|  |  |
| --- | --- |
|  | **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 **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе №** 5

**Название:** Исключения. Файлы

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

Студент \_\_\_ИУ6-23М\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**А.Д. Зеленский**\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**П. В. Степанов**\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

**Цель:** ознакомиться с базовыми принципами языка Java для работы с большими данными.

**Вариант 1:**

Выполнить задания на основе варианта 1 лабораторной работы 3, 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.

1. Определить класс Комплекс. Класс должен содержать несколько конструкторов. Реализовать методы для сложения, вычитания, умножения, деления, присваивания комплексных чисел. Создать два вектора размерности n из комплексных координат. Передать их в метод, который выполнит их сложение.
2. Определить класс Квадратное уравнение. Класс должен содержать несколько конструкторов. Реализовать методы для поиска корней, экстремумов, а также интервалов убывания/возрастания. Создать массив объектов и определить наибольшие и наименьшие по значению корни.

**Решение:**

Ниже представлен код класса первой программы, а на рисунке 1 – результат работы программы.

public Complex Div( Complex cB ) throws ArithmeticException{  
 Complex div = new Complex();  
 double dR, dDen;  
  
 if(Math.abs( cB.a ) >= Math.abs( cB.bi )) {  
 dR = cB.bi/cB.a;  
 dDen = cB.a + dR\*cB.bi;  
 if (dDen == Float.POSITIVE\_INFINITY || dDen == Float.NEGATIVE\_INFINITY) {  
 throw new ArithmeticException("Обнаружено деление на ноль");  
 }  
 div.a = (a + dR\*bi)/dDen;  
 div.bi = (bi - dR\*a)/dDen;  
 } else {  
 dR = cB.a/cB.bi;  
 if (dR == Float.POSITIVE\_INFINITY || dR == Float.NEGATIVE\_INFINITY) {  
 throw new ArithmeticException("Деление на ноль");  
 }  
 dDen = cB.bi + dR\*cB.a;  
 div.a = (dR\*a + bi)/dDen;  
 div.bi = (dR\*bi - a)/dDen;  
 }  
 return (div);  
}

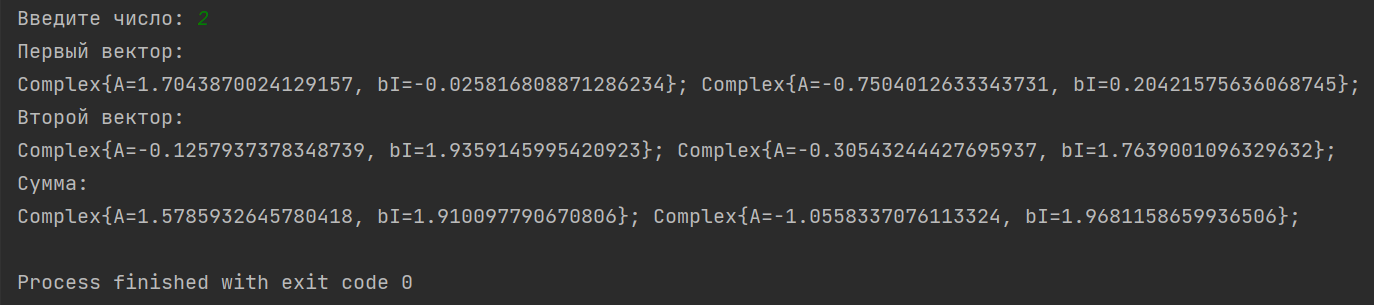


Рисунок 1 – Результат работы программы

Ниже представлен код класса второй программы, а на рисунке 2 – результат работы программы.

public void Root(){  
 this.d=this.b\*this.b-4\*this.a\*this.c;  
 if ((-b+Math.sqrt(d))/(2\*a) == Float.POSITIVE\_INFINITY || (-b+Math.sqrt(d))/(2\*a) == Float.NEGATIVE\_INFINITY) {  
 throw new ArithmeticException("Деление на ноль");  
 }  
 this.x1=(-b+Math.sqrt(d))/(2\*a);  
 this.x2=(-b-Math.sqrt(d))/(2\*a);  
}

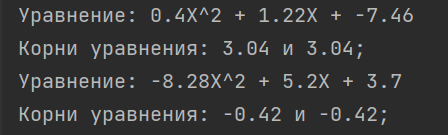


Рисунок 2 – Результат работы программы

**Вариант 2:**

Выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

Создать классы, спецификации которых приведены ниже. Определить конструкторы и методы setТип(), getТип(), toString(). Определить дополнительно методы в классе, создающем массив объектов. Задать критерий выбора данных и вывести эти данные на консоль.

