



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

О Т Ч Е Т

по лабораторной работе № 2

Вариант 18

Название: Арифметические операции

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

Д.Н. Хныкин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Цель работы

Получение навыков работы с арифметическими операциями языка программирования Java.

Выполнение

Задание 1:

1. Ввести n слов с консоли. Среди слов, состоящих только из цифр, найти слово-палиндром. Если таких слов больше одного, найти второе из них.
2. Используя оператор `switch`, написать программу, которая выводит на экран сообщения о принадлежности некоторого значения k интервалам $(-10k, 0]$, $(0, 5]$, $(5, 10]$, $(10, 10k]$.

Листинг выполнения подзадачи 1

```
package com.aranei;

import org.jetbrains.annotations.NotNull;
import java.util.Objects;
import java.util.Scanner;

public class Var1 {
    public String firstTask = "Ввести n слов с консоли. Среди слов, состоящих только из цифр, найти слово-палиндром. Если таких слов больше одного, найти второе из них";
    public String secondTask = "Используя оператор switch, написать программу, которая выводит на экран сообщения о принадлежности некоторого значения k интервалам (-10k, 0], (0, 5], (5, 10], (10, 10k]";

    public int polNum = 0;
    public String pol;
    public int total;
    String[] myArray;

    public void getTask(int task) {
        switch (task) {
            case 0:
                System.out.println(firstTask);
                break;
            case 1:
                System.out.println(secondTask);
                break;
        }
    }

    public void getArraySize() {
        System.out.println("Array size:");
        Scanner in = new Scanner(System.in);
        total = in.nextInt();
        myArray = new String[total];
    }

    public void getArray() {
        System.out.println("Array:");
        Scanner in = new Scanner(System.in);
        for (int i=0; i<total; i++){
            myArray[i] = in.next();
            if (polNum < 2) {
                if (checkDigits(myArray[i])){
                    if (checkPalindrom(myArray[i])) {
                        pol = myArray[i];
                        polNum++;
                    }
                }
            }
        }
    }
}
```

```

        System.out.println(pol);
    }

    public boolean checkDigits(@NotNull String someString) {
        return someString.chars().allMatch( Character::isDigit );
    }

    public static boolean checkPalindrom(String str) {
        return Objects.equals(str, new StringBuilder(str).reverse().toString());
    }

    public int k;

    public void getK() {
        System.out.println("Number:");
        Scanner in = new Scanner(System.in);
        k = in.nextInt();
    }

    public void checkInterval() {
        switch (k < 1 ? 0 : 1) {
            case 0:
                switch (k < -10_000 ? 0 : 1) {
                    case 0:
                        System.out.println(k + " is out of range");
                        break;
                    case 1:
                        System.out.println(k + " is in range (-10k, 0]");
                        break;
                }
                break;
            case 1:
                switch (k < 6 ? 0 : 1) {
                    case 0:
                        System.out.println(k + " is in range (0, 5]");
                        break;
                    case 1:
                        switch (k < 11 ? 0 : 1) {
                            case 0:
                                System.out.println(k + " is in range (5, 10]");
                                break;
                            case 1:
                                switch (k < 10_000 ? 0 : 1) {
                                    case 0:
                                        System.out.println(k + " is in range (10, 10k]");
                                        break;
                                    case 1:
                                        System.out.println(k + " is out of range");
                                        break;
                                }
                            }
                        }
                }
            }
        }
    }
}

```

Задание 2:

3. Вычислить определитель матрицы.
4. Построить матрицу, вычитая из элементов каждой строки матрицы ее среднее арифметическое.

Листинг выполнения подзадачи 2

#Var2.java

```

package com.aranei;

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

public class Var2 {
    public String firstTask = "Вычислить определитель матрицы";
}

```

```

    public String secondTask = "Построить матрицу, вычитая из элементов каждой строки матрицы
ее среднее арифметическое";

    double[][] matrix, matrix2;
    public int n;
    Random r = new Random();

    public void getTask(int task) {
        switch (task) {
            case 0:
                System.out.println(firstTask);
                break;
            case 1:
                System.out.println(secondTask);
                break;
        }
    }

    public void getMatrixSize() {
        System.out.println("Matrix size:");
        Scanner in = new Scanner(System.in);
        n = in.nextInt();
        matrix = new double[n][n];
    }

    public void setMatrix() {
        for (int i=0; i<matrix.length; i++){
            for (int j=0; j<matrix[i].length; j++){
                matrix[i][j] = r.nextDouble() * 100;
            }
        }
        matrix2 = matrix;
    }

    public void showMatrix() {
        System.out.println("Matrix:");
        for (double[] doubles : matrix) {
            for (double aDouble : doubles) {
                System.out.format(" %.3f ",aDouble);
            }
            System.out.println();
        }
    }

    double temp, tempMed;

    public void setMedMatrix() {
        for (int i=0; i<matrix2.length; i++){
            temp = 0;
            for (int j=0; j<matrix2[i].length; j++){
                temp += matrix2[i][j];
            }
            tempMed = temp/matrix2[i].length;
            for (int j=0; j<matrix2[i].length; j++){
                matrix2[i][j] -= tempMed;
            }
        }
        showMatrix2();
    }

    public void showMatrix2() {
        System.out.println("MatrixMed:");
        for (double[] doubles : matrix2) {
            for (double aDouble : doubles) {
                System.out.format(" %.3f ",aDouble);
            }
            System.out.println();
        }
    }

    public void calcMatrix(){
        Matrix matrixDet = new Matrix(matrix);
        System.out.format("Det: %.3f\n", matrixDet.determinant());
    }
}

```

#Matrix.java

package com.aranei;

```

import org.jetbrains.annotations.NotNull;

public class Matrix {
    private double[][] data = null;
    private int rows = 0, cols = 0;

    public Matrix(int rows, int cols) {
        data = new double[rows][cols];
        this.rows = rows;
        this.cols = cols;
    }

    public Matrix(double[][] data) {
        this.data = data.clone();
        rows = this.data.length;
        cols = this.data[0].length;
    }

    public static @NotNull Matrix subMatrix(Matrix matrix, int exclude_row, int exclude_col)
    {
        Matrix result = new Matrix(matrix.rows - 1, matrix.cols - 1);

        for (int row = 0, p = 0; row < matrix.rows; ++row) {
            if (row != exclude_row - 1) {
                for (int col = 0, q = 0; col < matrix.cols; ++col) {
                    if (col != exclude_col - 1) {
                        result.data[p][q] = matrix.data[row][col];

                        ++q;
                    }
                }
                ++p;
            }
        }

        return result;
    }

    public double determinant() {
        if (rows != cols) {
            return Double.NaN;
        }
        else {
            return _determinant(this);
        }
    }

    public double _determinant(@NotNull Matrix matrix) {
        if (matrix.cols == 1) {
            return matrix.data[0][0];
        }
        else if (matrix.cols == 2) {
            return (matrix.data[0][0] * matrix.data[1][1] -
                    matrix.data[0][1] * matrix.data[1][0]);
        }
        else {
            double result = 0.0;

            for (int col = 0; col < matrix.cols; ++col) {
                Matrix sub = subMatrix(matrix, 1, col + 1);

                result += (Math.pow(-1, 1 + col + 1) *
                           matrix.data[0][col] * _determinant(sub));
            }
            return result;
        }
    }
}

```

Ссылка на программное решение

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Aranei99/labsJava/tree/main/LB2>

Вывод

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