საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

ქულათა ჯამი: 15

ქულათა გადანაწილების ინსტრუქცია: 1 საკ -3ქ., 2 საკ -2ქ., 3 საკ -3ქ.,4 საკ-2 ქ., 5 საკ -5 ქ.

ბილეთი N=11

1. მოიყვანეთ არატრივიალური გამოსახულებების მაგალითები, რომლებიც ეკუთვნის ტიპებს:

a)((Char,Integer), String, [Double])

**(('C',’V’,7,8),"bar",”vashli”,[4.5,6.6])**

1. [(([Double],[Bool]),[Integer])]

**[(([5.5,7.7],[True,False]),[9,10])] მეორე წევრი -1**

1. (Double,Bool,(String,Integer))

**(1.1,6.2,False,True,("var",3,7))**

მოთხოვნა გამოსახულებების არატრივიალურობის შესახებ ნიშნავს, რომ გამოსახულებებში მონაწილე სიები უნდა შეიცავდნენ ერთ ელემენტზე მეტს.

1. როგორია შემდეგი მნიშვნელობების ტიპები:

([False,True],[’0’,’1’])

**Prelude>** :set +t

**Prelude>** ([False,True],['0','1'])

([False,True],"01")

it :: ([Bool], [Char])

[’a’,’b’,’c’]

**Prelude>** :set +t

**Prelude>** ['a','b','c']

"abc"

it :: [Char]

(“1”,”2”,”3”)

**Prelude>** :set +t

**Prelude>** ("1","2","3")

("1","2","3")

it :: ([Char], [Char], [Char])

1. განსაზღვრეთ შემდეგი ფუნქციების მხოლოდ ტიპები პოლიმორფიზმის გამოყენებით:
2. pair x y = (x,y)

**Prelude>** let pair x y = (x,y)

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

1. fun1 xs=(sum xs)\*(product xs)

**Prelude>** let fun1 xs=(sum xs)\*(product xs)

fun1 :: Num a => [a] -> a

c)length

**Prelude>** :type length

length :: [a] -> Int

d)[sum, product]

**Prelude>** :type [sum, product]

[sum, product] :: Num a => [[a] -> a]

თითოეული ფუნქცია გამოიძახეთ კონკრეტული მონაცემებისთვის. -1

1. მოცემულია ფუნქციის ტიპები:
2. fun :: Num a => a -> a

**Prelude>** let fun a = a\*a

fun :: Num a => a -> a

1. fun :: Ord a => (a, a) -> Bool

**Prelude>** let fun (a,c) = a>c

fun :: Ord a => (a, a) -> Bool

თითოეული ტიპისთვის მოიყვანეთ მაგალითი ფუნქციებისა, რომლებსაც ექნებათ მოცემული ტიპი.

1. მთელრიცხვა რიცხვითი კლასი -2

ეს კლასი შეიცავს ტიპებს, რომლებიც Num რიცხვითი კლასის ეგზემპლარებია, მაგრამ, გარდა ამისა, მათი მნიშვნელობები წარმოადგენს მთელი რიცხვებს, რის გამოც მხარდაჭერილია მთელრიცხვა გაყოფისა და მთელრიცხვა ნაშთის მეთოდები:  
  
div :: a->a->a  
  
mod::a->a->a  
  
პრაქტიკაში ეს ორი მეთოდი ხშირად იწერება საკუთარ არგუმენტთა შორის, რისთვისაც. მათი სახელები თავსდება ცალმაგ შექცეულ ბრჭყალებში. Int და Integer doრითადი ტიპები Integral კლასის ეგზემპლარებია. მაგალითად:  
  
> 7 div 2  
  
3  
  
>7'mod 2

1