



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
НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**
МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных.**

Отчет

по лабораторной работе №6

Дисциплина: Языки программирования для работы с большими данными.

Студент

ИУ6-23М

(Группа)

В.А Антонов

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

П.В. Степанов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

Лабораторная работа №6

Задание: Вариант 1

3. С использованием множества выполнить попарное суммирование произвольного конечного ряда чисел по следующим правилам: на первом этапе суммируются попарно рядом стоящие числа, на втором этапе суммируются результаты первого этапа и т.д. до тех пор, пока не останется одно число.

Ход работы: Код программы файла Element

```
public class Element {
    private int row;
    private int col;
    private int meaning;

    public Element(int row, int col, int meaning) {
        this.row = row;
        this.col = col;
        this.meaning = meaning;
    }

    public int getRow() {
        return row;
    }

    public void setRow(int row) {
        this.row = row;
    }

    public int getCol() {
        return col;
    }

    public void setCol(int col) {
        this.col = col;
    }

    public int getMeaning() {
        return meaning;
    }

    public void setMeaning(int meaning) {
        this.meaning = meaning;
    }

    @Override
    public String toString() {
        return "Element{" +
            "row=" + row +
            ", col=" + col +
            ", meaning=" + meaning +
            '}';
    }
}
```

Код программы файла Task_1_3

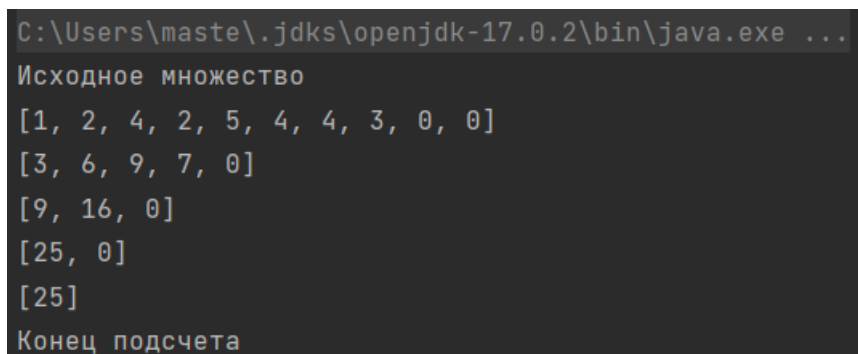
```
import java.util.Arrays;
import java.util.Random;

public class Task_1_3 {
    public static void main(String[] args) {
        int[] arr = generate(10, 5);
        System.out.println("Исходное множество");
        while (arr.length > 1) {
            System.out.println(Arrays.toString(arr));
            arr = modify(arr);
        }
        System.out.println(Arrays.toString(arr));
        System.out.println("Конец подсчета");
    }

    public static int[] generate(int count, int maxNumber) {
        int[] arr = new int[count];
        Random rand = new Random();
        for (int i = 0; i < count; i++) {
            arr[i] = rand.nextInt(maxNumber + 1);
        }
        return arr;
    }

    public static int[] modify(int[] arr) {
        int newSize = (int) Math.round(arr.length / 2.0);
        int[] arrAfterSum = new int[newSize];
        for (int i = 0; i < arr.length; i++) {
            arrAfterSum[i / 2] += arr[i];
        }

        return arrAfterSum;
    }
}
```



```
C:\Users\maste\.jdk\openjdk-17.0.2\bin\java.exe ...
Исходное множество
[1, 2, 4, 2, 5, 4, 4, 3, 0, 0]
[3, 6, 9, 7, 0]
[9, 16, 0]
[25, 0]
[25]
Конец подсчета
```

Рисунок 1. Результат выполнения программы

4. Сложить два многочлена заданной степени, если коэффициенты многочленов хранятся в объекте HashMap.

Ход работы: Код программы файла Task_1_4

```

import java.util.HashMap;
import java.util.Scanner;
public class Task_1_4 {
    public static void main(String[] args)
    {
        HashMap<Integer, Integer> d=new HashMap<>();
        Scanner f=new Scanner(System.in);
        System.out.print("Степень многочлена n= ");
        int n=f.nextInt();
        System.out.print("D(x)= c0 ");
        for(int i=1; i<=n; i++)
        {
            System.out.print(" + c"+i+"*x^"+i);
        }
        System.out.println();
        for(int i=0; i<=n; i++)
        {
            System.out.print("c"+i+"=");
            d.put(i, f.nextInt());
        }
        System.out.print("D(x)= "+d.get(0));
        for(int i=1; i<=n; i++){
            System.out.print(" + "+d.get(i)+"x^"+i);
        }
        System.out.println();
        HashMap<Integer, Integer> e=new HashMap<>();
        System.out.print("E(x)= a0 ");
        for(int i=1; i<=n; i++)
        {
            System.out.print(" + a"+i+"*x^"+i);
        }
        System.out.println();
        for(int i=0; i<=n; i++)
        {
            System.out.print("a"+i+"=");
            e.put(i, f.nextInt());
        }
        System.out.print("E(x)= "+e.get(0));
        for(int i=1; i<=n; i++)
        {
            System.out.print(" + "+e.get(i)+"x^"+i);
        }
        System.out.println();
        HashMap<Integer, Integer> h=new HashMap<>();
        h.put(0, d.get(0)+e.get(0));
        System.out.print("D(x)+E(x)= "+h.get(0));
        for(int i=1; i<=n; i++)
        {
            h.put(i, d.get(i)+e.get(i));
            System.out.print(" + "+h.get(i)+"x^"+i);
        }
    }
}

