



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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И  
СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Интеллектуальные системы анализа,  
обработки и интерпретации больших данных

## ОТЧЕТ

по лабораторной работе № 5

Вариант 13

Название: Исключения, файлы

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-22М

(Группа)

(Подпись, дата)

В.А.Ловцов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(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:

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

class NilDenominatorException extends Exception {
    public NilDenominatorException(String mess){
        super(mess);
    }
}

class Point {
    double x, y, z;

    public Point(double x, double y, double z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }

    public void printPoint() {
        System.out.print("(" + this.x + ", " + this.y + ", " +
this.z + ")", " ");
    }
}

class PointReader {
    public Point readPoint(Scanner scanner){
//        System.out.println("");
        System.out.print("\tx=");
        double x = scanner.nextDouble();
        System.out.print("\ty=");
        double y = scanner.nextDouble();
        System.out.print("\tz=");
```

```

        double z = scanner.nextDouble();

        return new Point(x, y, z);
    }
}

class Vector {
    Point a, b;

    public Vector(Point a, Point b) {
        this.a = a;
        this.b = b;
    }

    double getLen() {
        return Math.sqrt(this.getLenX() * this.getLenX() +
            this.getLenY() * this.getLenY() +
            this.getLenZ() * this.getLenZ());
    }

    public double getLenX() {
        return this.b.x - this.a.x;
    }

    public double getLenY() {
        return this.b.y - this.a.y;
    }

    public double getLenZ() {
        return this.b.z - this.a.z;
    }

    public double getScalar(Vector other) {
        return (this.getLenX() * other.getLenX() +
            this.getLenY() * other.getLenY() +
            this.getLenZ() * other.getLenZ());
    }

    public double getSin(Vector other) throws
    NilDenominatorException {
        double len_mul = this.getLen() * other.getLen();
        if (len_mul == 0)
            throw new NilDenominatorException("Произведение
длинн равно нулю! Деление на ноль");
        return this.getScalar(other) / (len_mul);
    }

    public double getVectMul(Vector other) throws
    NilDenominatorException {
        double sinus = this.getSin(other);
        if (sinus == 0)

```

```

        throw new NilDenominatorException("Синус равен
        нулю! Деление на ноль");
        return (this.getLen() * other.getLen()) /
this.getSin(other);
    }

    public Vector getVectorVector(Vector other) {
        double tmpXb = this.getLenY() * other.getLenZ();
        double tmpXa = this.getLenZ() * other.getLenY();

        double tmpYb = this.getLenZ() * other.getLenX();
        double tmpYa = this.getLenX() * other.getLenZ();

        double tmpZb = this.getLenX() * other.getLenY();
        double tmpZa = this.getLenY() * other.getLenX();

        Point tmpB = new Point(tmpXb, tmpYb, tmpZb);
        Point tmpA = new Point(tmpXa, tmpYa, tmpZa);

        return new Vector(tmpA, tmpB);
    }

    public double getMixMul(Vector other1, Vector other2) {
        Vector tmpBC = other1.getVectorVector(other2);
        return this.getScalar(tmpBC);
    }

    public boolean isComplanar(Vector other1, Vector other2) {
        return (this.getMixMul(other1, other2) == 0);
    }

    public boolean isCross(Vector other) throws
NilDenominatorException {
        double v1 = other.getVectMul(new Vector(other.a,
this.a));
        double v2 = other.getVectMul(new Vector(other.a,
this.b));

        double v3 = this.getVectMul(new Vector(this.a,
other.a));
        double v4 = this.getVectMul(new Vector(this.a,
other.b));

        return (v1 * v2 < 0 & v3 * v4 < 0);
    }

    public boolean isOrt(Vector other) {
        return (this.getScalar(other) == 0);
    }

    public void printVector() {

```

```

        this.a.printPoint();
        this.b.printPoint();
        System.out.println();
    }
}

public class Third_1 {
    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in)) {
            System.out.print("Введите n: ");
            int n = scanner.nextInt();

            scanner.nextLine();
            Vector[] array = new Vector[n];
            PointReader reader = new PointReader();
            for (int i = 0; i < n; i++) {
                System.out.println(i + 1 + " ");

