



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника
МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/12 Интеллектуальный анализ больших
данных в системах поддержки принятия решений

О Т Ч Е Т
по лабораторной работе № 2

Название: Арифметические операции

Дисциплина: Языки программирования для работы с большими данными

Студент ИУ6-23М
(Группа)

М.А. Гейне
(Подпись, дата) (И.О. Фамилия)

Преподаватель

П.В. Степанов
(Подпись, дата) (И.О. Фамилия)

Москва, 2023

ЗАДАНИЕ

В приведенных ниже заданиях необходимо вывести внизу фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания. Для получения последней даты и времени следует использовать класс Date.

Вариант 1.

4. Ввести n слов с консоли. Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.

5. Ввести n слов с консоли. Найти количество слов, содержащих только символы латинского алфавита, а среди них – количество слов с равным числом гласных и согласных букв.

Вариант 2.

Ввести с консоли n – размерность матрицы $a[n][n]$. Задать значения элементов матрицы в интервале значений от $-n$ до n с помощью датчика случайных чисел.

4. Найти сумму элементов матрицы, расположенных между первым и вторым положительными элементами каждой строки.

5. Транспонировать квадратную матрицу.

Создание проекта

Для ЛР2 был создан проект, аналогичный проекту ЛР1. Структура проекта приведена в листинге 1.

Листинг 1 – Структура проекта

```
Lab2/
├── src
│   ├── main
│   │   └── scala
│   │       ├── main.scala
│   │       ├── var1
│   │       │   ├── task1.scala
│   │       │   └── task2.scala
│   │       └── var2
│   │           ├── task1.scala
│   │           └── task2.scala
└── target
    ├── scala-3.2.2
    │   └── *output omitted*
    └── streams
        └── *output omitted*
```

183 directories, 94 files

Текст `main.scala` аналогичен программе верхнего уровня ЛР1, но скорректирован под задания ЛР2. В частности, был исключен разбор массива целых чисел, подаваемый в качестве аргумента командной строки. Исходный текст программы верхнего уровня приведён в листинге 2.

Листинг 2 – Текст главной программы

```
import java.util.Date

@main def main(args: String* ): Unit =
  val arglist = args.toList
  arglist match {
    case "-p"::tasks::tail => run_tasks(tasks)
    case "--help"::tail => print_help()
    case tail if tail.isEmpty => run_tasks("1234")
    case tail => illegal_args(tail.mkString(" "))
  }
  println(s"""|Developer: mikeGEINE
|Task recieved on: Fri Feb 17 15:39:00 MSK 2023
|Task completed (this run) on: ${new Date()}|""".stripMargin)

def run_tasks(tasks: String) =
  tasks.split("").foreach { key =>
    key match {
      case "1" =>{
        println("Variant 1")
        println("Question 4")
        var1.task1.main()
      }
      case "2" => {
        println("Variant 1")
        println("Question 5")
        var1.task2.main()
      }
      case "3" => {
        println("Variant 2")
        println("Question 4")
        var2.task1.main()
      }
      case "4" => {
        println("Variant 2")
        println("Question 5")
        var2.task2.main()
      }
      case arg => illegal_args("Task %s "+arg)
    }
    println("-----")
  }
```

```

def print_help() =
  println("""
  lab1 [OPTIONS]
  [OPTIONS]
  --help      Prints help message
  -p TASKS    Allows partial execution of tasks. TASK is a sting of digits
from 1 to 4.
  """)

def illegal_args(arg: Any) =
  println("Illegal arguments found: "+arg)
  println("Usage:")
  print_help()

```

Далее приведены решения задач ЛР2.

Задача 1

Текст задания: ввести n слов с консоли. Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.

Было решено, что слова вводятся одной строкой, отделённые друг от друга пробелами. Затем создаётся массив количеств различных символов, в нём находится позиция минимального элемента, а затем по этой позиции выбирается слово из оригинального списка. Код решения приведён в листинге 3.

Листинг 3 – Код решения задачи 1

```

package var1.task1
import bmstu.general._
import scala.io.StdIn

def main() =
  println("Put down your words separated by spaces:")
  val str = StdIn.readLine()
  val words = str.split(" ")
  val lengths = words.map(_._distinct.length()) // finds how many distinct
letters in each word
  // words.apply(lengths.indexOf(lengths.min))
  print("Found word: ")
  lengths.min |> (lengths.indexOf(_)) |> words.apply |> (println(_)) // get
minimum of distinct letters,
                                // get a corresponding position (first of), and retrieve the
word by index from original array

```

Результат работы программы приведён на рисунке 1.

```
BMSTU_BigData main
→ java -jar exec/lab1.jar -p 3
Variant 2
Question 4
Put down your words separated by spaces:
dog vs cat is not good
Found word: vs
-----
Developer: mikeGEINE
Task recieved on: Fri Feb 17 15:39:00 MSK 2023
Task completed (this run) on: Sun Feb 26 17:21:59 MSK 2023
```

Рисунок 1 – Решение задачи 1

Задача 2

Текст задания: ввести n слов с консоли. Найти количество слов, содержащих только символы латинского алфавита, а среди них – количество слов с равным числом гласных и согласных букв.

