



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 № 6

Вариант 18

Название: Коллекции в Java

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 коллекциями в Java.

Выполнение

Задание 1:

1. Ввести строки из файла, записать в список ArrayList. Выполнить сортировку строк, используя метод sort() из класса Collections.
2. Задана строка, состоящая из символов '(', ')', '[', ']', '{', '}'. Проверить правильность расстановки скобок. Использовать стек.

Листинг выполнения подзадачи 1

#n7.java

```
package com.aranei.var1;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.Scanner;

public class n7 {
    public void testV7() {
        try {
            File poem = new File("in/verse.txt");

            ArrayList<String> numbers = new ArrayList<>();

            Scanner text = new Scanner(poem, "UTF-8");

            while (text.hasNextLine()) {
                String line = text.nextLine();
                numbers.add(line);
            }
            numbers.sort(Comparator.naturalOrder());

        } catch (FileNotFoundException e) {
            System.out.println("Файл не найден");
            e.printStackTrace();
        }
    }
}
```

#n8.java

```
package com.aranei.var1;

import java.util.*;

public class n8 {
    public void testV80() {
        String str = "0{}[]{(())}";
        Stack<Character> stack;
        stack = new Stack<>();
        int flag = 0;
        char c;
        int count = 0;
        for (int i = 0; i < str.length(); i++) {
            c = str.charAt(i);
            count++;

            if (c == '(' || c == '{' || c == '[') {
                stack.push(c);
            }
        }
    }
}
```

```

        if (c == ')' || c == '}' || c == ']') {
            if (stack.isEmpty()) {
                System.out.println(count);
                flag = 1;
            }
            char top = stack.peek();
            if ((c == ')' && top == '(') || (c == '}' && top == '{') || (c == ']' && top == '[')) {
                stack.pop();
            } else {
                System.out.println(count);
                flag = 1;
            }
        }
    }
    if (flag == 0) {
        System.out.println("Success");
    }
}
}

```

Задание 2:

3. 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.
4. Дана матрица из целых чисел. Найти в ней прямоугольную подматрицу, состоящую из максимального количества одинаковых элементов. Использовать класс Stack.

Листинг выполнения подзадачи 2

#n82.java

```

package com.aranei.var2;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashSet;

public class n82 {
    public void testV82() {
        // Задаем поле из NxN клеток
        int[][] field = {
            {0, 0, 0, 0, 0, 0, 0, 0}, // 00000000
            {0, 1, 1, 0, 1, 0, 1, 0}, // 01101010
            {0, 1, 0, 0, 1, 0, 1, 0}, // 01001010
            {0, 0, 0, 0, 1, 0, 1, 0}, // 00001010
            {0, 0, 0, 0, 0, 0, 0, 0}, // 00000000
            {0, 1, 0, 0, 1, 1, 1, 0}, // 01001110
            {0, 1, 1, 0, 1, 1, 1, 0}, // 01101110
            {0, 0, 0, 0, 0, 0, 0, 0} // 00000000
        };
        System.out.println("Дано:");
        outArray(field);

        ArrayList<int[][]> figureList = new ArrayList<>();

        int[][] fieldTemplate = new int[field.length][field[0].length];
        clearArray(fieldTemplate);
    }
}

```

```

int show = 0;

HashSet<int[][]> figures = new HashSet<>();

for (int y = 0; y < field.length; y++) {
    for (int x = 0; x < field[y].length; x++) {
        if(field[y][x] == 1) {
            recursionCheck(x, y, fieldTemplate, field);
            if(show == 0){
                figureList.add(resizeArray(fieldTemplate));
//                outArray(figureList.get(show));
//                System.out.println();
                figures.add(figureList.get(show));
                show++;
            }
            if (checkRotate(figureList,resizeArray(fieldTemplate))){
                figureList.add(resizeArray(fieldTemplate));
                figures.add(figureList.get(show));
                show++;
            }
            clearArray(fieldTemplate);
        }
    }
}

System.out.println("Всего фигур: " + figures.size() + "\n");
for (int[][] f : figures) {
    outArray(f);
    System.out.println();
}

private void clearArray(int[][] array){
    for (int[] row: array)
        Arrays.fill(row, 0);
}

private void outArray(int[][] array){
    for (int[] ints : array) {
        System.out.println(Arrays.toString(ints));
    }
}

private void recursionCheck(int x, int y, int[][] figure, int[][] field){
    if (x>0 & x<field[0].length & y>0 & y<field.length)

    if (figure[y][x] != 1) {
        if(field[y][x] == 1) {
            figure[y][x] = field[y][x];
            recursionCheck(x+1, y, figure, field);
            recursionCheck(x, y+1, figure, field);
            recursionCheck(x-1, y, figure, field);
            recursionCheck(x, y-1, figure, field);
        }
    }
}

private int[][] resizeArray(int[][] array){
    //          xMax  yMax    xMin          yMin
    int[] minMax = {0, 0, array[0].length, array.length};

    for (int y = 0; y < array.length; y++) {
        for (int x = 0; x < array[y].length; x++) {
            if(array[y][x] == 1) {
                if(minMax[0] < x) minMax[0] = x;
                if(minMax[1] < y) minMax[1] = y;
                if(minMax[2] > x) minMax[2] = x;
                if(minMax[3] > y) minMax[3] = y;
            }
        }
    }

    //    outArray(array);
    //    System.out.println(Arrays.toString(minMax));
    int[][] result = new int[minMax[1] - minMax[3] + 1][minMax[0] - minMax[2] + 1];

    for (int y = minMax[3]; y < minMax[1] + 1; y++) {