1. Car: id, Марка, Модель, Год выпуска, Цвет, Цена, Регистрационный номер. Создать массив объектов. Вывести: a) список автомобилей заданной марки; b) список автомобилей заданной модели, которые эксплуатируются больше n лет; c) список автомобилей заданного года выпуска, цена которых больше указанной.
2. Product: id, Наименование, UPC, Производитель, Цена, Срок хранения, Количество. Создать массив объектов. Вывести: a) список товаров для заданного наименования; b) список товаров для заданного наименования, цена которых не превосходит заданную; c) список товаров, срок хранения которых больше заданного.

**Решение:**

Ниже представлен кусок кода первой программы, а на рисунке 3 – результат работы программы.

public String toString() {  
 return "Car{" +  
 ", id=" + id +  
 ", year=" + year +  
 ", price=" + price +  
 ", regNom=" + regNom +  
 ", brand='" + brand + '\'' +  
 ", model='" + model + '\'' +  
 ", color='" + color + '\'' +  
 '}';  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public void setYear(int year) {  
 this.year = year;  
 }  
  
 public void setPrice(int price) {  
 this.price = price;  
 }  
  
 public void setRegNom(String regNom) {  
 this.regNom = regNom;  
 }  
  
 public void setBrand(String brand) {  
 this.brand = brand;  
 }  
  
 public void setModel(String model) {  
 this.model = model;  
 }  
  
 public void setColor(String color) {  
 this.color = color;  
 }  
}  
  
class ModelNullException extends Exception {  
 String str;  
  
 public ModelNullException(){  
 super();  
 }  
 public ModelNullException(String str) {  
 this.str = str;  
 }  
  
 public static void GetYear(int num) throws ModelNullException {  
  
 if (num > 2023 | num < 1700) throw new ModelNullException("Введен некорректный год:" + num);  
  
 }  
 public void getMsg() {  
 System.err.println(str);  
 }  
}

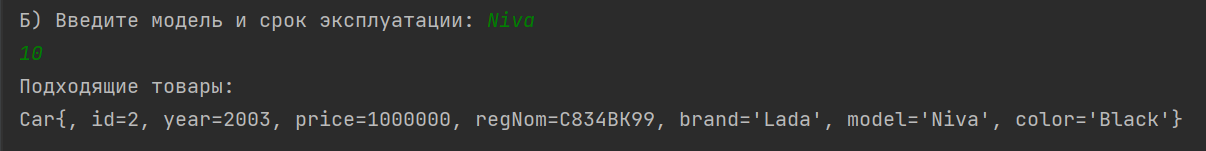


Рисунок 3 – Результат работы программы

Ниже представлен часть кода второй программы, а на рисунке 4 – результат работы программы.

class ModelNullException extends Exception {  
 String str;  
  
 public ModelNullException(){  
 super();  
 }  
 public ModelNullException(String str) {  
 this.str = str;  
 }  
  
 public static void GetYear(int num) throws ModelNullException {  
  
 if ( num < 0) throw new ModelNullException("Введен некорректный срок хранения:" + num);  
  
 }  
 public void getMsg() {  
 System.err.println(str);  
 }  
}

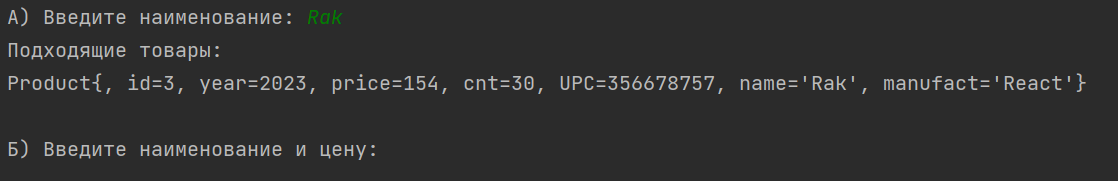


Рисунок 4 – Результат работы программы

**Вариант 3:**

В следующих заданиях требуется ввести последовательность строк из текстового потока и выполнить указанные действия. При этом могут рассматриваться два варианта:

· каждая строка состоит из одного слова;

· каждая строка состоит из нескольких слов.

Имена входного и выходного файлов, а также абсолютный путь к ним могут быть введены как параметры командной строки или храниться в файле.

