МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

Севастопольский государственный университет

кафедра Информационных систем

**Лисянский Александр Игоревич**

Институт информационных технологий и управления в технических системах

курс 1 группа ИC/м-11(о)

09.04.02 Информационные системы и технологии

Лабораторная работа №5

по дисциплине «Математические основы и парадигмы современных языков программирования»

«Исследование особенностей реализации операций каррирование, частичного применения функций и замыканий в языке Scala»

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

(дата)

Руководитель практикума

Строганов В.А.

(должность) (подпись) (инициалы, фамилия)

Севастополь 2016

1. **Цель работы**

Исследовать понятия каррирования, частичного применения и замыкания в языке Scala и получить практические навыки их применения.

1. **Постановка задачи**

Порядок выполнения работы

1. Реализовать на языке Scala функцию, возвращающую количество собственных вызовов.

2. Функция перевода десятичного числа в n-ичное представление.

Пример вызова: десятичноеЧислоВNичное(число, порядок\_числа).

3. Реализовать функцию, которая принимает один параметр – начальное число Z - и возвращает функцию, которая принимает один параметр – длину списка – и возвращает список заданной длины, содержащий случайные числа. Модуль разности случайных чисел и начального числа Z не должен превышать 5.

1. **Текст программы**

package main.scala  
  
import scala.util.Random  
  
/\*\*  
 \* Created by user on 13.11.2016.  
 \*/  
object lab5 {  
  
 def counterBuilder(): () => Int = {  
 var count: Int = 0  
 def callCounter(): Int = {  
 count += 1  
 count  
 }  
 callCounter  
 }  
  
 def десятичноеЧислоВNичное(c: Int, n: Int) : String = {  
 def numToStr(i: Int) : String = {  
 val s = i match {  
 case 10 => "A"  
 case 11 => "B"  
 case 12 => "C"  
 case 13 => "D"  
 case 14 => "E"  
 case 15 => "F"  
 case \_ => i+""  
 }  
 s  
 }  
 def numberToNumber2(num: Int, osn: Int, s: String) : String = {  
 var str = s  
 if(osn > num) {  
 str = numToStr(num) + str  
 str  
 } else {  
 str = numToStr(num % osn) + str  
 numberToNumber2(num/osn, osn, str)  
 }  
 }  
 numberToNumber2(c, n, "")  
 }  
  
 def listBuilder(z: Int)(length: Int): List[Int] = {  
 val random: Random = Random  
 def listBuilder(length: Int, acc: List[Int]): List[Int] = {  
 if(length == 0) acc  
 else {  
 val element = z + (random.nextInt() % 6)  
 listBuilder(length - 1, element :: acc)  
 }  
 }  
 listBuilder(length, Nil)  
 }  
  
  
 def main(args: Array[String]): Unit = {  
 val callCounter = counterBuilder()  
 println(callCounter())  
 println(callCounter())  
 val func = десятичноеЧислоВNичное(\_: Int,2)  
 println(func(18))  
 val func1 = десятичноеЧислоВNичное(\_: Int,16)  
 println(func1(31))  
 val listBuilderWith1Medium = listBuilder(1)(\_)  
 println(listBuilderWith1Medium(10))  
  
 }  
}

1. **Пример выполнения программы**

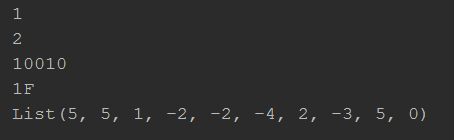


Рисунок 1 – Пример выполнения программы

**Вывод:**

В данной лабораторной работе были исследованы понятия каррирования, частичного применения и замыкания в языке Scala. Получены практические навыки их применения. Реализована функция, которая возвращает количество своих вызовов. При её написании использовались механизмы частичного применения функции и замыкания. Также реализована функция возведения в степень проводящая каррирование. Написана функция, возвращающая функцию, создающую список заданной длины, получающая параметром число, которое определяет границы членов списка так, что они не отличаются от него по модулю более чем на 5 единиц. Эта функция реализована при помощи механизмов каррирования, частичного применения функции и замыкания.