



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 09.04.01/12 Интеллектуальный анализ больших
данных в системах поддержки принятия решений.

О Т Ч Е Т

по лабораторной работе № 4

Вариант № 4

Название: внутренние классы и интерфейсы

Дисциплина: языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

А.А.Клушина

(И.О. Фамилия)

Преподаватель

П.В. Степанов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2024

Цель: изучить работу с исключениями и 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.

Код класса Main:

```
import java.util.InputMismatchException;

public class Matrix {
    private int[][] matrix;
    private int size;

    public Matrix(int size) {
        if (size <= 0) {
            throw new IllegalArgumentException("Размер матрицы
должен быть положительным целым числом");
        }
        this.size = size;
        this.matrix = new int[size][size];
    }

    public Matrix(int size, int[][] matrix) {
        if (size <= 0) {
            throw new IllegalArgumentException("Размер матрицы
должен быть положительным целым числом");
        }
        this.size = size;
        this.matrix = matrix;
    }

    public Matrix add(Matrix other) {
        if (other.size != size) {
            throw new IllegalArgumentException("Matrix sizes
must be equal for addition");
        }

        int[][] result = new int[size][size];
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                result[i][j] = this.matrix[i][j] +
other.matrix[i][j];
            }
        }
    }
}
```

```

        return new Matrix(size, result);
    }

    public Matrix subtract(Matrix other) {
        if (other.size != size) {
            throw new IllegalArgumentException("Матрицы должны
быть одной размерности");
        }

        int[][] result = new int[size][size];
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                result[i][j] = this.matrix[i][j] -
other.matrix[i][j];
            }
        }
        return new Matrix(size, result);
    }

    public Matrix multiply(Matrix other) {
        if (other.size != size) {
            throw new IllegalArgumentException("Матрицы должны
быть одной размерности");
        }

        int[][] result = new int[size][size];
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                int sum = 0;
                for (int k = 0; k < size; k++) {
                    sum += this.matrix[i][k] *
other.matrix[k][j];
                }
                result[i][j] = sum;
            }
        }
        return new Matrix(size, result);
    }

    public double calculateFirstNorm() {
        double norm = 0;
        for (int i = 0; i < size; i++) {
            double sum = 0;
            for (int j = 0; j < size; j++) {
                sum += Math.abs(matrix[i][j]);
            }
            norm = Math.max(norm, sum);
        }
        return norm;
    }

    public double calculateSecondNorm() {
        double norm = 0;

```

```

        for (int i = 0; i < size; i++) {
            double sum = 0;
            for (int j = 0; j < size; j++) {
                sum += Math.abs(matrix[j][i]);
            }
            norm = Math.max(norm, sum);
        }
        return norm;
    }

    public static void printMatrix(Matrix matrix) {
        System.out.println("Матрица:");
        for (int i = 0; i < matrix.size; i++) {
            for (int j = 0; j < matrix.size; j++) {
                System.out.print(matrix.matrix[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        try {
            Matrix[] matrices = new Matrix[3]; // Создаем массив
объектов

            // Создаем и инициализируем матрицы
            int[][] matrix1 = {{1, 2}, {3, 4}};
            matrices[0] = new Matrix(2, matrix1);

            int[][] matrix2 = {{5, 6}, {7, 8}};
            matrices[1] = new Matrix(2, matrix2);

            int[][] matrix3 = {{-1, -2}, {-3, -4}};
            matrices[2] = new Matrix(2, matrix3);

            // Находим матрицу с наименьшей первой и второй
нормами
            Matrix minMatrix = matrices[0];
            for (int i = 1; i < matrices.length; i++) {
                if (matrices[i].calculateFirstNorm() <
minMatrix.calculateFirstNorm() &&
                    matrices[i].calculateSecondNorm() <
minMatrix.calculateSecondNorm()) {
                    minMatrix = matrices[i];
                }
            }
            System.out.println("Матрица с наименьшей первой и второй
нормами:");
            printMatrix(minMatrix);
            System.out.println("Первая норма: " +
minMatrix.calculateFirstNorm());
            System.out.println("Вторая норма: " +
minMatrix.calculateSecondNorm());

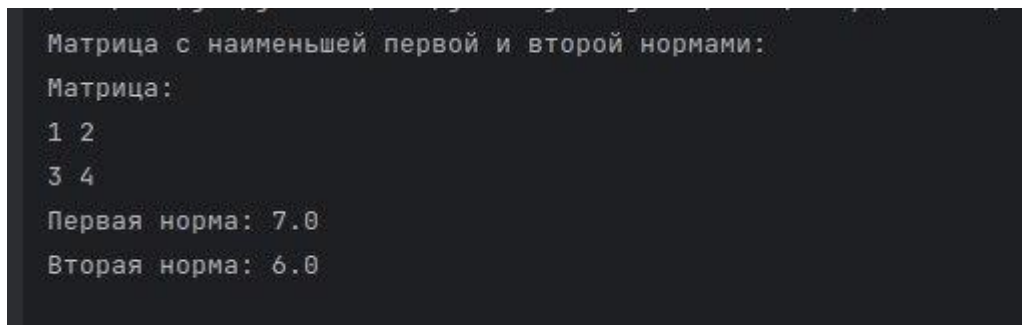
```

```

        } catch (IllegalArgumentException e) {
            System.err.println("Возникла ошибка: " +
e.getMessage());
        } catch (InputMismatchException e) {
            System.err.println("Ошибка ввода: " +
e.getMessage());
        } catch (Exception e) {
            System.err.println("Неожиданная ошибка: " +
e.getMessage());
        }
    }
}
}

