



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

Название: Исключения, файлы. Вариант 9

Дисциплина: Языки программирования для работы с большими
данными

Студент ИУ6-22М
(Группа)

К.Ю. Каташинский
(Подпись, дата) (И.О. Фамилия)

Преподаватель

П.В. Степанов
(Подпись, дата) (И.О. Фамилия)

Москва, 2024

Цель работы

Изучить работу с исключениями и файлами на языке программирования Java.

Ход работы

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.

Код программы представлен в листинге 1, результат – на рисунке 1.

Листинг 1 – Код программы задания 1

```
package Lab5;

class BinaryEquation {
    private int[] numbers;

    public BinaryEquation(int[] numbers) throws Exception {
        if (numbers == null) throw new Exception("numbers
is null");
        this.numbers = numbers;
    }

    public BinaryEquation(int a, int b, int c) throws
Exception {
        if (a == 0 || b == 0 || c == 0) throw new
Exception("a or b or c is zero");
        this.numbers = new int[]{a, b, c};
    }

    private int getD() {
        int a = numbers[0];
        int b = numbers[1];
        int c = numbers[2];
        return b * b - 4 * a * c;
    }
}
```

```

    }

    public double[] search() {
        int a = numbers[0];
        int b = numbers[1];
        return new double[]{
            (-b + Math.sqrt(getD()))/(2 * a),
            (-b - Math.sqrt(getD()))/(2 * a)
        };
    }

    public double searchCenter() {
        int a = numbers[0];
        int b = numbers[1];
        return (double) -b / (2 * a);
    }

    public double[][] searchIntervals() {
        int a = numbers[0];
        double point = searchCenter();
        if (a > 0) {
            return new double[][]{
                {Integer.MIN_VALUE, point},
                {point, Integer.MAX_VALUE}
            };
        }

        return new double[][]{
            {point, Integer.MAX_VALUE},
            {Integer.MIN_VALUE, point}
        };
    }
}

public class Variant19 {
    public static void main(String[] args) {

```

```

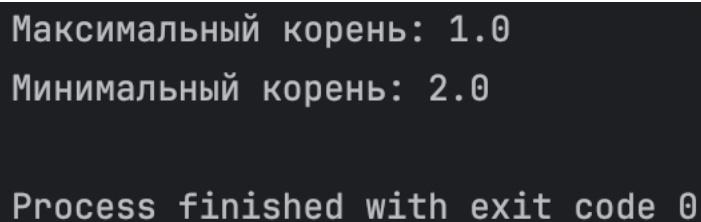
try {
    BinaryEquation[] bes = new BinaryEquation[]{
        new BinaryEquation(1, -3, 2),
        new BinaryEquation(1, 3, 2),
        new BinaryEquation(1, -5, 6),
        new BinaryEquation(1, 5, 6),
    };
    double[] arr = bes[0].search();
    double min = arr[0], max = arr[1];
    if (min < max) {
        double t = min;
        min = max;
        max = t;
    }
    for(int i = 1; i < bes.length; ++i) {
        double[] tempA = bes[i].search();
        double cMin = arr[0], cMax = arr[1];
        if (cMin < cMax) {
            double t = cMin;
            cMin = cMax;
            cMax = t;
        }
        if (cMin < min) {
            min = cMin;
        }
        if (cMax > max) {
            max = cMax;
        }
    }
    System.out.println("Максимальный корень: " +
max);

```

```

        System.out.println("Минимальный корень: " +
min);
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}

```



```

Максимальный корень: 1.0
Минимальный корень: 2.0

Process finished with exit code 0

```

Рисунок 1 – Результат задания 1

Задание 2. 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.

Код программы представлен в листинге 2, результат – на рисунке 2.

