|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 09.04.01/07 **Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |
| --- |
| **по лабораторной работе №6** |

**Название:**

Коллекции

**Дисциплина:** Языка программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | Д.Д. Капитонов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Вариант 1, номера 9 и 1.**

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

1. Определить множество на основе множества целых чисел. Создать методы для определения пересечения и объединения множеств.

**Код программы:**

*package dan.lab;*

*//1. Определить множество на основе множества целых чисел.*

*// Создать методы для определения пересечения и объединения множеств.*

*import java.io.\*;*

*import java.util.\*;*

*public class Lab\_9\_1\_1 {*

*public static boolean check\_in(ArrayList<Integer> m1, ArrayList<Integer> m2){*

*return m1.get(m1.size()-1) >= m2.get(0) || m1.get(0) <= m2.get(m2.size()-1);*

*}*

*public static ArrayList<Integer> concat(ArrayList<Integer> m1, ArrayList<Integer> m2){*

*System.out.println("Вариант 1 №1");*

*ArrayList<Integer> ret = new ArrayList<>();*

*int max= Math.max(m1.size(), m2.size());*

*System.out.println(max);*

*for (int i =0; i!= max; i++){*

*if (m1.size() > i && m2.size() > i){*

*if (m1.get(i) == m2.get(i)){*

*ret.add(m1.get(i));*

*} else{*

*ret.add(m1.get(i));*

*ret.add(m2.get(i));*

*}*

*} else{*

*if (m1.size() > i){*

*ret.add(m1.get(i));*

*} else {*

*ret.add(m2.get(i));*

*}*

*}*

*}*

*return ret;*

*}*

*public static void main(String[] args) throws IOException {*

*ArrayList<ArrayList<Integer>> mn\_vo = new ArrayList<>();*

*for (int j = 0; j!=6; j++){*

*ArrayList<Integer> cash = new ArrayList<>();*

*int i\_1 = (int) (Math.random() \* 10 + 1);*

*for (int i =0; i!=i\_1; i++){*

*cash.add(3\*(i\_1 - i));*

*}*

*Collections.sort(cash);*

*mn\_vo.add(cash);*

*}*

*System.out.println(mn\_vo);*

*System.out.println(check\_in(mn\_vo.get(1), mn\_vo.get(2)));*

*System.out.println(concat(mn\_vo.get(1), mn\_vo.get(2)));*

*System.out.println("Вариант 1 №9");*

*File file = new File("eng\_test.txt");*

*BufferedReader br = new BufferedReader(new FileReader(file));*

*String line = "";*

*HashSet<String> str = new HashSet<>();*

*while ((line = br.readLine())!= null){*

*for (String word:line.split(" ")){*

*if (!str.contains(word.toLowerCase())){*

*str.add(word.replace(".", "")*

*.replace(",","")*

*.replace(";", "").toLowerCase());*

*}*

*}*

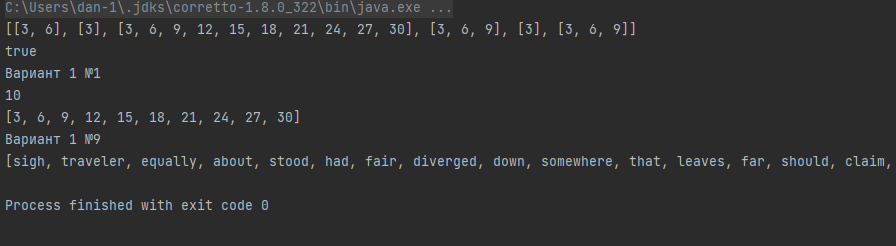
*}*

*System.out.println(str);*

*}*

*}*

**Результат выполнения программы:**

****

**Вариант 2, номера 9 и 10.**

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

10. На прямой гоночной трассе стоит N автомобилей, для каждого из которых известны начальное положение и скорость. Определить, сколько произойдет обгонов.

