ამოცანა: ვირტუალური ბიბლიოთეკა **Library**

ბიბლიოთეკაში ინახება წიგნები, გაზეთები და ჟურნალები. წიგნი ხასიათდება ავტორის გვარით და დასახელებით, ჟურნალი–დასახელებით, გამოშვების თვე და წელით, გაზეთი–დასახელებით და გამოშვების თარიღით. მონაცემთა ბაზა წარმოადგენს ამ ობიექტების სიას. შექმენით მონაცემთა ტიპი, რომელიც წარმოადგენს ბიბლიოთეკაში შესანახ ობიექტებს. განსაზღვრეთ შემდეგი ფუნქციები:

1) isPeriodic, ამოწმებს, რომ მისი არგუმენტი არის პერიოდული გამოცემა;

2) getByTytle, შენახული ობიექტებიდან (მონაცემთა ბაზიდან) იღებს ობიექტებს მოცემული დასახელებით;

3) getByMonth, იღებს მონაცემთა ბაზიდან პერიოდულ გამოცემებს, რომლებიც გამოდის თვეში ერთხელ და მითითებულ წელს (შევნიშნოთ, რომ გაზეთები გამოდის თვეში რამდენჯერმე);

4) getByMonths, ისევე მუშაობს, როგორც წინა ფუნქცია, მხოლოდ არგუმენტად ღებულობს თვეების სიას;

5) getAuthors, აბრუნებს ავტორების სიას იმ გამოცემებისა, რომლებიც ბიბლიოთეკაში ინახება.

ამოხსნა

**monacemTa Library tipi \_ virtualuri biblioTeka**

data Library = Book String String | Newspaper String Int Int Int | Journal String Int Int deriving (Show, Eq)

––ფუნქცია 1: ამოწმებს, არის თუ არა მისი არგუმენტი პერიოდული გამოცემა

--isPeriodic isPeriodic :: Library -> Bool

isPeriodic (Book \_ \_) = False

isPeriodic (Newspaper \_ \_ \_ \_) = True

isPeriodic (Journal \_ \_ \_) = True

––მიმართვა: isPeriodic (Book "Alex" "Pascal")

––ფუნქცია 2: იღებს ობიექტებს მოცემული დასახელებით

--getByTytle

getByTytle :: [Library] -> String -> [Library]

getByTytle [] \_ = []

getByTytle ((Book a x):xs) y = if x==y then (Book a x):getByTytle xs y else getByTytle xs y

getByTytle ((Newspaper x a b c):xs) y = if x==y then (Newspaper x a b c):getByTytle xs y else getByTytle xs y

getByTytle ((Journal x a b):xs) y = if x==y then (Journal x a b):getByTytle xs y else getByTytle xs y

––მიმართვა: getByTytle [(Book "Alex" "Pascal"),(Newspaper "Prog" 10 05 05),(Journal "Prog" 11 12)] "Prog"

––ფუნქცია 3: იღებს იმ პერიოდულ გამოცემებს, რომლებიც გამოდის მითითებულ წელს თვეში ერთხელ

--getByMonth (getbyMonth [(Book "Alex" "Pascal"),(Newspaper "Prog" 10 05 05),(Journal "Prog" 11 12)] 11 12)

getByMonth :: [Library] -> Int -> Int -> [Library]

getByMonth [] \_ \_ = []

getByMonth ((Book \_ \_):xs) x1 y1 = getByMonth xs x1 y1

getByMonth ((Newspaper a b x y):xs) x1 y1 = if x1==x && y1==y then (Newspaper a b x y):getByMonth xs x1 y1 else getByMonth xs x1 y1

getByMonth ((Journal a x y):xs) x1 y1 = if x==x1 && y==y1 then (Journal a x y):getByMonth xs x1 y1 else getByMonth xs x1 y1

––ფუნქცია 4: იღებს იმ პერიოდულ გამოცემებს, რომლებიც გამოდის მითითებულ წელს მითითებულ თვეებში (გადაეცემა თვეების სია)

--getByMonths (getbyMonths [(Book "Alex" "Pascal"),(Newspaper "Prog" 10 05 05),(Journal "Prog" 11 12)] [11,05] 12)

getByMonths :: [Library] -> [Int] -> Int -> [Library]

getByMonths \_ [] \_ = []

getByMonths a (x:xs) b = getByMonth a x b++getByMonths a xs b

––ამოცანა 5: აბრუნებს ავტორების სიას

--getAuthors (getAuthors [(Book "Alex" "Pascal"),(Newspaper "Prog" 10 05 05),(Journal "Prog" 11 12)])

getAuthors :: [Library] -> [String]

findEq :: [String] -> String -> [String]

findEq [] x = [x]

findEq (x:xs) y = if x/=y then x:findEq xs y else x:xs

getAuthors [] = []

getAuthors ((Book x \_):xs) = findEq (getAuthors xs) x

getAuthors ((Newspaper x \_ \_ \_):xs) = findEq (getAuthors xs) x

getAuthors ((Journal x \_ \_):xs) = findEq (getAuthors xs) x