wსაგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=5+4+5+5+4

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი =23

1. განსაზღვრეთ შემდეგი ფუნქცია ორი ხერხით: პირობითი ოპერატორისა და დაცული განტოლებების გამოყენებით:  მნიშვნელობის გამოსათვლელად. როგორ მიმართავთ ამ ფუნქციას? ფუნქციის ტიპი განსაზღვრეთ ცხადად.

let y x = if x >= (-3) && x <= 7 && x /= -2.5 then (1-x^2+8.2\*x)/(x+2.5)-4

else if x == -2.5 then 3.75\*x^2+10/((2-x)\*(1+4\*x))

else 5\*x-7+3.5\*x^3/(4\*(1.5+x))

y :: (Ord a, Fractional a) => a -> a

let y2 x | (x >= (-3) && x <= 7 && x /= -2.5) = (1-x^2+8.2\*x)/(x+2.5)-4

| (x == -2.5) = 3.75\*x^2+10/((2-x)\*(1+4\*x))

| otherwise = 5\*x-7+3.5\*x^3/(4\*(1.5+x))

y2 :: (Ord a, Fractional a) => a -> a

**Prelude>** y 5

-1.7333333333333334

it :: Double

**Prelude>** y 10

119.08695652173913

it :: Double

**Prelude>** y 1

-1.6571428571428575

it :: Double

**Prelude>** y2 1

-1.6571428571428575

it :: Double

**Prelude>** y (-2.5)

23.190586419753085

it :: Double

1. განსაზღვრეთ შემდეგი ფუნქციები λ ნოტაციის გამოყენებით და მიუთითეთ შესაბამისი ტიპები:

* ფუნქცია, რომელიც აბრუნებს სიის მეორე ელემენტს.
* let f = \xs -> xs!!1
* f :: [a] -> a
* **Prelude>** f [1,2,3,4]
* 2
* it :: Integer
* ფუნქცია, რომელიც გადაცემული სიისთვის აბრუნებს ლუწი ელემენტების კვადრატებს, კენტების -კუბებს.

let f = \xs -> map (\x -> if rem x 2==0 then x^2 else x^3)xs

f :: [Integer] -> [Integer]

**Prelude>** f [2,3,4,5]

[4,27,16,125]

it :: [Integer]

* ფუნქცია, რომელსაც გადაეცემა ელემენტი და სია და დააბრუნებს ახალ სიას, სადაც ბოლო ელემენტად ჩასმულია პირველი არგუმენტი. -1

1. განსაზღვრეთ ცხრილით მოცემული $ ფუნქცია **რამდენიმე** შაბლონის გამოყენებით:

|  |  |  |
| --- | --- | --- |
| a | b | a $$ b |
| False | False | True |
| False | True | True |
| True | False | False |
| True | True | True |

განსაზღვრეთ ფუნქციის ტიპი.

let fun a b = if a == b then True   
else if a == False then True

else False

fun :: Bool -> Bool -> Bool

function False False = True

function False True = True

function True True = True

function \_ \_ = False

function :: Bool -> Bool -> Bool

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

* იმ სამნიშნა რიცხვების სია, რომელთა ციფრთა ჯამი არის კენტი რიცხვი.
* [x|x<-[100..999],(x`mod`10+x`div`10`mod`10+x`div`100)`mod`2==1]
* შეადგინეთ ისეთი (x,y,z) სამეულების სია, სადაც თითოეული ცვლადი არის მარტივი, მნიშვნელობით 1–დან n-მდე და ჭეშმარიტია პირობა x+y==z.

let prime n = length [x|x<-[1..n], n`mod`x == 0] == 2

let sia n = [(x,y,z) | x<-[1..n],y<-[1..n],z<-[1..n],prime x, prime y, prime z, x+y==z]

sia :: (Num t, Enum t) => t -> [(t, t, t)]

* სია [22, 33, ... 2020].

[x^x|x<-[2..20]]

1. განსაზღვრეთ ფუნქცია ორი ხერხით: REPL გარემოში და საწყისი კოდის სახით. ფუნქცია გამოიძახეთ კონკრეტული მონაცემებისთვის:

* ფუნქცია, რომელიც აწყვილებს თავის ორ არგუმენტს.

f1 x y = (x,y) +2

f1 :: t -> t1 -> (t, t1)

* ფუნქცია, რომლის არგუმენტია სტრიქონი და რომელიც ამოწმებს, არის თუ არა არგუმენტი პალინდრომი.
* ფუნქცია isSorted, რომელიც შესასვლელზე ღებულობს სამ რიცხვს და აბრუნებს True, თუ ეს რიცხვები დალაგებულია ზრდადობით ან კლებადობით. +2

isSorted a b c = if (a < b && b < c) || (a > b && b >c) then True

else False