Ввод слов осуществляется в том же формате, что и в решении задачи 1. Для фильтрации слов используется регулярное выражение, которое выбирает буквы, не принадлежащие латинскому алфавиту: если в слове есть хоть один символ, удовлетворяющий этому выражению, то оно должно быть отброшено. Регулярные выражения используются и для подсчёта слов с равным числом гласных и согласных букв. В регулярном выражении перечислены все гласные буквы латинского алфавита; в слове ищутся все символы, удовлетворяющие регулярному выражению, подсчитывается их количество; функция проверки условия возвращает результат сравнения количества гласных букв с количеством букв в слове за вычетом гласных (т.е. согласных). Функция проверки условия равенства букв возвращает true или false и используется в функции фильтрации. Код решения приведён в листинге 4.

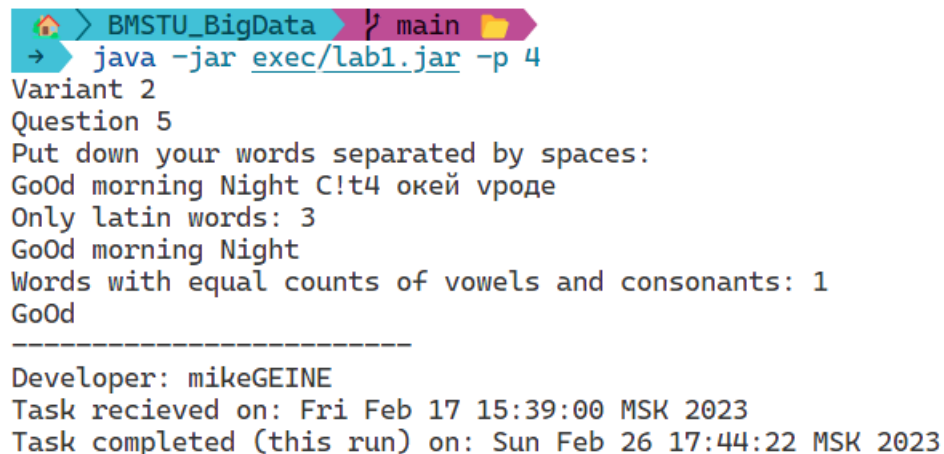
Листинг 4 – Код решения задачи 2

```
package var1.task2
import bmstu.general._
import scala.io.StdIn

def main() =
  println("Put down your words separated by spaces:")
  val str = StdIn.readLine()
  val words = str.split(" ")
  val latin_regex = ""[^a-zA-Z]"".r.unanchored
  val latin = words.filter(! latin_regex.matches(_)) // if there is no other
  chars then [a-zA-Z], then a word is all latin
  println(s"Only latin words: ${latin.length}")
  println(latin.mkString(" "))
  val latin_eq = latin.filter(vowel_check(_))
  println(s"Words with equal counts of vowels and consonants:
  ${latin_eq.length}")
  println(latin_eq.mkString(" "))

def vowel_check(word: String): Boolean =
  val vowels_regex = ""[eyuioaEYUIOA]"".r
  val vowels = vowels_regex.findAllIn(word).length
  vowels == (word.length - vowels)
```

Результат исполнения программы приведён на рисунке 2.



```
> BMSTU_BigData } main
→ java -jar exec/lab1.jar -p 4
Variant 2
Question 5
Put down your words separated by spaces:
GoOd morning Night C!t4 окей vpoде
Only latin words: 3
GoOd morning Night
Words with equal counts of vowels and consonants: 1
GoOd
-----
Developer: mikeGEINE
Task recieved on: Fri Feb 17 15:39:00 MSK 2023
Task completed (this run) on: Sun Feb 26 17:44:22 MSK 2023
```

Рисунок 2 – Решение задачи 2

Задача 3

Текст задания: ввести с консоли n – размерность матрицы $a[n][n]$. Задать значения элементов матрицы в интервале значений от $-n$ до n с помощью датчика случайных чисел. Найти сумму элементов матрицы, расположенных между первым и вторым положительными элементами каждой строки.

После ввода n с консоли задаётся матрица: создаётся 2-мерный массив размерности $n*n$, а затем заполняется элементами, генерируемые `random`.