```

```

        for (int x = minMax[2]; x < minMax[0] + 1; x++) {
            result[y-minMax[3]][x-minMax[2]] = array[y][x];
        }
    }
    //    outArray(result);
    return result;
}

private boolean checkRotate(ArrayList<int[][]> list, int[][] array){
    boolean add = true;
    for (int[][] ints : list) {
        if (ints.length == array.length | ints.length == array[0].length |
            ints[0].length == array.length | ints[0].length == array[0].length) {
            if (Arrays.deepEquals(ints, array)) add = false;
            if (Arrays.deepEquals(ints, arrayRotate(array))) add = false;
            if (Arrays.deepEquals(ints, arrayRotate(arrayRotate(array)))) add = false;
            if (Arrays.deepEquals(ints, arrayRotate(arrayRotate(arrayRotate(array)))) add = false;
        }
    }
    add = false;
}

//    System.out.println(add);
return add;
}

private int[][] arrayRotate(int[][] array){
    int[][] resultArray = new int[array[0].length][array.length];
    for (int i = 0; i < array.length; i++) {
        for (int j = 0; j < array[i].length; j++) {
            resultArray[j][array.length - i - 1] = array[i][j];
        }
    }
    return resultArray;
}
}

```

#n92.java

```

package com.aranei.var2;

import java.util.Arrays;
import java.util.Stack;

public class n92 {
    public void testN92() {
        int n = 5;
        int[][] field = new int[n][n];
        int[][] fieldTemplate = new int[field.length][field[0].length];
        clearArray(fieldTemplate);

        Stack<int[][]> stacks = new Stack<>();
        for (int i = 0; i != n; i++) {
            for (int j = 0; j != n; j++) {
                field[i][j] = (int) (Math.random() * 3 + 1);
                System.out.print(field[i][j] + " ");
            }
            System.out.println();
        }

        for (int i = 0; i != n; i++) {
            for (int j = 0; j != n; j++) {
                stacks.push(arrayCheck(i, j, field));
            }
        }

        int[][] maxArray = new int[1][1];
        while (!stacks.isEmpty()){
            int[][] local = stacks.pop();
            if ((maxArray.length + 1) * (maxArray[0].length + 1) < (local.length + 1) *
                (local[0].length + 1)) {
                maxArray = local;
            }
            //            System.out.println(Arrays.toString(s));
        }
        outArray(maxArray);
    }
}

```

```

private void clearArray(int[][] array){
    for (int[] row: array)
        Arrays.fill(row, 0);
}

private int[][] arrayCheck(int x, int y, int[][] field) {
    int ySize = 0;
    int xSize = 0;
    int[] s = {0, 0, 0};

    boolean check = true;

    for (int j = y; j < field.length; j++){
        for (int i = x; i < field[0].length; i++) {
            for (int j2 = y; j2 <= j; j2++) {
                if (field[j2][i] != field[y][x]) {
                    check = false;
                    j2 = j;
                }
            }
            if (check & ((xSize + 1) * (ySize + 1)) > s[2]) {
                s[0] = xSize + 1;
                s[1] = ySize + 1;
                s[2] = (xSize + 1) * (ySize + 1);
                System.out.println(Arrays.toString(s));
            }
            ySize++;
        }
        xSize++;
        ySize = 0;
    }
    check = true;
    xSize = 0;
}

//      System.out.println(Arrays.toString(s));
int[][] result = new int[s[1]][s[0]];
for (int i = 0; i < result.length; i++) {
    for (int j = 0; j < result[0].length; j++) {
        result[i][j] = field[y + i][x + j];
    }
}

return result;
}

private void outArray(int[][] array){
    for (int[] ints : array) {
        System.out.println(Arrays.toString(ints));
    }
    System.out.println();
}
}

```

Ссылка на программное решение

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Aranei99/labsJava/tree/main/LB6>

Вывод

При выполнении лабораторной работы были получены навыки работы с коллекциями в Java.