```

Работа программы показана на рисунке 1.



```

Матрица с наименьшей первой и второй нормами:
Матрица:
1 2
3 4
Первая норма: 7.0
Вторая норма: 6.0

```

Рисунок 1 – Работа программы

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 класса Main:

```

import java.util.InputMismatchException;
import java.util.Scanner;

class Matrix {
    private int rows;
    private int cols;
    private int[][] data;

    public Matrix(int rows, int cols) {
        this.rows = rows;
        this.cols = cols;
    }
}

```

```

        this.data = new int[rows][cols];
    }

    public Matrix() {
        this(2, 2);
    }

    public void fill() {
        Scanner input = new Scanner(System.in);
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                try {
                    System.out.print("Введите элемент матрицы ("
+ i + ", " + j + "): ");
                    data[i][j] = input.nextInt();
                } catch (InputMismatchException e) {
                    System.out.println("Ошибка: введено
некорректное значение. Попробуйте снова.");
                    input.nextLine(); // очистка буфера ввода
                    j--; // повторить ввод для этой ячейки
                }
            }
        }
    }

    public void printMatrix() {
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(data[i][j] + " ");
            }
            System.out.println();
        }
    }

    public void swapRowsWithMinMax(int k) {
        int minElement = data[0][k];
        int maxElement = data[0][k];
        int minRow = 0;
        int maxRow = 0;
        for (int i = 1; i < rows; i++) {
            if (data[i][k] < minElement) {
                minElement = data[i][k];
                minRow = i;
            }
            if (data[i][k] > maxElement) {
                maxElement = data[i][k];
                maxRow = i;
            }
        }
        int[] temp = data[minRow];
        data[minRow] = data[maxRow];
        data[maxRow] = temp;
    }
}

```

```

    public void squareMatrix() {
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                data[i][j] *= data[i][j];
            }
        }
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        try {
            System.out.print("Введите количество строк матрицы:");
            int rows = input.nextInt();
            System.out.print("Введите количество столбцов матрицы: ");
            int cols = input.nextInt();
            Matrix matrix = new Matrix(rows, cols);
            matrix.fill();
            System.out.println("Матрица:");
            matrix.printMatrix();
            System.out.print("Введите столбец k: ");
            int k = input.nextInt();
            matrix.swapRowsWithMinMax(k);
            System.out.println("Матрица после перемены строк:");
            matrix.printMatrix();
            matrix.squareMatrix();
            System.out.println("Матрица после возведения в квадрат:");
            matrix.printMatrix();
        } catch (InputMismatchException e) {
            System.out.println("Ошибка: введено некорректное значение для количества строк или столбцов матрицы.");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Ошибка: размер матрицы меньше введенного значения строк или столбцов.");
        } finally {
            input.close(); // закрытие потока ввода
        }
    }
}

```

Работа программы показана на рисунке 2.

```

Введите количество строк матрицы: 2
Введите количество столбцов матрицы: 3
Введите элемент матрицы (0, 0): 1
Введите элемент матрицы (0, 1): 2
Введите элемент матрицы (0, 2): 3
Введите элемент матрицы (1, 0): 4
Введите элемент матрицы (1, 1): 5
Введите элемент матрицы (1, 2): 6
Матрица:
1 2 3
4 5 6
Введите столбец k: 2
Матрица после перемены строк:
4 5 6
1 2 3
Матрица после возведения в квадрат:
16 25 36
1 4 9

