



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 6**

**Название:** Работа с исключениями и файлами

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

Студент

ИУ6-23М

(Группа)

\_\_\_\_\_  
(Подпись, дата)

В.А. Елисеев

(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:

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:

```
import java.util.ArrayList;

public class ChainFraction {
    int x;
    ArrayList<Integer> a;

    public ChainFraction(int x, ArrayList<Integer> a) {
        this.x = x;
        this.a = a;
    }

    public float add(ChainFraction b) {
        return this.value() + b.value();
    }

    public float sub(ChainFraction b) {
        return this.value() - b.value();
    }

    public float mul(ChainFraction b) {
        return this.value() * b.value();
    }

    public float div(ChainFraction b) {
        if (b.value() == 0 ) throw new ArithmeticException(s: "Can't divide by zero!");
        else return this.value() / b.value();
    }

    public float value() {
        float val = 0.0f;

        for (int i = a.size() - 1; i >= 0; i--) {
            val = x / (a.get(i) + val);
        }

        return val;
    }
}
```

## **Подзадача 2.**

Код программы:

```

public class Fraction {
    Integer _m;
    Integer _n;

    public Fraction(Integer m, Integer n) {
        this._m = m;
        this._n = n;

        if (this._n == 0) {
            throw new ArithmeticException(s: "Denominator can't be 0!");
        }
    }

    public Fraction(Fraction f) {
        this._m = f._m;
        this._n = f._n;
    }

    public Fraction add(Fraction b) {
        Integer r_m = this._m * b._n + b._m * this._n;
        Integer r_n = this._n * b._n;
        return new Fraction(r_m, r_n);
    }

    public Fraction sub(Fraction b) {
        Integer r_m = this._m * b._n - b._m * this._n;
        Integer r_n = this._n * b._n;
        return new Fraction(r_m, r_n);
    }

    public Fraction mul(Fraction b) {
        Integer r_m = this._m * b._m;
        Integer r_n = this._n * b._n;
        return new Fraction(r_m, r_n);
    }

    public Fraction div(Fraction b) {
        Integer r_m = this._m * b._n;
        Integer r_n = this._n * b._m;
        return new Fraction(r_m, r_n);
    }

    public float value() {
        return (float)this._m / this._n;
    }
}

```

## Задание 2:

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

### Подзадача 1.

Код программы:

```
public class InvalidHouseInfoException extends Exception {  
    public InvalidHouseInfoException(String field, String value) {  
        super(field + ": " + value);  
    }  
}
```

```
public House(  
    int id,  
    int number,  
    int square,  
    int floor,  
    int room_count,  
    String street,  
    String building_type,  
    int lifetime  
) throws InvalidHouseInfoException {  
    if (Stream.of(_id, _number, _square, _floor, _room_count, _lifetime, _street, _building_type).anyMatch(Objects::isNull))  
        throw new InvalidHouseInfoException(field: null, value: null);  
    if (square <= 0) throw new InvalidHouseInfoException(field: "_square", Integer.toString(square));  
    if (floor <= 0) throw new InvalidHouseInfoException(field: "_floor", Integer.toString(floor));  
    if (lifetime <= 0) throw new InvalidHouseInfoException(field: "_lifetime", Integer.toString(lifetime));  
  
    this._id = id;  
    this._number = number;  
    this._square = square;  
    this._floor = floor;  
    this._room_count = room_count;  
    this._street = street;  
    this._building_type = building_type;  
    this._lifetime = lifetime;  
}
```

