



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 № 5

**Вариант 18**

**Название:** Работа с исключениями и файлами

**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 обработки исключений в Java и навыков работы с файлами в Java.

## Выполнение

### Задание 1:

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.

Листинг выполнения подзадачи 1

```
#ComplexNum.java

package com.aranei.var1;

import java.util.ArrayList;

public class ComplexNum {
    private double real;
    private double imaginary;

    public ComplexNum() {
        this.real = 0;
        this.imaginary = 0;
    }

    public ComplexNum(double R, double I) {
        this.setReal(R);
        this.setImaginary(I);
    }

    public void setReal(double real) {
        this.real = real;
    }

    public void setImaginary(double imaginary) {
        this.imaginary = imaginary;
    }

    public double getReal() {
        return real;
    }

    public double getImaginary() {
        return imaginary;
    }

    public void copyComplex(ComplexNum Complex) {
        this.setReal(Complex.getReal());
        this.setImaginary(Complex.getImaginary());
    }

    public void plusComplex(ComplexNum Complex) {
        this.setReal(Complex.getReal() + this.getReal());
        this.setImaginary(Complex.getImaginary() + this.getImaginary());
    }
}
```

```

        public void minusComplex(ComplexNum Complex) {
            this.setReal(Complex.getReal() - this.getReal());
            this.setImaginary(Complex.getImaginary() - this.getImaginary());
        }

        public void mulComplex(ComplexNum Complex) {
            this.setReal(Complex.getReal() * this.getReal() - this.getImaginary() *
Complex.getImaginary());
            this.setImaginary(Complex.getImaginary() * this.getReal() + this.getImaginary() *
Complex.getReal());
        }

        public void divComplex(ComplexNum Complex) {
            double temp = Complex.getReal() * Complex.getReal() + Complex.getImaginary() *
Complex.getImaginary();
            this.setReal(Complex.getReal() * this.getReal() + this.getImaginary() *
Complex.getImaginary() / temp);
            this.setImaginary(Complex.getReal() * this.getImaginary() - this.getReal() *
Complex.getImaginary() / temp);

            if (temp == 0) throw new ArithmeticException("zero znamenatel");
        }

        public ArrayList<ComplexNum> vectorComplex(ArrayList<ComplexNum> vector1,
ArrayList<ComplexNum> vector2) {
            ArrayList<ComplexNum> result = new ArrayList<>();
            ComplexNum tempC = new ComplexNum();
            for (int i = 0; i < vector1.size(); i++) {
                tempC.setReal(vector1.get(i).getReal() + vector2.get(i).getReal());
                tempC.setImaginary(vector1.get(i).getImaginary() +
vector2.get(i).getImaginary());
                result.set(i, tempC);
            }
            return result;
        }
    }
}

```

### #KvadrUrav.java

```

package com.aranei.var1;

import java.util.ArrayList;

public class KvadrUrav {
    private double a;
    private double b;
    private double c;
    private double d;
    private ArrayList<Double> Xes;

    public KvadrUrav() {
        this.a = 0;
        this.b = 0;
        this.c = 0;
    }

    public KvadrUrav(double A, double B, double C) {
        this.setA(A);
        this.setB(B);
        this.setC(C);
        this.calcD();

        if (A == 0 | B == 0) throw new ArithmeticException("zero a or b");
    }

    public void setA(double A) {
        this.a = A;
        if (A == 0) throw new ArithmeticException("zero a");
    }

    public void setB(double B) {
        this.b = B;
        if (B == 0) throw new ArithmeticException("zero b");
    }

    public void setC(double C) {
        this.c = C;
    }
}