```

```

C:\Users\maste\.jdk\openjdk-17.0.2\bin\java.exe ...
Степень многочлена n= 3
D(x)= c0 + c1*x^1 + c2*x^2 + c3*x^3
c0=1
c1=2
c2=3
c3=4
D(x)= 1 + 2x^1 + 3x^2 + 4x^3
E(x)= a0 + a1*x^1 + a2*x^2 + a3*x^3
a0=5
a1=6
a2=7
a3=3
E(x)= 5 + 6x^1 + 7x^2 + 3x^3
D(x)+E(x)= 6 + 8x^1 + 10x^2 + 7x^3

```

Рисунок 2. Результат выполнения программы

Задание: Вариант 2

3. Во входном файле хранятся две разреженные матрицы A и B. Построить циклически связанные списки SA и SB, содержащие ненулевые элементы соответственно матриц A и B. Просматривая списки, вычислить: а) сумму $S = A + B$; б) произведение $P = A * B$.

Ход работы: Файл Matrix

```

0 1 0
0 2 0
1 0 0

1 0 0
0 0 1
3 0 1

```

Код программы файла Task_2_3

```

import java.io.File;
import java.io.FileNotFoundException;
import java.util.LinkedList;
import java.util.Scanner;
public class Task_2_3 {
    public static void main(String[] args) {

        File inp_file = new
File("C:\\Users\\maste\\IdeaProjects\\Lab6\\src\\Matrix.txt");

        Scanner scanner = null;

```

```

try {
    scanner = new Scanner(inp_file);
} catch (FileNotFoundException e) {
    e.printStackTrace();
}
boolean next_matrix = false;
int row_num = 1;
int col_num = 1;
LinkedList<Element> CA = new LinkedList<>();
LinkedList<Element> CB = new LinkedList<>();

while(scanner.hasNextLine()) {
    String line = scanner.nextLine();
    String[] nums = line.split(" ");
    if (nums.length < 2) {
        next_matrix = true;
        row_num = 1;
    } else {
        col_num = 1;
        for (String num : nums) {
            if(Integer.parseInt(num) != 0) {
                if (!next_matrix) {
                    CA.addLast(new Element(row_num, col_num,
Integer.parseInt(num)));
                } else {
                    CB.addLast(new Element(row_num, col_num,
Integer.parseInt(num)));
                }
                col_num++;
            }
            row_num++;
        }
    }
    System.out.println("_____Список CA_____");
    System.out.println(CA);
    System.out.println("_____Список CB_____");
    System.out.println(CB);

    //Сумма

    LinkedList<Element> sum_matrix = new LinkedList<>();
    int ind_a = 0;
    int ind_b = 0;
    boolean sum_flag = true;

    while(sum_flag) {
        if((CA.get(ind_a).getRow() == CB.get(ind_b).getRow()) &&
            (CA.get(ind_a).getCol() == CB.get(ind_b).getCol())) {
            sum_matrix.addLast(new Element(CA.get(ind_a).getRow(),
CA.get(ind_a).getCol(), CA.get(ind_a).getMeaning() +
CB.get(ind_b).getMeaning()));
            ind_a++;
            ind_b++;
        } else {
            if (CA.get(ind_a).getRow() == CB.get(ind_b).getRow()) {
                if (CA.get(ind_a).getCol() > CB.get(ind_b).getCol()) {
                    sum_matrix.addLast(new
Element(CB.get(ind_b).getRow(), CB.get(ind_b).getCol(),
CB.get(ind_b).getMeaning()));
                    ind_b++;
                } else {
                    sum_matrix.addLast(new
Element(CA.get(ind_a).getRow(), CA.get(ind_a).getCol(),

