საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=4+5+2+4+4

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი=19

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

GFunction :: Double -> Double -1

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

if x == (-8) then (7 \* x + 1 / (1 + x)) / (x \* x - 1) + 1 / 2 \* x else 3 \* x \* x - 5 / (2 \* (x + 8)) + 10

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

| x == (-8) = (7 \* x + 1 / (1 + x)) / (x \* x - 1) + 1 / 2 \* x

| otherwise = 3 \* x \* x - 5 / (2 \* (x + 8)) + 10

Gfunction 10

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

* ფუნქცია, რომელიც აბრუნებს გაორმაგებულ არგუმენტს.   
  double :: Int -> Int

double = \x -> x + x

* ფუნქცია, რომელსაც გადაეცემა ორი არგუმენტი და ითვლის მთელ ხარისხს.  
  power :: Int -> Int -> Int   
  power = \x y -> x ^ y
* ფუნქცია, რომელსაც გადაეცემა ორი წყვილი და დააბრუნებს შესაბამისი ელემენტების ჯამების წყვილს. მაგალითად, არგუმენტებია: (1,2), (3,4), შედეგია: (4,6).   
  addPoints :: (Int, Int) -> (Int, Int) -> (Int, Int)   
  addPoints = \(x1, y1) (x2, y2) -> (x1 + x2, y1 + y2)

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.

GFunction :: Bool -> Bool -> Bool +2  
GFunction False False = True   
GFunction False True = True   
GFunction True False = False   
GFunction True True = True  
სხვა ...

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

* იმ სამნიშნა რიცხვების სია, რომელთა ციფრთა ჯამი არის 3-ის ჯერადი რიცხვი;  
  sumsUpToThree = [102, 105..999] +2
* მოცემულ n რიცხვამდე ისეთი რიცხვების სია, რომელიც ორი მარტივი რიცხვის ჯამით; წარმოდგება.
* გამოსახულება , რომელიც პირველი 100 რიცხვის 12 + 22 + . . . + 1002 ჯამს ანგარიშობს.  
  sumSquares = sum [x ^ 2 | x <- [1..100]] +2

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

* ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.  
  swapElements (x, y) = (y, x)   
  swapElements (1, 2)
* ფუნქცია, რომელიც გადაცემული რიცხვების სიისთვის ამოწმებს თითოეული ელემენტის ლუწ-კენტობას და გვიბრუნებს ლოგიკური მნიშვნელობების სიას.  
  areEven :: [Int] -> [Bool]   
  areEven arr = map (\elem -> mod elem 2 == 0) arr  
  areEven [ 1, 4, 9, 8]
* ფუნქცია isTriangle, რომელიც განსაზღვრავს, შეიძლება თუ არა მოცემულ x, y და z სიგრძის მონაკვეთებზე აიგოს სამკუთხედი.

isTriangle :: (Int, Int, Int) -> Bool   
isTriangle (x, y, z) = (x + y) > z && (x + z) > y && (y + z) > x  
isTrainge (3, 5, 8)