                System.out.println("Точка A:");
                Point tmpA = reader.readPoint(scanner);
                System.out.println("Точка B:");
                Point tmpB = reader.readPoint(scanner);
                array[i] = new Vector(tmpA, tmpB);
            }
            //      System.out.print("Введите n: ");
            for (int i = 0; i < n - 2; i++) {
                for (int j = i + 1; j < n - 1; j++) {
                    for (int k = j + 1; k < n; k++) {
                        if (array[i].isComplanar(array[j],
array[k])) {

System.out.println("Компланарные:");
                                array[i].printVector();
                                array[j].printVector();
                                array[k].printVector();
                            }
                        }
                    }
                }
            }
        } catch (InputMismatchException e) {
            System.out.println("Неверный тип данных");
        } catch (OutOfMemoryError e) {
            System.out.println("Переполнение ОП");
        }
    }
}

```

Работа программы показана на рисунке 1.

```
/Library/Java/JavaVirtualMachines/jdk-2.
Введите n: 1,5
Неверный тип данных

Process finished with exit code 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:

```
import java.util.Arrays;
import java.util.InputMismatchException;
import java.util.Scanner;

class MatrixShapeException extends Exception {
    public MatrixShapeException(String mess){
        super(mess);
    }
}

class Matrix {
    int[][] mat;
    int n;

    public Matrix(int n) {
        this.n = n;
        mat = new int[n][n];
        for (int i = 0; i < n; i++)
            for (int j = 0; j < n ; j++) {
                mat[i][j] = (int) (Math.random() * 10);
            }
    }
}
```

```

public Matrix(int n, int val) {
    this.n = n;
    mat = new int[n][n];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n ; j++) {
            mat[i][j] = val;
        }
}

public Matrix(int[][] arr) {
    this.n = arr.length;
    mat = new int[this.n][this.n];
    for (int i = 0; i < n; i++)
        System.arraycopy(arr[i], 0, mat[i], 0, n);
}

public Matrix add(Matrix other) throws
MatrixShapeException {
    if (n != other.n)
        throw new MatrixShapeException("Несоответствие
размеров матриц");
    int[][] summa = new int[this.n][this.n];

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++) {
            summa[i][j] = this.mat[i][j] +
other.mat[i][j];
        }
    return new Matrix(summa);
}

public Matrix sub(Matrix other) throws
MatrixShapeException {
    if (n != other.n)
        throw new MatrixShapeException("Несоответствие
размеров матриц");
    int[][] sub = new int[this.n][this.n];

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++) {
            sub[i][j] = this.mat[i][j] - other.mat[i][j];
        }
    return new Matrix(sub);
}

public Matrix mul(Matrix other) throws
MatrixShapeException {
    if (n != other.n)
        throw new MatrixShapeException("Несоответствие
размеров матриц");
}

```

```

        int[][] mul = new int[this.n][this.n];

        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++) {
                int line_sum = 0;
                for (int k = 0; k < n; k++){
                    line_sum += this.mat[i][k] *
other.mat[k][j];
                }
                mul[i][j] = line_sum;
            }
        return new Matrix(mul);
    }

    public int l1() {
        int max = -99999999;
        for (int i = 0; i < n; i++) {
            int cur_sum = 0;
            for (int j = 0; j < n; j++) {
                cur_sum += mat[i][j];
            }
            if (cur_sum > max)
                max = cur_sum;
        }
        return max;
    }

    public int l2() {
        int max = -99999999;
        for (int j = 0; j < n; j++) {
            int cur_sum = 0;
            for (int i = 0; i < n; i++) {
                cur_sum += mat[i][j];
            }
            if (cur_sum > max)
                max = cur_sum;
        }
        return max;
    }

    public void printMat() {
        for (int i = 0; i < this.n; i++)
            System.out.println(Arrays.toString(this.mat[i]));
        System.out.println();
    }
}

public class Fourth_1 {
    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in)) {

