



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

Вариант № 7

Название: арифметические операции

Дисциплина: языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

Ф.А. Лучкин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2024

Цель: освоить принципы арифметических операций на языке программирования Java.

Задание 1: ввести n слов с консоли. Найти слово, состоящее только из различных символов. Если таких слов несколько, найти первое из них (необходимо вывести внизу фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания).

Код класса Main:

```
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.*;
import static java.lang.System.*;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(in);
        Random random = new Random();
        int n = random.nextInt(5) + 2;

        // INPUT RESULT
        out.printf("Input %d words\n", n);
        String[] words = new String[n];
        for (int i = 0; i < n; i++) {
            words[i] = scanner.next();
        }
        // INPUT RESULT
        out.printf("Your entered words: %s\n", Arrays.toString(words));

        // TASK
        out.print("Result: " + findUniqueSymbolsWord(words));

        // INFO
        DateFormat dateFormat = new SimpleDateFormat("yyyy.MM.dd HH:mm");
        Date date = new Date();
        out.printf("\nStudent: %s\nAssignment datetime: %s\nCompletion datetime: %s",
            "Luchkin", "2024.02.09 14:29", dateFormat.format(date));
    }

    private static String findUniqueSymbolsWord(String[] words) {
        for (String word : words) {
            if (hasOnlyUniqueCharacters(word)) {
                return word;
            }
        }
        return "No word with unique Symbols";
    }

    public static boolean hasOnlyUniqueCharacters(String word) {
        boolean[] characters = new boolean[128];
        for (char c : word.toCharArray()) {
            if (characters[c]) {
                return false;
            }
            characters[c] = true;
        }
        return true;
    }
}
```

Работа программы показана на рисунке 1.

```
Input 5 words
sdf sdf sdrtd 5f5 4f5 rty
Your entered words: [sdf sdf, sdrtd, 5f5, 4f5, rty]
Result: 4f5
Student: Luchkin
Assignment datetime: 2024.02.09 14:29
Completion datetime: 2024.02.12 23:30
```

Рисунок 1 – Работа программы

Задание 2: ввести n слов с консоли. Среди слов, состоящих только из цифр, найти слово-палиндром. Если таких слов больше одного, найти второе из них (необходимо вывести внизу фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания).

Код класса Main:

```
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.*;

import static java.lang.System.out;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Random random = new Random();
        int n = random.nextInt(5) + 2;

        // INPUT RESULT
        System.out.printf("Input %d words\n", n);
        String[] words = new String[n];
        for (int i = 0; i < n; i++) {
            words[i] = scanner.next();
        }
        // INPUT RESULT
        System.out.printf("Your entered words: %s\n", Arrays.toString(words));

        // TASK
        System.out.print("Result: " + findDigitPalindromeWord(words));

        // INFO
        DateFormat dateFormat = new SimpleDateFormat("yyyy.MM.dd HH:mm");
        Date date = new Date();
        out.printf("\nStudent: %s\nAssignment datetime: %s\nCompletion datetime: %s",
            "Luchkin", "2024.02.09 14:29", dateFormat.format(date));
    }

    private static String findDigitPalindromeWord(String[] words) {
        String result = "";
        int flag = 0;
        int i = 0;
        while (flag == 0 && i < words.length) {

```

```

        if (isDigitPalindrome(words[i])) {
            if (!result.isEmpty()) flag = 1;
            result = words[i];
        }
        i++;
    }
    if (!result.isEmpty()) {
        return result;
    } else {
        return "No Digit Palindrome Word";
    }
}

public static boolean isDigitPalindrome(String word) {
    if (!word.matches("\\d+")) {
        return false;
    }

    int left = 0;
    int right = word.length() - 1;

    while (left < right) {
        if (word.charAt(left) != word.charAt(right)) {
            return false;
        }
        left++;
        right--;
    }

    return true;
}
}

```

Работа программы показана на рисунке 2.

```

Input 5 words
rtr 12345 545 1221 12321
Your entered words: [rtr, 12345, 545, 1221, 12321]
Result: 1221
Student: Luchkin
Assignment datetime: 2024.02.09 14:29
Completion datetime: 2024.02.12 23:35

```

Рисунок 2 – Работа программы

Задание 3: 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. Повернуть матрицу на 90 (180, 270) градусов против часовой стрелки.

