## Задание. Добавить в свою программы задания t-5 возможности ввода исходных данных из файлов, а также сохранение результата в файл:

-- ЗАДАНИЕ 2. Функции для работы с типом List. Для введенного ранее типа List определите следующие функции:

-- 3) removeNegative, которая из списка целых (тип List Integer) удаляет отрицательные элементы.

data List a = Nil | Cons a (List a) deriving (Show, Read)

headList (Cons x \_) = x

headList Nil = error "headList: empty list"

tailList (Cons \_ y) = y

tailList Nil = error "tailList: empty list"

removeNegative Nil = Nil

removeNegative (Cons x y) | (x < 0) = tailList (Cons x y)

| otherwise = (Cons x (removeNegative y))

-- ЗАДАНИЕ 8. Иерархия должностей в некоторой организации образует древовидную структуру.

-- Каждый работник, однозначно характеризующийся уникальным именем, имеет несколько подчиненных.

-- Определите тип данных, представляющий такую иерархию и опишите следующие функции:

-- 1) getSubordinate, возвращающую список подчиненных указанного работника.

data Person = PersonName (String) [Person] deriving (Eq, Show, Read)

getName (PersonName n \_) = n

getNames = map getName

getSubordinatePersons (PersonName \_ p) = p

finder ((PersonName n p) :xs) name | (n == name) = getSubordinatePersons (PersonName n p)

| otherwise = finder xs name

getSubordinate (PersonName n p) name = getNames (finder [(PersonName n p)] name)

--Сохранение структуры в файл

save :: Person -> FilePath -> IO ()

save editor f = writeFile f $ show editor

main = do

-- ЗАДАНИЕ 2

-- Читаем файл

buf <- readFile "list.txt"

let array = read buf :: List

-- Получение иерархии от getAllSubordinate

let result = removeNegative array

-- Запись результата в переменную

let formatedResult = show result

-- Вывод в файл

writeFile "result1.txt" formatedResult

-- ЗАДАНИЕ 8

-- Читаем файл

buf <- readFile "persons.txt"

let tree = read buf :: Person

-- Получение иерархии от getAllSubordinate

let result = getSubordinate tree "a"

-- Запись результата в переменную

let formatedResult = show result

-- Вывод в файл

writeFile "result2.txt" formatedResult

Ввод:

*list.txt:* (Cons 1 (Cons (-3) (Cons 4 Nil)))

*persons.txt:* (PersonName "a"[PersonName "b" [PersonName "c" [], PersonName "d" [], PersonName "i" []], PersonName "f" []])

Вывод:

ghci> main

*result1.txt:* Cons 1 (Cons 4 Nil)

*result2.txt:* ["b","f"]