ФГАОУ ВО «Севастопольский государственный университет»

Институт информационных технологий   
и управления в технических системах

ОТЧЁТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

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

по дисциплине «Парадигмы современных языков программирования»

Выполнила:  
студентка группы ИС/м-21-1-з  
Ускова Екатерина Дмитриевна

Севастополь

2022

# ЦЕЛЬ РАБОТЫ

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

# Ход работы

Реализована функция, возвращающая количество собственных вызовов.

var count = 0  
def countFunc: Int = {  
 count += 1  
 count  
}

Вызов:

println(countFunc)  
println(countFunc)  
println(countFunc)  
println(countFunc)

Результат работы:

1

2

3

4

Реализована функция, используя механизм каррирования:

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

def decToN(n: Int)(x: Int): String = {  
 def numToLetter(num: Int): String = {  
 num match {  
 case 10 => "A"  
 case 11 => "B"  
 case 12 => "C"  
 case 13 => "D"  
 case 14 => "E"  
 case 15 => "F"  
 case \_ => num.toString  
 }  
 }  
 def toN(x: Int): String = {  
 if(x < n)  
 numToLetter(x)  
 else  
 toN(x / n) + numToLetter(x % n)  
 }  
 toN(x)  
}

Вызов:

println(decToN(2)(42))  
println(decToN(3)(42))  
println(decToN(4)(42))  
println(decToN(16)(42))

Результат работы:

101010

1120

222

2A

Используя каррирование, написана функция преобразования десятичного числа в двоичное.

def decToBin(x: Int): String = decToN(2)(x)

Вызов:

println(decToBin(42))

Результат работы:

101010

Реализована функция, 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 превышает 5.

def fun(start: Int): Int => List[Int] = {  
 def f(len: Int): List[Int] = {  
 var lst: List[Int] = Nil  
 for (\_ <- 1 to len) {  
 lst = start + scala.util.Random.nextInt(10) - 5 :: lst  
 }  
 lst  
 }  
 f  
}

Вызов:

scala.util.Random.setSeed(42)  
  
val funStart20 = fun(20)  
println(funStart20(10))

Результат работы:

List(18, 24, 23, 20, 20, 15, 19, 23, 18, 15)

# Вывод

Выполняя эту работу, я исследовала понятия каррирования, частичного применения и замыкания в языке Scala и получила практические навыки их применения.

# Приложение А

Листинг программного кода

package main.scala  
  
object Application {  
  
 var count = 0  
 def countFunc: Int = {  
 count += 1  
 count  
 }  
  
 def decToN(n: Int)(x: Int): String = {  
 def numToLetter(num: Int): String = {  
 num match {  
 case 10 => "A"  
 case 11 => "B"  
 case 12 => "C"  
 case 13 => "D"  
 case 14 => "E"  
 case 15 => "F"  
 case \_ => num.toString  
 }  
 }  
 def toN(x: Int): String = {  
 if(x < n)  
 numToLetter(x)  
 else  
 toN(x / n) + numToLetter(x % n)  
 }  
 toN(x)  
 }  
  
 def decToBin(x: Int): String = decToN(2)(x)  
  
 def fun(start: Int): Int => List[Int] = {  
 def f(len: Int): List[Int] = {  
 var lst: List[Int] = Nil  
 for (\_ <- 1 to len) {  
 lst = start + scala.util.Random.nextInt(10) - 5 :: lst  
 }  
 lst  
 }  
 f  
 }  
  
  
  
 def main (args: Array[String]): Unit = {  
  
// println(countFunc)  
// println(countFunc)  
// println(countFunc)  
// println(countFunc)  
  
// println(decToN(2)(42))  
// println(decToN(3)(42))  
// println(decToN(4)(42))  
// println(decToN(16)(42))  
//  
// println(decToBin(42))  
  
 scala.util.Random.setSeed(42)  
  
 val funStart20 = fun(20)  
 println(funStart20(10))  
 }  
}