|  |  |
| --- | --- |
| 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 №5** |

**Название:**

Исключения и файлы

**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 и 2, номер 1.**

Выполнить задания на основе варианта 1 лабораторной работы 4, контролируя состояние потоков ввода/вывода. При возникновении ошибок, связанных с корректностью выполнения математических операций, генерировать и обрабатывать исключительные ситуации. Предусмотреть обработку исключений, возникающих при нехватке памяти, отсутствии требуемой записи (объекта) в файле, недопустимом значении поля и т.д.

Выполнить задания из варианта 2 лабораторной работы 4, реализуя собственные обработчики исключений и исключения ввода/вывода.

**Код программы:**

*package daniil.lab;*

*import java.io.IOException;*

*import java.util.ArrayList;*

*import java.util.Scanner;*

*import java.util.SortedMap;*

*//1. Создать класс City (город)*

*// с внутренним классом, с помощью объектов которого можно хранить информацию о проспектах,*

*// улицах, площадях.*

*public class Lab5\_1\_2 {*

*public static class City{*

*String name;*

*ArrayList<Street> street\_list;*

*public City() {*

*street\_list = new ArrayList<>();*

*}*

*public City(String name) {*

*this.name = name;*

*street\_list = new ArrayList<>();*

*}*

*public void new\_street(String name, boolean square, boolean prospect, boolean street){*

*Street cash = new Street(name, square, prospect, street);*

*if (cash.name.equals("error")){*

*System.out.println("Новая улица не была добавлена");*

*}else{*

*street\_list.add(cash);*

*}*

*}*

*public String delenie(int n1, int n2){*

*try{*

*return Float.toString(n1/n2);*

*} catch (Exception er){*

*return "Воникла ошибка - "+ er.getMessage();*

*}*

*}*

*public static String check\_null(City c){*

*return "Объект пустой? - "+Boolean.toString(c==null);*

*}*

*@Override*

*public String toString() {*

*return "City{" +*

*"name='" + name + '\'' +*

*", street\_list=" + street\_list +*

*'}';*

*}*

*public static class Street{*

*String name;*

*boolean square;*

*boolean prospect;*

*boolean street;*

*public Street(String name, boolean square, boolean prospect, boolean street) {*

*if (!square && !prospect && !street){*

*System.out.println("Ошибка ввода названия улицы. Не выбран ни один её тип.");*

*this.name = "error";*

*} else{*

*this.name = name;*

*this.square = square;*

*this.prospect = prospect;*

*this.street = street;*

*}*

*}*

*public Street() {*

*}*

*@Override*

*public String toString() {*

*return "Street{" +*

*"name='" + name + '\'' +*

*", square=" + square +*

*", prospect=" + prospect +*

*", street=" + street +*

*'}';*

*}*

*}*

*}*

*interface Furniture{*

*public String getname();*

*}*

*public static abstract class Wardrobe implements Furniture{*

*String name;*

*int cost;*

*public String getname(){*

*return this.name;*

*}*

*public Wardrobe() {*

*}*

*public Wardrobe(String name, int cost) {*

*if (name.equals("") || cost == 0){*

*System.out.println("Введеные пустые или нулевые значения книги");*

*} else {*

*this.name = name;*

*this.cost = cost;*

*}*

*}*

*public void test(){*

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

*}*

*@Override*

*public String toString() {*

*return "Wardrobe{" +*

*"name='" + name + '\'' +*

*", cost=" + cost +*

*'}';*

*}*

*}*

*public static class Book\_war extends Wardrobe{*

*@Override*

*public void test(){*

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

*}*

*public Book\_war(String name, int cost) {*

*super(name, cost);*

*}*

*public Book\_war() {*

*}*

*}*

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

*City SPB = new City("Piter");*

*SPB.new\_street("Войковская", true, false, false);*

*SPB.new\_street("Чеборская", false, false, false);*

*System.out.println(SPB.toString());*

*System.out.println(City.check\_null(SPB));*

*System.out.println(SPB.delenie(1,0));*

*System.out.println(SPB.delenie(1,1));*

*Book\_war ward = new Book\_war("Onegin", 12);*

*System.out.println(ward.toString());*

*Scanner in = new Scanner(System.in);*

*System.out.println("Введите название книги и её стоимость");*

*try{*

*System.out.print("Название - ");*

*String name = in.nextLine();*

*System.out.print("Цена - ");*

*int cost = in.nextInt();*

*Book\_war t\_ward = new Book\_war(name, cost);*

*System.out.println(t\_ward.toString());*

*} catch (Exception er){*

*System.out.println("Введены не корректные данные - "+er.getMessage());*

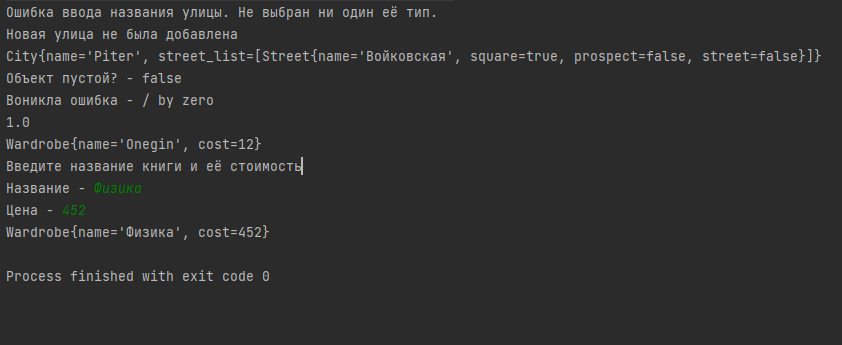
*Book\_war t\_ward = new Book\_war();*

*}*

*}*

*}*

**Результат выполнения программы:**

****

**Вариант 3 и 4, номера 9 и 10.**

9. Входной файл содержит совокупность строк. Строка файла содержит строку квадратной матрицы. Ввести матрицу в двумерный массив (размер матрицы найти). Вывести исходную матрицу и результат ее транспонирования.

