საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=5+4+5+3+4

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი=21

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

პირობითი ოპერატორით:

y x = if x>5 then (4\*x-5\*x^2+2)/(2+2\*x) - 1/(3\*(x-1))

else if x<=5 && x /=(-8) then 3\*x^2-(5/(2\*(x+8))+10

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

it::Double

დაცული განტოლებით:

y x | if x>5 = (4\*x-5\*x^2+2)/(2+2\*x) - 1/(3\*(x-1))

| x<=5 && x /=(-8) = 3\*x^2-(5/(2\*(x+8))+10

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

it::Double

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

* ფუნქცია, რომელიც აბრუნებს გაორმაგებულ არგუმენტს.

let fun = \a -> 2\*a

fun :: Integer -> Integer

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

let fun = \(a,b) -> a^b -1

fun :: (Integer, Integer) -> Integer

მთელ ხარისხში კონკრეტულად რა იგულისხმებოდა ვერ მივხვდი, ამიტომ ისეთი ფუნქცია დავწერე რომელსაც პირველი არგუმენტი აჰყავს ხარისხში მეორე არგუმენტის ხარისხში

* ფუნქცია, რომელსაც გადაეცემა ორი წყვილი და დააბრუნებს შესაბამისი ელემენტების ჯამების წყვილს. მაგალითად, არგუმენტებია: (1,2), (3,4), შედეგია: (4,6).

let fun = \(a,b) -> \(x,y) -> (a+x,b+y)

fun ::

(Integer, Integer) -> (Integer, Integer) -> (Integer, Integer)

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 == True && b == False then False else True

False $$ False = True

False $$ True = True

True $$ False = False

True $$ True = True

fun :: Bool -> Bool -> Bool

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

* იმ სამნიშნა რიცხვების სია, რომელთა ციფრთა ჯამი არის 3-ის ჯერადი რიცხვი;

map (\x-> if mod x 2 == 0 then x^2 else x^3) [1..10] [x | x <- [100..999] , (mod x 10 + div (mod x 100 ) 10 + div x 100) mod 3 == 0]

* მოცემულ n რიცხვამდე ისეთი რიცხვების სია, რომელიც ორი მარტივი რიცხვის ჯამით; წარმოდგება. -2
* გამოსახულება , რომელიც პირველი 100 რიცხვის 12 + 22 + . . . + 1002 ჯამს ანგარიშობს.

sum [x^2 | x <- [1..100]]

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

* ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.

let swap (x,y) = (y,x)

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

swap(2,7)

(7,2)

it :: (Integer, Integer)

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

let func y = [x `mod` 2 == 0 | x <- [1..y]]

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

func 5

[False,True,False,True,False]

it :: [Bool]

* ფუნქცია isTriangle, რომელიც განსაზღვრავს, შეიძლება თუ არა მოცემულ x, y და z სიგრძის მონაკვეთებზე აიგოს სამკუთხედი.

let isTriangle (x, y, z) = if x+y > z && x+z > y && y+z > x then True else False

isTriangle :: (Ord a, Num a) => (a, a, a) -> Bool

isTriangle(2, 4, 8)

False

it :: Bool

isTriangle(4, 5, 6)

True

it :: Bool