МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ ИМЕНИ АКАДЕМИКА М.Ф. РЕШЕТНЕВА”

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**Отчёт по Лабораторной работе № 3**По дисциплине “Функциональное программирование”

**Вариант № 12**

**Выполнил:** студент 1 курса  
Группы: БПИ22-02  
Трифонов Кирилл Вячеславович

**Преподаватель:**   
доцент каф. ИВТ, к.т.н

Богданов Константин Валериевич

**Красноярск 2023**

Задания на лабораторную работу 1.

*1. Определите следующие функции с использованием функций высшего порядка:*

1.1 Функция вычисления геометрического среднего элементов списка вещественных чисел   
с использованием функции foldr . Функция должна осуществлять только один проход по   
списку.

*--1.1--*

avGList :: [*Double*] -> *Double*

avGList x = foldr (\*) 1 x \*\* 1/len x

len [] = 0

len (x:xs)=1 + len xs

1.2 Функция, вычисляющая скалярное произведение двух списков ( используйте функции   
foldr и zipWith ).

*--1.2--*

scalar a b = foldr (+) 0 (zipWith (\*) a b)

1.3 Функция countNegat , возвращающая количество отрицательных элементов в списке.

*--1.3--*

countNegat x = len (getNegative x)

1.4 Функция quicksort , осуществляющая быструю сортировку списка по следующему ре-  
курсивному алгоритму. Для того, чтобы отсортировать список xs, из него выбирается   
первый элемент ( обозначим его x). Остальной список делится на две части: список, со-  
стоящий из элементов xs, меньших x и список элементов, больших x . Эти списки сор-  
тируются (здесь проявляется рекурсия, поскольку они сортируются этим же алгоритмов),   
а затем из них составляется результирующий список вида as ++ [x] ++ bs, где as и bs –  
отсортированные списки меньших и больших элементов соответственно.

*--1.4--*

quicksort [] = []

quicksort (x:xs) = quicksort(filter (<x) xs) ++ [x] ++ quicksort (filter (>=x) xs)

1.5 Определенная в предыдущем пункте функция quicksort сортирует список в порядке воз-  
растания. Обобщите ее: пусть она принимает еще один– функцию сравнения типа   
a -> a-> Bool и сортирует список в соответствие с нею.

*--1.5--*

bSort :: (a -> a -> *Bool*) -> [a] -> [a]

bSort a [] = []

bSort a (x:xs) = bSort a (filter (a x) xs) ++ [x] ++ bSort a (filter (not . a x) xs)

*2. Вернитесь к заданиям из лабораторной работы No 2 (вторая часть заданий) и реализуйте их с помощью функций высшего порядка. Постарайтесь полностью исключить из определений   
функций явный проход по списку.*

*-- Вторая часть Задание 12*

*-- 2.12.Функция, которая меняет знак всех положительных элементов списка чисел,*

*-- например: по [-1, 0, 5, -10, -20] дает [-1,0,-5,-10,-20]*

negative :: [*Int*] -> [*Int*]

negative= map (\n -> (abs n)\*(-1))

Функция negative, с помощью *map* применяет к каждому элементу массива операцию взятия модуля, затем отрицания.

*-- Вторая часть Задание 4*

*-- 2.4.Функция вычленения n -го элемента из заданного списка.*

vichlenit' x n = map snd(filter (\(a,b)-> a /=n) (zip [0..] x))

Функция vichlenit’, используя *zip* добавляет порядковый номер (для списка (1,2) будет (0,1),(1,2)), затем производится сравнение по первому элементу и выводится второй.