```



```

        System.out.print("Введите n >= 1: ");
        int n = scanner.nextInt();
        if (n < 1)
            throw new MatrixShapeException("n должно быть
не меньше 1");

        System.out.print("Введите m >= 3: ");
        int m = scanner.nextInt();
        if (m < 1)
            throw new MatrixShapeException("m должно быть
не меньше 3");

        scanner.nextLine();
        Matrix[] array = new Matrix[m];

        int l1_min = 999999999;
        int l2_min = 999999999;
        int l1_id = -1, l2_id = -1;

        for (int i = 0; i < m; i++) {
            array[i] = new Matrix(n);

            int l1 = array[i].l1();
            if (l1 < l1_min) {
                l1_min = l1;
                l1_id = i;
            }

            int l2 = array[i].l2();
            if (l2 < l2_min) {
                l2_min = l2;
                l2_id = i;
            }

            System.out.println("Matrix " + (i + 1));
            array[i].printMat();
        }

        System.out.println("Min l1: " + l1_min + ". It has
matrix number " + (l1_id + 1));
        System.out.println("Min l2: " + l2_min + ". It has
matrix number " + (l2_id + 1));

        Matrix add = array[0].add(array[1]);
        System.out.println("Matrix 1 add 2");
        add.printMat();
        Matrix sub = array[0].sub(array[2]);
        System.out.println("Matrix 1 subtract 3:");
        sub.printMat();

```

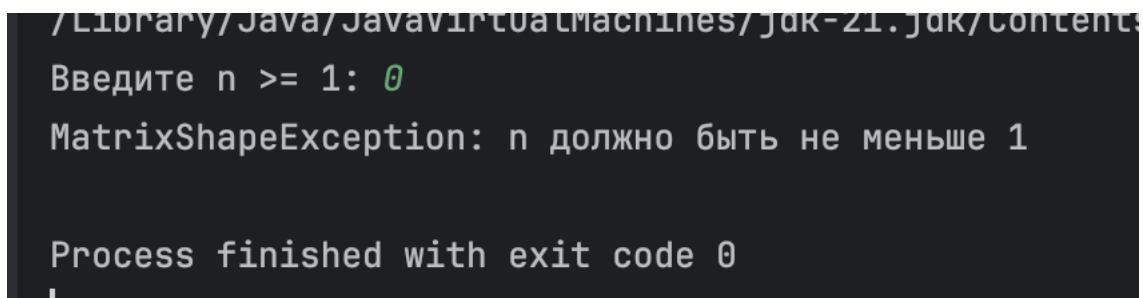
```

        Matrix mul = array[1].mul(array[2]);
        System.out.println("Matrix 2 mul 3: ");
        mul.printMat();

    } catch (MatrixShapeException e) {
        System.out.println(e);
    } catch (InputMismatchException e) {
        System.out.println("Неверный тип данных");
    } catch (OutOfMemoryError e) {
        System.out.println("Переполнение ОП");
    }
}
}

```

Работа программы показана на рисунке 2.



```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents
Введите n >= 1: 0
MatrixShapeException: n должно быть не меньше 1

Process finished with exit code 0

```

Рисунок 2 – Работа программы

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

Код:

```

import java.util.InputMismatchException;
import java.util.Objects;
import java.util.Scanner;
import java.util.concurrent.atomic.AtomicInteger;

class MedCardException extends Exception {
    public MedCardException(String mess) {
        super(mess);
    }
}

```

```

class PatientList {
    public Patient[] createPatients(int n, Scanner scanner)
throws MedCardException {
    Patient[] listPat = new Patient[n];
    for (int i = 0; i < n; i++) {
        System.out.println("Пациент " + i);
        System.out.print("\tФамилия: ");
        String surname = scanner.nextLine();

        System.out.print("\tИмя: ");
        String name = scanner.nextLine();

        System.out.print("\tОтчество: ");
        String father_name = scanner.nextLine();

        System.out.print("\tАдрес: ");
        String address = scanner.nextLine();

        System.out.print("\tТелефон: ");
        String phone = scanner.nextLine();

        System.out.print("\tНомер карты: ");
        int card = scanner.nextInt();
        scanner.nextLine();
        if (card < 0){
            throw new MedCardException("номер карты должен
быть больше нуля");
        }

        System.out.print("\tДиагноз: ");
        String diagnose = scanner.nextLine();

        System.out.println();

        listPat[i] = new Patient(surname, name,
father_name, address, phone, card, diagnose);
    }
    return listPat;
}

class Patient {
    private static final AtomicInteger count = new
AtomicInteger(0);
    int id;
    String surname;
    String name;
    String father_name;
    String address;
    String phone;