```

```

public double getA() {
    return a;
}

public double getB() {
    return b;
}

public double getC() {
    return c;
}

public double getD() {
    return d;
}

public double getEx() {
    return this.Xes.get(0);
}

public double getX1() {
    return this.Xes.get(1);
}

public double getX2() {
    return this.Xes.get(2);
}

public double getXs() {
    return this.Xes.get(3);
}

public void calcD() {
    this.d = this.getB() * this.getB() - 4 * this.getA() * this.getC();
}

public void calcXes() {
    if (this.getD() >= 0) {
        if (this.getD() == 0) {
            this.Xes.set(0, - this.getB() / (2 * this.getA())); // Экстремум
            this.Xes.set(1, - this.getB() / (2 * this.getA())); // Корень
            this.Xes.set(3, 1.0);
        }
        else {
            this.Xes.set(0, - this.getB() / (2 * this.getA())); // Экстремум
            this.Xes.set(1, - this.getB() - Math.sqrt(this.getD()) / (2 * this.getA())); // Корень 1
            this.Xes.set(2, - this.getB() + Math.sqrt(this.getD()) / (2 * this.getA())); // Корень 2
            this.Xes.set(3, 2.0);
        }
    }
}

public void fallRaise() {
    if (this.getA() > 0) {
        System.out.println("Fall to " + this.getEx());
        System.out.println("Raise from " + this.getEx());
    }
    else {
        System.out.println("Raise to " + this.getEx());
        System.out.println("Fall from " + this.getEx());
    }
}

public ArrayList<Double> vectorComplex(ArrayList<KvadrUrav> array) {
    ArrayList<Double> result = new ArrayList<>();
    result.set(0, array.get(0).getX1());
    result.set(1, array.get(0).getX1());
    for (KvadrUrav kvadrUrav : array) {
        if (kvadrUrav.getXs() == 2.0) {
            if (result.get(0) > kvadrUrav.getX2()) result.set(0, kvadrUrav.getX2());
            if (result.get(1) < kvadrUrav.getX2()) result.set(1, kvadrUrav.getX2());
        }
        if (result.get(0) > kvadrUrav.getX1()) result.set(0, kvadrUrav.getX1());
        if (result.get(1) < kvadrUrav.getX1()) result.set(1, kvadrUrav.getX1());
    }
    return result;
}
}
c

```

## Задание 2:

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

### Листинг выполнения подзадачи 2

#### #Car.java

```
package com.aranei.var2;

public class Car {
    private Integer id;
    private String mark;
    private String model;
    private Integer year;
    private String color;
    private Integer price;
    private String regNumber;

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getMark() {
        return mark;
    }

    public void setMark(String mark) throws Exception {
        checkString(mark);
        this.mark = mark;
    }

    public String getModel() {
        return model;
    }

    public void setModel(String model) throws Exception {
        checkString(model);
        this.model = model;
    }

    public Integer getYear() {
        return year;
    }

    public void setYear(Integer year) {
        this.year = year;
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) throws Exception {
        checkString(color);
        this.color = color;
    }

    public Integer getPrice() {
        return price;
    }

    public void setPrice(Integer price) {
        this.price = price;
    }

    public String getRegNumber() {
        return regNumber;
    }
}
```

```

        public void setRegNumber(String regNumber) throws Exception {
            checkString(regNumber);
            this.regNumber = regNumber;
        }

        @Override
        public String toString() {
            return "Car{" +
                "id=" + id +
                ", mark='" + mark + '\'' +
                ", model='" + model + '\'' +
                ", year=" + year +
                ", color='" + color + '\'' +
                ", price=" + price +
                ", regNumber='" + regNumber + '\'' +
                '}';
        }

        private void checkString(String value) throws Exception {
            if (value.equals("")) throw new Exception("Invalid value");
        }
    }

```

---

## #Product.java

```

package com.aranei.var2;

public class Product {
    private Integer id;
    private String name;
    private String UPC;
    private String manufacturer;
    private Integer price;
    private Integer shelfLife;
    private Integer amount;

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) throws Exception {
        checkString(name);
        this.name = name;
    }

    public String getUPC() {
        return UPC;
    }

    public void setUPC(String UPC) throws Exception {
        checkString(UPC);
        this.UPC = UPC;
    }

    public String getManufacturer() {
        return manufacturer;
    }

    public void setManufacturer(String manufacturer) throws Exception {
        checkString(manufacturer);
        this.manufacturer = manufacturer;
    }

    public Integer getPrice() {
        return price;
    }

    public void setPrice(Integer price) {
        this.price = price;
    }
}

```

```

    }

    public Integer getShelfLife() {
        return shelfLife;
    }

    public void setShelfLife(Integer shelfLife) {
        this.shelfLife = shelfLife;
    }

    public Integer getAmount() {
        return amount;
    }

    public void setAmount(Integer amount) {
        this.amount = amount;
    }

    @Override
    public String toString() {
        return "Product{" +
            "id=" + id +
            ", name='" + name + '\'' +
            ", UPC='" + UPC + '\'' +
            ", manufacturer='" + manufacturer + '\'' +
            ", price=" + price +
            ", shelfLife=" + shelfLife +
            ", amount=" + amount + '\'' +
            '}';
    }

    private void checkString(String value) throws Exception {
        if (value.equals("")) throw new Exception("Invalid value");
    }
}

```

### Задание 3:

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.