10. Входной файл хранит квадратную матрицу по принципу: строка представляет собой число. Определить размерность. Построить 2-мерный массив, содержащий матрицу. Вывести исходную матрицу и результат ее поворота на 90 градусов по часовой стрелке.

9. Прочитать строки из файла и поменять местами первое и последнее слова в каждой строке.

10. Ввести из текстового файла, связанного с входным потоком, последовательность строк. Выбрать и сохранить m последних слов в каждой из последних n строк

**Код программы:**

*package daniil.lab;*

*import java.io.\*;*

*import java.util.ArrayList;*

*import java.util.Arrays;*

*import java.util.Scanner;*

*import java.util.stream.Collectors;*

*import java.util.stream.IntStream;*

*public class Lab5\_3\_4 {*

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

*File file = new File("test.txt");*

*FileOutputStream fout = new FileOutputStream(file);*

*String greetings = "1 2 3 4 5 6\r\n7 8 9 1 2 3\n4 5 6 7 8 9\n";*

*fout.write(greetings.getBytes());*

*fout.close();*

*FileInputStream fileInputStream = new FileInputStream(file);*

*String zap = "";*

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

*ArrayList<ArrayList<String>> mass\_str = new ArrayList<>();*

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

*String line = "";*

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

*ArrayList<String> mass\_str\_c = new ArrayList<>();*

*mass.add(IntStream.of(Arrays.stream(line.split(" ")).mapToInt(Integer::valueOf).toArray()).boxed().collect(Collectors.toCollection(ArrayList::new)));*

*mass\_str\_c.addAll(Arrays.asList(line.split(" ")));*

*mass\_str.add(mass\_str\_c);*

*}*

*ArrayList<ArrayList<Integer>> mass\_t = new ArrayList<>();*

*for (int i=0; i!=mass.get(0).size(); i++){*

*ArrayList<Integer> mass\_1 = new ArrayList<>();*

*for (int j=0; j!=mass.size(); j++){*

*mass\_1.add(mass.get(j).get(i));*

*}*

*mass\_t.add(mass\_1);*

*}*

*System.out.println(mass);*

*System.out.println(mass\_t);*

*System.out.println("Размерность x - "+mass.size()+" y - "+mass.get(0).size());*

*ArrayList<ArrayList<Integer>> mass\_90 = new ArrayList<>();*

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

*for (int i=0; i!=mass\_t.size(); i++){*

*ArrayList<Integer> mass\_1 = new ArrayList<>();*

*for (int j=mass\_t.get(0).size()-1;j>=0; j--){*

*mass\_1.add(mass\_t.get(i).get(j));*

*}*

*mass\_90.add(mass\_1);*

*}*

*System.out.println(mass\_90);*

*String cash = "";*

*String new\_str\_file = "";*

*for (int i=0; i!=mass\_str.size(); i++){*

*cash = mass\_str.get(i).get(mass\_str.get(i).size()-1);*

*mass\_str.get(i).set(mass\_str.get(i).size()-1,mass\_str.get(i).get(0));*

*mass\_str.get(i).set(0, cash);*

*new\_str\_file+= String.join(" ", mass\_str.get(i));*

*new\_str\_file+="\n";*

*}*

*System.out.println(new\_str\_file);*

*File file\_n = new File("new\_f.txt");*

*FileOutputStream new\_f = new FileOutputStream(file\_n);*

*new\_f.write(new\_str\_file.getBytes());*

*new\_f.close();*

*Scanner in = new Scanner(System.in);*

*System.out.println(mass\_str);*

*System.out.print("Введите m (слов) - ");*

*int m = in.nextInt();*

*System.out.print("Введите n (строк) - ");*

*int n = in.nextInt();*

*ArrayList<ArrayList<String>> mass\_str\_l\_nm = new ArrayList<>();*

*for (int i=mass\_str.size()-n; i!=mass\_str.size(); i++){*

*ArrayList<String> mass\_s = new ArrayList<>();*

*for (int j=mass\_str.get(i).size()-m; j!=mass\_str.get(i).size(); j++){*

*mass\_s.add(mass\_str.get(i).get(j));*

*}*

*mass\_str\_l\_nm.add(mass\_s);*

*}*

*String str\_nm = "";*

*for (int i=0; i!=mass\_str\_l\_nm.size();i++){*

*str\_nm+= String.join(" ", mass\_str\_l\_nm.get(i));*

*str\_nm+="\n";*

*}*

*System.out.println(mass\_str\_l\_nm);*

*File file\_n\_nm = new File("new\_f\_nm.txt");*

*FileOutputStream new\_f\_nm = new FileOutputStream(file\_n\_nm);*

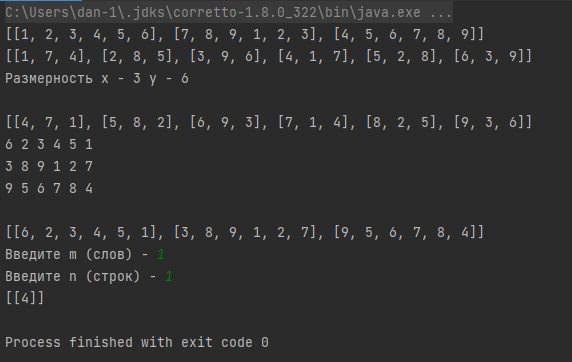
*new\_f\_nm.write(str\_nm.getBytes());*

*new\_f\_nm.close();*

*}*

*}*

**Результат выполнения программы:**

****

**Вывод:** научились работать с исключениями и файлами.