### Подзадача 2.

Код программы:

```
public class InvalidPatronymicException extends Exception {  
    InvalidPatronymicException(String patronymic) {  
        super(patronymic);  
    }  
}
```

```
public class InvalidCardNumberException extends Exception {
    public InvalidCardNumberException(String card) {
        super(card);
    }
}
```

```
public Phone(
    int id,
    String last_name,
    String name,
    String patronymic,
    String address,
    String card_number,
    double debit,
    double credit,
    int intercity_calls,
    int intracity_calls
) throws InvalidCardNumberException, InvalidPatronymicException {
    if (!_card_number.matches(regex: "[0-9]{4}([0-9]{4}){3}$")) throw new InvalidCardNumberException(_card_number);
    if (!_patronymic.endsWith(suffix: "вич") || !_patronymic.endsWith(suffix: "вна")) throw new InvalidPatronymicException(patronymic);

    this._id = id;
    this._last_name = last_name;
    this._name = name;
    this._patronymic = patronymic;
    this._address = address;
    this._card_number = card_number;
    this._debit = debit;
    this._credit = credit;
    this._intercity_calls = intercity_calls;
    this._intracity_calls = intracity_calls;
}
```

### Задание 3:

6. В каждой строке стихотворения Анны Ахматовой подсчитать частоту повторяемости каждого слова из заданного списка и вывести эти слова в порядке возрастания частоты повторяемости.

7. В каждом слове стихотворения Николая Заболоцкого заменить первую букву слова на прописную.

### Подзадача 1.

Код программы:

```

public static void main(String[] args) throws Exception {
    String fileLocation = "poem.txt";
    Pattern pattern = Pattern.compile(regex: "[a-zA-Z]{1,}");

    try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(fileLocation)), charsetName: "UTF-8"))) {
        for (String line; (line = br.readLine()) != null; ) {
            if (!line.isEmpty()) {
                System.out.println(line);

                line = line.toLowerCase();
                HashMap<String, Integer> map = new HashMap<String, Integer>();
                for (String arg : args) map.put(arg, value: 0);

                Matcher matcher = pattern.matcher(line);
                while (matcher.find()) {
                    String word = matcher.group();
                    if (Arrays.stream(args).anyMatch(word::equals)) map.put(word, map.get(word) + 1);
                }

                map = map.entrySet().stream().sorted(Entry.comparingByValue()).collect(Collectors.toMap(Entry::getKey, Entry::getValue, (e1, e2) -> e1, LinkedHashMap::new));
                System.out.println(map);
            }
            System.out.println();
        }
    } catch (Exception e) {
        System.out.println(e.toString());
        System.exit(status: 0);
    }
}

```

**Результат работы программы:**

```
Я научилась просто, мудро жить,  
{на=0, и=0, я=1}  
  
Смотреть на небо и молиться Богу,  
{я=0, на=1, и=1}  
  
И долго перед вечером бродить,  
{на=0, я=0, и=1}  
  
Чтоб утомить ненужную тревогу.  
{на=0, и=0, я=0}  
  
Когда шуршат в овраге лопухи  
{на=0, и=0, я=0}  
  
И никнет гроздь рябины желто-красной,  
{на=0, я=0, и=1}  
  
Слагаю я веселые стихи  
{на=0, и=0, я=1}  
  
О жизни тленной, тленной и прекрасной.  
{на=0, я=0, и=1}  
  
Я возвращаюсь. Лижет мне ладонь  
{на=0, и=0, я=1}  
  
Пушистый кот, мурлыкает умильней,  
{на=0, и=0, я=0}  
  
И яркий загорается огонь  
{на=0, я=0, и=1}  
  
На башенке озерной лесопильни.  
{и=0, я=0, на=1}
```

## Подзадача 2.

Код программы:



```

Run | Debug
public static void main(String[] args) throws IOException {
    String fileLocation = "poem2.txt";

    try (OutputStreamWriter writer = new OutputStreamWriter(new FileOutputStream(name: "poem2_out.txt"), charsetName: "UTF-8")) {
        try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(fileLocation)), charsetName: "UTF-8"))) {
            for (String line; (line = br.readLine()) != null;) {
                String[] words = line.split(regex: " ");
                ArrayList<String> list = new ArrayList<String>();

                for (String word : words) {
                    if (word.length() > 1) {
                        String newWord = word.substring(beginIndex: 0, endIndex: 1).toUpperCase() + word.substring(beginIndex: 1);
                        list.add(newWord);
                    }
                    else list.add(word.toUpperCase());
                }

                writer.write(list.stream().map(Object::toString).collect(Collectors.joining(delimiter: " ")) + "\n");
            }
        } catch (Exception e) {
            System.out.println(e.toString());
            System.exit(status: 0);
        }

        writer.close();
    }
}