**Код программы:**

**package dan.lab;**

**//**

**//9. Дана матрица из целых чисел. Найти в ней прямоугольную подматрицу,**

**// состоящую из максимального количества одинаковых элементов. Использовать класс Stack.**

**//**

**// 10. На прямой гоночной трассе стоит N автомобилей, для каждого из которых**

**// известны начальное положение и скорость. Определить, сколько произойдет обгонов.**

**//**

**import com.sun.org.apache.xalan.internal.xsltc.trax.SmartTransformerFactoryImpl;**

**import java.util.\*;**

**public class Lab9\_10\_2 {**

**public static void main(String[] args) {**

**System.out.println("Вариант 2№9");**

**int n = 5;**

**HashSet<Integer> num\_list = new HashSet<>();**

**int[][] mass = new int[n][n];**

**Stack<Integer> matr\_stack = new Stack<>();**

**for (int i = 0; i != n; i++) {**

**for (int j = 0; j != n; j++) {**

**mass[i][j] = (int) (Math.random() \* 3 + 1);**

**System.out.print(mass[i][j] + " ");**

**matr\_stack.push(mass[i][j]);**

**num\_list.add(mass[i][j]);**

**}**

**System.out.println("");**

**}**

**System.out.println("Num\_list = " + num\_list);**

**System.out.println("Matr\_stack = " + matr\_stack);**

**System.out.println(matr\_stack.search(3));**

**int[] max\_p\_i\_j\_c = new int[3];**

**max\_p\_i\_j\_c[0] = 0;**

**max\_p\_i\_j\_c[1] = 0;**

**max\_p\_i\_j\_c[2] = 0;**

**for (int num\_add\_i = 0; num\_add\_i != n + 1; num\_add\_i++) {**

**for (int num\_add\_j = 0; num\_add\_j != n + 1; num\_add\_j++) {**

**int[] mass\_numbers\_c = new int[3];**

**if (num\_add\_i != n || num\_add\_j != n) {**

**for (int i = 0; i != num\_add\_i; i++) {**

**for (int j = 0; j != num\_add\_j; j++) {**

**switch (mass[i][j]) {**

**case 1:**

**mass\_numbers\_c[0]++;**

**break;**

**case 2:**

**mass\_numbers\_c[1]++;**

**break;**

**case 3:**

**mass\_numbers\_c[2]++;**

**break;**

**}**

**System.out.print(mass[i][j] + " ");**

**}**

**System.out.println("");**

**}**

**System.out.println(num\_add\_i + " " + num\_add\_j);**

**System.out.println(Arrays.toString(mass\_numbers\_c));**

**int max = Math.max(Math.max(mass\_numbers\_c[0], mass\_numbers\_c[1]), Math.max(mass\_numbers\_c[1], mass\_numbers\_c[2]));**

**if (max > max\_p\_i\_j\_c[2]) {**

**max\_p\_i\_j\_c[2] = max;**

**max\_p\_i\_j\_c[0] = num\_add\_i;**

**max\_p\_i\_j\_c[1] = num\_add\_j;**

**}**

**}**

**}**

**}**

**System.out.println(Arrays.toString(max\_p\_i\_j\_c));**

**for (int i = 0; i != max\_p\_i\_j\_c[0]; i++) {**

**for (int j = 0; j != max\_p\_i\_j\_c[1]; j++) {**

**System.out.print(mass[i][j] + " ");**

**}**

**System.out.println("");**

**}**

**System.out.println("Вариант 2№10");**

**int road\_lenght = (int)(Math.random()\*2000+1000);**

**System.out.println("Длинна трассы = "+ road\_lenght);**

**ArrayList<Car> car\_mass = new ArrayList<>();**

**for (int i=0; i!=n; i++){**

**car\_mass.add(new Car((int)(Math.random()\*16+100),(int)(Math.random()\*2+10),(int)(Math.random()\*2+20), road\_lenght));**

