**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической Кибернетики и Информационных Технологий

****

**Отчет по лабораторной работе**

по предмету «Функциональное программирование»

на тему:

«Язык программирования «Scala»»

Выполнил: студент группы

БВТ1802

Ткачев Александр Владимирович

Руководитель:

Мосева Марина Сергеевна

Москва 2020

Вариант 10

1. Посчитать количество использований цифры для записи всех чисел от 1 до n
2. Записать выбранное количество случайных элементов массива в новый массив

Каждую задачу решить в функциональном и императивном стиле.

package Lab9Bin

import scala.collection.mutable.ListBuffer

import scala.util.Random;

import scala.util.control.Breaks.\_

object functions {

  val FirstFunctional = (number: Int, digit: Int) =>

    (for (i <- 0 to number) yield i.toString.count(\_ == digit.toString.charAt(0))).sum

  val SecondFunctional = (list: List[Any], count: Int) =>

    Random.shuffle(list).take(count)

  def FirstImperative(number: Int, digit: Int): Int = {

    var count = 0

    for (i <- 1 to number) {

      count += i.toString.count(\_ == digit.toString.charAt(0))

    }

    return count

  }

  def SecondImperative(list: List[Any], count: Int): ListBuffer[Any] = {

    var result = ListBuffer[Any]()

    for (i <- 1 to count) {

      result += list(Random.nextInt(list.length))

    }

    return result

  }

}

object program extends App {

  val intList = List.range(1, 100)

  val stringList = List("this", "is", "a", "string", "list", "for", "tests")

  val floatList = List(0.1, 2.923, 0.2382, 100.231, 172.0, 231.5, 1000.02)

  val charList = List('h', 'e', 'l', 'l', 'o', 'w', 'o', 'r', 'l', 'd', '!');

  var boolList = List(true, false, false, true, false, true)

  breakable {

    while (true) {

      println("Введите 1 для работы со списком целых чисел")

      println("Введите 2 для работы со списком строк")

      println("Введите 3 для работы со списком дробных чисел")

      println("Введите 4 для работы со списком символов")

      println("Введите 5 для работы со списком логических значений")

      println("Введите 6 для выхода из программы")

      var input = scala.io.StdIn.readInt

      println()

      var list = List[Any]()

      input match {

        case 1 => list = intList

        case 2 => list = stringList

        case 3 => list = floatList

        case 4 => list = charList

        case 5 => list = boolList

        case 6 => break

        case other => println("Некорректный ввод")

      }

      // 1. Посчитать количество импользований цифры для записи всех чисел от 1 до n

      println(functions.FirstFunctional(11, 1))

      println(functions.FirstImperative(11, 1))

      // 2. Записать выбранное количество случайных элементов массива в новый массив

      println(functions.SecondFunctional(list, 5))

      println(functions.SecondImperative(list, 5))

      println("\n")

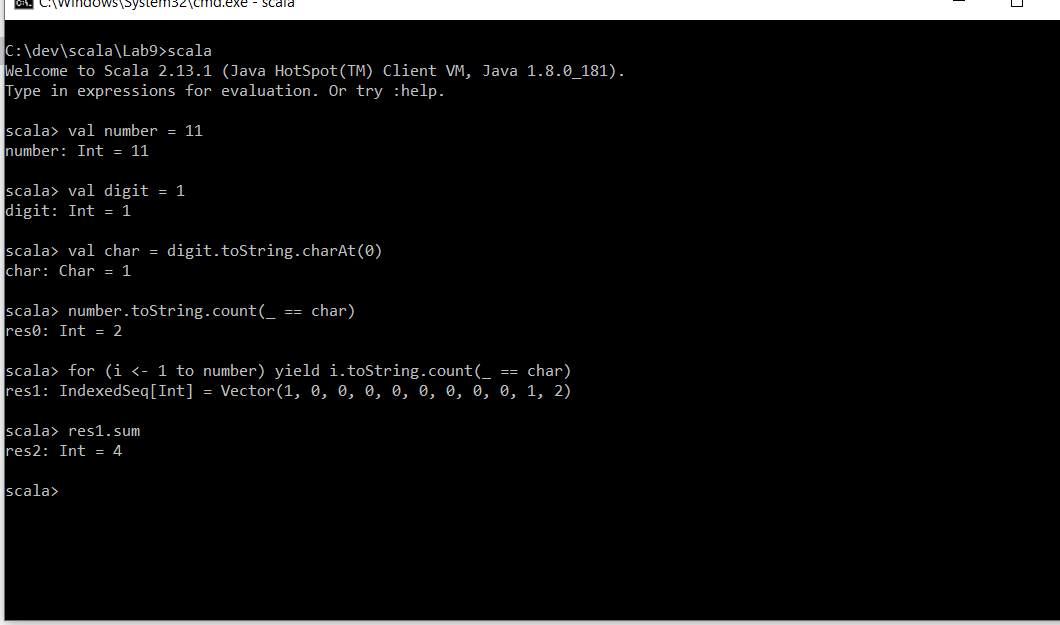
    }

  }

}

Анализ функциональных решений в REPL:

Задача 1: посчитать количество использований цифры в записи всех чисел от 1 до n



val number = 11 – верхняя граница диапазона

val digit = 1 – цифра, вхождения которой нужно посчитать

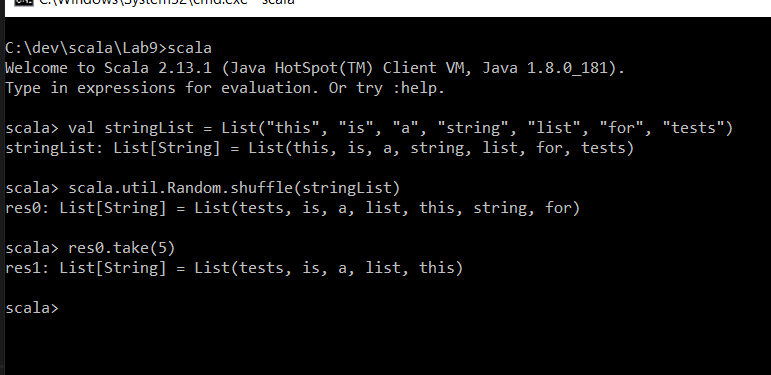
val char = digit.toString.charAt(0) – перевод цифры в ascii-символ

number.toString.count(\_ == char) – перевод числа в строку и подсчет вхождений символа char

for ( i <- 1 to number) yield i.toString.count( \_ == char) – for-генератор (возвращает коллекцию) – возвращает число вхождений цифры в каждом числе от 1 до number

res1.sum – сложение всех элементов контейнера, который вернул for-генератор

Задача 2: сформировать массив из n случайных элементов переданного массива



stringList – список строк, из которого будем брать случайные элементы

scala.util.Random.shuffle(stringList) – возвращает список с элементами в другом порядке

res0.take(5) – возвращает список из 5 элементов перемешанного старого списка

Результаты:

