საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=5+5+5+4+3

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ონისე ჩაგანავა

ბილეთი =25

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 2\*x^3-7\*x/(3\*(2-x^2))

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

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

**\*Main>** y (-5)

-210.96153846153845

it :: Double

**\*Main>** y(-10)

28.522727272727273

it :: Double

**დაცული განტოლებებით:**

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

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

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

**\*Main>** y (-5)

-210.96153846153845

it :: Double

**\*Main>** y (20)

16000.117252931323

it :: Double

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

* ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.

**Prelude>** map(\(x,y)->(y,x)) [(14.5,15.5)]

[(15.5,14.5)]

it :: [(Double, Double)]

* 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 [6, 5, 12]

[36,125,144]

it :: [Integer]

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

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

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

**Prelude>** fun [999, 666] [88, 99]

[999,666,88,99]

it :: [Integer]

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;

[a100+b10+d|a<-[1..9],b<-[0..9],d<-[0..9],a/=b&&a/=d&&b/=d ]

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

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

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

* ფუნქცია, რომელიც აბრუნებს გაორმაგებულ არგუმენტს.

let fun x = x \* 2 +1

fun :: Num a => a -> a

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

let max3 a b c = max a (max b c) +2

max3 :: Ord a => a -> a -> a -> a

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