```

```

CA.get(ind_a).getMeaning()));
        ind_a++;
    }
    } else {
        if (CA.get(ind_a).getRow() > CB.get(ind_b).getRow()) {
            sum_matrix.addLast(new
Element(CB.get(ind_b).getRow(), CB.get(ind_b).getCol(),
CB.get(ind_b).getMeaning()));
            ind_b++;
        } else {
            sum_matrix.addLast(new
Element(CA.get(ind_a).getRow(), CA.get(ind_a).getCol(),
CA.get(ind_a).getMeaning()));
            ind_a++;
        }
    }
}
if(ind_a == CA.size()) {
    sum_flag = false;
    for (int i = ind_b; i < CB.size(); i++) {
        sum_matrix.addLast(new Element(CB.get(i).getRow(),
CB.get(i).getCol(), CB.get(i).getMeaning()));
    }
} else if(ind_b == CB.size()){
    sum_flag = false;
    for (int i = ind_a; i < CA.size(); i++) {
        sum_matrix.addLast(new Element(CA.get(i).getRow(),
CA.get(i).getCol(), CA.get(i).getMeaning()));
    }
}

}
System.out.println("_____Сумма_____");
System.out.println(sum_matrix);

//Умножение( A[i, k] * B[k, j])
LinkedList<Element> mul_matrix = new LinkedList<>();
int answer = 0;

for (int i = 1; i < 4; i++) {
    for (int j = 1; j < 4; j++) {
        answer = 0;
        for (int k = 1; k < 4; k++) {

            for (int l = 0; l < CA.size(); l++) {
                if((CA.get(l).getRow() == i) && (CA.get(l).getCol()
== k)){

                    for (int m = 0; m < CB.size(); m++) {
                        if(CB.get(m).getRow() == k &&
CB.get(m).getCol() == j){
                            answer += CA.get(l).getMeaning() *
CB.get(m).getMeaning();
                            break;
                        }
                    }
                    break;
                }
            }
        }
    }
    if(answer != 0) {
        mul_matrix.addLast(new Element(i, j, answer));
    }
}

```

```

    }
    System.out.println("_____Умножение_____");
    System.out.println(mul_matrix);
}
}

```

```

C:\Users\maste\jdk\openjdk-17.0.2\bin\java.exe ...
-----Список СА-----
[Element{row=1, col=2, meaning=1}, Element{row=2, col=2, meaning=2}, Element{row=3, col=1, meaning=1}]
-----Список СВ-----
[Element{row=1, col=1, meaning=1}, Element{row=2, col=3, meaning=1}, Element{row=3, col=1, meaning=3},
-----Сумма-----
[Element{row=1, col=1, meaning=1}, Element{row=1, col=2, meaning=1}, Element{row=2, col=2, meaning=2},
-----Умножение-----
[Element{row=1, col=3, meaning=1}, Element{row=2, col=3, meaning=2}, Element{row=3, col=1, meaning=1}]

```

Рисунок 3. Результат выполнения программы

Во входном файле хранятся наименования некоторых объектов. Построить список С1, элементы которого содержат наименования и шифры данных объектов, причем элементы списка должны быть упорядочены по возрастанию шифров. Затем “сжать” список С1, удаляя дублирующие наименования объектов.

Ход работы: Код программы файла Obj

```

public class Obj implements Comparable<Obj>{
    private String name;
    private int code;

    public Obj() {
    }

    public Obj( String name, int code) {
        this.name = name;
        this.code = code;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }

    public int getCode() {
        return code;
    }
    public void setCode(String surname) {
        this.code = code;
    }

    @Override
    public String toString() {

```



```

        return "Objects{" +
            ", Наименование:" + name + '\'' +
            ", Шифр:" + code + '\'' +
            '}';
    }

    @Override
    public int compareTo(Obj o) {
        return this.getCode() - o.getCode();
    }
}

```

Код программы файла Obj

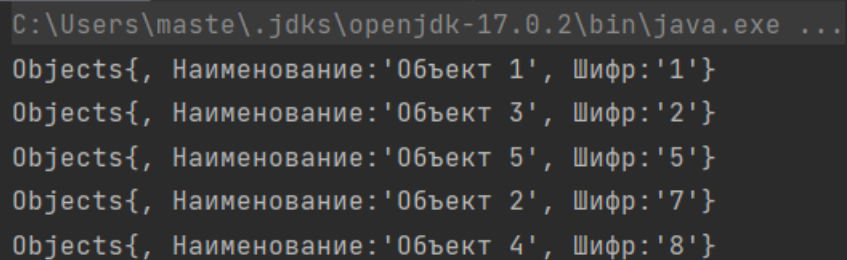
```

import java.util.ArrayList;
import java.util.TreeSet;

public class Task_2_4<p1> {
    public static void main(String[] args) {
        ArrayList<Obj> sour = new ArrayList<>();
        Obj p1 = new Obj("Объект 1", 1);
        Obj p2 = new Obj("Объект 2", 7);
        Obj p3 = new Obj("Объект 3", 2);
        Obj p4 = new Obj("Объект 4", 8);
        Obj p5 = new Obj("Объект 5", 5);
        sour.add(p1);
        sour.add(p2);
        sour.add(p3);
        sour.add(p4);
        sour.add(p5);

        TreeSet<Obj> Total = new TreeSet<>(sour);
        for(Obj o: Total)
            System.out.println(o);
    }
}

```



```

C:\Users\maste\.jdk\openjdk-17.0.2\bin\java.exe ...
Objects{, Наименование: 'Объект 1', Шифр: '1'}
Objects{, Наименование: 'Объект 3', Шифр: '2'}
Objects{, Наименование: 'Объект 5', Шифр: '5'}
Objects{, Наименование: 'Объект 2', Шифр: '7'}
Objects{, Наименование: 'Объект 4', Шифр: '8'}

```

Рисунок 4. Результат выполнения программы

Вывод: лабораторная работа была выполнена в соответствии с заданием и полученные верные результаты работ программ