#### Листинг выполнения подзадачи 3

##### #Pushkin.java

```

package com.aranei.var3;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.HashMap;
import java.util.Scanner;

public class Pushkin {
    public void testP() {
        try {
            File verse = new File("src/com/aranei/var3/verse.txt");

            HashMap<String, Integer> words = new HashMap<>();
            HashMap<Character, Integer> characters = new HashMap<>();

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

            while (text.hasNextLine()) {
                String line = text.nextLine().toLowerCase();
            }
        }
    }
}

```

```

        String[] totalWords = line.split("[^a-яA-Я]+");

        for (String w:totalWords) {
            if (w.equals("")) continue;

            int count = words.getDefault(w, 0);
            words.put(w, count + 1);

            for (Character c:w.toCharArray()) {
                int c_count = characters.getDefault(c, 0);
                characters.put(c, c_count + 1);
            }
        }
        System.out.println("Слова:");
        words.forEach((key, value) -> System.out.println(key + " : " + value + ";"));

        System.out.println("Буквы:");
        characters.forEach((key, value) -> System.out.println(key + " " + value + ";"));

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

```

---

### #Matrix.java

```

package com.aranei.var3;

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

public class Matrix {
    public void testM() {
        try {
            File file = new File("src/com/aranei/var3/matrixFile.txt");

            ArrayList<ArrayList<Integer>> numbers = new ArrayList<>();
            int size = 0;

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

            while (text.hasNextLine()) {
                String line = text.nextLine().toLowerCase();
                ArrayList<Integer> temp = new ArrayList<>();
                String[] totalNumbers = line.split("[^0-9]+");
                for (String w:totalNumbers) {
                    if (w.equals("")) continue;
                    temp.add(Integer.parseInt(w));
                }
                numbers.add(temp);
                size++;
            }

            System.out.println("Размеры:" + numbers.size() + "x" + numbers.get(0).size());

            for (ArrayList<Integer> number : numbers) {
                for (int j = 0; j < numbers.get(0).size(); j++) {
                    System.out.print(number.get(j) + " ");
                }
                System.out.println();
            }

            System.out.println("Транспонирование:");

            for (int i = 0; i < numbers.get(0).size(); i++) {
                for (ArrayList<Integer> number : numbers) {
                    System.out.print(number.get(i) + " ");
                }
                System.out.println();
            }
        }
    }
}

```



```

    }

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

```

### Задание 4:

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

Листинг выполнения подзадачи 4

#Comments.java

```
package com.aranei.var4;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

public class Comments {
    public void testC() {
        try {

            // Получаем путь файла
            Path inputPath = Paths.get("in/MainT.java");

            // Читаем все строки файла с кодом программы
            byte[] content = Files.readAllBytes(inputPath);
            String program = new String(content);

            // Обрабатываем строки - удаляем комментарии
            String out = program.trim()
                .replaceAll("/\\*([^\r\n"] +
                    ")|\\*+([^\r\n"] +
                    "])))*\\*+/", "") // Многострочные
                .replaceAll("//(.*)*", ""); // Однострочные

            // Создаем новую директорию, если не существует
            Files.createDirectories(Paths.get("result"));

            // Задаем путь для нового файла для вывода
            Path outputPath = Paths.get("result/CommentsResult.java");

            // Записываем в новый файл получившийся код
            Files.write(outputPath, out.getBytes());
            System.out.println("Файл done");
        } catch (FileNotFoundException e) {
            System.out.println("Файл не найден");
            e.printStackTrace();
        } catch (IOException e) {
            System.out.println("Произошла ошибка I/O");
            e.printStackTrace();
        }
    }
}
```

### #SwapWords.java

```
package com.aranei.var4;

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

```

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

            ArrayList<ArrayList<String>> numbers = new ArrayList<>();
            int size = 0;

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

            while (text.hasNextLine()) {
                String line = text.nextLine();
                ArrayList<String> temp = new ArrayList<>();
                String[] totalNumbers = line.split("[^a-zA-Я]+");
                for (String w:totalNumbers) {
                    if (w.equals("")) continue;
                    temp.add(w);
                }
                numbers.add(temp);
                size++;
            }

            for (int i = 0; i < numbers.size(); i++) {
                System.out.print(numbers.get(i).get(numbers.get(i).size()-1) + " ");
                for (int j = 1; j+1 < numbers.get(i).size(); j++) {
                    System.out.print(numbers.get(i).get(j) + " ");
                }

                System.out.print(numbers.get(i).get(0) + " ");
                System.out.println();
            }

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

```

### Ссылка на программное решение

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Aranei99/labsJava/tree/main/LB5>

### Вывод

При выполнении лабораторной работы были получены навыки обработки исключений в Java и навыки работы с файлами в Java.