```

Рисунок 2 – Работа программы

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

Код класса Main:

```

import java.util.InputMismatchException;
import java.util.Scanner;
import java.util.ArrayList;

public class Abiturient {
    private int id;
    private String surname;
    private String firstName;
    private String middleName;
    private String address;
    private String phoneNumber;
    private int[] grades;

    public Abiturient(int id, String surname, String firstName,
String middleName, String address, String phoneNumber, int[]
grades) {
        this.id = id;
        this.surname = surname;
        this.firstName = firstName;

```



```

        this.middleName = middleName;
        this.address = address;
        this.phoneNumber = phoneNumber;
        this.grades = grades;
    }

    public double calculateAverageGrade() {
        int sum = 0;
        for (int grade : grades) {
            sum += grade;
        }
        return (double) sum / grades.length;
    }

    public int getId() {
        return id;
    }

    public String getFullName() {
        return surname + " " + firstName + " " + middleName;
    }

    public int[] getGrades() {
        return grades;
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        ArrayList<Abiturient> abiturients = new ArrayList<>();
        abiturients.add(new Abiturient(1, "Чернавина", "Ксения",
            "Максимовна", "Ярославль", "123456789", new int[]{80, 70, 60}));
        abiturients.add(new Abiturient(2, "Клушина",
            "Анастасия", "Алексеевна", "Москва", "987654321", new int[]{90,
            85, 75}));
        abiturients.add(new Abiturient(3, "Тюкин", "Иван",
            "Дмитриевич", "Владивосток", "555555555", new int[]{50, 55,
            70}));

        try {
            // а) список абитуриентов, имеющих
            неудовлетворительные оценки
            System.out.println("Список абитуриентов, имеющих
            неудовлетворительные оценки:");
            for (Abiturient abiturient : abiturients) {
                for (int grade : abiturient.getGrades()) {
                    if (grade < 60) {

                        System.out.println(abiturient.getFullName());
                        break;
                    }
                }
            }
        }
    }

```

```

        // b) список абитуриентов, средний балл у которых
        выше заданного
        System.out.print("Введите балл: ");
        double minAverageGrade = input.nextDouble();
        System.out.println("Список абитуриентов, средний
        балл у которых выше " + minAverageGrade + ":");
        for (Abiturient abiturient : abiturients) {
            if (abiturient.calculateAverageGrade() >
            minAverageGrade) {

                System.out.println(abiturient.getFullName());
            }
        }

        // c) выбрать заданное число n абитуриентов, имеющих
        самый высокий средний балл
        System.out.print("Введите количество абитуриентов:
        ");

        int n = input.nextInt();
        abiturients.sort((a, b) ->
        Double.compare(b.calculateAverageGrade(),
        a.calculateAverageGrade()));
        System.out.println("Топ " + n + " абитуриентов с
        максимальным средним баллом:");
        for (int i = 0; i < n && i < abiturients.size();
        i++) {

            System.out.println(abiturients.get(i).getFullName());
        }

        // полный список абитуриентов с полупроходным баллом
        double passingGrade = 60.0;
        System.out.println("Список абитуриентов с
        полупроходным баллом:");
        for (int i = n; i < abiturients.size(); i++) {
            if (abiturients.get(i).calculateAverageGrade()
            >= passingGrade) {

                System.out.println(abiturients.get(i).getFullName());
            }
        }
    } catch (InputMismatchException e) {
        System.out.println("Ошибка: введено некорректное
        значение.");
    } finally {
        input.close(); // закрытие потока ввода
    }
}
}

```

Работа программы показана на рисунке 3.

```
/usr/lib/jvm/java-11/bin/java -javaagent:/home/nusya/.loc
Введите автора: Достоевский
Книги Достоевский:
Название: Преступление и наказание
Автор(ы): Достоевский
Издательство: Москва
Год: 2000
Количество страниц: 300
Цена: 20.0 руб.
Переплет: Твердый

Введите название издательства: АСТ
Книги АСТ:
Название: 1984
Автор(ы): Оруэлл
Издательство: АСТ
Год: 2010
Количество страниц: 400
Цена: 30.0 руб.
Переплет: Твердый

Введите год издательства: 2000
Книги изданные после 2000:
Название: Дракула
Автор(ы): Стокер
Издательство: Издательство
Год: 2005
Количество страниц: 250
Цена: 15.0 руб.
Переплет: Мягкий

Название: 1984
Автор(ы): Оруэлл
Издательство: АСТ
Год: 2010
Количество страниц: 400
Цена: 30.0 руб.
Переплет: Твердый
```

Рисунок 3 – Работа программы

Задание 4: Найти и вывести слова текста, для которых последняя буква одного слова совпадает с первой буквой следующего слова.

