



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 14**

**Название:** Работа с исключениями и файлами

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

Студент

ИУ6-23М

(Группа)

\_\_\_\_\_  
(Подпись, дата)

В.Е. Санталов

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

**Цель работы:** получение навыков обработки исключения в Java и навыков работами с файлами в Java.

### Задание 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.

### Подзадача 1.

Код программы:

```
public void mul(Matrix1 matrix) throws Exception {
    int size = matrix.getValue().length;
    Integer[][] newMatrix = new Integer[size][size];

    if (this.value == null) throw new NullPointerException();
    if (matrix.getValue()[0].length != this.value[0].length) throw new Exception(message: "Размерность матриц не одинакова");

    if (size == this.value.length) {
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                int sum = 0;
                for (int k = 0; k < matrix.getValue().length; k++) {
                    sum += this.value[i][k] * matrix.getValue()[k][j];
                }
                newMatrix[i][j] = sum;
            }
        }
    }
    else System.out.println(x: "Matrix sizes are not equal");

    this.value = newMatrix;
}
```

```

public Integer findNorm(int normNumber) {
    int size = this.value.length;

    Integer[] normArray = new Integer[size];

    if (this.value == null) throw new NullPointerException();

    for (int i = 0; i < size; i++) {
        int sum = 0;
        for (int j = 0; j < size; j++) {
            sum += this.value[i][j];
        }
        normArray[i] = sum;
    }

    Arrays.sort(normArray);
    System.out.println(normArray);

    if (normNumber < size) return normArray[size - normNumber - 1];
    else throw new ArrayIndexOutOfBoundsException();
}

```

## Подзадача 2.

Код программы:

```

public Matrix2(int n, int m, int bound, int bias) throws Exception {
    Integer[][] value = new Integer[n][m];
    Random rand = new Random();

    if (n == 0 || m == 0) throw new Exception(message: "Размерность матрицы не может быть равна нулю");

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            value[i][j] = rand.nextInt(bound) + bias;
        }
    }

    this.value = value;
}

```

```

static public Matrix2 swapMaxRows(Matrix2 matrix, int column) {
    int maxElem = -1000;
    int minElem = 1000;

    int maxElemRow = 0;
    int minElemRow = 0;

    if (matrix.value.length > column + 1) throw new ArrayIndexOutOfBoundsException();

    for (int i = 0; i < matrix.value.length; i++) {
        int val = matrix.value[column][i];

        if (val > maxElem) {
            maxElem = val;
            maxElemRow = i;
        }

        if (val < minElem) {
            minElem = val;
            minElemRow = i;
        }
    }

    Integer[] maxRow = matrix.value[maxElemRow];
    matrix.value[maxElemRow] = matrix.value[minElemRow];
    matrix.value[minElemRow] = maxRow;

    return matrix;
}

```

## Задание 2:

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

## Подзадача 1.

Код программы:

```

public class ZeroScoresException extends Exception {
    public ZeroScoresException(String errorMessage) {
        super(errorMessage);
    }
}

```

```

public Abiturient(int id, String name, String surname, String address, String phone, int[] scores) throws ZeroScoresException {
    this.id = id;
    this.name = name;
    this.surname = surname;
    this.address = address;
    this.phone = phone;
    this.scores = scores;

    if (scores.length == 0) throw new ZeroScoresException(name + " " + surname);
}

```

## Подзадача 2.

Код программы:

```

public class ZeroPageCount extends Exception {
    public ZeroPageCount(int pageCount) {
        super(Integer.toString(pageCount));
    }
}

```

```

public class ZeroScoresException extends Exception {
    public ZeroScoresException(String errorMessage) {
        super(errorMessage);
    }
}

```

```

public Book(int id, String name, String author, int year, int pageCount, int price, String cover) throws NegativeYearException, ZeroPageCount {
    this.id = id;
    this.name = name;
    this.author = author;
    this.year = year;
    this.pageCount = pageCount;
    this.price = price;
    this.cover = cover;

    if (year < 0) throw new NegativeYearException(year);
    if (pageCount <= 0) throw new ZeroPageCount(pageCount);
}

```

## Задание 3:

4. Найти и вывести слова текста, для которых последняя буква одного слова совпадает с первой буквой следующего слова.
5. Найти в строке наибольшее число цифр, идущих подряд.

## Подзадача 1.

Код программы:

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.ArrayList;

public class Laba5_4 {
    Run | Debug
    public static void main(String[] args) throws IOException {
        String fileLocation = System.getProperty("key: \"user.dir\") + \"\\src\\\" + args[0];
        ArrayList<String> list = new ArrayList<String>();

        try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(fileLocation)), charsetName: \"UTF-8\"))) {
            for (String line; (line = br.readLine()) != null;) {
                list.add(line);
            }
        } catch (Exception e) {
            System.out.println(x: \"File not found\");
            System.exit(status: 0);
        }

        String string = \"\";

        for (String word : list) {
            String fword = word.toLowerCase();
            if (fword.charAt(index: 0) == fword.charAt(fword.length() - 1)) {
                string += word + \" \";
            }
        }