```

## Результат работы программы:

```

≡ poem2.txt

1  я увидел во сне можжевеловый куст,
2  я услышал вдали металлический хруст,
3  аметистовых ягод услышал я звон,
4  и во сне, в тишине, мне понравился он.
5
6  Я почуял сквозь сон легкий запах смолы.
7  Отогнув невысокие эти стволы,
8  Я заметил во мраке древесных ветвей
9  чуть живое подобье улыбки твоей.
10
11 Можжевеловый куст, можжевеловый куст,
12 Остывающий лепет изменчивых уст,
13 легкий лепет, едва отдающий смолой,
14 Проколовший меня смертоносной иглой!
15
16 В золотых небесах за окошком моим
17 Облака проплывают одно за другим,
18 Облетевший мой садик безжизнен и пуст...
19 да простит тебя бог, можжевеловый куст!

```

```
poem2_out.txt

1  Я Увидел Во Сне Можжевеловый Куст,
2  Я Услышал Вдали Металлический Хруст,
3  Аметистовых Ягод Услышал Я Звон,
4  И Во Сне, В Тишине, Мне Понравился Он.
5
6  Я Почуял Сквозь Сон Легкий Запах Смолы.
7  Отогнув Невысокие Эти Стволы,
8  Я Заметил Во Мраке Древесных Ветвей
9  Чуть Живое Подобье Улыбки Твоей.
10
11 Можжевеловый Куст, Можжевеловый Куст,
12 Остывающий Лепет Изменчивых Уст,
13 Легкий Лепет, Едва Отдающий Смолой,
14 Проколовший Меня Смертоносной Иглой!
15
16 В Золотых Небесах За Окошком Моим
17 Облака Проплывают Одно За Другим,
18 Облетевший Мой Садик Безжизнен И Пуст...
19 Да Простит Тебя Бог, Можжевеловый Куст!
```

#### Задание 4:

1. Из файла удалить все слова, содержащие от трех до пяти символов, но при этом из каждой строки должно быть удалено только максимальное четное количество таких слов.
2. Прочитать текст Java-программы и удалить из него все “лишние” пробелы и табуляции, оставив только необходимые для разделения операторов.

#### Подзадача 1.

Код программы:

```

public class Task1 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        try (OutputStreamWriter writer = new OutputStreamWriter(new FileOutputStream(name: "out.txt"), charsetName: "UTF-8")) {
            try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(pathname: "text.txt")), charsetName: "UTF-8"))) {
                for (String line; (line = br.readLine()) != null; ) {
                    ArrayList<String> words = new ArrayList<String>(Arrays.asList(line.split(regex: " ")));
                    ArrayList<String> list = new ArrayList<String>();

                    for (String word : words) {
                        if (word.matches(regex: "[a-zA-Z]{3,5}")) {
                            list.add(word);
                        }
                    }

                    System.out.println("Line: " + line + ", detected words: " + list);

                    System.out.print(s: "Deleted: ");
                    for (int i = 0; i < list.size() - list.size() % 2; i++) {
                        System.out.print(list.get(i) + " ");
                        words.remove(list.get(i));
                    }

                    System.out.println(x: "\n");

                    writer.write(String.join(delimiter: " ", words) + "\n");
                }
            } catch (Exception e) {
                System.out.println(e.toString());
                System.exit(status: 0);
            }

            writer.close();
        }
    }
}

```

Projects are imported into

## Результат работы программы:

```

≡ text.txt
1  kal joran djasd adddaa dsadasd asdasd
2  asdasd asdkjasd asdkj aaa aaaaa aaaaaa
3  mmmm mmm mmmm mmmmmmm m mm mm m mmm

```

```

≡ out.txt
1  djasd adddaa dsadasd asdasd
2  asdasd asdkjasd aaaaa aaaaaa
3  mmmmmmm m mm mm m