Листинг 2 – Код программы задания 2

```

package Lab5;

class BinaryMatrix {
    private int n, m;
    private boolean[][] matrix;
    public BinaryMatrix(boolean[][] matrix, int n, int m)
throws Exception {
        if (matrix == null) throw new Exception("matrix is
null");
        if (n == 0 || m == 0) throw new Exception("n or m
is zero");

```

```

        this.matrix = matrix;
        this.n = n;
        this.m = m;
    }

    public BinaryMatrix(int n, int m) throws Exception {
        if (n == 0 || m == 0) throw new Exception("n or m
is zero");
        this.matrix = new boolean[n][m];
    }

    public boolean[][] get() {
        return this.matrix;
    }

    public void sum(BinaryMatrix bm) {
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                if (!this.matrix[i][j] && !bm.matrix[i][j])
{
                    this.matrix[i][j] = false;
                    continue;
                }
                this.matrix[i][j] = true;
            }
        }
    }

    public void multiply(BinaryMatrix bm) {
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                if (this.matrix[i][j] && bm.matrix[i][j]) {
                    this.matrix[i][j] = true;
                    continue;
                }
            }
        }
    }

```

```

        this.matrix[i][j] = false;
    }
}

public void inverse() {
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            this.matrix[i][j] = !this.matrix[i][j];
        }
    }
}

public int countTrue() {
    int count = 0;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            if (this.matrix[i][j]) {
                ++count;
            }
        }
    }
    return count;
}

public void sort() {
    int[][] arr = new int[n][2];
    for (int i = 0; i < n; ++i) {
        int count = 0;
        for (int j = 0; j < m; ++j) {
            if (this.matrix[i][j]) {
                ++count;
            }
        }
    }
}

```

```

        arr[i][0] = count;
        arr[i][1] = i;
    }
    for (int i = 0; i < n - 1; ++i) {
        for (int j = 0; j < n - i - 1; ++j) {
            if (arr[j][0] > arr[j + 1][0]) {
                var t = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = t;
            }
        }
    }
    var newMatrix = new boolean[n][m];
    for (int i = 0; i < n; ++i) {
        var pos = arr[i][1];
        newMatrix[i] = this.matrix[pos].clone();
    }
    this.matrix = newMatrix;
}

}

public class Variant110 {
    public static void main(String[] args) {
        try {
            BinaryMatrix bm = new BinaryMatrix(new
boolean[][]{
                {true, true, true},
                {false, true, false},
                {false, false, false}
            }, 3, 3);
            bm.sort();
            var r = bm.get();

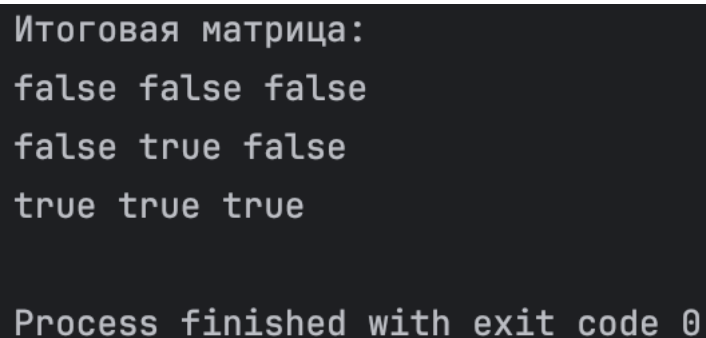
```



```

        System.out.println("Итоговая матрица: ");
        for (int i = 0; i < 3; ++i) {
            for (int j = 0; j < 3; ++j) {
                System.out.printf(r[i][j] + " ");
            }
            System.out.println();
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}

```



```

Итоговая матрица:
false false false
false true false
true true true

Process finished with exit code 0

```

Рисунок 2 – Результат задания 2

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

Код программы представлен в листинге 3, результат – на рисунке 3.

Листинг 3 – Код программы задания 3

```

package Lab5;

import java.util.Objects;

class ProductException extends Exception {
    public ProductException() { super(); }
    public ProductException(String message) {
        super(message); }
    public ProductException(String message, Throwable
        cause) { super(message, cause); }
}

```

```

        public ProductException(Throwable cause) {
super(cause); }
}
class Product {
    private final int id;
    private int price;
    private int count;
    private final int days;
    private final String name;
    private final String upc;
    private final String creator;

    public Product(int id, int price, int count, int days,
String name, String upc, String creator) throws
ProductException {
        if (id <= 0 || price <= 0 || count <= 0 || days <=
0 || Objects.equals(name, "") ||
            Objects.equals(upc, "") ||
Objects.equals(creator, "")) {
            throw new ProductException("some params are not
present");
        }
        this.id = id;
        this.price = price;
        this.count = count;
        this.days = days;
        this.name = name;
        this.upc = upc;
        this.creator = creator;
    }

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

