საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=3+0+3+1+4

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი=10

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

**func :: (Ord a, Fractional a) => a -> a +3**

**func x**

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

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

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

**func1 x**

**If x > 5 then (4\*x-5\*(x^2)+2))/(2+2\*x)) - 1/3(x -1) else**

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

**If x = (-8) then (7\*x + 1/1+x)/x^2 - 1 + 1/2\*x**

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

* ფუნქცია, რომელიც აბრუნებს გაორმაგებულ არგუმენტს. 0

**list = [x\*2 | x <- list] -**

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

**pow x y=fix(\r a i->if i>=y then a else r(a\*x)(i+1))1 0 -**

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

**addTuples :: (Num a) => [(a, a)] -> [a] -**

**addTuples :: [] = []**

**addTuples ((a, b) : xs) = (a+b) : addTuples xs**

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.

**($) :: Bool -> Bool -> Bool +3**

**($) False False = True**

**($) False True = True**

**($) True False = False**

**($) True True =True**

**----------------------------**

**(>>=) :: Bool -> Bool -> Bool**

**(>>=) False False = True**

**(>>=) False True = True**

**(>>=) True False = False**

**(>>=) True True =True**

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

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

**[x | x <- [100..999], (x ‘div’ 100 + (x’div’10) ‘mod’ 10 + x’mod’ 10) ‘mod’ 3 === 0 +1**

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

**sumOfSquares :: Integral a => a -> a -> a**

**sumOfSquares lo hi**

**| lo > hi = error "error"**

**| otherwise = sum $ map (^2) [lo..hi]**

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

* ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.

**fun x = (snd x, fst x) +2**

* ფუნქცია, რომელიც გადაცემული რიცხვების სიისთვის ამოწმებს თითოეული ელემენტის ლუწ-კენტობას და გვიბრუნებს ლოგიკური მნიშვნელობების სიას.
* ფუნქცია isTriangle, რომელიც განსაზღვრავს, შეიძლება თუ არა მოცემულ x, y და z სიგრძის მონაკვეთებზე აიგოს სამკუთხედი.

**isTriangle x y z +2**

**| x + y > z && y + z > x && x + z > y**

**| otherwise = False**

**-------------------------------------------------**

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