        BufferedWriter writer = new BufferedWriter(new FileWriter(args[1]));
        writer.write(string);
        writer.close();
    }
}

```

## Результат работы программы:

```

3 > src > ≡ file.txt
1   AbbA
2   Bob
3   Bruh
4   Chicken
5   Cow
6   Submarine

```

```

3 > ≡ out_file.txt
1   AbbA Bob

```

## Подзадача 2.

Код программы:

```

import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;

public class Laba5_5 {
    Run | Debug
    public static void main(String[] args) throws IOException {
        String fileLocation = System.getProperty(key: "user.dir") + "\\src\\" + args[0];
        String string = "";

        try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(fileLocation)), charsetName: "UTF-8"))) {
            string = br.readLine();
        } catch (Exception e) {
            System.out.println(x: "File not found");
            System.exit(status: 0);
        }

        System.out.println(string);

        String maxSequence = "";
        String currentSequence = "";
        int maxSequenceLen = 0;
        int currentLen = 0;

        for (int i = 0; i < string.length(); i++) {
            if (Character.isDigit(string.charAt(i))) {
                currentSequence += string.charAt(i);
                currentLen++;

                if (currentLen > maxSequenceLen) {
                    maxSequenceLen = currentLen;
                    maxSequence = currentSequence;
                }
            } else {
                currentLen = 0;
                currentSequence = "";
            }
        }

        BufferedWriter writer = new BufferedWriter(new FileWriter(args[1]));
        writer.write(maxSequence + "\n");
        writer.write("Sequence length: " + maxSequenceLen);
        writer.close();
    }
}

```

## Результат работы программы:

```

3 > src > file2.txt
1  jdhksakjhd1273ghkjsgdkjhasgd12873t6adsgjhdg812dg182idg192dgb1378d120p897df12368de1811113d0816dg3b120p8d312d0p12gdbdgiaks

```

```

3 > out_file2.txt
1  1811113
2  Sequence length:7

```

## Задание 4:

1. В файле, содержащем фамилии студентов и их оценки, записать прописными буквами фамилии тех студентов, которые имеют средний балл более "7".
2. Файл содержит символы, слова, целые числа и числа с плавающей запятой. Определить все данные, тип которых вводится из командной строки.

## Подзадача 1.

Код программы:

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.InputStreamReader;
import java.nio.file.Files;

public class Laba5_4 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        String fileLocation = System.getProperty(key: "user.dir") + "\\src\\";
        BufferedWriter writer = new BufferedWriter(new FileWriter(fileLocation + "out.txt"));

        try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(fileLocation + "students.txt")), charsetName: "UTF-8"))) {
            for (String line; (line = br.readLine()) != null;) {
                String[] splitLine = line.split(regex: ":");

                String student = splitLine[0];
                String[] stringScores = splitLine[1].split(regex: " ");
                double sumScores = 0.0;

                for (String score : stringScores) {
                    int numScore = Integer.parseInt(score);
                    sumScores += numScore;
                }

                sumScores /= stringScores.length;

                if (sumScores > 7.0) {
                    student = student.substring(beginIndex: 0, endIndex: 1).toUpperCase() + student.substring(beginIndex: 1);
                }

                writer.write(student + ":" + String.join(delimiter: " ", stringScores) + "\n");
            }
        } catch (Exception e) {
            System.out.println(x: "File not found");
            System.exit(status: 0);
        }

        writer.close();

        Files.delete(new File(fileLocation + "students.txt").toPath());
        new File(fileLocation + "out.txt").renameTo(new File(fileLocation + "students.txt"));
    }
}
```

Результат работы программы:

```
4 > src > ≡ students.txt
1 Eliseev:10 6 9 5 9 6 9 7 8
2 pupkin:2 9 4 1 9 7 4 5
3 Dudka:8 8 8 8 8 8 8 8
4 trubnik:7 9 6 9 4 2 5 7 3 1
```

## Подзадача 2.

Код программы:



```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStreamReader;

public class Laba5_5 {
    static String parseType(String smth) {
        try {
            Integer.parseInt(smth);
            return "Integer";
        } catch (Exception a) {
            //System.out.println(smth + "is not double");
            try {
                Double.parseDouble(smth);
                return "Double";
            } catch (Exception b) {
                //System.out.println(smth + "is not integer");
                return smth.length() == 1 ? "Char" : "String";
            }
        }
    }

    Run | Debug
    public static void main(String[] args) {
        String fileLocation = System.getProperty(key: "user.dir") + "\\src\\";

        try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(fileLocation + "pile.txt")), charsetName: "UTF-8"))) {
            String line = br.readLine();
            //System.out.println(line);
            for (String smth : line.split(regex: " ")) {
                String type = parseType(smth);
                //System.out.println(smth + " is a " + type);

                if (type.equals(args[0])) {
                    System.out.print(smth + " ");
                }
            }
        } catch (Exception e) {
            System.exit(status: 0);
        }
    }
}

```

**Ссылка на программное решение:**

<https://github.com/Time2HackJS/BigDataLanguages/tree/master/lr5>

**Вывод:** в ходе лабораторной работы были получены навыки обработки исключений в Java и навыки работы с файлами в Java.