1. Определить частоту повторяемости букв и слов в стихотворении Александра Пушкина.
2. 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.

**Решение:**

Ниже представлен код класса первой программы, а на рисунке 5 – результат работы программы.

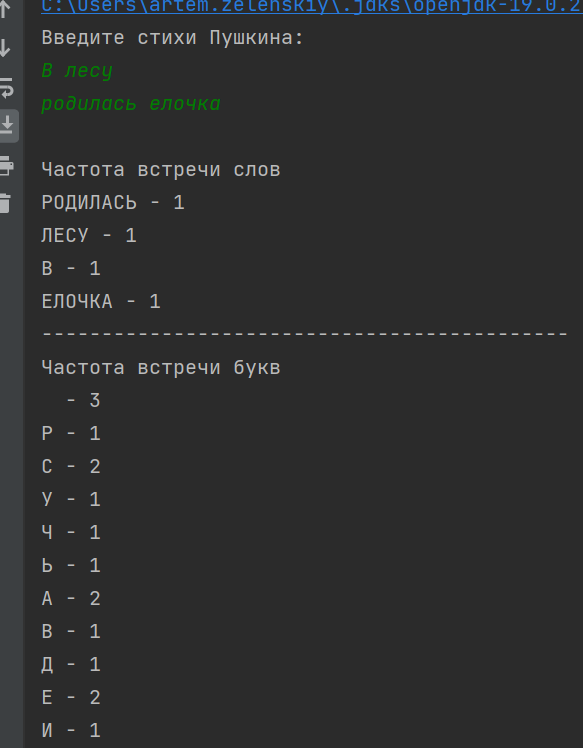
import java.util.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 ArrayList<String> words = new ArrayList<String>();  
 StringBuilder oneLongString =new StringBuilder();  
 HashMap<String,Integer> hasWord =new HashMap<>();  
 HashMap<Character,Integer> hasChar =new HashMap<>();  
 System.out.println("Введите стихи Пушкина: ");  
 String s = null;  
  
 // Запоминаем стихи  
 while ( true ) {  
 s = scanner.nextLine().replaceAll("[.,?!:-;]", "").toUpperCase();  
 if ( s.isEmpty() )  
 break;  
 words.addAll(Arrays.asList(s.split(" ")));  
 oneLongString.append(s);  
 }  
  
 for (int i=0; i<words.size(); i++) {  
 if (words.get(i).isEmpty()) words.remove(i);  
 }  
  
 //Хэшируем слова  
 for ( String str : words )  
 if ( hasWord.containsKey(str)) hasWord.put(str,hasWord.get(str)+1);  
 else hasWord.put(str,1);  
  
 System.out.println("Частота встречи слов");  
 for (String str : hasWord.keySet())  
 System.out.println(str+ " - " + hasWord.get(str));  
  
 //Хэшируем буквы  
 for (int i=0; i< oneLongString.length(); i++) {  
 char ch = oneLongString.charAt(i);  
 if (hasChar.containsKey(ch)) hasChar.put(ch, hasChar.get(ch) + 1);  
 else hasChar.put(ch, 1);  
 }  
  
 System.out.println("--------------------------------------------");  
 System.out.println("Частота встречи букв");  
 for (Character str : hasChar.keySet())  
 System.out.println(str+ " - " + hasChar.get(str));  
 scanner.close();  
 }  
}  


Рисунок 5 – Результат работы программы

Ниже представлен код класса второй программы, а на рисунке 6 – результат работы программы.

import java.io.FileReader;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class Main {

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

String fileName="C:\\Users\\artem.zelenskiy\\IdeaProjects\\Lab\_5\\Lab\_5\_6\\src\\test.txt";

FileReader fr= new FileReader("C:\\Users\\artem.zelenskiy\\IdeaProjects\\Lab\_5\\Lab\_5\_6\\src\\test.txt");

Scanner scanner = new Scanner(fr);

int x = 0;

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

while (scanner.hasNextLine()) {

x++;

String s = scanner.nextLine().replaceAll("[.,?!:-;]", "");

if ( s.isEmpty() )

break;

strMatrix.addAll(Arrays.asList(s.split(" ")));

}

System.out.println("Исходная матрица размером " + x +"x"+ x);

int [][] matrix=new int [x][x];

System.out.println("Исходная матрица: ");

for(int i=0; i<x; i++){

for(int j=0; j<x; j++){

matrix[i][j]= Integer.parseInt(strMatrix.get(i\*x+j));

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

}

System.out.println("");