**}**

**System.out.println(car\_mass);**

**int count =0;**

**for (int i=0; i!=n; i++){**

**for (int j=0; j!=n; j++){**

**if (car\_mass.get(i).time\_for\_road > car\_mass.get(j).time\_for\_road){**

**count++;**

**}**

**}**

**}**

**System.out.println("Кол-во обгонов = "+ count);**

**}**

**public static class Car{**

**int speed;**

**int length;**

**int leng\_from\_start;**

**double time\_for\_road;**

**public Car(int speed, int length, int leng\_from\_start, int road\_lenght) {**

**this.speed = speed;**

**this.length = length;**

**this.leng\_from\_start = length + leng\_from\_start;**

**this.time\_for\_road = time\_count(this.speed, road\_lenght);**

**}**

**public double time\_count(int speed, int road\_lenght){**

**return (double) road\_lenght/ (double)speed;**

**}**

**@Override**

**public String toString() {**

**return "Car{" +**

**"speed=" + speed +**

**", length=" + length +**

**", leng\_from\_start=" + leng\_from\_start +**

**", time\_for\_road=" + time\_for\_road +**

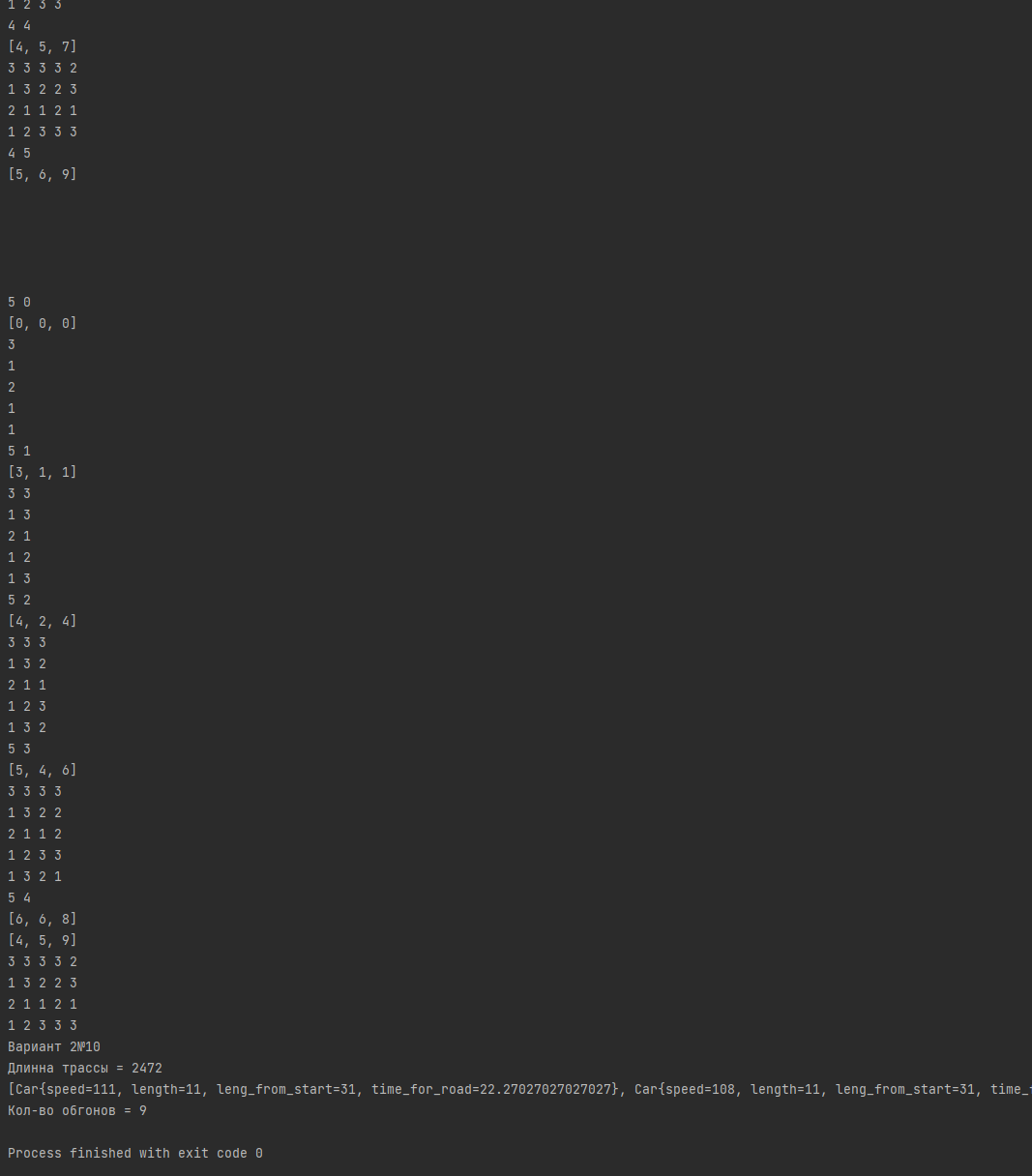
**'}';**

**}**

**}**

**}**

**Результат выполнения программы:**

****

**Вывод:** научились работать с коллекциями.