



МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

Вариант 10

Дисциплина: Языки программирования для работы с большими данными

Москва, 2022

Цель работы:

Получение навыков работы с арифметическими операциями языка программирования Java.

Выполнение:

Задание 1:

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.

1. Ввести n строк с консоли, найти самую короткую и самую длинную строки. Вывести найденные строки и их длину.
2. Используя оператор switch, написать программу, которая выводит на экран сообщения о принадлежности некоторого значения k интервалам $(-10k, 5]$, $[0, 10]$, $[5, 15]$, $[10, 10k]$.

Задание 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.

1. Упорядочить строки (столбцы) матрицы в порядке возрастания значений элементов k-го столбца (строки).
2. Найти максимальный элемент(ы) в матрице и удалить из матрицы все строки и столбцы, его содержащие.

Листинг выполнения задачи 1 и 2

```
package com.company;
import java.io.InputStream;
import java.util.*;
import java.util.regex.*;

import java.util.Random;
import java.util.ArrayList;
import java.util.Scanner;

public class Main {

    public static void println(Object obj) {
        System.out.println(obj);
    }
}
```

```

public static void print(Object obj) {
    System.out.print(obj);
}

public static void println() {
    System.out.println();
}

public static String[] toStringArray(Collection<String> list) {
    String[] result = new String[list.size()];
    Iterator<String> it = list.iterator();
    for (int i = 0; i < list.size(); i++) {
        result[i] = it.next();
    }
    return result;
}

public static String[] readInput(Scanner sc) {
    List<String> input = new ArrayList<String>();
    print("Введите строки через пробел, для окончания введите q: ");
    while (sc.hasNext()) {
        String word = sc.next();
        if (word.equals("q")) {
            break;
        } else {
            input.add(word);
        }
    }
    sc.close();
    return toStringArray(input);
}

public static String[] readInput(InputStream is) {
    return readInput(new Scanner(is));
}

public static String[] readInput() {
    return readInput(System.in);
}

public static int[] toIntArray(Collection<Integer> list) {
    int[] result = new int[list.size()];
    Iterator<Integer> it = list.iterator();
    for (int i = 0; i < list.size(); i++) {
        result[i] = it.next();
    }
    return result;
}

public static int[] toIntArray(String[] sourceStrings) {
    List<Integer> numbers = new ArrayList<Integer>();
    for (int i = 0; i < sourceStrings.length; i++) {
        if (isNumericWord(sourceStrings[i])) {
            numbers.add(Integer.parseInt(sourceStrings[i]));
        }
    }
    int[] result = toIntArray(numbers);
    return result;
}

public static void findShortestAndLongestString(String[] sourceStrings) {
    String longest = sourceStrings[0];
    String shortest = sourceStrings[0];
    for (int i = 1; i < sourceStrings.length; i++) {
        if (shortest.length() > sourceStrings[i].length()) {

```

```

        shortest = sourceStrings[i];
    } else if (longest.length() < sourceStrings[i].length()) {
        longest = sourceStrings[i];
    }
}
println("Самая короткая строка (длина): " + shortest + "("
    + shortest.length() + ")");
println("Самая длинная строка (длина): " + longest + "("
    + longest.length() + ")");
}

public static boolean isNumericWord(String word) {
    Pattern numericPattern = Pattern.compile("[+-]?[0-9]+");
    Matcher numericMatcher = numericPattern.matcher(word);
    if (numericMatcher.matches()) {
        return true;
    }
    return false;
}

public static void isBelongToInterval(int[] kSourceValues,
    String... intervals) {
    print("Вхождения в интервалы " + Arrays.toString(intervals) + ": ");
    for (String s : intervals) {
        for (int k : kSourceValues) {
            String interval = null;
            switch (s) {
                case "(-10000, 0]":
                    if (k > -10000 && k <= 0) {
                        interval = s;
                    }
                    break;
                case "(-10000, 5]":
                    if (k > -10000 && k <= 5) {
                        interval = s;
                    }
                    break;
                case "(0, 5]":
                    if (k > 0 && k <= 5) {
                        interval = s;
                    }
                    break;
                case "[0, 10]":
                    if (k >= 0 && k <= 10) {
                        interval = s;
                    }
                    break;
                case "(5, 10]":
                    if (k > 5 && k <= 10) {
                        interval = s;
                    }
                    break;
                case "[5, 15]":
                    if (k >= 5 && k <= 15) {
                        interval = s;
                    }
                    break;
                case "(10, 10000]":
                    if (k > 10 && k <= 10000) {
                        interval = s;
                    }
                    break;
                case "[10, 10000]":
                    if (k >= 10 && k <= 10000) {
                        interval = s;
                    }
            }
        }
    }
}

