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=13

1. მოიყვანეთ არატრივიალური გამოსახულებების მაგალითები, რომლებიც ეკუთვნის ტიპებს: a)((Char,Integer), String, [Double])
2. [(([Double],[Bool]),[Integer])]
3. (Double,Bool,(String,Integer))

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

1. როგორია შემდეგი მნიშვნელობების ტიპები:

([False,True],[’0’,’1’])

[’a’,’b’,’c’]

(“1”,”2”,”3”)

1. განსაზღვრეთ შემდეგი ფუნქციების მხოლოდ ტიპები პოლიმორფიზმის გამოყენებით:
2. pair x y = (x,y)
3. fun1 xs=(sum xs)\*(product xs)

c)length

d)[sum, product]

თითოეული ფუნქცია გამოიძახეთ კონკრეტული მონაცემებისთვის.

1. მოცემულია ფუნქციის ტიპები:
2. fun :: Num a => a -> a
3. fun :: Ord a => (a, a) -> Bool

თითოეული ტიპისთვის მოიყვანეთ მაგალითი ფუნქციებისა, რომლებსაც ექნებათ მოცემული ტიპი.

1. მთელრიცხვა რიცხვითი კლასი

**1.** a) (('k',5),"abb",[2.3,2.5,5.7])

b) [(([2.3,2.4],[True,False]),[4,5,6]),(([7.5,2.1],[True,False]),[6,5,9])]

c) (2.4,True,("kdh",6))

**2.** a) ([Bool], [Char])

b) [Char]

c) [[Char]]

**5.** მთელ რიცხვთა კლასი(Integral) წარმოადგენს კლასს, რომელიც შეიცავს ტიპებს. ეს ტიპები კი თავისთავად არის Num რიცხვითი კლასის ეგზემპლიარები. ასევე მათი მნიშვნელობები არის მთელი რიცხვები, ამიტომაც ამ კლასს აქვს მთელრიცხვა გაყოფის( div ) და მთელრიცხვა ნაშთის( mod ) მეთოდების მხარდაჭერა.

მთელრიცხვა გაყოფის მეთოდი div :: a->a-> a

მთელრიცხვა ნაშთის მეთოდი mod :: a->a->a

პრაქტიკულად თუ გამვიყენებთ ამ მეთოდებს, ისინი უნდა ჩავწეროთ თავიანთ არგუმენტებს შორის, ხოლო თვითონ მეთოდის სახელი ცალმაგ შექცეულ ბრჭყალში:

მაგ: 10 `div` 3 პასუხია 3

მაგ: 11 `mod` 2 პასუხია 1

მთელ რიცხვთა კლასის ეგზემპლიარებია int და integer ძირითადი ტიპები.

**3.** a) t -> t1 -> (t, t1) მაგალითად: pair 5 6 დააბრუნებს წყვილს (5,6)

b) Num a => [a] -> a გამოძახება? -0.5

c) [a] -> Int მაგალითად: length [3,4,5] დააბრუნებს 3-ს

d) Num a => [[a] -> a] a გამოძახება? -0.5

**4.** a) fun x = x + 2

b) -1