Random генерирует числа от 0 до верхней границы (исключая её), подаваемой в качестве параметра. Для генерации чисел в нужном нам диапазоне верхнюю границу генерации random выберем $n*2+1$, а после генерации числа вычтем из него n. После генерации матрица печатается в консоль, а после в каждой строке ищутся позиции первого и второго положительных элементов и считается сумма элементов между этими позициями (не включая их). 0 не считается положительным. Если нет ни одного положительного элемента, то считается сумма всей строки матрицы. Если нет второго положительного элемента, то считается сумма от первого элемента до конца строки. Текст решения задачи приведён в листинге 5.

Листинг 5 – Код решения задачи 3

```
package var2.task1
import bmstu.general._
import scala.io.StdIn
import scala.util.Random

def main() =
  println("Enter size of a matrix: ")
  val n = StdIn.readInt()
  println("The Matrix:")
  val matrix = createMatrix(n)
  printMatrix(matrix)
  println(s"Sum of elements: ${matrix.map(sumBetweenNegatives(_)).sum}")

def createMatrix(n: Int): Array[Array[Int]] =
  val random = new Random()
  val matrix = Array.ofDim[Int](n, n)
  for (i <- 0 until n; j <- 0 until n) {
    matrix(i)(j) = random.nextInt(2 * n + 1) - n
  }
  matrix

def printMatrix(matrix: Array[Array[Int]]): Unit =
  for (row <- matrix) {
    println(row.mkString(", "))
  }

def sumBetweenNegatives(row: Array[Int]): Int =
  val firstPos = row.indexWhere(_ > 0, from = 0)
  if firstPos >= 0 then
    val secondPos = row.indexWhere(_ > 0, from = firstPos + 1)
    if secondPos > 0 then
      row.slice(firstPos + 1, secondPos).sum
    else
      row.slice(firstPos + 1, row.length).sum
  else
    row.sum
```


Результат работы программы приведён на рисунке 3.

```
→ java -jar exec/lab2.jar -p 3
Variant 2
Question 4
Enter size of a matrix:
6
The Matrix:
-5, -5, 3, -1, -1, -5
-2, 2, 3, -3, 1, 5
1, -2, -2, -1, 6, 0
-6, -1, 1, -4, 4, 4
-4, -1, -6, 4, 4, -2
-3, -2, 3, -6, 4, 3
Sum of elements: -22
-----
Developer: mikeGEINE
Task recieved on: Fri Feb 17 15:39:00 MSK 2023
Task completed (this run) on: Fri Mar 03 14:50:47 MSK 2023
```

Рисунок 3 – Решение задачи 3

Задача 4

Текст задания: ввести с консоли n – размерность матрицы $a[n][n]$. Задать значения элементов матрицы в интервале значений от $-n$ до n с помощью датчика случайных чисел. Транспонировать квадратную матрицу.

Матрица генерируется и печатается так же, как и в задаче 3. Для транспонирования матрицы используется метод транспонирования из стандартного набора методов массивов Scala. Код решения задачи 4 приведён в листинге 6.

Листинг 6 – Код решения задачи 4

```
package var2.task2
import bmstu.general._
import scala.io.StdIn
import scala.util.Random

def main() =
  println("Enter size of a matrix: ")
  val n = StdIn.readInt()
  println("The Matrix:")
  val matrix = createMatrix(n)
  printMatrix(matrix)
  println(s"Transposed matrix:")
  printMatrix(matrix.transpose)

def createMatrix(n: Int): Array[Array[Int]] =
  val random = new Random()
  val matrix = Array.ofDim[Int](n, n)
  for (i <- 0 until n; j <- 0 until n) {
    matrix(i)(j) = random.nextInt(2 * n + 1) - n
  }
  matrix

def printMatrix(matrix: Array[Array[Int]]): Unit =
  for (row <- matrix) {
    println(row.mkString(", "))
  }
```

Результат работы программы приведён на рисунке 4.

```
➔ java -jar exec/lab2.jar -p 4
Variant 2
Question 5
Enter size of a matrix:
3
The Matrix:
-2, 1, -1
0, 0, 2
0, 1, -1
Transposed matrix:
-2, 0, 0
1, 0, 1
-1, 2, -1
-----
Developer: mikeGEINE
Task recieved on: Fri Feb 17 15:39:00 MSK 2023
Task completed (this run) on: Fri Mar 03 14:56:30 MSK 2023
```

Рисунок 4 – Решение задачи 4

ВЫВОД

Написана консольная программа на основе ЛР1, решающая задачи ЛР2. В ходе решения поставленных задач изучены способы работы со строками, массивами и регулярными выражениями.

Освоена работа с датами с использованием класса Date. Изучены функция filter, методы регулярных выражений, различия между anchored и unanchored регулярными выражениями.

Изучен способ работы с random. Изучены методы создания многомерных массивов и их обработки. Изучены методы поиска в массиве, выбора промежутка массива.