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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ 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 №6

Название: Лабораторная работа №6

Дисциплина: Языки программирования для работы с большими данными

Студент	ИУ6-22М		М.И. Замула
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			П.В. Степанов
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2023 г.

Вариант: 1

Задание 8

Задана строка, состоящая из символов '(', ')', '[', ']', '{', '}'. Проверить правильность расстановки скобок. Использовать стек.

Выполнение

```
package lab6;

import java.util.Stack;

public class Lab6_1_8 {
    public static void main(String[] args) {
        String str = "([]{()})";

        if (isValid(str)) {
            System.out.println("Правильная расстановка скобок");
        } else {
            System.out.println("Неправильная расстановка скобок");
        }
    }

    public static boolean isValid(String str) {
        Stack<Character> stack = new Stack<>();

        for (char c : str.toCharArray()) {
            if (c == '(' || c == '[' || c == '{') {
                stack.push(c);
            } else if (c == ')' && !stack.isEmpty() && stack.peek() == '(') {
                stack.pop();
            } else if (c == ']' && !stack.isEmpty() && stack.peek() == '[') {
                stack.pop();
            } else if (c == '}' && !stack.isEmpty() && stack.peek() == '{') {
                stack.pop();
            } else {
                return false;
            }
        }

        return stack.isEmpty();
    }
}
```

Результаты

```
Правильная расстановка скобок
```

Задание 9

Задан файл с текстом на английском языке. Выделить все различные слова. Слова, отличающиеся только регистром букв, считать одинаковыми. Использовать класс HashSet.

Выполнение

```java
package lab6;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class Lab6_1_9 {
    public static void main(String[] args) {
        File file = new File("lab6_1_9.txt"); // замените на имя вашего файла
        Set<String> words = new HashSet<>();

        try (Scanner scanner = new Scanner(file)) {
            while (scanner.hasNext()) {
                String word = scanner.next().toLowerCase();
                words.add(word);
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

        System.out.println("Уникальные слова в файле:");
        System.out.println(words);
    }
}
```

lab6_1_9.txt

```
To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles
And by opposing end them. To die—to sleep,
No more; and by a sleep to say we end
The heart-ache and the thousand natural shocks
That flesh is heir to: 'tis a consummation
Devoutly to be wish'd. To die, to sleep;
To sleep, perchance to dream—ay, there's the rub,
For in that sleep of death what dreams may come,
When we have shuffled off this mortal coil,
Must give us pause—there's the respect
That makes calamity of so long life:
For who would bear the whips and scorns of time,
Th'oppressor's wrong, the proud man's contumely,
The pangs of despised love, the law's delay,
The insolence of office, and the spurns
That patient merit of th'unworthy takes,
When he himself might his quietus make
With a bare bodkin? Who would fardels bear,
To grunt and sweat under a weary life,
But that the dread of something after death,
The undiscover'd country from whose bourn
No traveller returns, puzzles the will
And makes us rather bear those ills we have
Than fly to others that we know not of?
Thus conscience does make cowards of us all;
```

```
And thus the native hue of resolution
Is sicklied o'er with the pale cast of thought,
And enterprises of great pith and moment,
With this regard their currents turn awry,
And lose the name of action.
```

Результат

Уникальные слова в файле:

[them., 'tis, whips, when, resolution, come,, dream—ay,, pangs, cast, suffer, would, pith, sleep;, pause—there's, wish'd., die—to, traveller, to:, pale, devoutly, there's, give, natural, thus, in, lose, is, man's, whose, something, himself, must, shocks, flesh, office,, awry,, be, against, nobler, be,, of?, turn, dread, sea, long, sleep, law's, currents, native, does, by, have, death,, after, so, insolence, undiscover'd, mind, a, spurns, may, weary, makes, sleep,, time,, bear,, great, off, delay,, the, fly, o'er, regard, action., to, under, thousand, but, country, sicklied, die,, despised, conscience, arrows, moment,, enterprises, heart-ache, that, his, life,, wrong,, whether, than, all;, bear, from, takes,, bare, life:, bourn, us, those, others, proud, puzzles, fortune,, might, cowards, shuffled, this, thought,, merit, th'unworthy, respect, take, more;, perchance, rub,, name, know, arms, who, no, death, rather, for, their, slings, dreams, troubles, we, question:, grunt, not, calamity, and, heir, consummation, patient, of, th'oppressor's, end, coil,, ills, outrageous, scorns, love,, make, mortal, opposing, or, will, bodkin?, fardels, say, quietus, with, what, sweat, contumely,, hue, returns,, he]

Вариант: 2

Задание 8

На клетчатом листе бумаги закрашена часть клеток. Выделить все различные фигуры, которые образовались при этом. Фигурой считается набор закрашенных клеток, достижимых друг из друга при движении в четырёх направлениях. Две фигуры являются различными, если их нельзя совместить поворотом на угол, кратный 90 градусам, и параллельным переносом. Используйте класс HashSet.

