საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=5+5+5+3+4

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი =22

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

ა)პირობითი ოპერატორების გამოყენებით

**fun :: (Ord a, Fractional a) => a -> a**

**fun y x = if x >= (-2) && x\=0.4 then (3\*x^2 -7\*x -3)/(5\*x -2) else if x==0.4 then (4\*x^3 +(100/(3\*x)-5) else 7\*(3.5+2\*x)-(1/(2.5\*(1-x)**

**ფუნქციის ტიპი: fun :: (Ord a, Fractional a) => a -> a**

**ფუნქციის მიმართვა :f un 5**

**ბ) დაცული განტოლებების გამოყენებით**

**fun :: (Ord a, Fractional a) => a -> a**

**Fun x | x>=(-2)= (3\*x^2 -7\*x -3)/(5\*x -2)**

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

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

**ფუნქციის ტიპი fun :: (Ord a, Fractional a) => a -> a**

**ფუნქციის მიმართვა :f un 5**

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

* ფუნქცია, რომელსაც გადაეცემა ორი არგუმენტი და ითვლის ნამდვილ ხარისხს.
* **(\x y->x\*\*y) 2 3**
* **8.0**
* **it :: Double**
* ფუნქცია, რომელიც ორჯერ ითვლის მნიშვნელობას არგუმენტზე.
* **(\f x->f(f x))(\*2) 3**
* **12**
* **it :: Integer**
* ფუნქცია, რომელიც გადაცემული რიცხვების სიისთვის ამოწმებს თითოეული ელემენტის ლუწ-კენტობას და გვიბრუნებს ლოგიკური მნიშვნელობების სიას.
* **map (\x->x`mod`2==0)[1,2,3]**
* **[False,True,False]**
* **it :: [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.

**პირველი შაბლონი: მეორე შაბლონი:**

**(##) ::Bool->Bool->Bool**

**False ## False =True fun False False=True**

**False ## True =False fun False True =False**

**True ## False =False fun True False =False**

**True ## True =True fun True True =True**

**ფუნქციის ტიპი:(##) ::Bool->Bool->Bool**

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ: +3

* ხუთნიშნა რიცხვების სია, სადაც ყველა ციფრი ერთნაირია;

**let func = [ x | x <- [10000..99999], x `mod` 10 == x `div` 10000 && x `mod` 10 == (x `div` 1000) `mod` 100 && x` mod` 10 == (x `div` 100) ` mod` 10] სწორე არაა**

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

**x=product[x<-[1..1000],x`mod`3==0| |x`mod`7==0]**

* სია [22, 44,...2020].
* **x=[x<-[2,4..20],x^x]**

1. განსაზღვრეთ ფუნქცია ორი ხერხით: REPL გარემოში და საწყისი კოდის სახით. ფუნქცია გამოიძახეთ კონკრეტული მონაცემებისთვის: =+4

* ფუნქცია, რომელიც გადაცემული სიისთვის აბრუნებს ლუწი ელემენტების კვადრატებს, კენტების -კუბებს.

**Func ::(Num->a)[a]->a --** func :: Integral a => a -> a

**Func x=if x`mod`2==0 then x^2 else x^3**

**Func as=map func as ორივე ფუნქციას ერთიდაიგივე სახელი?**

**მეორე ვარიანტი;**

**fun :: [Integer] -> [Integer]**

**let fun =\xs -> map (\x -> if x`mod` 2==0 then x^2 else x^3)xs**

swore versia: (\xs -> map (\x -> if x`mod` 2==0 then x^2 else x^3)xs )[1,3,5,2]

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

**Module test where**

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

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

**მეორე ვარიანტი:**

**(\xs ys ->xs++ys) [2,3] [3,7]**

**[2,3,3,7]**

**it :: [Integer]**

* ფუნქცია isRectangular, რომელიც პარამეტრად ღებულობს სიბრტყეზე სამი წერტილის კოორდინატებს და აბრუნებს True–ს, თუ მათ მიერ შედგენილი სამკუთხედი არის მართკუთხა სამკუთხედი.
* **----**
* **Module test where**
* **isRectangular a b c=a^2==b^2+c^2||a^2+b^2==c^2||b^2==c^2+a^2**