```

```

        break;
    default:
        System.err.println("Ошибка интервала: " + s);
        return;
    }
    if (interval != null) {
        print(k + " ∈ " + interval + ", ");
    }
}
println();
}

public static int getRandom(int min, int max) {
    Random ran = new Random();
    int n = max - min + 1;
    int i = ran.nextInt() % n;
    if (i < 0)
        i = -i;
    return min + i;
}

public static void main(String[] args) {
    // Лаб2, вариант 1. Ввести n строк с консоли.

    // String[] sourceStrings = readInput();
    // if (sourceStrings.length == 0) {
    //     System.out.println("Нет строк. Программа завершена.");
    //     System.exit(0);
    // }
    //
    // // 1. Найти самую короткую и самую длинную строки. Вывести найденные строки и
    // их длину.
    // findShortestAndLongestString(sourceStrings);
    //
    // // 10. Используя оператор switch, написать программу, которая выводит на
    // экран сообщения о принадлежности
    // // некоторого значения k интервалам (-10k, 0], (0, 5], (5, 10], (10, 10k].*/
    //
    // int[] kSourceValues = toIntArray(sourceStrings);
    // isBelongToInterval(kSourceValues, "(-10000, 0]", "(0, 5]", "(5, 10]", "(10,
    10000]");

    /* Лаб2, 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. */
    // 1. Упорядочить строки (столбцы) матрицы в порядке возрастания значений
    элементов k-го столбца (строки).

    int q = 0;
    Scanner in = new Scanner(System.in);
    System.out.print("Input a number: ");
    int n = in.nextInt();
    in.close();

    int[][] a = new int[n][n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            a[i][j] = getRandom(-100, 100);
        }
    }
    System.out.println("-----Created Matrix (Random)-----");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            System.out.print(a[i][j] + "\t");

```

```

    }
    System.out.println();
}
int[][] sorted_a = new int[n][n];
for(int i = 0; i < a.length; i++) {
    System.arraycopy(a[i], 0, sorted_a[i], 0, a[i].length);
}
// bubble sort
for (int i = 0; i < n * n - 1; i++) {
    for (int j = i; j < n * n; j++) {
        if (sorted_a[(int) i / n][(int) i % n] > sorted_a[(int) j / n][(int) j
% n]) {
            q = sorted_a[(int) i / n][(int) i % n];
            sorted_a[(int) i / n][(int) i % n] = sorted_a[(int) j / n][(int) j
% n];
            sorted_a[(int) j / n][(int) j % n] = q;
        }
    }
}
System.out.println("-----New Matrix-----");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        System.out.print(sorted_a[i][j] + "\t");
    }
    System.out.println();
}

// 10. Найти максимальный элемент(ы) в матрице и удалить из матрицы все строки
и столбцы, его содержащие
int max = -n;

ArrayList<Integer> rows = new ArrayList<Integer>();
ArrayList<Integer> column = new ArrayList<Integer>();

for (int i = 0; i < a.length; i++) {
    for (int j = 0; j < a.length; j++) {
        if(a[i][j] > max) {
            max = a[i][j];
        }
    }
}
System.out.println("\nMax: " + max);

// rows and columns that must be deleted
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        if(max == a[i][j]) {
            if(!rows.contains(j))
                rows.add(j);
            if(!column.contains(i))
                column.add(i);
        }
    }
}

int[][] newArr = new int[n - rows.size()][n - column.size()];
System.out.println("Size: " + (n - rows.size()) + "x" + (n - column.size()));

int newArrI = 0;
int newArrJ = 0;
// Net matrix without max element rows and columns
for (int i = 0; i < n; i++) {
    if(!rows.contains(i)) {
        newArrJ = 0;
        for (int j = 0; j < n; j++) {

```

```

        if(!column.contains(j)) {
            newArr[newArrI][newArrJ] = a[j][i];
            newArrJ++;
        }
        newArrI++;
    }
}

System.out.println("-----Matrix without max element rows and columns-----");

for (int i = 0; i < newArr.length; i++) {
    for (int j = 0; j < newArr[i].length; j++) {
        System.out.print(newArr[j][i] + "\t");
    }
    System.out.println();
}
}
}

```

Введите строки через пробел, для окончания введите q: f7fg87 fg74g fg478 f4g8 q
 Самая короткая строка (длина): f4g8(4)
 Самая длинная строка (длина): f7fg87(6)
 Вхождения в интервалы $[(-10000, 0], (0, 5], (5, 10], (10, 10000]]$:
 Process finished with exit code 0

Рисунок 1 - Результат выполнения кода решения задачи 1

```

Input a number: 3
-----Created Matrix (Random)-----
85  -61 28
73  -29 46
-14 97  -39
-----New Matrix-----
-61 -39 -29
-14 28  46
73  85  97

Max: 97
Size: 2x2
-----Matrix without max element rows and columns-----
85  28
73  46

Process finished with exit code 0

```

Рисунок 2 - Результат выполнения кода решения задачи 2

Ссылка на программное решение:

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Wingo11/BigDataLanguages/tree/Lab2/src>

Вывод:

При выполнении лабораторной работы были получены навыки работы с арифметическими операциями языка программирования Java.