```

```

    }
    public void setCount(int count) {
        this.count = count;
    }
    public String getName() {
        return this.name;
    }
    public int getPrice() {
        return this.price;
    }
    public int getCount() {
        return this.count;
    }
    public int getDays() {
        return this.days;
    }
    @Override
    public String toString() {
        return String.format("%d - %s - %s - %s - %d",
this.id, this.name, this.upc, this.creator, this.days);
    }
}
public class Variant29 {
    public static void main(String[] args) {
        try {
            var products = new Product[]{
                new Product(1, 2, 3, 4, "n1", "u2",
"c3"),
                new Product(2, 2, 3, 10, "n1", "u3",
"c3")
            };

```

```

var name = "n1";
var price = 5;
var days = 7;
System.out.println("a");
for (Product pr : products) {
    if (pr.getName().equals(name)) {
        System.out.println(pr);
    }
}
System.out.println("6");
for (Product pr : products) {
    if (pr.getName().equals(name) &&
pr.getPrice() <= price) {
        System.out.println(pr);
    }
}
System.out.println("B");
for (Product pr : products) {
    if (pr.getDays() > days) {
        System.out.println(pr);
    }
}
} catch (ProductException e) {
    System.out.println(e.getMessage());
}
}
}

```

```
a
1 - n1 - u2 - c3 - 4
2 - n1 - u3 - c3 - 10
6
1 - n1 - u2 - c3 - 4
2 - n1 - u3 - c3 - 10
В
2 - n1 - u3 - c3 - 10

Process finished with exit code 0
```

Рисунок 3 – Результат задания 3

Задание 4. Выполнить задания из варианта 2 лабораторной работы 3, реализуя собственные обработчики исключений и исключения ввода/вывода.

Код программы представлен в листинге 4, результат – на рисунке 4.

Листинг 4 – Код программы задания 4

```
package Lab5;
import java.util.Objects;
class TrainException extends Exception {
    public TrainException() { super(); }
    public TrainException(String message) { super(message); }
}
    public TrainException(String message, Throwable cause)
{ super(message, cause); }
    public TrainException(Throwable cause) { super(cause); }
}
}
class Train {
    private final String destination;
    private final String number;
    private int departureTime;
    private final int generalCount;
```

```

    private final int compartmentCount;
    private final int reservedCount;
    private final int suiteCount;

    public Train(String destination, String number, int
departureTime, int generalCount, int compartmentCount, int
reservedCount, int suiteCount) throws TrainException {
        if (Objects.equals(destination, "") ||
Objects.equals(number, "") ||
            departureTime <= 0 || generalCount <= 0 ||
compartmentCount <= 0 || reservedCount <= 0 || suiteCount
<= 0) {
            throw new TrainException("some params are not
present");
        }
        this.destination = destination;
        this.number = number;
        this.departureTime = departureTime;
        this.generalCount = generalCount;
        this.compartmentCount = compartmentCount;
        this.reservedCount = reservedCount;
        this.suiteCount = suiteCount;
    }

    public void setDepartureTime(int departureTime) {
        this.departureTime = departureTime;
    }

    public String getDestination() {
        return this.destination;
    }

    public int getDepartureTime() {
        return this.departureTime;
    }
}

