



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

О Т Ч Е Т

по лабораторной работе №5

Название: Исключения и файлы

Дисциплина: Языка программирования для работы с большими
данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

Д.Д.

Капитонов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(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 нулевой? - "+Boolean.toString(c==null);
}
@Override
public String toString() {
    return "City{" +
        "name=" + name + "\n" +
        ", 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 + "\n" +
            ", 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("Введенные 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("Чебокская", 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:

```

Ошибка ввода названия улицы. Не выбран ни один её тип.
Новая улица не была добавлена
City{name='Piter', street_list=[Street{name='Войковская', square=true, prospect=false, street=false}]}
Объект пустой? - false
Воникла ошибка - / by zero
1.0
Wardrobe{name='Onegin', cost=12}
Введите название книги и её стоимости
Название - Физика
Цена - 452
Wardrobe{name='Физика', cost=452}

Process finished with exit code 0

```

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("Введи m (слов) - ");
int m = in.nextInt();
System.out.print("Введи 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_fm.close();

    }
}
```

Результат выполнения программы:

```
C:\Users\dan-1\.jdk\corretto-1.8.0_322\bin\java.exe ...
[[1, 2, 3, 4, 5, 6], [7, 8, 9, 1, 2, 3], [4, 5, 6, 7, 8, 9]]
[[1, 7, 4], [2, 8, 5], [3, 9, 6], [4, 1, 7], [5, 2, 8], [6, 3, 9]]
Размерность x - 3 y - 6

[[4, 7, 1], [5, 8, 2], [6, 9, 3], [7, 1, 4], [8, 2, 5], [9, 3, 6]]
6 2 3 4 5 1
3 8 9 1 2 7
9 5 6 7 8 4

[[6, 2, 3, 4, 5, 1], [3, 8, 9, 1, 2, 7], [9, 5, 6, 7, 8, 4]]
Введите m (слов) - 1
Введите n (строк) - 1
[[4]]

Process finished with exit code 0
```

Вывод: научились работать с исключениями и файлами.