



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

Название: Арифметические операции

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-22М

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Т.И. Кадыров

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2024

**Цель работы:** освоить базовые принципы работы с арифметическими операциями на языке Java.

**Вариант: 8.**

**Задание 1:** Ввести  $n$  слов с консоли. Среди слов, состоящих только из цифр, найти слово-палиндром. Если таких слов больше одного, найти второе из них.

Код решения приведен в листинге 1.

Листинг 1 — реализация решения

```
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        scanner.nextLine();

        String firstPalindrome = null;
        String secondPalindrome = null;

        for (int i = 0; i < n; i++) {
            String word = scanner.nextLine();

            if (isPalindrome(word) && containsOnlyDigits(word)) {
                if (firstPalindrome == null) {
                    firstPalindrome = word;
                } else if (secondPalindrome == null) {
                    secondPalindrome = word;
                }
            }
        }

        if (secondPalindrome != null) {
            System.out.println(secondPalindrome);
            return;
        }
        System.out.println(firstPalindrome);
        System.out.println(getSign());
    }

    private static boolean isPalindrome(String word) {
        String reversed = new StringBuilder(word).reverse().toString();
        return word.equals(reversed);
    }

    private static boolean containsOnlyDigits(String word) {
```

```

        return word.matches("\\d+");
    }

    private static String getSign() {
        String lastName = "Kadyrov";
        SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy hh:mm");
        Date taskStartDateTime = new Date(124, 1, 9, 12, 0);
        String startDateTime = formatter.format(taskStartDateTime);
        long workInterval = 20 * 24 * 60 * 60 * 1000L;
        Date TaskEndDatetime = new Date(taskStartDateTime.getTime() + workInterval);
        String endDateTime = formatter.format(TaskEndDatetime);
        return lastName + "-" + startDateTime + "-" + endDateTime;
    }
}

```

**Задание 2:** Используя оператор switch, написать программу, которая выводит на экран сообщения о принадлежности некоторого значения k интервалам  $(-10k, 0]$ ,  $(0, 5]$ ,  $(5, 10]$ ,  $(10, 10k]$ .

Код решения приведен в листинге 2.

Листинг 2 — реализация решения

```

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);
        int k = in.nextInt();

        switch (getInterval(k)) {
            case 1:
                System.out.println("(-10k;0]");
                break;
            case 2:
                System.out.println("(0;5]");
                break;
            case 3:
                System.out.println("(5;10]");
                break;
            case 4:
                System.out.println("(10;10k]");
                break;
            default:
                System.out.println("Other interval");
                break;
        }
        System.out.println(getSign());
    }
}

```

```

private static String getSign() {
    String lastName = "Kadyrov";
    SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy hh:mm");
    Date taskStartDateTime = new Date(124, 1, 9, 12, 0);
    String startDateTime = formatter.format(taskStartDateTime);
    long workInterval = 20 * 24 * 60 * 60 * 1000L;
    Date TaskEndDatetime = new Date(taskStartDateTime.getTime() + workInterval);
    String endDateTime = formatter.format(TaskEndDatetime);
    return lastName + "-" + startDateTime + "-" + endDateTime;
}

private static int getInterval(int k) {
    if (k > -10 * k && k <= 0) {
        return 1;
    } else if (k > 0 && k <= 5) {
        return 2;
    } else if (k > 5 && k <= 10) {
        return 3;
    } else if (k > 10 && k <= 10 * k) {
        return 4;
    } else {
        return 0;
    }
}
}

```

### Задание 3: Вычислить определитель матрицы.

Код решения приведен в листинге 3.

Листинг 3 — реализация решения

```

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

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int[][] matrix = generateMatrix(n);
        System.out.println("Matrix is:");
        printMatrix(matrix);
        System.out.printf("Determinant: %d", calculateDeterminant(matrix));
    }

    private static int calculateDeterminant(int[][] matrix) {
        int n = matrix.length;

        if (n == 1) {
            return matrix[0][0];
        }

        if (n == 2) {
            return matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0];
        }
    }
}

```

```

    }

    int determinant = 0;

    for (int i = 0; i < n; i++) {
        determinant += Math.pow(-1, i) * matrix[0][i] * calculateDeterminant(minor(matrix, 0, i));
    }

    return determinant;
}

private static int[][] minor(int[][] matrix, int row, int col) {
    int n = matrix.length - 1;
    int[][] minor = new int[n][n];

    int minorRow = 0, minorCol;
    for (int i = 0; i < matrix.length; i++) {
        if (i == row) {
            continue;
        }
        minorCol = 0;
        for (int j = 0; j < matrix[i].length; j++) {
            if (j == col) {
                continue;
            }
            minor[minorRow][minorCol] = matrix[i][j];
            minorCol++;
        }
        minorRow++;
    }

    return minor;
}

private static int[][] generateMatrix(int n) {
    int[][] matrix = new int[n][n];
    Random random = new Random();
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            matrix[i][j] = random.nextInt(2 * n + 1) - n;
        }
    }
    return matrix;
}

private static void printMatrix(int[][] matrix) {
    for (int[] row : matrix) {
        for (int value : row) {
            System.out.print(value + " ");
        }
        System.out.println();
    }
}
}

```

**Задание 4:** Построить матрицу, вычитая из элементов каждой строки матрицы ее среднее арифметическое.

Код решения приведен в листинге 4.

Листинг 4 — реализация решения

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

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int[][] matrix = generateMatrix(n);
        System.out.println("Original matrix is:");
        printMatrix(matrix);
        System.out.println("Custom matrix is:");
        printMatrix(generateCustomMatrix(matrix));
    }

    private static int[][] generateCustomMatrix(int[][] source) {
        int size = source[0].length;
        int[][] customMatrix = new int[size][size];

        for (int i = 0; i < size; i++) {
            int sum = 0;
            for (int j = 0; j < size; j++) {
                sum += source[i][j];
            }

            int average = sum / size;

            for (int j = 0; j < size; j++) {
                customMatrix[i][j] = source[i][j] - average;
            }
        }

        return customMatrix;
    }

    private static int[][] generateMatrix(int n) {
        int[][] matrix = new int[n][n];
        Random random = new Random();
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = random.nextInt(2 * n + 1) - n;
            }
        }
        return matrix;
    }

    private static void printMatrix(int[][] matrix) {
        for (int[] row : matrix) {
```

```
    for (int value : row) {  
        System.out.print(value + " ");  
    }  
    System.out.println();  
}  
}  
}
```

**Вывод:** в ходе лабораторной работы были освоены базовые принципы работы с арифметическими операциями на языке Java.