```

```

    public int getGeneralCount() {
        return this.generalCount;
    }
    @Override
    public String toString() {
        return String.format("%s - %s - %d", this.number,
this.destination, this.departureTime);
    }
}
public class Variant210 {
    public static void main(String[] args) {
        try {
            var trains = new Train[]{
                new Train("Moscow", "1", 11, 50, 1, 10,
1),
                new Train("London", "2", 5, 1, 50, 10,
1)

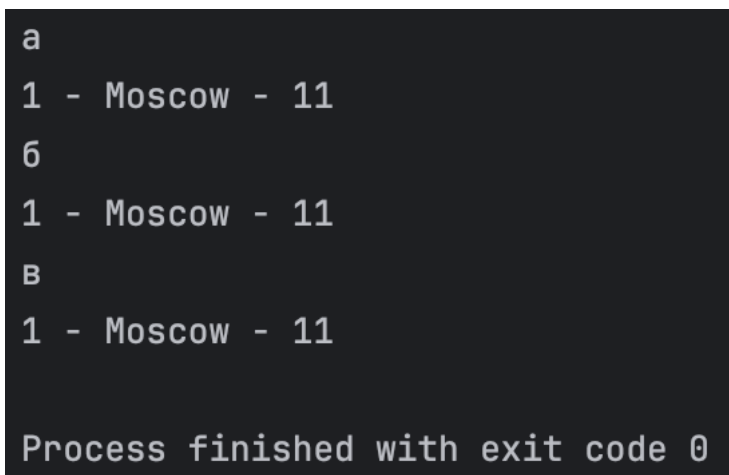
            };
            var destination = "Moscow";
            var departureTime = 10;
            var generalCount = 0;
            System.out.println("a");
            for (Train tr : trains) {
                if
(tr.getDestination().equals(destination)) {
                    System.out.println(tr);
                }
            }
            System.out.println("б");
            for (Train tr : trains) {

```

```

        if (tr.getDestination().equals(destination)
&& tr.getDepartureTime() > departureTime) {
            System.out.println(tr);
        }
    }
    System.out.println("В");
    for (Train tr : trains) {
        if (tr.getDestination().equals(destination)
&& tr.getGeneralCount() > generalCount) {
            System.out.println(tr);
        }
    }
} catch (TrainException e) {
    System.out.println(e.getMessage());
}
}
}

```



```

а
1 - Moscow - 11
б
1 - Moscow - 11
в
1 - Moscow - 11
Process finished with exit code 0

```

Рисунок 4 – Результат задания 4

Задание 5. 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.

Код программы представлен в листинге 5, результат – на рисунке 5.

Листинг 5 – Код программы задания 5

```
package Lab5;

import java.io.FileReader;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Arrays;

public class Variant39 {

    public static void main(String[] args) {

        var list = new ArrayList<ArrayList<String>>();

        try (BufferedReader br = new BufferedReader(new
FileReader("src/Lab5/input_39.txt"))) {

            for (String line; (line = br.readLine()) !=
null; ) {

                list.add(new
ArrayList<>(Arrays.asList(line.split(" ")))));

            }

        } catch (Exception e) {

            System.out.println(e.getMessage());

        }

        try (var fw = new
FileWriter("src/Lab5/output_39.txt")) {

            for (int i = 0; i < list.size(); ++i) {

                for (int j = 0; j <list.getFirst().size();
++j) {

                    fw.write(list.get(i).get(j) + " ");

                }

                fw.write("\r\n");

            }

            fw.write("\r\n");

        }

    }

}
```

```

        for (int i = 0; i < list.getFirst().size();
++i) {
            for (int j = 0; j < list.size(); ++j) {
                fw.write(list.get(j).get(i) + " ");
            }
            fw.write("\r\n");
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}

```

1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	16	17	18	19	20
5	21	22	23	24	25
6					
7	1	6	11	16	21
8	2	7	12	17	22
9	3	8	13	18	23
10	4	9	14	19	24
11	5	10	15	20	25
12					

Рисунок 5 – Результат задания 5

Задание 6. 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.

Код программы представлен в листинге 6, результат – на рисунке 6.