```

```

int card;
String diagnose;

public Patient(
    String surname, String name, String fatherName,
    String address, String phone, int card, String diagnose
) {
    this.id = count.incrementAndGet();
    this.surname = surname;
    this.name = name;
    this.father_name = fatherName;
    this.address = address;
    this.phone = phone;
    this.card = card;
    this.diagnose = diagnose;
}

public void print() {
    System.out.println("Пациент id: " + this.id );
    System.out.print("    |  Фамилия: " + this.surname );

    System.out.print("    |  Имя: " + this.name );

    System.out.print("    |  Отчество: " + this.father_name
);

    System.out.print("    |  Адрес: " + this.address );

    System.out.print("    |  Телефон: " + this.phone );

    System.out.print("    |  Номер карты: " + this.card );

    System.out.println("    |  Диагноз: " + this.diagnose );

    System.out.println();
}

}

public class Third_2 {
    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in)) {
            System.out.print("Введите n: ");
            int n = scanner.nextInt();
            scanner.nextLine();
            if (n < 1){
                throw new MedCardException("n должно быть
больше нуля");
            }

            System.out.print("Введите искомый диагноз: ");
            String diag = scanner.nextLine();

```

```

        System.out.println("Введите интервал медицинских
карт: ");
        System.out.print("\tНачало: ");
        int start = scanner.nextInt();
        if (start < 0){
            throw new MedCardException("начало интервала
номеров должно быть больше нуля");
        }
        System.out.print("\tКонец: ");
        int stop = scanner.nextInt();
        scanner.nextLine();
        if (start >= stop){
            throw new MedCardException("конец интервала
должен быть больше его начала");
        }

        PatientList patientCreator = new PatientList();
        Patient[] patients =
patientCreator.createPatients(n, scanner);

        for (int i = 0; i < n; i++) {
            patients[i].print();
        }

        System.out.println("Пациенты с диагнозом \" " +
diag + "\"");
        for (int i = 0; i < n; i++) {
            if (Objects.equals(patients[i].diagnose,
diag))
                patients[i].print();
        }

        System.out.println("Пациенты с номерами медкарт в
диапазоне " + start + " - " + stop);
        for (int i = 0; i < n; i++) {
            if (patients[i].card >= start &
patients[i].card <= stop)
                patients[i].print();
        }
    } catch (MedCardException e) {
        System.out.println(e);
    } catch (InputMismatchException e) {
        System.out.println("Неверный тип данных");
    } catch (OutOfMemoryError e) {
        System.out.println("Переполнение ОП");
    }
}
}

```

Работа программы показана на рисунке 3.

```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java
Введите n: 4
Введите искомый диагноз: xxx
Введите интервал медицинских карт:
    Начало: -5
MedCardException: начало интервала номеров должно быть больше нуля

Process finished with exit code 0
```