}

for (int i = 0; i < x; i++) {

for (int j = i+1; j < x; j++) {

int temp = matrix[i][j];

matrix[i][j] = matrix[j][i];

matrix[j][i] = temp;

}

}

System.out.println("Транс матрица: ");

for(int i=0; i<x; i++){

for(int j=0; j<x; j++){

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

}

System.out.println("");

}

}}

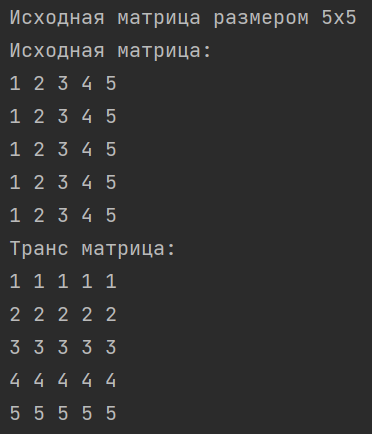


Рисунок 6 – Результат работы программы

**Вариант 4:**

При выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File

1. Из текста Java-программы удалить все виды комментариев.
2. Прочитать строки из файла и поменять местами первое и последнее слова в каждой строке.

**Решение:**

Ниже представлен кусок кода первой программы.

import java.io.\*;  
import java.nio.file.\*;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Scanner;  
import java.io.FileNotFoundException;  
  
 public class Main {  
 public static void main(String[] args) throws Exception {  
 FileReader fr= new FileReader("C:\\Users\\artem.zelenskiy\\IdeaProjects\\Lab\_5\\Lab\_5\_7\\src\\Java\_code.txt");  
 Scanner scanner = new Scanner(fr);  
 StringBuilder code = new StringBuilder();  
 while (scanner.hasNextLine()) {  
 String s = scanner.nextLine();  
 code.append(s);  
 code.append("\n");  
 }  
 fr.close();  
  
 System.out.println(code);  
 System.out.println(code.indexOf("\*/"));  
 System.out.println("------------------------");  
  
 int n=code.length();  
  
 while (n>=0){  
 if (code.indexOf("//")>0 && code.indexOf("\n")>0){  
 code.delete(code.indexOf("//"), code.indexOf("\n", code.indexOf("//")));  
 }  
  
 if (code.indexOf("/\*")>=0 && code.indexOf("\*/")>=0){  
 code.delete(code.indexOf("/\*"), code.indexOf("\*/", code.indexOf("/\*"))+2);  
 }  
 n=n-code.indexOf("\n")-1;  
  
 }  
 Path directory= Files.createDirectory((Paths.get("C:\\Users\\artem.zelenskiy\\IdeaProjects\\Lab\_5\\Lab\_5\_7\\src\\file1")));  
 FileWriter nFile = new FileWriter("C:\\Users\\artem.zelenskiy\\IdeaProjects\\Lab\_5\\Lab\_5\_7\\src\\file1\\file1.txt");  
 nFile.write(String.valueOf(code));  
 nFile.close();  
 System.out.println(code);  
 }  
}

Ниже представлен часть кода второй программы.

import java.io.\*;

import java.nio.file.\*;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class Main {

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

FileReader fr= new FileReader("C:\\Users\\artem.zelenskiy\\IdeaProjects\\Lab\_5\\Lab\_5\_8\\src\\test.txt");

Scanner scanner = new Scanner(fr);

StringBuilder code = new StringBuilder();

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

System.out.println("Первоначальный текст \n");

while (scanner.hasNextLine()) {

String s = scanner.nextLine();

System.out.println(s);

str.clear();

str.addAll(Arrays.asList(s.split(" ")));

String st= str.get(str.size()-1);

str.set(str.size()-1, str.get(0));

str.set(0, st);

for (String s1:str){

code.append(s1);

code.append(" ");

}

code.append("\n");

}

fr.close();

Path directory= Files.createDirectory((Paths.get("C:\\Users\\artem.zelenskiy\\IdeaProjects\\Lab\_5\\Lab\_5\_8\\src\\file1")));

FileWriter nFile = new FileWriter("C:\\Users\\artem.zelenskiy\\IdeaProjects\\Lab\_5\\Lab\_5\_8\\src\\file1\\file1.txt");

nFile.write(String.valueOf(code));

nFile.close();

System.out.println("\n \n Измененный текст\n");

System.out.println(code);

}

}

**Вывод:** были получены базовые представления о работе языка java сисключениями, файлами. Были написаны программы согласно вариантам.