Код класса Main:

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Main {
    public static void main(String[] args) {
        try {
```

```

        BufferedReader reader = new BufferedReader(new
FileReader("input.txt"));
        BufferedWriter writer = new BufferedWriter(new
FileWriter("output.txt"));
        String text;
        while ((text = reader.readLine()) != null) {

            String[] words = text.split("\\s+");

            for (int i = 0; i < words.length - 1; i++) {
                String currentWord = words[i];
                String nextWord = words[i + 1];
                System.out.println(currentWord);
                System.out.println(nextWord);

                if (currentWord.charAt(currentWord.length()
- 1) == nextWord.charAt(0)) {
                    writer.write(currentWord + " " +
nextWord + System.lineSeparator());
                }
            }

            reader.close();
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Работа программы показана на рисунке 4.

```
7031 / c:/jvm/java-11/bin/java -javaagent:/home/nosysa/.com
Я
Анастасия
Анастасия
Клушина
Клушина
студентка
студентка
МГТУ
МГТУ
Баумана
Я
хочу
хочу
спать
спать
и
и
хочу
хочу
есть
учу
ученый
ученый
свет

Process finished with exit code 0
|
```

Рисунок 4 – Работа программы

Задание 5: Найти в строке наибольшее число цифр, идущих подряд.

Код класса Main;

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Main {
    public static void main(String[] args) {
        try {
            BufferedReader reader = new BufferedReader(new
            FileReader("input.txt"));
            BufferedWriter writer = new BufferedWriter(new
            FileWriter("output.txt"));

            String text = reader.readLine();

            int maxDigits = 0;
            int curDigits = 0;

            for (int i = 0; i < text.length(); i++) {
                char c = text.charAt(i);
```

```

        if (Character.isDigit(c)) {
            curDigits++;
        } else {
            if (curDigits > maxDigits) {
                maxDigits = curDigits;
            }
            curDigits = 0;
        }
    }

    if (curDigits > maxDigits) {
        maxDigits = curDigits;
    }

    writer.write("Наибольшее число цифр, идущих подряд:
" + maxDigits);

    reader.close();
    writer.close();
} catch (IOException e) {
    System.err.println("Ошибка при чтении/записи
файла");
}
}
}

```

Работа программы показана на рисунках 5-6.

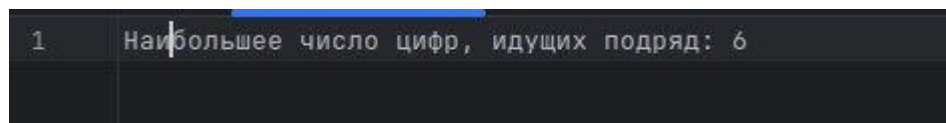


Рисунок 5 – Работа программы

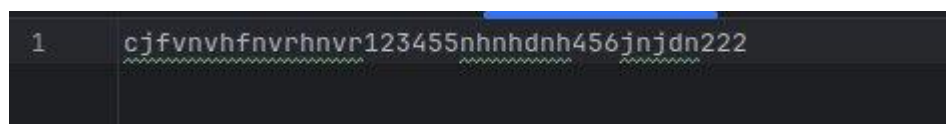


Рисунок 6 – Работа программы

Задание 6: В файле, содержащем фамилии студентов и их оценки, записать прописными буквами фамилии тех студентов, которые имеют средний балл более “7”.

Код класса Main:

```
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;

public class Main {
    public static void main(String[] args) {
        try {
            // Создаем новую директорию
            File directory = new File("results");
            directory.mkdir();

            // Создаем файл с фамилиями студентов и оценками
            File studentsFile = new File(directory,
"students.txt");
            studentsFile.createNewFile();

            // Записываем в файл данные студентов
            BufferedWriter writer = new BufferedWriter(new
FileWriter(studentsFile));
            writer.write("Клушина 7.5\n");
            writer.write("Тюкин 8.0\n");
            writer.write("Чернавина 6.9\n");
            writer.write("Платонов 9.5\n");
            writer.close();

            // Создаем файл для записи результатов
            File output = new File(directory, "output.txt");
            output.createNewFile();

            // Чтение данных из файла
            BufferedReader reader = new BufferedReader(new
FileReader(studentsFile));
            BufferedWriter resultWriter = new BufferedWriter(new
FileWriter(output));

            String line;
            while ((line = reader.readLine()) != null) {
                String[] data = line.split("\\s+");

                if (data.length > 1) {
                    if (data[1].contains(".")) {
                        double score =
Double.parseDouble(data[1]);
                        if (score > 7) {
resultWriter.write(data[0].toUpperCase() + "\n");
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }

    reader.close();
    resultWriter.close();

