საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=5+5+4+5+5

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი =24

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

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

u x = if x==0.4 then 4\*x^3+100/(3\*x)-5

else if x>=(-2) then (3\*x^2-7\*x-3)/(5\*x-2)

else 7\*(3.5+2\*x)-1/(2.5\*(1-x))

u x | x==0.4 = 4\*x^3+100/(3\*x)-5

| x>=(-2) = (3\*x^2-7\*x-3)/(5\*x-2)

|otherwise = 7\*(3.5+2\*x)-1/(2.5\*(1-x))

u 0.4

u 3

u (-3)

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

* ფუნქცია, რომელსაც გადაეცემა ორი არგუმენტი და ითვლის ნამდვილ ხარისხს.

(\x y -> x\*\*y) :: Floating a => a -> a -> a

(\x y -> x\*\*y) 3 2

* ფუნქცია, რომელიც ორჯერ ითვლის მნიშვნელობას არგუმენტზე.

fun :: (t -> t) -> t -> t

let fun f x = f (f x)

* ფუნქცია, რომელიც გადაცემული რიცხვების სიისთვის ამოწმებს თითოეული ელემენტის ლუწ-კენტობას და გვიბრუნებს ლოგიკური მნიშვნელობების სიას.

trueFalse :: Integral a => [a] -> [Bool]

trueFalse xs = map (\x -> even x) xs

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

|  |  |  |
| --- | --- | --- |
| a | b | a⇒b |
| False | False | True |
| False | True | False |
| True | False | False |
| True | True | True |

განსაზღვრეთ ფუნქციის ტიპი.

Function :: Bool -> Bool -> Bool

function True a = a

function False a = not a

სხვა შაბლონები +4

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

* ხუთნიშნა რიცხვების სია, სადაც ყველა ციფრი ერთნაირია;

[x\*10000+x\*1000+x\*100+x\*10+x|x<-[1..9]]

* გამოსახულება, რომლითაც დაითვლით 1000–ზე ნაკლები 3–ის ან 7–ის ჯერადი ნატურალური რიცხვების ნამრავლი ?

[x|x<-[1..999],x `mod` 3==0,x`mod`7==0]

* სია [22, 44,...2020].

[x^x|x<-[2,4..20]]

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

* ფუნქცია, რომელიც გადაცემული სიისთვის აბრუნებს ლუწი ელემენტების კვადრატებს, კენტების -კუბებს.

let f = map (\x -> if even x then x\*x else x\*x\*x )

f :: [Integer] -> [Integer]

f = map (\x -> if even x then x\*x else x\*x\*x )

f [1,2,3]

* ფუნქცია, რომელსაც გადაეცემა ორი სია და დააბრუნებს მათ გაერთიანებას.

let f = (\xs ys -> xs ++ ys)

f :: [a] -> [a] -> [a]

f = (\xs ys -> xs ++ ys)

f [‘e’,’w’] [‘a’,’n’]

* ფუნქცია isRectangular, რომელიც პარამეტრად ღებულობს სიბრტყეზე სამი წერტილის კოორდინატებს და აბრუნებს True–ს, თუ მათ მიერ შედგენილი სამკუთხედი არის მართკუთხა სამკუთხედი.

isRectangular (x1,y1) (x2,y2) (x3,y3) = if (l1^2+l2^2 == l3^2 || l1^2+l3^2 == l2^2 || l2^2+l3^2 == l1^2)

where l1 = sqrt ((x1-x2)^2+(y1-y2)^2)

l2 = sqrt ((x1-x3)^2+(y1-y3)^2)

l3 = sqrt ((x2-x3)^2+(y2-y3)^2)

isRectangular (1,2) (5,7) (5,2)