Рисунок 3 – Работа программы

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

Код:

```
import java.util.InputMismatchException;
import java.util.Objects;
import java.util.Scanner;
import java.util.concurrent.atomic.AtomicInteger;

class AbiturientException extends Exception {
    public AbiturientException(String mess) {
        super(mess);
    }
}

class AbiturientList {
    public Abiturient[] createAbiturients(int n, Scanner
scanner) {
        Abiturient[] listAb = new Abiturient[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Абитуриент " + i);
            System.out.print("\tФамилия: ");
            String surname = scanner.nextLine();

            System.out.print("\tИмя: ");
            String name = scanner.nextLine();

            System.out.print("\tОтчество: ");
            String father_name = scanner.nextLine();

            System.out.print("\tАдрес: ");
```

```

        String address = scanner.nextLine();

        System.out.print("\tТелефон: ");
        String phone = scanner.nextLine();

        System.out.print("\tОценка по математике: ");
        int math = scanner.nextInt();
        System.out.print("\tОценка по русскому: ");
        int russian = scanner.nextInt();
        System.out.print("\tОценка по информатике: ");
        int informatics = scanner.nextInt();
        scanner.nextLine();

        Marks mark = new Marks(math, russian,
informatics);

        System.out.println();

        listAb[i] = new Abiturient(surname, name,
father_name, address, phone, mark);
    }
    return listAb;
}
}

class Marks {
    int math;
    int russian;
    int informatics;

    public Marks(int math, int russian, int informatics) {
        this.math = math;
        this.russian = russian;
        this.informatics = informatics;
    }

    public double getMean(){
        return (double) (math + russian + informatics) / 3;
    }

    public String toString() {
        return "(мат: " + this.math + "; рус: " + this.russian
+ "; инф: " + this.informatics + ")";
    }

    public boolean hasBad() {
        return (this.math < 3 | this.russian < 3 |
this.informatics < 3);
    }
}

```

```

class Abiturient {
    private static final AtomicInteger count = new
AtomicInteger(0);
    int id;
    String surname;
    String name;
    String father_name;
    String address;
    String phone;
    Marks marks;

    public Abiturient(
        String surname, String name, String fatherName,
String address, String phone, Marks marks
    ) {
        this.id = count.incrementAndGet();
        this.surname = surname;
        this.name = name;
        this.father_name = fatherName;
        this.address = address;
        this.phone = phone;
        this.marks = marks;
    }

    public void print() {
        System.out.println("Абитуриент id: " + this.id );
        System.out.print("    |  Фамилия: " + this.surname );

        System.out.print("    |  Имя: " + this.name );

        System.out.print("    |  Отчество: " + this.father_name
);

        System.out.print("    |  Адрес: " + this.address );

        System.out.print("    |  Телефон: " + this.phone );

        System.out.print("    |  Оценки: " +
this.marks.toString());

        System.out.println();
    }
}

public class Fourth_2 {
    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in)) {

            System.out.print("Введите число абитуриентов: ");
            int m = scanner.nextInt();

```



```

        scanner.nextLine();

        if (m < 1){
            throw new AbiturientException("m должно быть
больше нуля");
        }

        System.out.print("Искомый средний балл: ");
        double mean_mark = scanner.nextDouble();
        if (mean_mark <= 0){
            throw new AbiturientException("средний балл
должен быть больше нуля");
        }
        System.out.print("Введите топ-абитуриентов (n):
");

        int n = scanner.nextInt();
        scanner.nextLine();

        if (n < 1){
            throw new AbiturientException("n должно быть
больше нуля");
        }

        AbiturientList abiturientCreator = new
AbiturientList();
        Abiturient[] abiturients =
abiturientCreator.createAbiturients(m, scanner);

        for (int i = 0; i < m; i++){
            abiturients[i].print();
        }

        System.out.println("Абитуриенты с плохими
оценками:");
        for (int i = 0; i < m; i++){
            if (abiturients[i].marks.hasBad())
                abiturients[i].print();
        }

        System.out.println("Абитуриенты со средним баллом
выше " + mean_mark);
        for (int i = 0; i < m; i++){
            if (abiturients[i].marks.getMean() >
mean_mark)
                abiturients[i].print();
        }

        double[] mean_marks = new double[m];
        for (int i = 0; i < m; i++){
            for (int j = i; j < m; j++) {