    System.out.println("Результаты записаны в файл " +
output.getPath());
    } catch (IOException e) {
        System.err.println("Ошибка при работе с файлами");
    }
}
}

```

Работа программы показана на рисунке 7-8.

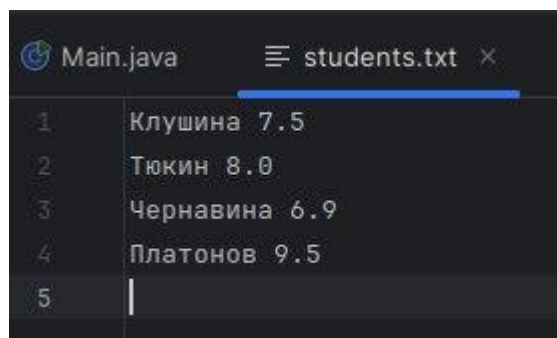


Рисунок 7 – Исходные данные

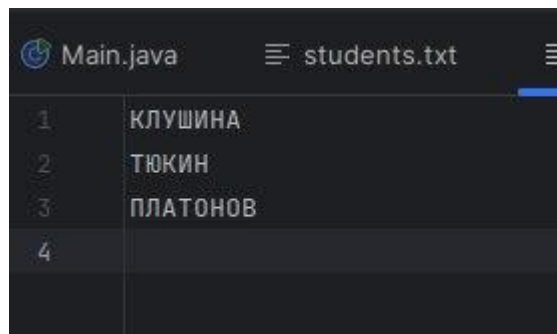


Рисунок 8 – Работа программы

Задание 8: Файл содержит символы, слова, целые числа и числа с плавающей запятой. Определить все данные, тип которых вводится из командной строки.

Код класса Main:

```
import java.io.File;
```



```

import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // Создание новой директории
        File directory = new File("results");
        if (!directory.exists()) {
            directory.mkdir();
            System.out.println("Директория создана: results");
        }

        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите тип данных (char, String,
int, double):");
        String dataType = scanner.nextLine();

        try {
            File inputFile = new File("input.txt");
            File outputFile = new File("results/output.txt");
            Scanner fileScanner = new Scanner(inputFile);
            FileWriter writer = new FileWriter(outputFile);

            while (fileScanner.hasNext()) {
                if (dataType.equals("char")) {
                    String data = fileScanner.next();
                    if (data.length() == 1) {
                        writer.write(data + " ");
                    }
                } else if (dataType.equals("String")) {
                    String data = fileScanner.next();
                    writer.write(data + " ");
                } else if (dataType.equals("int")) {
                    if (fileScanner.hasNextInt()) {
                        int data = fileScanner.nextInt();
                        writer.write(data + " ");
                    } else {
                        fileScanner.next();
                    }
                } else if (dataType.equals("double")) {
                    if (fileScanner.hasNextDouble()) {
                        double data = fileScanner.nextDouble();
                        writer.write(data + " ");
                    } else {
                        fileScanner.next();
                    }
                }
            }

            fileScanner.close();
            writer.close();
        }
    }
}

```

```

        System.out.println("Результаты сохранены в файле
results/output.txt");

    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Работа программы показана на рисунках 9 – 10.

```

/usr/lib/jvm/java-11/bin/java -javaagent:/home/husya/.
Введите тип данных (char, String, int, double):
int
Результаты сохранены в файле results/output.txt

Process finished with exit code 0
|

```

Рисунок 9 – Работа программы

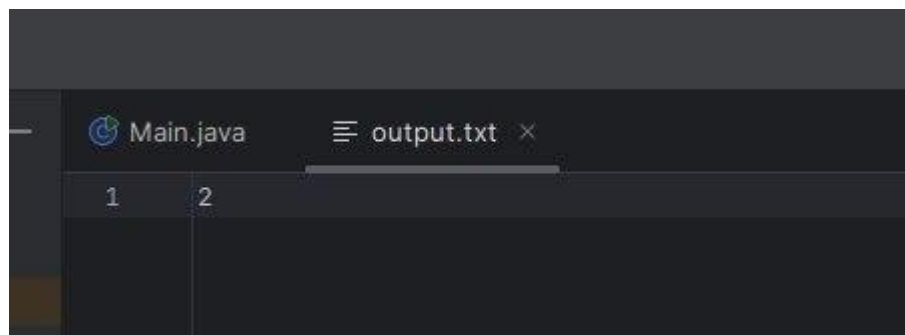


Рисунок 10 – Работа программы

Вывод: были изучена работа с исключениями и файлами в java.