```

## Подзадача 2.

### Код программы:

```

public class Task2 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        try (OutputStreamWriter writer = new OutputStreamWriter(new FileOutputStream(name: "out2.txt"), charsetName: "UTF-8")) {
            try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(pathname: "program.txt")), charsetName: "UTF-8"))) {
                for (String line; (line = br.readLine()) != null; ) {
                    String fLine = line.replaceAll(regex: " ", replacement: " ");
                    if (fLine.startsWith(prefix: " ")) fLine = fLine.substring(beginIndex: 1);

                    if (!fLine.isEmpty()) writer.write(fLine.replaceAll(regex: "+", replacement: " ") + "\n");
                }
            } catch (Exception e) {
                System.out.println(e.toString());
                System.exit(status: 0);
            }

            writer.close();
        }
    }
}

```

## Результат работы программы:

```
program.txt
1  import java.io.BufferedReader;
2  import java.io.File;
3  import java.io.FileInputStream;
4  import java.io.FileOutputStream;
5  import java.io.InputStreamReader;
6  import java.io.OutputStreamWriter;
7  import java.util.ArrayList;
8  import java.util.Arrays;
9
10 public class Task1 {
11     public static void main(String[] args) throws Exception {
12         try (OutputStreamWriter writer = new OutputStreamWriter(new FileOutputStream("out.txt"), "UTF-8")) {
13             try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File("text.txt")), "UTF-8"))) {
14                 for (String line; (line = br.readLine()) != null; ) {
15                     ArrayList<String> words = new ArrayList<String>(Arrays.asList(line.split(" ")));
16                     ArrayList<String> list = new ArrayList<String>();
17
18                     for (String word : words) {
19                         if (word.matches("[a-zA-Z]{3,5}$")) {
20                             list.add(word);
21                         }
22                     }
23
24                     System.out.println("Line: " + line + ", detected words: " + list);
25
26                     System.out.print("Deleted: ");
27                     for (int i = 0; i < list.size() - list.size() % 2; i++) {
28                         System.out.print(list.get(i) + " ");
29                         words.remove(list.get(i));
30                     }
31
32                     System.out.println("\n");
33
34                     writer.write(String.join(" ", words) + "\n");
35                 }
36             } catch (Exception e) {
37                 System.out.println(e.toString());
38                 System.exit(0);
39             }
40
41             writer.close();
42         }
43     }
44 }
```

```

1  import java.io.BufferedReader;
2  import java.io.File;
3  import java.io.FileInputStream;
4  import java.io.FileOutputStream;
5  import java.io.InputStreamReader;
6  import java.io.OutputStreamWriter;
7  import java.util.ArrayList;
8  import java.util.Arrays;
9  public class Task1 {
10 public static void main(String[] args) throws Exception {
11     try (OutputStreamWriter writer = new OutputStreamWriter(new FileOutputStream("out.txt"), "UTF-8")) {
12         try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File("text.txt")), "UTF-8"))) {
13             for (String line; (line = br.readLine()) != null;) {
14                 ArrayList<String> words = new ArrayList<String>(Arrays.asList(line.split(" ")));
15                 ArrayList<String> list = new ArrayList<String>();
16                 for (String word : words) {
17                     if (word.matches("[a-zA-Z]{3,5}$")) {
18                         list.add(word);
19                     }
20                 }
21                 System.out.println("Line: " + line + ", detected words: " + list);
22                 System.out.print("Deleted: ");
23                 for (int i = 0; i < list.size() - list.size() % 2; i++) {
24                     System.out.print(list.get(i) + " ");
25                     words.remove(list.get(i));
26                 }
27                 System.out.println("\n");
28                 writer.write(String.join(" ", words) + "\n");
29             }
30         } catch (Exception e) {
31             System.out.println(e.toString());
32             System.exit(0);
33         }
34         writer.close();
35     }
36 }
37 }

```

### Ссылка на программное решение:

<https://github.com/ArMaxik/BigDataLanguages/tree/main/lr5>

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