Код класса Main:

```

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 = 0;
while (n <= 0) {
    try {
        System.out.print("Enter matrix dimension: ");
        n = Integer.parseInt(scanner.nextLine());
        if (n <= 0) System.out.println("Matrix dimension must be > 0!");
    } catch (NumberFormatException e) {
        System.out.println("IT'S NOT A INTEGER!");
    }
}

int[][] matrix = generateMatrix(n);
System.out.println("Source matrix:");
printMatrix(matrix);

for (int i=1; i<4; i++) {
    matrix = rotateMatrix(matrix);
    System.out.printf("Matrix after rotating %d degrees  
counterclockwise:\n", i * 90);
    printMatrix(matrix);
}

public 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;
}

public static int[][] rotateMatrix(int[][] matrix) {
    int n = matrix.length;
    int[][] rotatedMatrix = new int[n][n];

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            rotatedMatrix[i][j] = matrix[j][n - i - 1];
        }
    }
    return rotatedMatrix;
}

public static void printMatrix(int[][] matrix) {
    int n = matrix.length;

    for (int[] ints : matrix) {
        for (int j = 0; j < n; j++) {
            System.out.print(ints[j] + "\t");
        }
        System.out.println();
    }
}
}

```

Работа программы показана на рисунке 3.

```

Enter matrix dimension: 5
Source matrix:
5   2   1  -5   1
-2  -2  -2  -1   1
-1  -3   2   0  -4
2   -2  -2  -5   5
1   3   2  -5   2

Matrix after rotating 90 degrees counterclockwise:
1   1  -4   5   2
-5  -1   0  -5  -5
1  -2   2  -2   2
2  -2  -3  -2   3
5  -2  -1   2   1

Matrix after rotating 180 degrees counterclockwise:
2  -5   2   3   1
5  -5  -2  -2   2
-4   0   2  -3  -1
1  -1  -2  -2  -2
1  -5   1   2   5

Matrix after rotating 270 degrees counterclockwise:
1   2  -1  -2   5
3  -2  -3  -2   2
2  -2   2  -2   1
-5  -5   0  -1  -5
2   5  -4   1   1

```

Рисунок 3 – Работа программы

Задание 4: 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.

Вычислить определитель матрицы.

Код модуля Main:

```

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 = 0;
        while (n <= 0) {
            try {

```

```

        System.out.print("Enter matrix dimension: ");
        n = Integer.parseInt(scanner.nextLine());
        if (n <= 0) System.out.println("Matrix dimension must be > 0!");
    } catch (NumberFormatException e) {
        System.out.println("IT'S NOT A INTEGER!");
    }
}

int[][] matrix = generateMatrix(n);
System.out.println("Source matrix:");
printMatrix(matrix);
int determinant = calculateDeterminant(matrix);
System.out.println("Matrix Determinant: " + determinant);
}

public 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;
}

public static int calculateDeterminant(int[][] matrix) {
    int n = matrix.length;
    if (n == 1) return matrix[0][0];
    int determinant = 0;
    for (int j = 0; j < n; j++) {
        determinant += matrix[0][j] * getCofactor(matrix, 0, j);
    }
    return determinant;
}

public static int getCofactor(int[][] matrix, int row, int col) {
    int n = matrix.length;
    int subMatrixRow = 0;
    int subMatrixColumn = 0;
    int[][] subMatrix = new int[n - 1][n - 1];

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i != row && j != col) {
                subMatrix[subMatrixRow][subMatrixColumn++] = matrix[i][j];
                if (subMatrixColumn == n - 1) {
                    subMatrixRow++;
                    subMatrixColumn = 0;
                }
            }
        }
    }

    return (int) Math.pow(-1, row + col) * calculateDeterminant(subMatrix);
}

public static void printMatrix(int[][] matrix) {
    int n = matrix.length;
    for (int[] element : matrix) {
        for (int j = 0; j < n; j++) {
            System.out.print(element[j] + "\t");
        }
        System.out.println();
    }
}
}

```

Работа программы показана на рисунке 4.

```
Enter matrix dimension: 5
Source matrix:
-3  -5  0  5  3
-3  -5  -2 -2  5
 4   4  2  -2 -3
-2  5  -4 -4  3
 5  -5  5  -1  2
Matrix Determinant: -123
```

Рисунок 4 – Работа программы

Ссылка на git-репозиторий: https://github.com/FedorLuchkin/Java_bmstu

Вывод: были освоены принципы арифметических операций на языке программирования Java.