```

```

        if (abiturients[i].marks.getMean() <
abiturients[j].marks.getMean()){
            Abiturient tmp = abiturients[i];
            abiturients[i] = abiturients[j];
            abiturients[j] = tmp;
        }
    }

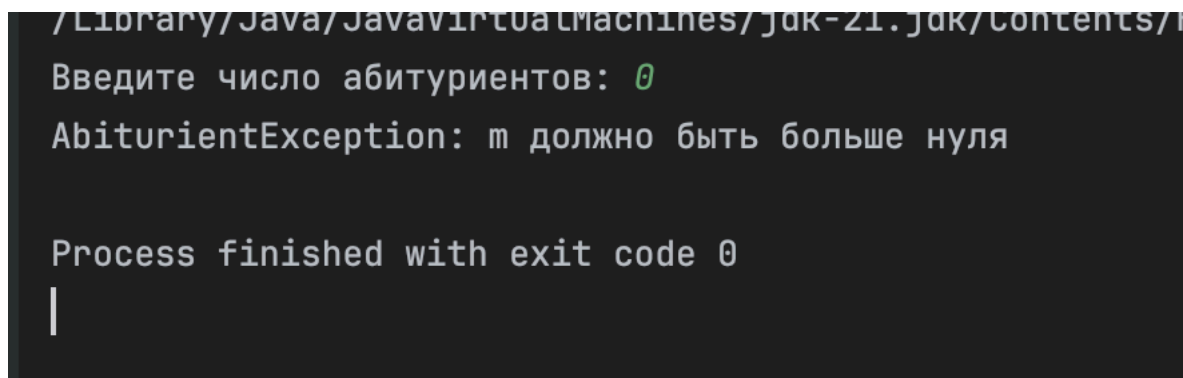
    System.out.println("Список абитуриентов топ-" + n);
    for (int i = 0; i < n; i++){
        abiturients[i].print();
    }

    System.out.println("Абитуриенты с полупроходным
баллом:");
    double half_entrance_mark = abiturients[n-
1].marks.getMean();
    for (int i = 0; i < m; i++){
        if (abiturients[i].marks.getMean() ==
half_entrance_mark)
            abiturients[i].print();
    }

    } catch (AbiturientException e) {
        System.out.println(e);
    } catch (
InputMismatchException e) {
        System.out.println("Неверный тип данных");
    } catch (OutOfMemoryError e) {
        System.out.println("Переполнение ОП");
    }
}
}

```

Работа программы показана на рисунке 4.



```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/
Введите число абитуриентов: 0
AbiturientException: m должно быть больше нуля

Process finished with exit code 0
|

```

Рисунок 4 – Работа программы

**Задание 5:** ввести последовательность строк из текстового потока и выполнить указанные действия. При этом могут рассматриваться два варианта:

- каждая строка состоит из одного слова;
- каждая строка состоит из нескольких слов.

Имена входного и выходного файлов, а также абсолютный путь к ним могут быть введены как параметры командной строки или храниться в файле. В каждой строке найти слова, начинающиеся с гласной буквы.

Код:

```
import java.util.Scanner;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

class FileLineReader {
    public static void readLinesFromFile(String filePath) {
        try (BufferedReader reader = new BufferedReader(new
FileReader(filePath))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.err.println("Ошибка при чтении файла: " +
e.getMessage());
        }
    }

    // public static void main(String[] args) {
    //     String filePath = "путь_к_файлу.txt"; // Укажите
    //     путь к файлу, который нужно прочитать
    //     readLinesFromFile(filePath);
    // }

    public class third_3 {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Введите путь до файла: ");
```

```

String filename = scanner.nextLine();

try (BufferedReader reader = new BufferedReader(new
FileReader(filename))) {
    String line;
    while ((line = reader.readLine()) != null) {
        String[] words = line.split("\\s+");
        System.out.println("Строка: " + line);
        System.out.print("\tСлова, начинающиеся с
гласной буквы: ");
        boolean found = false;
        for (String word : words) {
            if (word.matches("[AEIOUYaeiouy]\\w*")) {
                System.out.print(word + ", ");
                found = true;
            }
        }
        if (!found)
            System.out.println("Не найдены");
        else
            System.out.println();
//        System.out.println(line);
    }
} catch (IOException e) {
    System.err.println("Ошибка при чтении файла: " +
e.getMessage());
}

scanner.close();
}
}

```

Работа программы показана на рисунках 5 – 6.

