Министерство науки и высшего образования Российской Федерации

Новосибирский Государственный технический университет

Кафедра автоматизированных систем управления



**Отчет по лабораторной работе 2**

**по дисциплине «Функциональное программирование»**

**Вариант 3**

Выполнил

студент группы АВТ-812:

Березин Дмитрий

Преподаватель:

Достовалов Дмитрий Николаевич,

к.т.н., доцент кафедры АСУ

г. Новосибирск

2021 г.

Содержание

[1 Цель работы 3](#_Toc83996303)

[2 Текст задания 3](#_Toc83996304)

[3 Текст программы 3](#_Toc83996305)

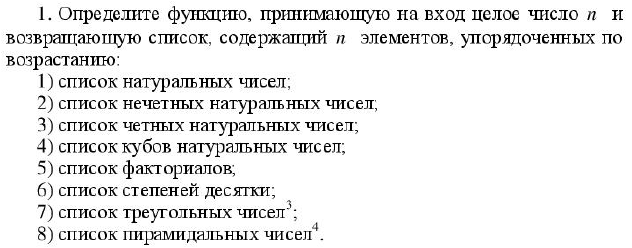
[4 Результаты тестов 5](#_Toc83996306)

[5 Вывод 6](#_Toc83996307)

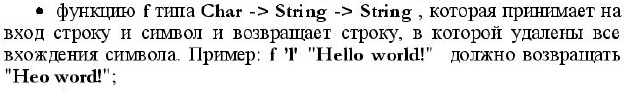
# 1 Цель работы

Научиться определять рекурсивные функции. Получить представление о механизме сопоставления с образцом. Приобрести навыки определения функций для обработки списков.

# 2 Текст задания







# 3 Текст программы

Main.hs

module Lab2 where

{--Natural numbers--}

list1 :: Integer -> [Integer]

list1 0 = [ ]

list1 x = list1 (x-1)++(x:[])

{--Odd numbers --}

list2 :: Integer -> [Integer]

list2 0 = [ ]

list2 x = list2 (x-1)++(2\*x-1:[])

{--Even numbers --}

list3 :: Integer -> [Integer]

list3 0 = [ ]

list3 x = list3 (x-1)++(2\*x:[])

{--Natural numbers in the 3rd degree --}

list4 :: Integer -> [Integer]

list4 0 = [ ]

list4 x = list4 (x-1)++((x\*x\*x):[])

{--Factorial --}

list5 :: Integer -> [Integer]

factorial 0 = 1

factorial x = x\*factorial(x-1)

list5 0 = [ ]

list5 x = list5 (x-1)++[factorial(x)]

{--Degrees 10 --}

list6 :: Integer -> [Integer]

degree10 1 = 1

degree10 x = 10\*degree10(x-1)

list6 0 = [ ]

list6 x = list6 (x-1)++[degree10(x+1)]

{--Triangle numbers --}

list7 :: Integer -> [Integer]

triangle 1 = 1

triangle(x) = x+triangle(x-1)

list7 0 = [ ]

list7 x = list7 (x-1)++[triangle(x)]

{--Pyramid numbers --}

list8 :: Integer -> [Integer]

pyramid 1 = 1

pyramid(x) = x+pyramid(x-1)

p 1 = 1

p x = pyramid(x) + p(x-1)

list8 0 = [ ]

list8 x = list8 (x-1)++[p(x)]

{--Delete characters from string --}

delete :: Char -> String -> String

delete c [] = []

delete c (x:xs) = if x == c then delete c xs else x:delete c xs

Здесь к каждой функции перед ее определением приведен краткий комментарий, описывающий выполняемую ею операцию.

# 4 Результаты тестов

Тестирование первой части задания:

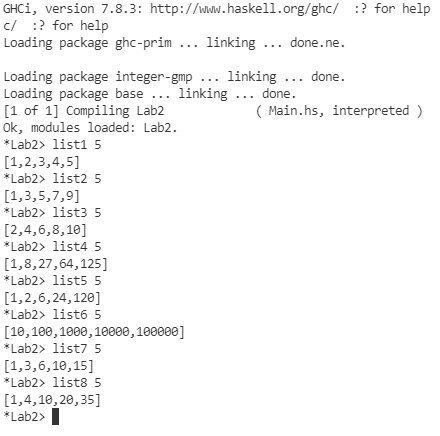


Рисунок 1 – тесты для первой части задания

На рисунке 1 показаны функции, возвращающие списки чисел, удовлетворяющих заданным условиям. Номер в конце функции соответствует порядковому номеру задания из списка в первой части.

Тестирование второй части задания:

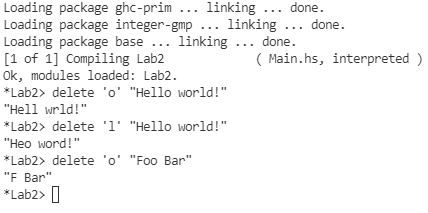


Рисунок 2 – тестирование второй части задания

На втором рисунке представлены тесты для функции, которая удаляет заданный символ из заданной строки.

# 5 Вывод

В ходе выполнения лабораторной работы я научился создавать и использовать рекурсивные функции на языке Haskell. Также мною была реализована и протестирована функция для обработки списков. Я получил представление о механизме сопоставления с образцом.