



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

ОТЧЕТ

по лабораторной работе № 2

Название: Арифметические операции

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

Э.А. Гаджиев

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Файл Program.java:

```
import java.lang.reflect.Array;
import java.util.*;

public class Program {
    public static void main(String[] args) {
        task_1();
        task_2();
        task_3();
        task_4();
    }

    public static void task_1() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Input number of words: ");
        int n = scanner.nextInt();

        String[] words = new String[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Input " + (i+1) + " word: ");
            words[i] = scanner.next();
        }

        ArrayList<HashSet<String>> wordSets = new ArrayList<HashSet<String>>(n);
        for (int i = 0; i < words.length; i++) {
            String[] splitedWord = words[i].split("");
            wordSets.add(new HashSet<String>());
            for (int j = 0; j < splitedWord.length; j++) {
                wordSets.get(i).add(splitedWord[j]);
            }
        }

        int minUniqueCount = wordSets.get(0).size();
        // System.out.println(minUniqueCount);
        int minUniqueIdx = 0;

        System.out.println(wordSets.toString());
        for (int i = 1; i < n; i++) {

            if (wordSets.get(i).size() < minUniqueCount) {
                minUniqueIdx = i;
            }
        }

        System.out.println("Result: " + words[minUniqueIdx]);
    }

    public static void task_2() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Input number of words: ");
        int n = scanner.nextInt();

        String[] words = new String[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Input " + (i+1) + " word: ");
        }
    }
}
```

```

        words[i] = scanner.next();
    }

//    ArrayList<String> latinWords = new ArrayList<String>();
    System.out.println("Latin words: ");
    for (int i = 0; i < words.length; i++) {
        boolean isLatin = true;
        for (int j = 0; j < words[i].length(); j++) {
            char currChar = words[i].charAt(j);
            isLatin = isLatin && (((currChar >= 'a') && (currChar <= 'z')) || ((currChar >= 'A') && (currChar
<= 'Z')));
        }
        if (isLatin) {
            System.out.print(words[i]);
//            latinWords.add(words[i]);
            int soglCount = 0;
            int glCount = 0;
            for (int j = 0; j < words[i].length(); j++) {
                String currWord = words[i].toLowerCase(Locale.ROOT);
                if ((currWord.charAt(j) == 'a') ||
                    (currWord.charAt(j) == 'e') ||
                    (currWord.charAt(j) == 'i') ||
                    (currWord.charAt(j) == 'o') ||
                    (currWord.charAt(j) == 'y')) {
                    glCount++;
                } else {
                    soglCount++;
                }
            }
            if (soglCount == glCount) {
                System.out.print("(!) ");
            } else {
                System.out.print(" ");
            }
        }
    }

//    for (int i = 0; i < latinWords.size(); i++) {
//        int soglCount = 0;
//        int glCount = 0;
//        for (int j = 0; j < latinWords.get(i).length(); j++) {
//        }
//    }

}

// Ввести с консоли n - размерность матрицы. 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
// каждый строки.
public static void task_3() {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Input size of matrix: ");

```

```

int mtrxSize = scanner.nextInt();
int[][] mtrx = new int[mtrxSize][mtrxSize];

for (int i = 0; i < mtrxSize; i++) {
    for (int j = 0; j < mtrxSize; j++) {
        Random rand = new Random();
        mtrx[i][j] = rand.nextInt(2) == 1 ? -rand.nextInt(mtrxSize) : rand.nextInt(mtrxSize);
        System.out.printf("%2d ", mtrx[i][j]);
    }
    System.out.println();
}

for (int i = 0; i < mtrxSize; i++) {
    int firstPositiveIdx = -1;
    int secondPositiveIdx = -1;
    int sum = 0;

    for (int j = 0; j < mtrxSize; j++) {
        if ((firstPositiveIdx == -1) && (mtrx[i][j] >= 0)) {
            firstPositiveIdx = j;
        } else if ((firstPositiveIdx != -1) && (secondPositiveIdx == -1) && (mtrx[i][j] >= 0)) {
            secondPositiveIdx = j;
        }
    }

    if (firstPositiveIdx != -1 && secondPositiveIdx != -1) {
        for (int j = firstPositiveIdx + 1; j < secondPositiveIdx; j++) {
            sum += mtrx[i][j];
        }
    }

    System.out.println("The sum of elements of " + i + " row is " + sum);
}

}

// Транспонировать квадратную матрицу
public static void task_4() {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Input size of matrix: ");

    int mtrxSize = scanner.nextInt();
    int[][] mtrx = new int[mtrxSize][mtrxSize];

    System.out.println("Original matrix:");
    for (int i = 0; i < mtrxSize; i++) {
        for (int j = 0; j < mtrxSize; j++) {
            Random rand = new Random();
            mtrx[i][j] = rand.nextInt(2) == 1 ? -rand.nextInt(mtrxSize) : rand.nextInt(mtrxSize);
            System.out.printf("%2d ", mtrx[i][j]);
        }
        System.out.println();
    }

    int[][] transposedMtrx = new int[mtrxSize][mtrxSize];

```

```
        for (int i = 0; i < mtrxSize; i++) {
            for (int j = 0; j < mtrxSize; j++) {
                transposedMtrx[i][j] = mtrx[j][i];
            }
        }

        System.out.println("Transposed matrix:");
        for (int i = 0; i < mtrxSize; i++) {
            for (int j = 0; j < mtrxSize; j++) {
                System.out.printf("%2d ", transposedMtrx[i][j]);
            }
            System.out.println();
        }
    }
}
```