Выполнение

```java
package lab6;

import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import java.util.Scanner;
import java.util.Set;

public class Lab6_2_8 {

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

        System.out.print("Введите количество строк в листе бумаги: ");
        int rows = scanner.nextInt();

        System.out.print("Введите количество столбцов в листе бумаги: ");
        int cols = scanner.nextInt();

        boolean[][] grid = new boolean[rows][cols];

        System.out.println("Введите расположение закрашенных клеток:");

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                int cellValue = scanner.nextInt();
                grid[i][j] = (cellValue == 1);
            }
        }

        Map<String, Set<String>> figures = new HashMap<>();

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                if (grid[i][j]) {
                    Set<String> cells = new HashSet<>();
                    String figure = findFigure(grid, i, j, rows, cols,
cells);
                    if (figures.containsKey(figure)) {
                        figures.get(figure).addAll(cells);
                    } else {
                        figures.put(figure, cells);
                    }
                }
            }
        }

        System.out.println("Уникальные фигуры на клетчатом листе бумаги:");

        int count = 1;
        for (Map.Entry<String, Set<String>> entry : figures.entrySet()) {
            System.out.println("Фигура " + count++);
            System.out.println("Координаты закрашенных клеток: " +
entry.getValue());
        }
    }

    private static String findFigure(boolean[][] grid, int i, int j, int
rows, int cols, Set<String> cells) {
        if (i < 0 || i >= rows || j < 0 || j >= cols || !grid[i][j]) {
            return "";
        }
```

```
        grid[i][j] = false;
        cells.add(i + "," + j);

        String left = findFigure(grid, i, j - 1, rows, cols, cells);
        String up = findFigure(grid, i - 1, j, rows, cols, cells);
        String right = findFigure(grid, i, j + 1, rows, cols, cells);
        String down = findFigure(grid, i + 1, j, rows, cols, cells);

        return "L" + left + "U" + up + "R" + right + "D" + down;
    }
}
```

Результат

```
Введите количество строк в листе бумаги: 4
Введите количество столбцов в листе бумаги: 4
Введите расположение закрашенных клеток:
1 0 0 1
1 1 0 0
0 0 1 1
0 0 0 1
Уникальные фигуры на клетчатом листе бумаги:
Фигура 1
Координаты закрашенных клеток: [2,2, 2,3, 3,3]
Фигура 2
Координаты закрашенных клеток: [0,0, 1,0, 1,1]
Фигура 3
Координаты закрашенных клеток: [0,3]
```

Задание 9

Дана матрица из целых чисел. Найти в ней прямоугольную подматрицу, состоящую из максимального количества одинаковых элементов. Использовать класс Stack.

Выполнение

```
package lab6;

import java.util.Stack;

public class Lab6_2_9 {
    public static void main(String[] args) {
        int[][] matrix = {
                {1, 1, 1, 0},
                {1, 0, 0, 1},
                {1, 1, 1, 1},
                {0, 1, 1, 0}
        };

        int[] result = findMaxRectangularSubmatrix(matrix);

        System.out.println("Max rectangular submatrix: ");
        for (int i = result[0]; i <= result[2]; i++) {
            for (int j = result[1]; j <= result[3]; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static int[] findMaxRectangularSubmatrix(int[][] matrix) {
        int numRows = matrix.length;
        int numCols = matrix[0].length;
        int[] result = new int[4]; // [rowStart, colStart, rowEnd, colEnd]
        int maxArea = 0;

        for (int i = 0; i < numRows; i++) {
            int[] heights = new int[numCols];
            for (int j = 0; j < numCols; j++) {
                heights[j] = matrix[i][j];
            }
            int[] subResult = findMaxRectangularSubarray(heights);
            int area = (subResult[2] - subResult[0] + 1) * (subResult[3] -
subResult[1] + 1);
            if (area > maxArea) {
                maxArea = area;
                result[0] = i - (subResult[2] - subResult[0]);
                result[1] = subResult[1];
                result[2] = i;
                result[3] = subResult[3];
            }
        }

        return result;
    }

    public static int[] findMaxRectangularSubarray(int[] heights) {
        int numCols = heights.length;
        Stack<Integer> stack = new Stack<>();
        int[] left = new int[numCols];
        int[] right = new int[numCols];

        for (int i = 0; i < numCols; i++) {
            while (!stack.empty() && heights[stack.peek()] >= heights[i]) {
                stack.pop();
            }
```

```
            left[i] = (stack.empty() ? 0 : stack.peek() + 1);
            stack.push(i);
        }

        stack.clear();

        for (int i = numCols - 1; i >= 0; i--) {
            while (!stack.empty() && heights[stack.peek()] >= heights[i]) {
                stack.pop();
            }
            right[i] = (stack.empty() ? numCols - 1 : stack.peek() - 1);
            stack.push(i);
        }

        int maxArea = 0;
        int[] result = new int[4]; // [rowStart, colStart, rowEnd, colEnd]

        for (int i = 0; i < numCols; i++) {
            int area = heights[i] * (right[i] - left[i] + 1);
            if (area > maxArea) {
                maxArea = area;
                result[0] = 0;
                result[1] = left[i];
                result[2] = heights[i] - 1;
                result[3] = right[i];
            }
        }

        return result;
    }
}
```

Результаты

```
Max rectangular submatrix:
1 1 1 1
```