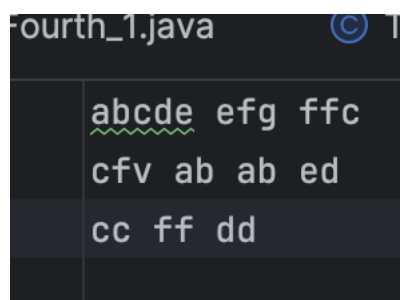


Рисунок 5 – Работа программы

```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home
Введите путь до файла: 3_3.txt
Строка: abcde efg ffc
        Слова, начинающиеся с гласной буквы: abcde, efg,
Строка: cfv ab ab ed
        Слова, начинающиеся с гласной буквы: ab, ab, ed,
Строка: cc ff dd
        Слова, начинающиеся с гласной буквы: Не найдены
```

Рисунок 6 – Работа программы

**Задание 6:** ввести последовательность строк из текстового потока и выполнить указанные действия. При этом могут рассматриваться два варианта:

- каждая строка состоит из одного слова;
- каждая строка состоит из нескольких слов.

Имена входного и выходного файлов, а также абсолютный путь к ним могут быть введены как параметры командной строки или храниться в файле. Найти и вывести слова текста, для которых последняя буква одного слова совпадает с первой буквой следующего слова.

Код:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class fourth_3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введите путь до файла: ");
        String filename = scanner.nextLine();
        StringBuilder text = new StringBuilder();
        try (BufferedReader reader = new BufferedReader(new
FileReader(filename))) {
            //
                String line;

                while ((line = reader.readLine()) != null) {
```

```

        text.append(line).append("\n");
    }

    } catch (IOException e) {
        System.err.println("Ошибка при чтении файла: " +
e.getMessage());
    }

    String content = text.toString();
    System.out.println("Текст файла:\n\n" + content);

//    String text = scanner.nextLine();
    String[] words = content.split("\\s+");

    System.out.println("Слова, для которых последняя буква
одного слова совпадает с первой буквой следующего:");

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

        if (!currentWord.isEmpty() && !nextWord.isEmpty())
        {
            char lastChar =
Character.toLowerCase(currentWord.charAt(currentWord.length()
- 1));
            char firstChar =
Character.toLowerCase(nextWord.charAt(0));

            if (lastChar == firstChar) {
                System.out.println(currentWord + " " +
nextWord);
            }
        }
    }

    scanner.close();
}
}

```

**Работа программы показана на рисунке 7.**

```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Applications
Введите путь до файла: 3_3.txt
Текст файла:

abcde efg ffc
cfv ab ab ed
cc ff dd

Слова, для которых последняя буква одного слова совпадает с первой буквой следующего:
abcde efg
ffc cfv

Process finished with exit code 0

```

Рисунок 7 – Работа программы

**Задание 7:** при выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File. Прочитать текст Java-программы и в каждом слове длиннее двух символов все строчные символы заменить прописными.

Код:

```

import java.io.*;
import java.util.Scanner;

public class Third_4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Введите путь к файлу Java-
программы:");
        String inputFilePath = scanner.nextLine();

        File inputFile = new File(inputFilePath);

        if (!inputFile.exists() || !inputFile.isFile()) {
            System.err.println("Указанный файл не существует
или не является файлом.");
            return;
        }

        try (BufferedReader reader = new BufferedReader(new
FileReader(inputFile))) {
            StringBuilder processedContent = new
StringBuilder();

            String line;

```

```

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

            for (String word : words) {
                if (word.length() > 2) {
processedContent.append(word.toUpperCase()).append(" ");
                } else {
                    processedContent.append(word).append("
");
                }
            }
            processedContent.append("
");

            String outputDirectoryPath = "output_directory";
            File outputDirectory = new
File(outputDirectoryPath);
            outputDirectory.mkdir();

