საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25= 5+5+5+4 +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. როგორ მიმართავთ ამ ფუნქციას? ფუნქციის ტიპი განსაზღვრეთ ცხადად.

პირობითი ოპერატორებით:

y x = if x>=0 then 2x^3-7x/(3(2-x^2))

else if(x<0 && x/=(-5)) then (15-3x^2)/(10+2x)+1/(4(1-x))

else 5\*(3+x/(x^2+1))-225

\*Main> y (-5)

-210.96153846153845

it :: Double

\*Main> y 1

-0.3333333333333335

it :: Double

\*Main> y 7

686.3475177304964

it :: Double

დაცული გნტოლებებით:

y x | x>=0 = 2x^3-7x/(3(2-x^2))

|(x<0 && x/=(-5)) = (15-3x^2)/(10+2x)+1/(4(1-x))

| otherwise = 5\*(3+x/(x^2+1))-225

\*Main> y (-5)

-210.96153846153845

it :: Double

\*Main> y 7

686.3475177304964

it :: Double

\*Main> y (-11)

29.020833333333332

it :: Double

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

* ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.

o ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.

Prelude> map((x,y)->(y,x)) [(4,5)]

[(5,4)]

it :: [(Integer, Integer)]

Prelude> funmap((x,y)->(y,x)) [('k','l')]

Prelude> map((x,y)->(y,x)) [('k','l')]

[('l','k')]

it :: [(Char, Char)]

* ფუნქცია, რომელიც გადაცემული სიისთვის აბრუნებს 3-ის ჯერადი ელემენტების კვადრატებს, დანარჩენი ელემენტების კუბებს.

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

fun :: [Integer] -> [Integer]

Prelude> fun [1,2,3]

[1,8,9]

it :: [Integer]

Prelude> fun [10, 3, 5, 6]

[1000,9,125,36]

it :: [Integer]

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

Prelude> let fun = (xs) -> (ys) -> ((xs)++(ys))

fun :: [a] -> [a] -> [a]

Prelude> fun [1,2,3] [4,5]

[1,2,3,4,5]

it :: [Integer]

Prelude> fun ['a','b'] ['d','z']

"abdz"

it :: [Char]

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

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

განსაზღვრეთ ფუნქციის ტიპი.

fun a b = if a==b then False

else True

fun :: Eq a => a -> a -> Bool

f False False=False

f False True=True

f True False=True

f True True=False

f :: Bool -> Bool -> Bool

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

* სამნიშნა რიცხვების სია, სადაც ყველა ციფრი განსხვავებულია;

[a\*100+b\*10+z|a<-[1..9],b<-[0..9],z<-[0..9],a/=b&&a/=z&&b/=z ] +2

* გამოსახულება, რომლითაც დაითვლით 1000000–ზე ნაკლები 3–ის და 7–ის ჯერადი ნატურალური რიცხვების ნამრავლს და რაოდენობას?
* სია [22, 32, ... 202 ] +2

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

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

* ფუნქცია, რომელიც აბრუნებს გაორმაგებულ

double x = 2 \* x +2

let double x = x \* 2

* ფუნქცია max3, რომელიც სამი მთელი რიცხვიდან აბრუნებს მათ შორის უდიდესს.

let maxthree x y z = max x (max y z) +2

maxthree x y z = max x (max y z)

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

let isRectangular x y z = if ((x^2)+(y^2))==z^2 then True else False არ არის საკმ.პირობა +1