საგნის დასახელება: ფუნქციონალური პროგრამირება

სპეციალობა: კომპიუტერული მეცნიერება

ლექტორი: ნათელ არჩვაძე

დრო: 2 სთ

ქულათა ჯამი: 25=4+5+3+1+3

ქულათა გადანაწილების ინსტრუქცია: 5 საკითხი, თითოეული 5 ქულა.

ბილეთი=16

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

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

Else if x < (-5) && x/=(-8) then 3\*x^2-5/(2(x+8))+10

Else (7\*x+1/(1+x))/(x\*2-1)+1/(2\*x)

Y2x|(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

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

[1 of1] compiling main

Ok,modules loaded: main,

\*main>y1

(ord a, Fractional a) =>a -> a სინტ. შეცდ -1

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

* ფუნქცია, რომელიც აბრუნებს გაორმაგებულ არგუმენტს.
* პასუხი ( \ x->2 \* x)

\*Test> ( \ x ->2 \* x) 6

12

ტიპი : Num a = > a -> a

* ფუნქცია, რომელსაც გადაეცემა ორი არგუმენტი და ითვლის მთელ ხარისხს.
* პასუხი ( \ x y->x^y)

\*Test> ( \ x y -> x^y ) 2 3

8

ტიპი : (Num a , integral – b ) => a -> b -> a

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

პასუხი ( \ (a , b ) ( c , d ) -> ( a + c , b + d ) )

\*Test> ( \ (a , b ) ( c , d ) -> ( a + c , b + d ) ) (1 , 2 ) ( 3 , 4 )

(4 , 6)

ტიპი : ( Num t1 , Num t ) => (t , t1 ) -> (t, t1 ) – ( t, t1)

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.

3. let funct a b = if a == b then True else False -1

funct :: Eq a => a -> a -> Bool

function \_ \_ = True -1

function False True = True

function True False = False

function :: Bool -> Bool -> Bool +

1. სიის კონსტრუქტორის გამოყენებით შეადგინეთ:

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

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

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

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

* ფუნქცია, რომელიც მოცემულ წყვილში ადგილებს უცვლის ელემენტებს.

**Prelude>** let f = \ (a,b) -> (b,a) +2

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

<interactive>:1:20: ?

The function `(b, a)' is applied to one argument,

but its type `(t1, t0)' has none

In the expression: (b, a) f :: (t1, t) -> (t, t1)

In the expression: \ (a, b) -> (b, a) f :: (t1, t) -> (t, t1)

In an equation for `f':

f = \ (a, b) -> (b, a) f :: (t1, t) -> (t, t1)

**Prelude>** f(11,21)

(21,11)

**Prelude>** it:: (integer.integer)

<interactive>:1:14: ?

Illegal symbol '.' in type

Perhaps you intended -XRankNTypes or similar flag

to enable explicit-forall syntax: forall <tvs>. <type>

* **Prelude>**
* 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 a b c = a+b>c && b+c>a აკლია პირობა +1

\*Test> istriangle 3 4 5

True

It : : bool

\*Test> istriangle 3 4 9

False

It : : bool

\*Test> ( \ a b c -> a+b>c && a+c>b && b+c>a ) 4 5 6

true