МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

Севастопольский государственный университет

кафедра Информационных систем

Институт информационных технологий и управления в технических системах

курс 1 группа ИC/б-11-о

Лисянский Александр Игоревич

09.03.02 Информационные системы (уровень магистра)

**ОТЧЁТ**

о лабораторном практикуме №1

«Исследование способов реализации рекурсивных алгоритмов в

функциональном программировании»

Отметка о зачёте \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

Ст.преподаватель Строганов В.А.

(должность) (подпись) (инициалы, фамилия)

Севастополь

2016

Цель работы

Исследовать различные способы реализации рекурсивных алгоритмов в функциональном программировании. Реализовать и исследовать рекурсивные функции обработки списков на языке Scala. Научится работать в среде разработки IntelliJ IDEA Community Edition и создавать проекты на Scala.

Вариант задания

Функция возвращает новый список, в котором каждый элемент является список из чисел от 0 до модуля соответствующего элемента входного списка. Пример: List(2, -6, 10) преобразуется в List(List(0, 1, 2), List(0, 1, 2, 3, 4, 5, 6), List(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10))

Ход работы

В ходе выполнения лабораторной работы в соответствии с вариантом задания разработана программа на языке Scala с прямой и хвостовой рекурсии.

Листинг программы

package main.scala

/\*\*

\* Created by user on 18.09.2016.

\*/

object Application {

def createAbsList(in: Int): List[List[Int]] = {

List(List.range(0, Math.abs(in) + 1))

}

def moduleList(lst: List[Int]): List[Any] = {

def moduleListAcc(\_\_list: List[Int], acc: List[Any]): List[Any] = {

if (\_\_list != Nil) {

moduleListAcc(\_\_list.tail,acc:::createAbsList(\_\_list.head))

}

else

acc

}

moduleListAcc(lst, Nil)

}

def moduleListRec(lst: List[Int]): List[Any] = {

if (lst == Nil)

Nil

else

createAbsList(lst.head) ::: moduleListRec(lst.tail)

}

def main(args: Array[String]): Unit = {

var list = List.range(0,6500,2);

println(moduleList(list))

println(moduleListRec(list))

}

}

Выводы

В ходе выполнения данной лабораторной работы была разработана программа на языке Scala в соответствии с вариантом задания. В программе были применены 2 рекурсивных подхода: прямая рекурсия и хвостовая рекурсия. В соответствии с заданием было определено граничное значение количества входных элементов, при котором стек вызовов рекурсии не переполняется.