Ord** uyesi (ayni tipten) alir ve bir *ordering* geri getirir.

data Ordering = LT | EQ | GT

Show

Show'un uyesi olan bir degeri alip bize "Gosterir".

Read

Show sinifinin tam tersidir

Enum

Sirali kiyasli tipler bu sinifa uyedir. Bu sinifa dahil tipler *()*, *Bool*, *Char*, *Ordering*, *Int*, *Integer*, *Float*, *Double*

['A'..'Z']

[3 .. 5]

succ 'b'

Bounded

uyulerinin ust ve alt sinirlari vardir.

Num, Integral, Floating

Numeric tipsinifidir. Uyeleri sayi gibi davranabilirler **Num** sinifi butun sayilari icine alirken, **Integral** sinifinda sadece **Int** ve **Integer**lar vardir. Benzer bir sekilde **Float** ve **Double** tipleri **Floating** tip sinifindadir.

fromIntegral

Tum bu tipler ve tip siniflari karmasasi icerisinde, problemlerimizin cozumu olan bu fonksiyon farkli tiplerle islem yapabilmemizi sagliyor.
fromIntegral (length "length bize Int getirecek") + pi

Daha Detayli Bilgi Icin

www.learnyouahaskell.com/types-and-typeclasses

Biraz Sayılar

negate $x = -x$

abs $x = \text{if } x < 0 \text{ then } -x \text{ else } x$

signum $x \mid x < 0 = -1$
 $\mid x == 0 = 0$

 $\mid x > 0 = 1$

divMod

fonksiyonu bize sonuc olarak (bolum, kalan) seklinde 2'li-tup verir

$x \text{ 'div' } y = \text{fst } (x \text{ 'divMod' } y)$

$x \text{ 'mod' } y = \text{snd } (x \text{ 'divMod' } y)$

Biraz Daha Sayılar ve Fonksiyonlar :)

floor?

floor' :: Float -> Integer

floor' = read . takeWhile (/= '.') . show

Until

until :: (a -> Bool) -> (a -> a) a -> a

until p f x = if p x then x else until p f (f x)

until (>100) (*7) 1 == 343

Subtract

subtract x y = y-x

O Zaman List

Verileri çekmek veya tasamak için kullanılan yapılardan biridir

```
[undefined,undefined] :: [a]  
[sin,cos,tan] :: Floating a => [a -> a]  
[[1,2,3],[4,5]] :: Num a => [[a]]  
["Seminer Odasi",'M',231]
```

Liste Oluşturmak

```
1:2:3:[] == [1,2,3]
```

Liste Elemanı Çağırma

```
[0,1,2,3,4,5]!!3 == 3
```

Listelerin Özellikleri

$[a]$ tipindeki bir liste aşağıdaki 3 formdan birindedir

- tanımsız liste $:: [a]$
- boş liste $[] :: [a]$
- $(x:xs)$ formunda olan bir listede $x :: a$ 'dir ve $xs :: [a]$

Dolayısıyla 3 Cesi liste vardır

- Sonlu listeler
- Parçalı listeler
- Sonsuz listeler

Listelerle Çalışan Fonksiyonlar

- head
- tail
- last
- reverse
- length
- sum
- take n
- iterate
- zip
- zipWith

Kume Notasyonu!?

$[x*x \mid x \leftarrow [1..10], \text{isPrime } x]$

$[(i,j) \mid i \leftarrow [1..5], j \leftarrow [1..5]]$

Kelime Frekansi Problemi

Elimizde bir paragraf ,uzun bir string, oldugunu dusunelim. Bu paragrafta en cok tekrar eden n tane kelimeyi bize verecek programi nasil yazabiliriz?