            String outputFilePath = outputDirectoryPath +
"/processed_java_program.java";
            File outputFile = new File(outputFilePath);

            try (BufferedWriter writer = new
BufferedWriter(new FileWriter(outputFile))) {
                writer.write(processedContent.toString());
                System.out.println("Результаты обработки
сохранены в файл: " + outputFilePath);
            } catch (IOException e) {
                System.err.println("Ошибка при записи в файл:
" + e.getMessage());
            }
            } catch (IOException e) {
                System.err.println("Ошибка при чтении файла: " +
e.getMessage());
            }

            scanner.close();
        }
    }
}

```

**Работа программы показана на рисунке 8.**



```

1  import java.io.*;

2  import java.util.Scanner;

3

4  public class THIRD_4 {
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);

7

8          System.out.println("ВВЕДИТЕ ПУТЬ К ФАЙЛУ JAVA-ПРОГРАММЫ:");
9          String inputFilePath = scanner.nextLine();

10

11         File inputFile = new File(inputFilePath);

12

13         if (!inputFile.exists() || !inputFile.isFile()) {
14             System.err.println("УКАЗАННЫЙ ФАЙЛ НЕ СУЩЕСТВУЕТ ИЛИ НЕ ЯВЛЯЕТСЯ ФАЙЛОМ.");
15             return;
16         }

17

18         try (BufferedReader reader = new BufferedReader(new FileReader(inputFile))) {
19             StringBuilder processedContent = new StringBuilder();

20

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

```

Рисунок 8 – Работа программы

**Задание 8:** при выполнении следующих заданий для вывода результатов создавать новую директорию и файл средствами класса File. В файле, содержащем фамилии студентов и их оценки, записать прописными буквами фамилии тех студентов, которые имеют средний балл более “7”.

Код:

```

import java.io.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Fourth_4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Введите путь к файлу с фамилиями
студентов и их оценками:");
        String inputFilePath = scanner.nextLine();

```

```

        File inputFile = new File(inputFilePath);

        if (!inputFile.exists() || !inputFile.isFile()) {
            System.err.println("Указанный файл не существует
или не является файлом.");
            return;
        }

        List<String> studentsAboveSeven = new ArrayList<>();

        try (BufferedReader reader = new BufferedReader(new
FileReader(inputFile))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split("\\s+");
                String name = parts[0];
                double averageGrade =
Double.parseDouble(parts[1]);

                if (averageGrade > 7) {
studentsAboveSeven.add(name.toUpperCase());
                }
            }

            String outputDirectoryPath = "output_directory";
            File outputDirectory = new
File(outputDirectoryPath);
            outputDirectory.mkdir();

            String outputFilePath = outputDirectoryPath +
"/students_above_seven.txt";
            File outputFile = new File(outputFilePath);

            try (BufferedWriter writer = new
BufferedWriter(new FileWriter(outputFile))) {
                for (String student : studentsAboveSeven) {
                    writer.write(student + "\n");
                }
                System.out.println("Фамилии студентов с
оценкой более 7 сохранены в файл: " + outputFilePath);
            } catch (IOException e) {
                System.err.println("Ошибка при записи в файл:
" + e.getMessage());
            }
        } catch (IOException | NumberFormatException e) {
            System.err.println("Ошибка при чтении файла: " +
e.getMessage());
        }

        scanner.close();

```

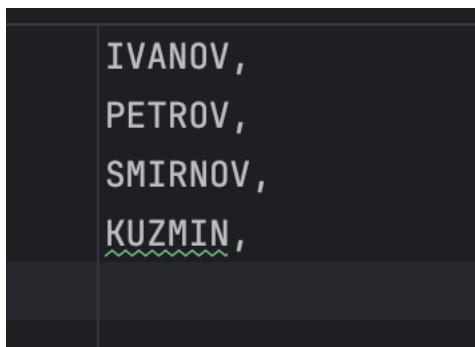
```
}  
}
```

Работа программы показана на рисунках 9 – 10.



```
Ivanov, 9.5  
Petrov, 8.2  
Smirnov, 7.8  
Kuzmin, 7.1  
Lebedev, 6.9|
```

Рисунок 9 – Работа программы



```
IVANOV,  
PETROV,  
SMIRNOV,  
KUZMIN,
```

Рисунок 10 – Работа программы

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