Листинг 6 – Код программы задания 6

```

package Lab5;

import java.io.FileReader;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.util.ArrayList;

public class Variant310 {

    public static void main(String[] args) {
        var list = new ArrayList<String>();

        try (BufferedReader br = new BufferedReader(new
FileReader("src/Lab5/input_310.txt"))) {
            for (String line; (line = br.readLine()) !=
null; ) {
                list.add(line);
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }

        var dimension = (int)Math.sqrt(list.size());
        var newList = new ArrayList<ArrayList<String>>();
        var clockWisedList = new
ArrayList<ArrayList<String>>();

        for (int i = 0; i < dimension; ++i) {
            var temp = new ArrayList<String>();
            for (int j = 0; j < dimension; ++j) {
                temp.add(list.get(i * dimension + j));
            }
            newList.add(temp);
            clockWisedList.add((ArrayList<String>)
temp.clone());
        }
    }
}

```

```

        for (int i = 0; i <
clockWisedList.getFirst().size(); ++i) {
            for (int j = 0; j < clockWisedList.size(); ++j)
{

clockWisedList.get(j).set(clockWisedList.getFirst().size()
- 1 - i, newList.get(i).get(j));
                }
            }
            try (var fw = new
FileWriter("src/Lab5/output_310.txt")) {
                for (int i = 0; i < newList.size(); ++i) {
                    for (int j = 0; j <
newList.getFirst().size(); ++j) {
                        fw.write(newList.get(i).get(j) + " ");
                    }
                    fw.write("\r\n");
                }
                fw.write("\r\n");
                for (int i = 0; i < clockWisedList.size(); ++i)
{
                    for (int j = 0; j <
clockWisedList.getFirst().size(); ++j) {
                        fw.write(clockWisedList.get(i).get(j) +
" ");
                    }
                    fw.write("\r\n");
                }
            } catch (Exception e) {
                System.out.println(e.getMessage());
            }
}

```

```

    }
}

```

1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	16	17	18	19	20
5	21	22	23	24	25
6					
7	21	16	11	6	1
8	22	17	12	7	2
9	23	18	13	8	3
10	24	19	14	9	4
11	25	20	15	10	5
12					

Рисунок 6 – Результат задания 6

Задание 7. Прочитать строки из файла и поменять местами первое и последнее слова в каждой строке.

Код программы представлен в листинге 7, результат – на рисунке 7.

Листинг 7 – Код программы задания 7

```

package Lab5;

import java.io.FileReader;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Arrays;

public class Variant49 {
    public static void main(String[] args) {
        var list = new ArrayList<ArrayList<String>>();
        try (BufferedReader br = new BufferedReader(new
FileReader("src/Lab5/input_49.txt"))) {

```

```

        for (String line; (line = br.readLine()) !=
null; ) {
            list.add(new
ArrayList<>(Arrays.asList(line.split(" ")));
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    try (var fw = new
FileWriter("src/Lab5/output_49.txt")) {
        for (ArrayList<String> line : list) {
            var temp = line.getFirst();
            line.set(0, line.getLast());
            line.set(line.size() - 1, temp);
            for (String s : line) {
                fw.write(s + " ");
            }
            fw.write("\r\n");
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}

```

1	5	2	3	4	1
2	10	7	8	9	6
3	15	12	13	14	11
4	20	17	18	19	16
5	25	22	23	24	21
6					

Рисунок 7 – Результат задания 7

Задание 8. Ввести из текстового файла, связанного с входным потоком, последовательность строк. Выбрать и сохранить *m* последних слов в каждой из последних *n* строк.

Код программы представлен в листинге 8, результат – на рисунке 8.

Листинг 8 – Код программы задания 8

```
package Lab5;

import java.io.FileReader;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Arrays;

public class Variant410 {

    public static void main(String[] args) {
        var list = new ArrayList<ArrayList<String>>();
        try (BufferedReader br = new BufferedReader(new
FileReader("src/Lab5/input_410.txt"))) {
            for (String line; (line = br.readLine()) !=
null; ) {
                list.add(new
ArrayList<>(Arrays.asList(line.split(" ")))));
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }

        int m = 3, n = 2;
        try (var fw = new
FileWriter("src/Lab5/output_410.txt")) {
            for (int i = list.size() - n; i < list.size();
++i) {
                var line = list.get(i);
```

```

        for (int j = line.size() - m; j <
line.size(); ++j) {
            fw.write(list.get(i).get(j) + " ");
        }
        fw.write("\r\n");
    }
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}
}

```

1	18 19 20
2	23 24 25
3	

Рисунок 8 – Результат задания 8

Вывод

В ходе выполнения лабораторной работы была изучена работа с исключениями и файлами на языке программирования Java.