საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=5+4+5+4+5

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი =23

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

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

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

else 3 \* x \* x - 7 \* x - 3 / (5 \* x - 2)

func :: (Floating a, Ord a) => a -> a

func1 x

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

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

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

func1 :: (Floating a, Ord a) => a -> a

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

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

f1 = (\x y -> x \*\* y)

f1 :: Double -> Double -> Double

* ფუნქცია, რომელიც ორჯერ ითვლის მნიშვნელობას არგუმენტზე. +4
* ფუნქცია, რომელიც გადაცემული რიცხვების სიისთვის ამოწმებს თითოეული ელემენტის ლუწ-კენტობას და გვიბრუნებს ლოგიკური მნიშვნელობების სიას.

f2 = \xs ->[if even x then True else False| x <- xs]

f2 :: [Integer] -> [Bool]

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.

fs1 True True = True

fs1 False False = True

fs1 \_ \_ = False

fs1 :: Bool -> Bool -> Bool

fs2 a b = if a == b then True else False

fs2 :: Eq a => a -> a -> Bool

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

* ხუთნიშნა რიცხვების სია, სადაც ყველა ციფრი ერთნაირია;

fun = [ x | x <- [1..99999], length (tail (subsequences (show x) +4

`intersect` subsequences (show tail x))) > 0 ]

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

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

* სია [22, 44,...2020].

fun = [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.

f1 xs = [if even x then x \* x else x \* x \* x| x <- xs]

f1 :: Integral a => [a] -> [a]

\*Midterm> f1 [1,2,3,4]

[1,4,27,16]

let f1 xs = [if even x then x \* x else x \* x \* x| x <- xs]

f :: Integral a => [a] -> [a]

\*Midterm> f [2,3,4,5,6]

[4,27,16,125,36]

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

f2 xs ys = xs ++ ys

f2 :: [a] -> [a] -> [a]

\*Midterm> f2 [1,2] [3,4,5]

[1,2,3,4,5]

let f xs ys = xs ++ ys

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

\*Midterm> f [3,4,5] [1,2]

[3,4,5,1,2]

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

f3 [x1, y1] [x2, y2] [x3,y3] =

((x2 - x1)\*\*2 + (y2 - y1)\*\*2)+ ((x3 - x1)\*\*2 +

(y3 - y1)\*\*2) == ((x3 - x2)\*\*2 + (y3 - y2)\*\*2)

f3 :: (Eq a, Floating a) => [a] -> [a] -> [a] -> Bool

Let f3 [x1, y1] [x2, y2] [x3,y3] =

((x2 - x1)\*\*2 + (y2 - y1)\*\*2)+ ((x3 - x1)\*\*2 +

(y3 - y1)\*\*2) == ((x3 - x2)\*\*2 + (y3 - y2)\*\*2)

f3 :: (Eq a, Floating a) => [a] -> [a] -> [a] -> Bool

f3 [2,4] [-1,0] [2,0]

False

წესით True უნდა დააბრუნოს როგორც ვხვდები რაღაც მათემატიკური შეცდომა მაქვს

უნდა შემოწმდეს ყველა შესაძლო ტოლობა, იმიტომ რომ არ იცი ჰიპოტენუზა რომელია.